High-Order Moment Methods for Thermal Radiative Transfer

by

Samuel Stephen Olivier

A dissertation submitted in partial satisfaction of the

requirements for the degree of

Doctor of Philosophy

in

Applied Science and Technology

and the Designated Emphasis

in

Computational and Data Science and Engineering

in the

Graduate Division

of the

University of California, Berkeley

Committee in charge:

Professor Per-Olof Persson, Chair
Professor Phillip Colella
Professor Panayiotis Papadopoulos
Dr. Terry Haut

Spring 2022

High-Order Moment Methods for Thermal Radiative Transfer

Abstract

High-Order Moment Methods for Thermal Radiative Transfer

by

Samuel Stephen Olivier

Doctor of Philosophy in Applied Science and Technology

and the Designated Emphasis in

Computational and Data Science and Engineering

University of California, Berkeley

Professor Per-Olof Persson, Chair

Numerically modeling the high energy density regimes characteristic of astrophysical phenomena and inertial confinement fusion (ICF) requires simultaneously modeling hydrodynamics and thermal radiative transfer (TRT). Recently, high-order finite element discretizations of the hydrodynamics equations using high-order (curved) meshes have been shown to have improved robustness and computational performance over low-order methods. Due to the tightly coupled nature of these radiation-hydrodynamics simulations, high-order methods compatible with curved meshes are also desired for TRT.

This dissertation develops high-order, moment-based methods for solving the radiation transport equation, a crucial component of modeling TRT. Moment methods are a class of scale and model-bridging algorithms for solving kinetic equations, such as the radiation transport equation, in the context of multiphysics simulations. An efficient and robust iterative scheme is found by coupling the transport equation to a reduced-dimensional model derived from its statistical moments. The moment equations are closed such that, upon iterative convergence, the reduced-dimensional model is capable of reproducing the physics of the high-dimensional transport equation. Moment methods are attractive in the context of high energy density physics (HEDP) simulations as they provide significant algorithmic flexibility, efficient and robust iterative convergence, and a means to isolate the expensive, high-dimensional transport equation from the evolution of the stiff hydrodynamic multiphysics.

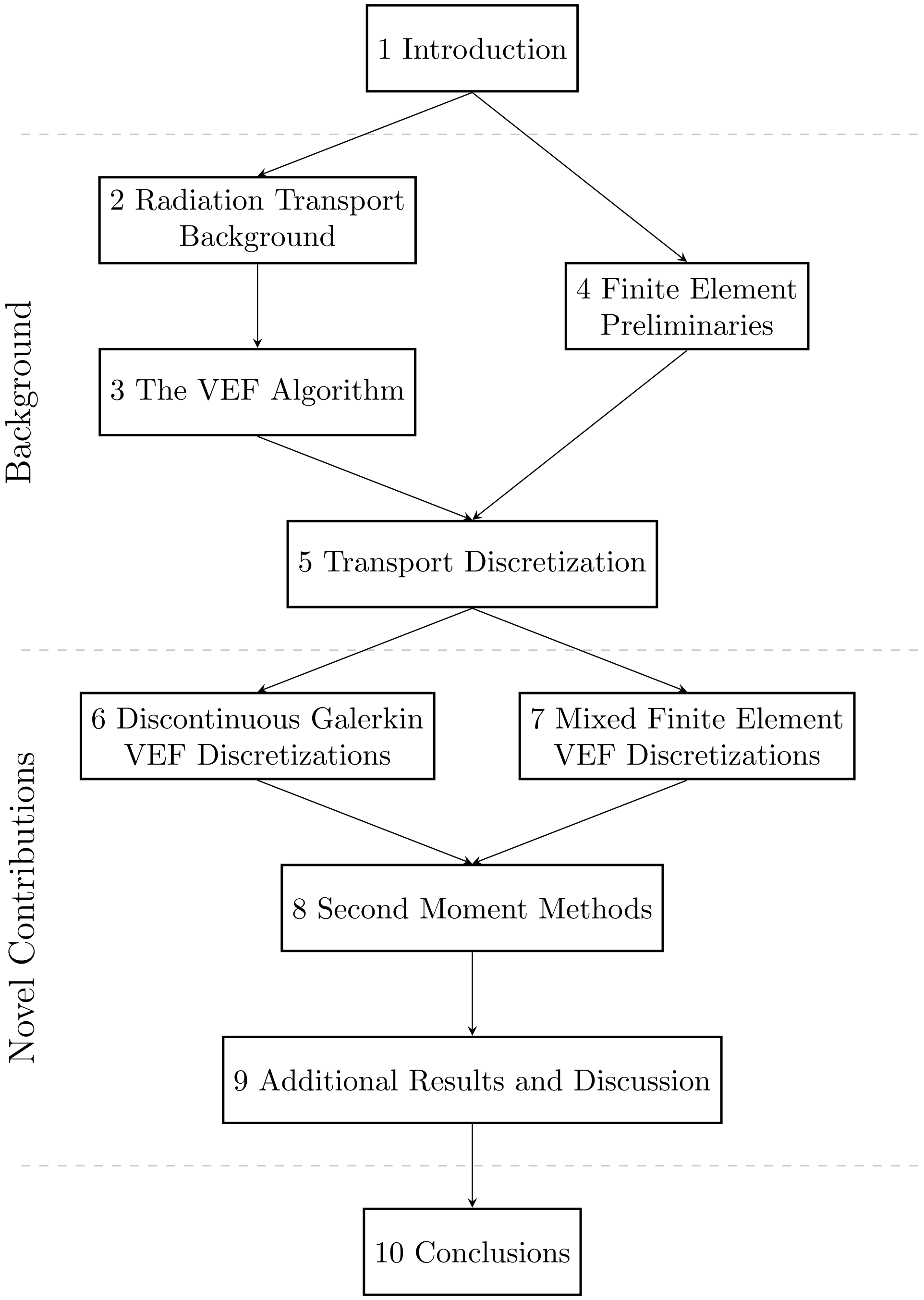
The Variable Eddington Factor (VEF) method is a moment-based transport algorithm where the choice of closure causes the moment system to have an unusual, non-symmetric structure. This makes the development of discretizations for the VEF moment system and their corresponding scalable preconditioned iterative solvers difficult. The flexibility provided by moment methods is leveraged to design discretizations for the VEF moment system that are capable of employing existing linear solver technology. We present Discontinuous Galerkin (DG), continuous finite element (CG), and mixed finite element discretizations that all have high-order accuracy, compatibility with curved meshes, and efficient preconditioned iterative solvers. When paired with a high-order DG discretization of the Discrete Ordinates (S_N) transport equations, the resulting methods form efficient and robust algorithms for solving the radiation transport equation.

We also investigate the use of Second Moment Methods (SMMs), a class of moment methods closely related to the VEF method. SMMs avoid the difficult-to-solve VEF moment system through a clever choice of closure, leading to an iterative scheme where only radiation diffusion must be inverted at each iteration. By leveraging a mathematical connection between SMM and VEF, the VEF methods presented in this dissertation are converted to SMMs to derive novel DG, CG, and mixed finite element-based algorithms. The resulting methods also form robust and efficient transport algorithms while avoiding the non-symmetric solvers that VEF methods require.

This work demonstrates that the algorithmic flexibility allowed by moment methods can be used to design efficient algorithms for radiation transport. In addition, this dissertation serves as the foundation for the design of efficient, high-order, moment-based radiation-hydrodynamics algorithms.

Contents

Interdependence Table


1 Introduction
2 Radiation Transport Background
3 The VEF Algorithm
4 Finite Element Preliminaries
5 Transport Discretization
6 Discontinuous Galerkin VEF Discretizations
7 Mixed Finite Element VEF Discretizations
8 Second Moment Methods
9 Additional Results and Discussion
10 Conclusions
Background
Novel Contributions

Acronyms

AMG	Algebraic Multigrid
BCG	Bi-Conjugate Gradient Method
BiCGStab	Stabilized Bi-Conjugate Gradient Method
BR2	Second Method of Bassi and Rebay
CG	continuous finite element
DG	Discontinuous Galerkin
DSA	Diffusion Synthetic Acceleration
FEM	finite element method
GMG	Geometric Multigrid
GMRES	Generalized Minimal Residual Method
GPU	graphics processing unit
HEDP	high energy density physics
HPC	high performance computer
HRT	Hybridized Raviart Thomas
ICF	inertial confinement fusion
IMC	Implicit Monte Carlo
IP	Interior Penalty
JFNK	Jacobian-free Newton Krylov
LDG	Local Discontinuous Galerkin
LLNL	Lawrence Livermore National Laboratory
LOBPCG	Locally Optimal Block Preconditioned Conjugate Gradient
MDLDG	Minimal Dissipation Local Discontinuous Galerkin
MINRES	Minimal Residual Method
MMS	Method of Manufactured Solutions
NIF	National Ignition Facility
QD	Quasidiffusion
RT	Raviart Thomas
$\mathbf{S}_N$	Discrete Ordinates
SI	Source Iteration
SMM	Second Moment Method
TRT	thermal radiative transfer
USC	Uniform Subspace Correction
VEF	Variable Eddington Factor

List of Symbols

Below is a list of the symbols and mathematical notation used throughout this document.

Mathematical Symbols

$\in$	indicates set membership. $x \in A$ means that x is an element of the set A
$\subset$	indicates subset. $A \subset B$ if all elements of A are also members of B
$\emptyset$	the emtpy set defined as the unique set that has no elements
$\cap$	intersection of two sets. $A \cap B$ has only the elements that belong to both A and B
$\cup$	union of two sets. $A \cup B$ has all the elements of A and B
$\forall$	for all
$\mathbb{R}$	the set of real numbers
dim	denotes spatial dimensionality ($1 \leq \dim \leq 3$)
$\mathcal{D}$	denotes the computational domain, $\mathcal{D} \subset \mathbb{R}^{\dim}$
$\partial\mathcal{D}$	the boundary of the domain, $\mathcal{D}$
$\mathbb{S}^2$	the unit sphere
$\mathbf{e}_i$	the canonical Cartesian basis
$\frac{\mathrm{d}}{\mathrm{d}x}$	total derivative with respect to x
$\frac{\partial}{\partial x}$	partial derivatives with respect to x
∇	the spatial gradient $\nabla = \frac{\partial}{\partial x}\mathbf{e}_x + \frac{\partial}{\partial y}\mathbf{e}_y + \frac{\partial}{\partial z}\mathbf{e}_z$
$\nabla\cdot$	the divergence operator
$:$	scalar contraction of two tensors
$\mathbf{x}$	the spatial variable such that $\mathbf{x} \in \mathcal{D}$
$\mathrm{d}\mathbf{x}$	differential volume element such that $\mathrm{d}\mathbf{x} = \mathrm{d}x\,\mathrm{d}y\,\mathrm{d}z$
$\mathrm{d}s$	differential surface element (e.g. for integrating over an embedded surface)
t	time variable
$\mathcal{O}$	of the order
$\equiv$	defined as

Finite Element Notation

$\hat{K}^{\dim}$	the dim-dimensional reference element, $\hat{K}^{\dim} = [0,1]^{\dim}$
$\mathcal{P}_k(\hat{K}^1)$	space of univariate polynomials of degree less than or equal to k
$\mathcal{Q}_{m,n}(\hat{K})$	tensor product polynomial space of $\mathcal{P}_m(\hat{K}^1)$ and $\mathcal{P}_n(\hat{K}^1)$

$\mathcal{Q}_{\ell,m,n}(\hat{K})$	tensor product polynomial space of $\mathcal{P}_\ell(\hat{K}^1)$, $\mathcal{P}_m(\hat{K}^1)$, and $\mathcal{P}_n(\hat{K}^1)$
$\mathcal{Q}_p(\hat{K})$	tensor product space of equal degree in each coordinate
$\ell_i(\boldsymbol{\xi})$	a nodal basis function for a polynomial space defined on $\hat{K}$
$\boldsymbol{\xi}$	referential coordinate such that $\boldsymbol{\xi} \in \hat{K}$
K	a finite element
$\mathcal{F}$	denotes a face in the mesh
$\mathbf{T}$	reference to physical space transformation such that $K = \mathbf{T}(\hat{K})$
$\mathbf{F}$	Jacobian matrix of an element transformation
J	Jacobian of an element transformation such that $J = \det(\mathbf{F})$
$\mathbf{T}_\mathcal{F}$, $\mathbf{F}_\mathcal{F}$, $J_\mathcal{F}$	transformation, Jacobian, and determinant corresponding to the embedded surface, $\mathcal{F}$
h	characteristic mesh element length
$\mathbb{Q}_p(K)$	space of mapped polynomials defined by composing $\mathcal{Q}_p(\hat{K})$ with the inverse mesh transformation
$\mathbb{D}_k(K)$	element-local polynomial space for the Raviart Thomas space RT_k
$\mathbf{n}$	outward unit normal vector
$\mathbf{n}_K$	outward normal with respect to element K
$\hat{\mathbf{n}}$	outward unit normal vector to a surface on the reference element $\hat{K}$
$\mathbf{t}$	unit vector tangential to a surface
b_i	denotes a global basis function for a finite element space
κ	interior penalty parameter *or* condition number of an algebraic system of equations
$\mathcal{T}$	collection of elements K called a tessellation of the domain, $\mathcal{D}$
$\llbracket \cdot \rrbracket$	the jump of a function across a face between two elements
$\{\!\{ \cdot \}\!\}$	the average of a function evaluated on each side of a face
∇_h	the local gradient defined by applying the gradient locally on each element
$\hat{\nabla}$	gradient in reference space
Γ	Set of unique faces in the mesh
Γ_0	Set of unique interior faces in the mesh
Γ_b	Set of unique faces on the boundary of the mesh, $\Gamma = \Gamma_0 \cup \Gamma_b$
$L^2(\mathcal{D})$	the space of square-integrable functions
$H^1(\mathcal{D})$	the space of square-integrable functions with square-integrabl gradient
$H(\mathrm{div}; \mathcal{D})$	the space of square-integrable vector-valued functions with square-integrable divergence
Y_p	the degree-p Discontinuous Galerkin space

X_p	the vector-valued degree-p Discontinuous Galerkin space ($X_p = [Y_p]^{\mathrm{dim}}$)
V_p	the degree-p continuous finite element space
W_p	the degree-p vector-valued finite element space where each component of the vector belongs to V_p ($W_p = [V_p]^{\mathrm{dim}}$)
RT_p	the order p Raviart Thomas finite element space
Λ_p	the interior trace of the order p Raviart Thomas space, RT_p

Radiation Transport

h	Planck's constant
c	speed of light
$\boldsymbol{\Omega}$	the direction-of-flight variable. $\boldsymbol{\Omega} \in \mathbb{S}^2$
ν	photon frequency
E	neutron energy
ψ	the angular flux
$\bar{\psi}$	the inflow boundary function
$J_n^{\pm}$	half-range partial currents of the angular flux
J_{in}	the inflow current computed from the inflow boundary function, $\bar{\psi}$
ϕ	the zeroth angular moment of the angular flux
φ	denotes the scalar flux solution of the moment
$\boldsymbol{J}$	the first angular moment of the angular flux
$\mathbf{P}$	the second angular moment of the angular flux
$\mathbf{E}$	the Eddington tensor defined as the ratio of the second and zeroth angular moments of the angular flux
E_b	the Eddington boundary factor used to form boundary conditions for the VEF method
$\mathbf{T}$	SMM correction tensor
β	the SMM boundary correction factor
Q_0	the zeroth moment of the fixed-source, q
$\boldsymbol{Q}_1$	the first moment of the fixed-source, q
σ_s	the scattering macroscopic cross section
σ_a	the absorption macroscopic cross section
σ_t	the total interaction macroscopic cross section such that $\sigma_t = \sigma_s + \sigma_a$

Acknowledgments

I am deeply grateful for the support and guidance of many people. Foremost, I would like to thank Dr. Terry Haut, my external advisor from Lawrence Livermore National Laboratory, for his expertise, mentorship, endless encouragement, and infectious enthusiasm for all things math. I would not be the confident mathematician I am today without his calming guidance or our many discussions over the past four years. By Terry's example, I learned to value and enjoy collaboration almost as much as the technical details.

I would also like to thank the members of the "VEF Biweekly Meeting", Professor Dmitriy Anistratov (North Carolina State University) and Drs. Terry Haut, Ben Yee (LLNL), Ben Southworth (LANL), and James Warsa (LANL), who helped sustain three years of productive and insightful meetings that often acted as the sounding board for the ideas presented here. Their expertise saved me countless hours. In particular, I want to thank Professor Anistratov for lending his immense experience with the topics discussed in this dissertation and for his encyclopedic ability to always know a reference to a related idea in the literature.

I am grateful to the "LDRD code team" at LLNL consisting of Drs. Terry Haut, Ben Yee, and Milan Holec, for welcoming me into their productive research environment. Their related efforts, with Ben Yee in particular, expanded the reach, impact, and scope of my work. I would also like to thank LLNL for their generous computing allowance and the MFEM team for providing the software foundation that inspired me to learn as much as I could about finite elements and made the implementation of the enclosed work enjoyable.

My sincere gratitude goes to Professor Jim Morel at Texas A&M University for introducing me to this topic as part of an undergraduate research course. I am truly grateful for his guidance in the earliest stages of my professional development and for identifying the fruitful and engaging research problem that ultimately culminated in this dissertation and the many skills and connections I gained along the way. Jim also brought the second moment method to my attention in my final year of graduate school, which I consider to be the most exciting and promising outcome of my work.

I am very fortunate to have been supported by a Department of Energy Computational Science Graduate Fellowship which afforded me the freedom and autonomy to pursue this research and collaborate so broadly. I am also thankful for the Applied Science & Technology program at UC Berkeley which provided the flexibility to explore all my passions and interests regardless of department or college and design a unique degree I am proud to have. I am truly lucky to have been able to have this experience and sincerely appreciate all those named above (and many more!) for their part in my success.

Chapter 1

Introduction

Thermal radiative transfer (TRT) is a dominant mechanism of energy transfer in the high energy density regimes found in astrophysical phenomena and inertial confinement fusion (ICF) experiments. Thermal radiation is emitted by all matter at a temperature greater than absolute zero. The human body, for example, emits radiation in the infrared region allowing our bodies to be visible on an infrared camera. In an ICF experiment, temperatures and pressures are high enough that thermal radiation is emitted in the "soft X-ray" region and is energetic and abundant enough to alter the pressure exerted on matter, impacting its motion. This tightly coupled interplay between the motion of matter and thermal radiative heat transfer is characterized by the field of radiation-hydrodynamics [1, 2]. Here, we focus on the "radiation" part of radiation-hydrodynamics.

Kinetic models of photon transport phenomena are regarded as first-principles models for TRT. These models are believed to be a key component in reducing the gap between simulation and experiment observed in high energy density physics (HEDP) experiments [3]. Kinetic models are capable of capturing the physics that cheaper models (e.g. radiation diffusion) miss but at the cost of orders of magnitude more computational work and memory usage. In fact, the kinetic TRT package used in radiation-hydrodynamics simulations of the National Ignition Facility (NIF) often occupies 90% of the runtime and memory usage of the entire simulation. Algorithms that reduce the cost of modeling TRT can thus have a significant impact on the cost of the entire simulation, allowing scientists to perform faster design iterations and realize higher accuracy models for the same electricity bill. Existing codes are extremely optimized so reductions in time-to-solution must come from the development of novel algorithms.

In this dissertation, numerical algorithms for efficiently solving the kinetic description of radiation's interaction with matter are developed, the aim being to design methods that can be readily extended and incorporated into the radiation-hydrodynamics codes used to model NIF. The algorithms are centered around the use of the radiation moment equations to accelerate the iterative solution of the kinetic equation. Iterative acceleration is achieved through a bidirectional coupling: the kinetic equation *informs* the moment system through closures while the moment system *drives* the kinetic equation by computing the slow-to-converge physics [4]. Such algorithms are attractive in the context of radiation-hydrodynamics since the moment system can be directly coupled to the hydrodynamics equations providing separation between the expensive kinetic equation and the evolution of stiff multiphysics [5].

In this chapter, we further motivate the need for this research, provide an overview of the approach along with the gap in the literature this dissertation fills, discuss the objectives and scope of the research, and conclude with an outline of the content in this document.

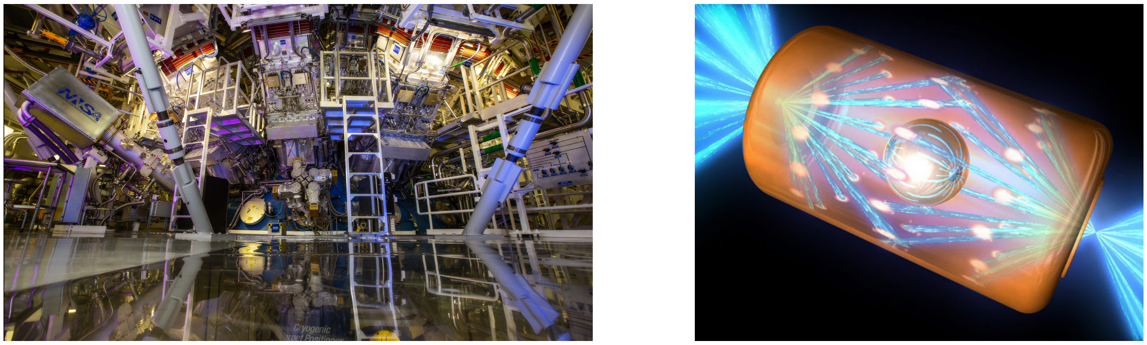
1.1 Motivation

1.1.1 The National Ignition Facility

NIF is a laser-based ICF research facility located at the Lawrence Livermore National Laboratory (LLNL). At the time of writing this dissertation, NIF houses the world's largest and most energetic laser consisting of 192 beam lines. The target bay of NIF is shown in Fig. 1.1a inside of which the 192 beam lines converge onto the interior surfaces of a dime-sized, cylindrical halhraum (depicted in Fig. 1.1b) positioned at the center of the target chamber. The laser heats the walls of the halhraum to extreme temperatures, creating an X-ray oven that bathes a BB-sized capsule of frozen hydrogen isotopes. The X-rays burn the fuel capsule initiating an ablation that compresses the hydrogen to densities and pressures comparable to those seen at the center of the sun. These conditions cause the hydrogen to fuse into helium and release tremendous energy. In August 2021, NIF was able to achieve a burning plasma where the fusion reaction was partially sustained by energy released by the fusion process [6]. This achievement represents a 10x improvement over previous attempts and is an important step toward achieving ignition, where the energy released by the fusion reaction exceeds that of the laser energy used to seed the fusion reaction.

The primary physical processes involved in a NIF experiment are: TRT, plasma physics, and nuclear reactions. Thermal radiation emitted by the walls of the halhraum is the primary driver of the ablation that initiates the fusion reaction. As the fuel heats, it also emits radiation which alters the distribution of energy and motion of the compression. The extreme temperatures and pressures present inside the halhraum mean matter exists in an ionized, plasma state where electrons are stripped free leaving a positively charged nucleus. This separation of charge induces electric and magnetic fields which greatly expand the range of possible motions and significantly complicate the study of the plasma's behavior [7]. Finally, nuclear reactions are responsible for the production of fusion energy and the release of reaction byproducts, such as neutrons, that are an invaluable component for measuring the yield of a NIF experiment.

The numerical simulation of these physical processes, along with many others, comprise a simulation suite that allows scientists to more efficiently design experiments and gain insight into the physics of ICF with reduced reliance on expensive and time-consuming physical experimentation. The demand for increasingly predictive models has made the numerical



(a) (b)

Figure 1.1: (a) a photograph of the National Ignition Facility's target bay showing the laser entrance ports and diagnostics for measuring the properties of the experiment. (b) a depiction of the halhraum placed in the center of the target chamber shown in (a). The lasers impinge on the interior surface of the halhraum by entering through two openings on each side of the cylindrical halhraum. The walls of the halhraum heat to extreme temperatures, releasing X-rays that bathe the hydrogen fuel at the center of the halhraum.

tools themselves topics of significant research spurring research investigations such as the one presented here in this dissertation.

1.1.2 Thermal Radiative Transfer

The focus of this dissertation is the development of numerical methods for simulating the release and absorption of thermal radiation. We seek to model radiation as it moves through and interacts with a participating medium. Interactions of interest include matter absorbing and emitting radiation as well as scattering events where a photon's direction of travel and frequency are altered. The fundamental quantity of interest is the radiation's intensity, $I(\mathbf{x}, \mathbf{\Omega}, \nu, t)$, which is a function of position, $\mathbf{x}$, direction of flight, $\mathbf{\Omega}$, frequency, ν, and time t. It represents the expected amount of energy per unit area, per unit solid angle, per unit frequency bandwidth, and per unit time. The seven-dimensional phase space is depicted in Fig. 1.2. The intensity is mathematically described by the integro-partial differential equation known as the Boltzmann transport equation. Coupling the transport equation to a description of the conservation and exchange of energy between radiation and matter yields the equations of thermal radiative transfer.

The emission of radiation is a highly nonlinear process with emissions scaling according to the fourth power of the temperature of the material [8]. The probabilities of the occurrence of absorption and scattering events also often have nonlinear dependence on the material's temperature. In addition, stimulated scattering, where the probability of scattering depends on the density of nearby radiation, leads to quadratic nonlinearities in the transport equation [9]. The above nonlinearities combined with the incredible expense associated with the

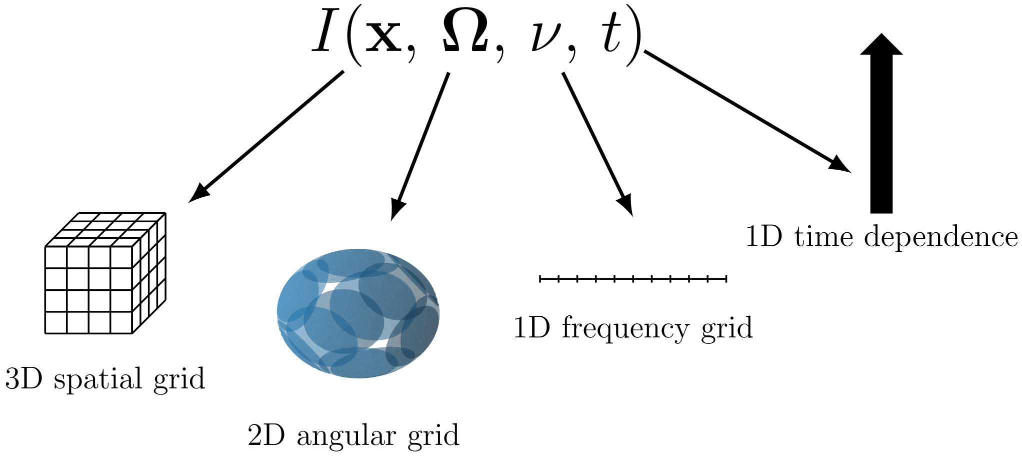


Figure 1.2: A depiction of radiation transport's seven-dimensional phase space.

transport equation's seven-dimensional phase space make the numerical simulation of TRT taxing on even the largest supercomputers.

Numerical methods for solving the Boltzmann transport equation are classified into stochastic and deterministic methods. Stochastic methods use analog simulation to sample the movement of particles and their interactions. By sampling enough particles, the correct intensity can be found. Stochastic methods sample the solution's continuous dependence in angle and frequency and are thus considered the most accurate. However, they are very expensive due to the slow elimination of stochastic error. Deterministic methods discretize each variable of the seven-dimensional phase space producing a large system of algebraic equations that must be solved efficiently. In practice, deterministic methods use iterative solution methods to solve the resulting algebraic system in order to avoid the memory cost associated with storing a system of equations corresponding to the entire phase space. In addition, well-designed iterative methods can significantly reduce the computational cost of solving the discrete transport equations. However, the design of such methods is a highly non-trivial task and has been the focus of significant research tracing back to the 1960s [10].

In this dissertation, we focus on deterministic methods and in particular those that employ the Discrete Ordinates (S_N) angular model. In this approach, the transport equation is collocated at a discrete set of directions and integrations over the angular variable are approximated with a suitable quadrature rule. While a stochastic method such as Implicit Monte Carlo (IMC) [11] has undoubtedly better better angular and frequency resolution (when stochastic variance is sufficiently reduced), it is typically infeasible to perform more than one nonlinear absorption-emission iteration per time step. In addition, IMC often struggles to preserve the diffusion limit when strong material discontinuities are present. Despite the known drawbacks of discrete frequency groups and ray effects, S_N methods are the de facto choice for kinetic models of TRT in HEDP simulations as they are computationally tractable enough to fully converge the nonlinear iteration at each time step, provide the solution in the entire phase space, preserve the diffusion limit, and can achieve iterative efficiency independent of the material parameters. This allows S_N methods to be both faster and more accurate than IMC in many problems of interest.

In the context of radiation-hydrodynamics simulations, numerical models of TRT are used to compute the energy and momentum deposition of the radiation field onto the matter it is interacting with. In this case, we are interested in the amount of energy and momentum imparted to the material due to radiation traveling in all directions and with any frequency. In other words, radiation's effect on matter is communicated through integrals of the intensity over the direction and frequency variables. The resulting integrated quantities of the intensity are known as the moments. The fact that radiation and matter couple through a limited number of moments makes moment methods a natural choice for radiation-hydrodynamics simulations.

1.1.3 High-Order Finite Elements and Curved Meshes

Recent trends in computer architecture, namely the ending of Moore's Law[1], indicate computers will be increasingly parallel and dependent on domain specific architectures such as graphics processing units (GPUs). This is especially evident in the Top500 list[2]: from November 2011 to November 2021, the average number of CPU cores per socket rose from 6 to 27. In that same time span, the number of heterogeneous architectures increased from 39 to 150. Furthermore, it has been observed that floating point throughput is improving faster than memory latency and bandwidth. Thus, data movement will become increasingly expensive relative to computation. For GPU-accelerated computers, data movement is further compounded by the need to transfer data to and from the GPU.

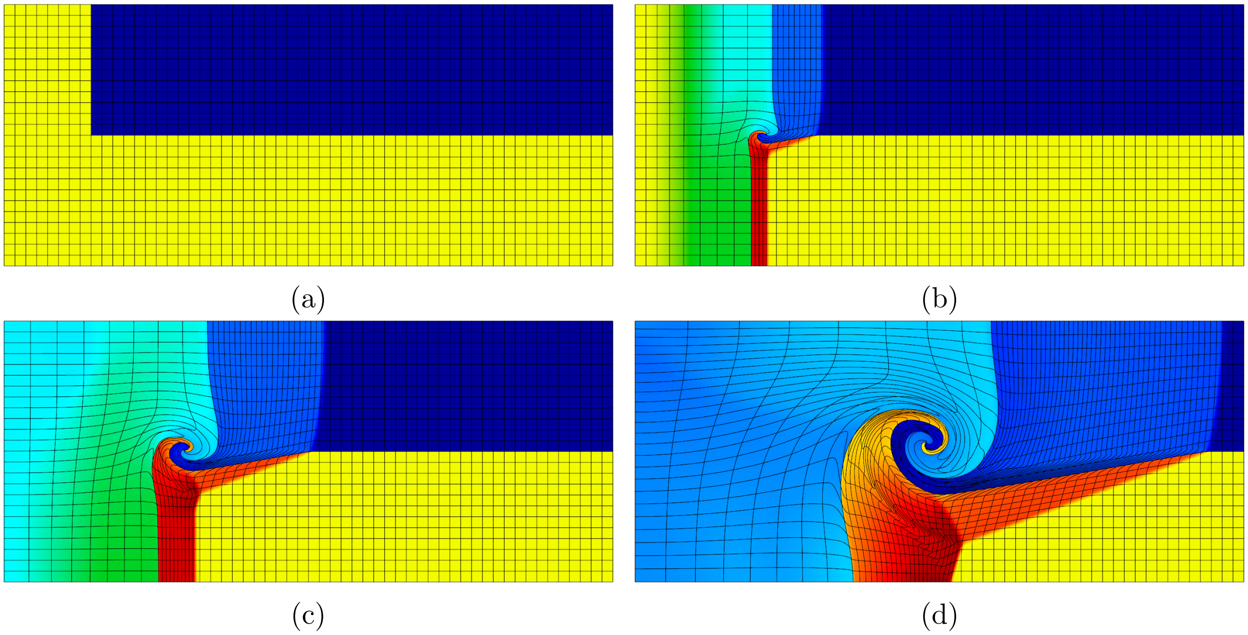
In light of these trends, the Department of Energy's Center for Efficient Exascale Discretizations (CEED) within the Exascale Computing Project (ECP) has targeted high-order finite element methods as one of it's main research thrusts. Compared to low-order methods, high-order methods are more accurate for the same number of unknowns (on smooth problems) and have better data reuse and locality. In other words, high-order methods have a higher floating point operation to memory access ratio[3] that makes them more amenable to efficient implementation on emerging high performance computer (HPC) architectures.

In particular, the next-generation radiation-hydrodynamics code for modeling NIF at LLNL is based around the use of high-order finite elements on high-order (curved) meshes. In hydrodynamics simulations, high-order methods that use curved meshes have been shown to provide greater robustness (especially in the presence of significant mesh distortions), symmetry preservation, and strong scaling when compared to low-order methods [12–14]. In this framework, the material's velocity is represented with continuous finite elements and the thermodynamic variables are represented with discontinuous finite elements. This

[1] the empirical observation that the number of transistors in an integrated circuit doubles every two years.

[2] `https://www.top500.org/`: a ranked list of the world's 500 most powerful supercomputers.

[3] this ratio is called the *arithmetic intensity*.



(a)

(b)

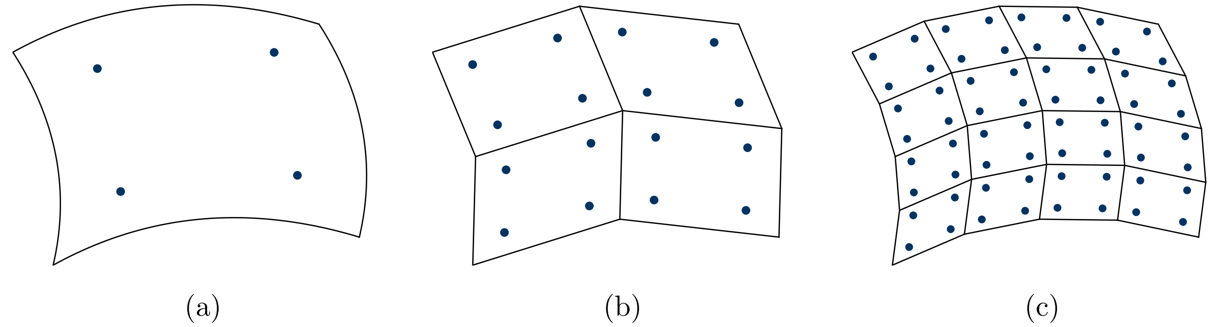
(c)

(d)

Figure 1.3: Selected time steps of the triple point problem. A third-order finite element representation is used to describe the mesh. A Lagrangian hydrodynamics approach is used where the mesh deforms with the materials in order to preserve material interfaces. This leads to the severely distorted curved elements depicted at the final time step. It is on such a mesh that we would like to solve radiation transport. Images taken from `https://computing.llnl.gov/projects/blast/triple-point-shock-interaction`.

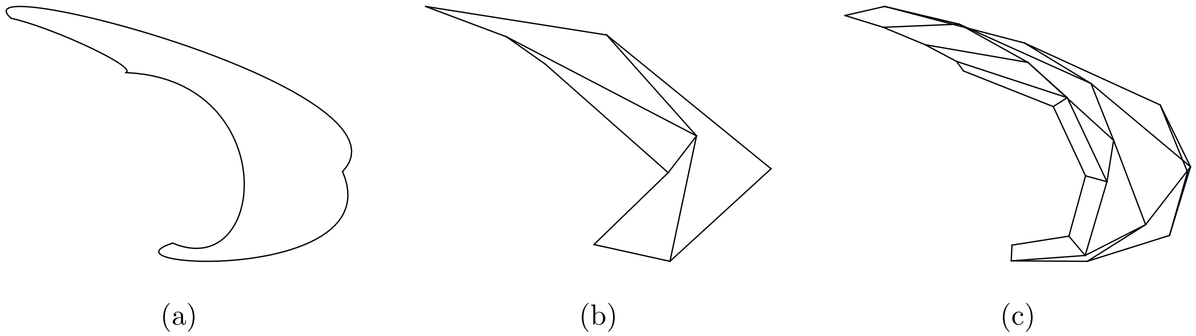
approach ensures the interfaces in the mesh remain continuous while allowing thermodynamic conservation to hold locally on each element. Figure 1.3 shows the evolution of the mesh associated with a third-order Lagrangian hydrodynamics simulation known to as the "triple point problem." In Lagrangian simulations, the mesh moves with the materials in order to preserve material interfaces exactly. Thus, as the materials deform, so does the mesh, leading to the deformed curved interfaces seen in the final mesh of Fig. 1.3.

Coupling TRT to high-order hydrodynamics requires a transport method that is in some sense compatible with curved meshes. One possibility is to leverage existing transport methods by approximating the high-order mesh by refining it and using straight-edged elements. This approach, referred to as the low-order refined approach, is depicted for an example quadratic quadrilateral element in Fig. 1.4. Note that this approach necessarily increases the number of unknowns, depicted as the nodes in each element, in the problem. Haut *et al.* [15] showed that realistic meshes generated from a high-order Lagrangian hydrodynamics code required a significant number of refinements to avoid simulation failure arising from inverted elements. In the context of the already memory and computation intensive seven-dimensional transport problem, this option can be impractical. In addition, under severe



(a) (b) (c)

Figure 1.4: Depictions of the low-order refined process. (a) shows a quadratic quadrilateral element that has four unknowns. (b) and (c) show linear approximations to the geometry of the element in (a) that use one and two refinements, respectively. In order for the curved surfaces to be accurately captured, refinements are required. However, this necessarily increases the number of unknowns which can be impractical in the context of the already memory intensive radiation transport solve.



(a) (b) (c)

Figure 1.5: Depictions of the low-order refined process on a very distorted, cubic element. In this case, refining the high-order geometry shown in (a) leads to elements with poor aspect ratios, inverted elements, and elements that overlap. This is an example where naively refining the element would lead to simulation failure and is a motivating example for the need to solve on the high-order mesh.

mesh distortions the low-order refined approach may not always produce an admissible mesh. Figure 1.5 shows an example of a distorted third-order element where the low-order refined process leads to an ill-defined mesh containing inverted and overlapping elements as well as elements with poor aspect ratios. In such case, a method that solves on the high-order mesh could continue whereas a method reliant on the low-order refined mesh would cause the simulation to fail.

Radiation transport methods compatible with curved meshes are desired in order to avoid the necessary increase in unknowns and reduced robustness associated with the low-order refined approach. It is also possible that high-order methods could be beneficial in terms of accuracy and multiphysics compatibility with high-order hydrodynamics. However, these claims of increased accuracy and compatibility have yet to be demonstrated in large-scale, tightly coupled simulations. Discontinuous finite element representations of the energy density (scalar flux in nuclear engineering terminology) are preferred in order to have immediate multiphysics compatibility with the framework of [12]. High-order discontinuous finite element methods for radiation transport have received attention recently with the development of Discontinuous Galerkin (DG) discretizations of the S_N transport equations compatible with curved meshes in [15, 16] and corresponding Diffusion Synthetic Acceleration (DSA) methods in [17, 18]. However, high-order moment-based transport algorithms compatible with curved meshes have not yet been developed.

1.2 Moment Methods for Radiation Transport

Moment methods are a class of iterative schemes for solving kinetic equations. Example applications include radiation transport [19], plasma physics [20], and ocean modeling [21]. Chacón *et al.* [4] provides an excellent survey of their 60 year history. These methods are characterized by iteratively coupling the kinetic equation to its statistical moments. Closures of the moment system are used such that, upon iterative convergence, the moment system is capable of reproducing the physics of the kinetic equation. In the case of radiation transport, the closures are weak functions of the solution, allowing the design of efficient and robust iterative schemes.

In the radiation transport literature, moment methods have been shown to allow significant algorithmic flexibility in that any valid discretization of the moment system will yield a rapidly converging algorithm [22]. This is in stark contrast to preconditioning schemes, such as DSA, which place severe restrictions on the discretization of the moment system in order to guarantee iterative efficiency [23]. In the case where the transport and moment discretizations are not algebraically consistent, referred to as an “independent” moment method [22, 24], the discrete moment and transport solutions will differ on the order of the spatial discretization error. Thus, the two solutions will be equivalent only in the limit as the mesh is refined. However, even in an under-resolved problem, moment methods still produce a “transport solution” in that the moment algorithm produces a discrete solution to an equivalent reformulation of the transport equation. Furthermore, moment methods generally preserve the thick diffusion limit [25] and have conservation even if the transport discretization in isolation does not. Such properties are particularly useful in the context of multiphysics calculations since the lower-dimensional moment system can be directly coupled to the other physics components in place of the high-dimensional transport equation. Most importantly, moment methods allow the transport and moment discretizations to be chosen independently meaning each can be designed to be in some sense optimal for their intended use.

In this section, we introduce the two classes of moment methods investigated in this dissertation. Relevant literature is discussed in order to identify the research gaps this dissertation fills.

1.2.1 The Variable Eddington Factor Method

The Variable Eddington Factor (VEF) method [26, 27], also known as Quasidiffusion (QD) [19], is a particular class of moment method that employs multiplicative closures. The closures are built from transport-dependent functionals that are bounded and possess small functional derivatives with respect to the transport solution. This allows the design of robust and efficient iterative schemes but comes with the costs of the moment system no longer being self-adjoint and, due to the nature of multiplicative closures, the resulting iteration being nonlinear. VEF has been applied to a wide range of transport and multiphysics problems including (but not limited to) nuclear reactor eigenvalue problems [28], nuclear reactor kinetics [29], and TRT [30]. In addition, VEF performs well in problems having both optically thick and thin regions and treats anisotropic scattering equally well [31, 32].

VEF-based moment algorithms have been designed to improve efficiency in relation to all seven dimensions of the transport equation. Ghassemi and Anistratov [33] showed that different order temporal discretizations can be applied to the transport and VEF equations. Ongoing work suggests that time-stepping stability and accuracy can be maintained when just one transport inversion is performed per time step [34]. Anistratov and Coale [35] used data compression techniques to reduce storage costs in time-dependent calculations. In astrophysics, VEF is used to simplify the implementation of coupling TRT to hydrodynamics and to avoid the memory cost of solving the time-dependent transport equation [36–38]. Davis *et al.* [39] used a short characteristics discretization of the transport equation. Olivier and Morel [40] and Lou *et al.* [41] designed a spatial discretization of the VEF equations to increase multiphysics compatibility. This algorithm was used to form the basis of an efficient radiation-hydrodynamics method in [42]. Yee *et al.* [43] showed that robust convergence is maintained even when positivity-preserving methods are used inside the iteration. Anistratov [44] solved the multigroup TRT equations by using a VEF method with multiple levels in frequency. The stability of this algorithm was analyzed in [45]. It is also well-known that the multigroup eigenvalue problem can be solved with only the need for eigenvalue iterations on the one-group VEF equations [10].

The above techniques rely on the efficient solution of the discretized VEF moment equations. VEF methods reduce the overall cost of the simulation by trading inversions of the high-dimensional transport equation with inversions of the lower-dimensional VEF equations. In all of VEF's applications, the solution of the discretized VEF equations is buried under multiple nested loops corresponding to time integration, Newton iterations, eigenvalue iterations, multi-group iterations, and/or fixed-point iterations. The efficient iterative solution of the VEF equations is then crucial to the efficiency of the overall algorithm and is a prerequisite for the practicality of any VEF method.

The unusual structure of the VEF equations and their lack of self-adjointness make the development of discretizations and their corresponding preconditioned iterative solvers difficult. Previous work on discretizing the VEF equations includes finite volume [24, 31, 36, 46, 47], finite difference [48], mixed finite element [40–42, 49, 50], continuous finite element [22, 51], and discontinuous finite element [52] techniques. While considerable effort has been placed into the discretization of the VEF equations, to our knowledge, existing methods either rely on expensive and unscalable preconditioners such as block incomplete LU (BILU) factorization, cannot be solved with iteration counts independent of the mesh size, or do not mention solvers entirely. In addition, none of these methods are compatible with curved meshes or achieve higher than second-order accuracy in space.

Warsa and Anistratov [22] showed that VEF methods with and without algebraic consistency converged equivalently as long as the transport-dependent closures were properly represented. In particular, computing the VEF closures using finite element interpolation and S_N angular quadrature enabled rapid convergence for the independent discretizations they considered. Thus, an independent discretization of the VEF moment system has the potential to provide the rapid convergence of an algebraically consistent VEF method while enabling significant algorithmic flexibility. This flexibility can be used to define VEF methods that leverage existing linear solver technology and/or have increased multiphysics compatibility.

1.2.2 The Second Moment Method

The Second Moment Method (SMM) of Lewis and Miller [53] is another moment method designed for the radiation transport equation. Where VEF uses nonlinear, multiplicative closures to form the VEF moment system, the SMM moment system is formed using linear, additive closures. The original concept was to close the radiation diffusion system by means of transport-dependent corrections that vanish when the transport solution is a linear function in angle [10]. In this way, the SMM algorithm is able to form a transport method where the moment system is the symmetric radiation diffusion operator with additional transport-dependent source terms. This avoids the difficulty of developing discretizations and preconditioners for the non-symmetric VEF operator required by VEF methods and allows use of simpler iterative solvers such as the conjugate gradient method.

A Fourier analysis of the iterative convergence of SMM was conducted by Cefus and Larsen [54] where SMM was shown to have equivalent convergence as DSA. It was also shown that VEF and SMM will converge equivalently when in close enough vicinity of the solution. Furthermore, they showed that SMM can be viewed as a VEF method that has been linearized about a linearly anisotropic solution. These results suggest the benefits of the VEF algorithm will extend to SMM.

Since the SMM moment system is the simple radiation diffusion equation, discrete SMMs can leverage existing discretization and scalable solver technology immediately. However, the representation of the SMM correction sources shares many of the same discretization difficulties seen for the VEF moment system. Investigation into discrete SMMs is limited in the literature with the only previous work being Stehle *et al.* [55] and Anistratov *et al.* [56]. Stehle *et al.* [55] developed a domain decomposition method where an SMM is used to couple transport and diffusion domains through interface conditions. The SMM moment system and correction sources were discretizated to be algebraically consistent with the upstream corner balance (UCB) [57] S_N transport method. Anistratov *et al.* [56] investigated a multilevel in energy algorithm for neutron transport problems. The discretization of the moment system was designed to be consistent with a lowest-order DG discretization of the S_N transport equations. This method can be viewed as a linearization of the DG VEF method from [52]. Neither of these methods attain higher than second-order accuracy or are compatible with curved meshes. Furthermore, independent SMMs have not yet been developed.

1.3 Objectives and Scope

The goal of this work is to develop computationally efficient moment methods for solving the kinetic radiation transport equation that can be readily incorporated into the radiation-hydrodynamics codes used to model NIF. In particular, we seek to design discretizations for the moment system that have:

1. high-order accuracy,
2. compatibility with curved meshes, and
3. efficient preconditioned iterative solvers.

These objectives are achieved using the independent approach. The flexibility of this approach allows use of any valid discretization of the moment equations as long as the closures are properly represented. Thus, we pursue the extension of discretization techniques developed for the model elliptic problem to the unusual structures of the VEF moment system and the SMM correction sources. Our hypothesis is that these discretization techniques and their associated preconditioned iterative solvers will also be effective for the VEF and SMM moment systems allowing the design of efficient moment methods.

We limit the scope of this investigation to the steady-state, mono-energetic, linear Boltzmann equation with isotropic scattering. This simplified model is derived from the TRT system by neglecting time and frequency dependence and replacing the temperature coupling with isotropic scattering. This model problem emulates a single Newton iteration for each discrete frequency group in a full TRT simulation and thus represents a first step toward developing methods for the much more complicated TRT and radiation-hydrodynamics systems. The efficacy of this proxy problem has already been demonstrated as one of the methods presented in this dissertation has been successfully used in a frequency-independent TRT context [34].

1.4 Outline

The structure of this document is depicted graphically on page vi. We begin in Chapter 2 with background on radiation transport. We present the equations of thermal radiative transfer and the simplifications that result in the steady-state Boltzmann equation that serves as our model problem. The chapter concludes with a derivation of the radiation diffusion approximation.

Chapter 3 derives the VEF and SMM moment systems by applying suitable closures to the angular moments of the model transport problem defined in Chapter 2. We present the techniques used in this document for solving the coupled transport-moment system and discuss the mathematical properties of the VEF and SMM closures, including their close connection through linearization. These properties are used to further motivate the need for the specialized discretization techniques developed here.

Chapter 4 provides an introduction to the finite element method. The key high-level ideas of weak forms and Sobolev spaces are motivated. We then define the finite element spaces used in subsequent chapters to discretize the transport equation and moment systems. These spaces are defined on a computational mesh and are built as piecewise polynomial functions on each element in the mesh. We provide implementation details on the representation of high-order meshes and the associated transformations needed to perform numerical integration over arbitrary elements. The chapter concludes with a brief description of preconditioned iterative solvers.

The transport discretization used by both the VEF and SMM algorithms is defined in Chapter 5. We use the S_N angular model along with a DG discretization in space. The efficient solution procedure known as the transport sweep is defined. Summaries of recent advances related to solving the S_N transport equations on high-order meshes are provided. This includes a graph algorithm to sweep on curved meshes with reentrant faces, discussion of numerically integrating the inflow conditions on curved faces, and the extension of positivity-preserving flux fixup methods to high-order solution representations. The chapter concludes with a discussion of the discrete moment algorithm. The closures for the moment system are computed using the S_N angular quadrature and finite element interpolation. We also present the construction of the transport equation's scattering source from the solution of the moment system.

Chapters 6–8 contain the novel contributions of this dissertation. Discretizations of the moment system are presented that, when combined with the transport discretization from Chapter 5, lead to efficient high-order methods for solving the Boltzmann transport equation. Each chapter concludes with numerical results demonstrating the efficacy of the methods presented therein using the Method of Manufactured Solutions (MMS), an asymptotic thick diffusion limit stress test problem on both an orthogonal and curved mesh, a multi-material proxy problem from TRT referred to as the crooked pipe problem, and a parallel weak scaling study of the preconditioned iterative solvers used to invert the moment systems.

In Chapter 6, a DG discretization is applied to the VEF moment system. We extend the unified analysis of DG methods for elliptic problems presented by Arnold *et al.* [58] to the VEF moment system to derive a family of discretizations for the second-order form of the VEF equations. This family of methods is efficiently preconditioned by the recently developed Uniform Subspace Correction (USC) preconditioner in Pazner and Kolev [59]. The DG framework is also used to derive a continuous finite element discretization of the VEF equations closely related to the method of Warsa and Anistratov [22].

Chapter 7 applies mixed finite element techniques to the VEF moment system. Such methods are chosen to match as closely as possible the methods used in the radiation diffusion package associated with the hydrodynamics code of [12]. We present three discretizations of which two can be effectively preconditioned using the standard techniques developed for the solution of mixed finite element discretizations of elliptic problems.

Chapter 8 shifts focus to SMM. The connection between SMM and VEF established in Chapter 3 is leveraged to convert the discrete VEF algorithms developed in Chapters 6 and 7 to SMMs. This includes deriving DG, continuous finite element, and mixed finite element-based SMMs. The resulting discrete moment systems are effectively preconditioned by their associated standard techniques on the simpler problem of radiation diffusion.

Chapter 9 serves two purposes. First, we use the DG VEF method to investigate generalities of the VEF method. This includes investigations into the effect of the choice of the initial guess for the preconditioned iterative solvers used to invert the moment systems and the use of Anderson acceleration to mitigate the slower convergence observed on meshes with reentrant faces. Second, we draw comparisons between all the methods presented in Chapters 6–8. Performance on the common benchmarks of the thick diffusion limit, crooked pipe problem, and weak scaling are presented side-by-side in order to facilitate their comparison.

Finally, Chapter 10 summarizes the findings in this dissertation. Conclusions from each of Chapters 6–9 are presented. In addition, directions for future work concerning the extension of the methods presented here to a full radiation-hydrodynamics algorithm are included.

Chapter 2

Radiation Transport Background

The purpose of this chapter is to motivate and define the simplified transport problem that serves as the model for the development of the moment methods derived in later chapters. To that end, the general kinetic Boltzmann equation is introduced and then particularized to the equations of thermal radiative transfer which describe thermal photons in the absence of material motion and heat conduction. We pay particular attention to the notational differences present in the fields of nuclear engineering, astrophysics, and HEDP. The simplifications of isotropic scattering and frequency and time-independence are applied to derive the model problem. The chapter concludes with a derivation of the radiation diffusion approximation.

2.1 The Boltzmann Transport Equation

A discussion of the Boltzmann transport equation must begin with the definition of the particle distribution function, f. The set of all possible positions $\mathbf{x}$ and velocities $\mathbf{v}$ is called the system's phase space. The particle distribution function represents the expected number of particles at each point in the phase space at a given time. In other words, $f(\mathbf{x}, \mathbf{v}, t)\,\mathrm{d}\mathbf{x}\,\mathrm{d}\mathbf{v} =$ the expected number of particles in the differential phase space volume at a time t.

The Boltzmann transport equation describes the evolution of the particle distribution function over time. In general, the Boltzmann equation is written as:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \left(\frac{\partial f}{\partial t}\right)_{\text{force}} + \left(\frac{\partial f}{\partial t}\right)_{\text{collision}} , \tag{2.1}$$

where the force and collision terms are application-dependent sources or sinks for the particle distribution function [7, Chapter 7]. The force term represents the effect of an external influence (i.e. not caused by the particles themselves) and the collision term represents the effect of particles colliding with each other or a background material. If a force $\mathbf{F}$ acts on the particles over a time period of $\mathrm{d}t$, the particles' position and velocity will change by an amount $\mathrm{d}\mathbf{x} = \mathbf{v}\,\mathrm{d}t$ and $\mathrm{d}\mathbf{v} = \frac{\mathbf{F}}{m}\,\mathrm{d}t$ where m is the mass of the particle. Taking the total derivative of f in the phase space:

$$\begin{aligned} \mathrm{d}f &= \frac{\partial f}{\partial t}\,\mathrm{d}t + \nabla f \cdot \mathrm{d}\mathbf{x} + \nabla_{\mathbf{v}} f \cdot \mathrm{d}\mathbf{v} \\ &= \frac{\partial f}{\partial t}\,\mathrm{d}t + \nabla f \cdot \mathbf{v}\,\mathrm{d}t + \nabla_{\mathbf{v}} f \cdot \frac{\mathbf{F}}{m}\,\mathrm{d}t\,, \end{aligned} \tag{2.2}$$

where $\nabla_{\mathbf{v}} = \frac{\partial}{\partial v_x}\mathbf{e}_x + \frac{\partial}{\partial v_y}\mathbf{e}_y + \frac{\partial}{\partial v_z}\mathbf{e}_z$ denotes the gradient in velocity space. Dividing by $\mathrm{d}t$ yields:

$$\frac{\mathrm{d}f}{\mathrm{d}t} = \frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{\mathbf{F}}{m} \cdot \nabla_{\mathbf{v}} f\,. \tag{2.3}$$

Thus, the force term $\left(\frac{\partial f}{\partial t}\right)_{\text{force}} = \frac{\mathbf{F}}{m} \cdot \nabla_{\mathbf{v}} f$ so that the general form of the Boltzmann equation can be equivalently be written:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f + \frac{\mathbf{F}}{m} \cdot \nabla_{\mathbf{v}} f = \left(\frac{\partial f}{\partial t}\right)_{\text{collision}}\,. \tag{2.4}$$

For charged particle transport, the particles experience external electromagnetic forces and frequently interact with each other through Coulomb collisions. In this case, particles with charge q experience the force $\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$ due to the electric field, $\mathbf{E}$, and magnetic field, $\mathbf{B}$. The Coulombic collision term is described by the (quite complicated) Fokker-Planck operator [7, Eq. 7.25].

Here, we focus on neutral particle transport (i.e. photons or neutrons). Since these particles do not have charge, there are no external electromagnetic forces. Furthermore, particle-particle collisions are exceedingly rare and it is thus commonplace to ignore them [60]. The collision term then only includes the interaction of the neutral particle with the background medium. These interactions are broadly classified into absorption and scattering events. The probabilities of these events occurring per unit distance traveled are called cross sections for neutrons and opacities for photons and are governed by nuclear physics. The most general form of the Boltzmann transport equation for a neutral particle is then:

$$\frac{\partial f}{\partial t} + \mathbf{v} \cdot \nabla f = \left(\frac{\partial f}{\partial t}\right)_{\text{absorption}} + \left(\frac{\partial f}{\partial t}\right)_{\text{scattering}} + \left(\frac{\partial f}{\partial t}\right)_{\text{source}}\,, \tag{2.5}$$

where we have split the collision term into absorption and scattering terms and included an additional term representing an external source of particles. For neutrons, the external source could be a fixed-source of particles such as a radioactive material that emits neutrons. For photons, we consider the source to be the thermal emission of photons. Thus, the photon external source is dependent on the material's temperature. We note that the case of the general Boltzmann equation under the assumption that particles do not interact with each other is called the linear Boltzmann equation.

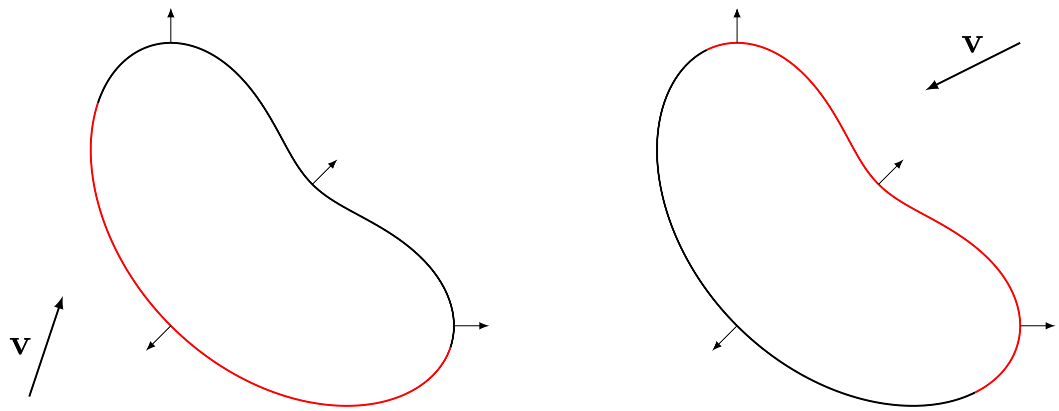


Figure 2.1: A kidney bean shaped domain depicted with a selection vectors normal to its boundary. The inflow region corresponding to $\mathbf{v} \cdot \mathbf{n} < 0$ is colored in red for two example directions of $\mathbf{v}$. The value of the distribution function in the phase space must be supplied at each position corresponding to the inflow region shown in red.

2.2 Boundary and Initial Conditions

Suppose that the Boltzmann transport equation is solved in a spatial domain $\mathcal{D}$ with boundary $\partial\mathcal{D}$. To solve the transport equation, the initial and boundary conditions for the particle distribution function must be provided. The initial condition corresponds to specifying $f(\mathbf{x}, \mathbf{v}, 0)$: the expected number of particles in the phase space at time $t = 0$. Letting $\mathbf{n}$ represent the outward unit normal to the boundary of the domain, the boundary condition specifies $f(\mathbf{x}, \mathbf{v}, t)$ for each $\mathbf{x} \in \partial\mathcal{D}$ satisfying $\mathbf{v} \cdot \mathbf{n} < 0$ for all values of t. Figure 2.1 depicts an example domain and its boundary along with a selection of normal vectors. The region $\mathbf{v} \cdot \mathbf{n} < 0$ where the boundary conditions must be provided are shown in red. The case where

$$f(\mathbf{x}, \mathbf{v}, t) = \bar{f}(\mathbf{x}, \mathbf{v}, t)\,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \mathbf{v} \cdot \mathbf{n} < 0\,, \tag{2.6}$$

is called an inflow boundary condition with $\bar{f}$ the inflow boundary function. The special case $\bar{f} \equiv 0$ is called a vacuum boundary condition. Other types of boundary conditions commonly used in radiation transport are provided in [60, §1.3].

2.3 Navigating Notation

Application of the Boltzmann transport equation to modeling radiation is home to three sets of notation: nuclear engineering, astrophysics, and HEDP. Here, we define the notation used to describe the radiation field and its moments. The hope of this section is to provide the means for newcomers to this subject to navigate these three fields.

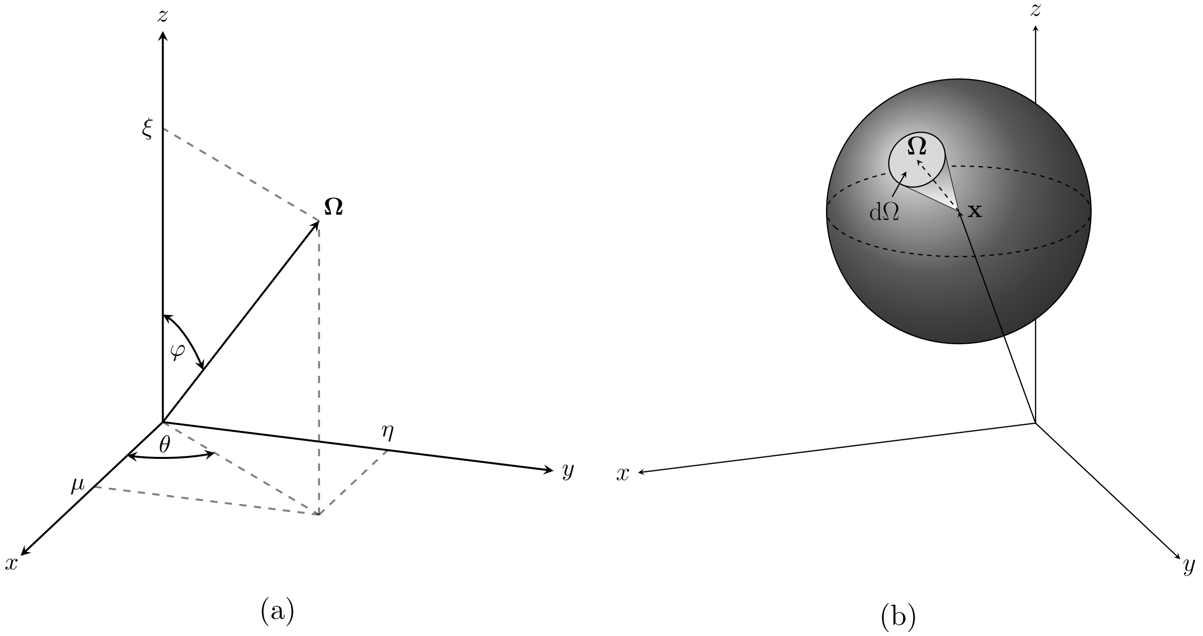


Figure 2.2: (a) a depiction of the spherical coordinate system used for the direction of particle travel variable, $\mathbf{\Omega}$. Here, $\theta \in [0, 2\pi]$ is the azimuthal angle and $\varphi \in [0, \pi]$ the polar angle. (b) A depiction of the direction-of-flight portion of the phase space at a fixed location $\mathbf{x}$, frequency ν, and time t. At each position $\mathbf{x}$, particles can travel in any direction on the unit sphere.

2.3.1 f, ψ, and I

In radiation transport, the velocity variable is represented using a direction-of-flight variable, $\mathbf{\Omega}$, and an energy variable. The angular variable, $\mathbf{\Omega}$, is a direction on the unit sphere, $\mathbb{S}^2$, and is described using the spherical coordinate system depicted in Fig. 2.2a where polar and azimuthal angles are used to define the direction $\mathbf{\Omega}$ in three-dimensional space. Figure 2.2b depicts the direction-of-flight portion of the phase space. Particles at any point in time, space, and frequency can travel in any direction on the unit sphere. The differential phase space volume associated with the angular variable is the cone $\mathrm{d}\Omega$ about the direction $\mathbf{\Omega}$.

In the case of neutrons, the energy variable is $E = \frac{1}{2}mv^2$ where $v = |\mathbf{v}|$ is the speed of the particle. For photons, $E = h\nu$ with h Planck's constant and ν the photon frequency. Since h is a constant, the photon distribution is usually presented as a function of frequency. These distribution functions are related through:

$$\iiint f(\mathbf{x}, \mathbf{v}, t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\mathbf{v} \, \mathrm{d}t = \iiiint f(\mathbf{x}, \mathbf{\Omega}, E, t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\Omega \, \mathrm{d}E \, \mathrm{d}t = \iiiint f(\mathbf{x}, \mathbf{\Omega}, \nu, t) \, \mathrm{d}\mathbf{x} \, \mathrm{d}\Omega \, \mathrm{d}\nu \, \mathrm{d}t \, . \tag{2.7}$$

That is, each representation of the velocity variable represents the same expected number of particles. Although $\mathbf{\Omega}$ is a three-dimensional vector, it is defined in spherical coordinates using only two parameters. This is possible since $\mathbf{\Omega}$ is a unit vector. Thus, $f(\mathbf{x}, \mathbf{\Omega}, E, t)$ and $f(\mathbf{x}, \mathbf{\Omega}, \nu, t)$ are both still functions of seven independent variables.

For nuclear reactors, neutrons are the primary particle of interest and the goal is to understand the fission power produced by the system. To that end, the transport equation is cast in terms of the *angular flux*, ψ. The angular flux represents the particle path length density in the phase space. In other words, $\psi \,\mathrm{d}\mathbf{x}\,\mathrm{d}\Omega\,\mathrm{d}E\,\mathrm{d}t$ represents the expected distance traveled by particles located in the phase space element $\mathrm{d}\mathbf{x}\,\mathrm{d}\Omega\,\mathrm{d}E$ in the time interval $\mathrm{d}t$. In relation to the distribution function, f, the angular flux is defined as the product of the particle speed and the distribution function:

$$\psi(\mathbf{x}, \mathbf{\Omega}, E, t) \equiv v f(\mathbf{x}, \mathbf{\Omega}, E, t)\,. \tag{2.8}$$

This definition is the natural choice for computing reaction rates. If σ represents the probability of a neutron inducing a reaction per unit length traveled, $\sigma\psi\,\mathrm{d}\mathbf{x}\,\mathrm{d}\Omega\,\mathrm{d}E\,\mathrm{d}t$ represents the expected number of reactions induced by neutrons traveling in the phase space element $\mathrm{d}\mathbf{x}\,\mathrm{d}\Omega\,\mathrm{d}E$ in the time $\mathrm{d}t$. The reaction rate is computed by integrating $\sigma\psi$ over the domain of the reaction, $\mathcal{D}$, all angles on the unit sphere, $\mathbb{S}^2$, and all energies $E \in [0, \infty)$. Thus, the number of reactions per unit time is computed as

$$\int_{\mathcal{D}} \int_{\mathbb{S}^2} \int_0^\infty \sigma\psi \,\mathrm{d}E\,\mathrm{d}\Omega\,\mathrm{d}\mathbf{x}\,. \tag{2.9}$$

Radiation transport in astrophysics and HEDP is often concerned with the energy transfer and deposition associated with thermal photons. The Boltzmann equation is then typically cast in terms of the *intensity*, I. The intensity is defined with power units, energy per unit time, but is also expressed per unit area, per unit sold angle, and per unit frequency bandwidth. It is defined as

$$I(\mathbf{x}, \mathbf{\Omega}, \nu, t) \equiv ch\nu f(\mathbf{x}, \mathbf{\Omega}, \nu, t)\,, \tag{2.10}$$

where c is the speed of light. Since photons travel at the speed of light, the particle speed is $v \equiv c$. Due to this, the intensity can be thought of as the energy-track length rate density in the phase space. Thus, if σ_a represents the probability per unit length of a material absorbing a photon, the product $\sigma_a I$ represents the rate energy is deposited by photons at each position, direction, frequency, and time. The power deposited in the material is computed with

$$\int_{\mathcal{D}} \int_{\mathbb{S}^2} \int_0^\infty \sigma_a I \,\mathrm{d}\nu\,\mathrm{d}\Omega\,\mathrm{d}\mathbf{x}\,. \tag{2.11}$$

A photon's energy is related to the magnitude of its momentum, p, through $E = pc$. A photon traveling in direction $\mathbf{\Omega}$ then has momentum

$$\boldsymbol{p} = \mathbf{\Omega}\frac{E}{c} = \mathbf{\Omega}\frac{h\nu}{c}\,. \tag{2.12}$$

We then have that $\frac{\sigma_a}{c}\mathbf{\Omega} I$ represents the rate of momentum deposition in the material due to photon absorption at any point in the phase space. The rate of momentum deposited into the material by radiation traveling in all directions and all frequencies is computed with:

$$\frac{1}{c}\int_{\mathcal{D}}\int_{\mathbb{S}^2}\int_0^\infty \sigma_a\,\mathbf{\Omega} I\,\mathrm{d}\nu\,\mathrm{d}\Omega\,\mathrm{d}\mathbf{x}\,. \tag{2.13}$$

These examples show that the choice of $\psi \equiv vf$ and $I \equiv ch\nu f$ are purely notational conveniences designed to aid in the computation of each field's quantities of interest. In particular, the intensity and angular flux are informally related through

$$\psi = \frac{I}{h\nu} \quad \text{or} \quad I = E\psi\,. \tag{2.14}$$

Here, the informality arises from the angular flux depending on the particle energy and the intensity depending on the particle frequency. The intent of providing Eq. 2.14 is to give a simple way to convert between the notations used in nuclear engineering, astrophysics, and HEDP. McClarren [61, §2.1.5] provides further commentary on these notational differences.

2.3.2 Moments of the Distribution Function

Integration over the velocity variable plays an important role in both neutron and photon transport. In the above, we integrated over position-velocity space to determine the reaction rate and the energy and momentum deposition rates. These integrated quantities are called the moments of the distribution function. For most purposes, the direction particles are traveling is immaterial for computing reaction and deposition rates. Thus, defining variables that represent integrations of the angular flux over the direction of travel can simplify the above calculations. Let,

$$\phi(\mathbf{x}, E, t) = \int_{\mathbb{S}^2} \psi(\mathbf{x}, \mathbf{\Omega}, E, t)\,\mathrm{d}\Omega\,, \tag{2.15}$$

be the *scalar flux*. The reaction rate is then $\int_{\mathcal{D}}\int_0^\infty \sigma\phi(\mathbf{x}, E, t)\,\mathrm{d}E\,\mathrm{d}\mathbf{x}$. In this way, the scalar flux represents the track length rate density of neutrons at energy E traveling in any direction. We will also use the *current* defined as:

$$\boldsymbol{J}(\mathbf{x}, E, t) = \int_{\mathbb{S}^2} \mathbf{\Omega}\,\psi(\mathbf{x}, \mathbf{\Omega}, E, t)\,\mathrm{d}\Omega\,. \tag{2.16}$$

Letting $\mathbf{n}$ denote a unit vector normal to a differential area $\mathrm{d}A$, $\mathbf{\Omega}\cdot\mathbf{n}\,\psi(\mathbf{x}, \mathbf{\Omega}, E, t)\,\mathrm{d}A$ represents the rate at which particles cross $\mathrm{d}A$ going in the direction $\mathbf{\Omega}$ at energy E. Thus, the current represents the net number of particles crossing $\mathrm{d}A$ regardless of $\mathbf{\Omega}$. We call these terms the zeroth and first moments of the angular flux due to their definition as $\int_{\mathbb{S}^2} \mathbf{\Omega}^i\,\psi\,\mathrm{d}\Omega$ where $i = 0$ for the scalar flux and $i = 1$ for the current.

For photons, the astrophysics and HEDP notations differ in their definitions of the moments. In astrophysics, the zeroth, first, and second moments are defined as averages over solid angle:

$$J(\mathbf{x},\nu,t) = \frac{1}{4\pi}\int_{\mathbb{S}^2} I(\mathbf{x},\boldsymbol{\Omega},\nu,t)\,\mathrm{d}\Omega\,, \tag{2.17a}$$

$$\boldsymbol{H}(\mathbf{x},\nu,t) = \frac{1}{4\pi}\int_{\mathbb{S}^2} \boldsymbol{\Omega}\, I(\mathbf{x},\boldsymbol{\Omega},\nu,t)\,\mathrm{d}\Omega\,, \tag{2.17b}$$

$$\mathbf{K}(\mathbf{x},\nu,t) = \frac{1}{4\pi}\int_{\mathbb{S}^2} \boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\, I(\mathbf{x},\boldsymbol{\Omega},\nu,t)\,\mathrm{d}\Omega\,. \tag{2.17c}$$

Alternatively, the HEDP community often uses

$$E(\mathbf{x},\nu,t) = \frac{1}{c}\int_{\mathbb{S}^2} I(\mathbf{x},\boldsymbol{\Omega},\nu,t)\,\mathrm{d}\Omega\,, \tag{2.18a}$$

$$\boldsymbol{F}(\mathbf{x},\nu,t) = \int_{\mathbb{S}^2} \boldsymbol{\Omega}\, I(\mathbf{x},\boldsymbol{\Omega},\nu,t)\,\mathrm{d}\Omega\,, \tag{2.18b}$$

$$\mathbf{P}(\mathbf{x},\nu,t) = \frac{1}{c}\int_{\mathbb{S}^2} \boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\, I(\mathbf{x},\boldsymbol{\Omega},\nu,t)\,\mathrm{d}\Omega\,. \tag{2.18c}$$

These moments are referred to as the energy density, flux, and pressure, respectively. Note that the flux does not have the $1/c$ factor that the energy density and pressure have. The astrophysical notation of J, $\boldsymbol{H}$, and $\mathbf{K}$ simplifies the radiation transport equation by removing factors of 4π and c at the expense of introducing those factors into the radiation-hydrodynamics equations [1]. We elect to use the HEDP notation of E, $\mathbf{F}$, and $\mathbf{P}$ when discussing photon transport and the nuclear engineering notation of ϕ and $\boldsymbol{J}$ when discussing neutron transport.

2.4 The Equations of Thermal Radiative Transfer

For thermal photons we consider the linear Boltzmann transport equation given by Eq. 2.5 cast in terms of the intensity. Multiplying by $ch\nu$, Eq. 2.5 becomes

$$\frac{\partial I}{\partial t} + (\boldsymbol{\Omega}c)\cdot\nabla I = \left(\frac{\partial I}{\partial t}\right)_{\text{collision}} + \left(\frac{\partial I}{\partial t}\right)_{\text{source}}\,, \tag{2.19}$$

where we have used the photon velocity $\mathbf{v} = \boldsymbol{\Omega}c$ and that $I = ch\nu f$. Dividing by c gives the standard form of the transport equation

$$\frac{1}{c}\frac{\partial I}{\partial t} + \boldsymbol{\Omega}\cdot\nabla I = \frac{1}{c}\left(\frac{\partial I}{\partial t}\right)_{\text{collision}} + \frac{1}{c}\left(\frac{\partial I}{\partial t}\right)_{\text{source}}\,. \tag{2.20}$$

We model scattering and absorption events with the background media by setting:

$$\frac{1}{c}\left(\frac{\partial I}{\partial t}\right)_{\text{collision}} = \int_{\mathbb{S}^2}\int_0^\infty \sigma_s(\boldsymbol{\Omega}'\to\boldsymbol{\Omega},\nu'\to\nu) I(\mathbf{x},\boldsymbol{\Omega}',\nu',t)\,\mathrm{d}\nu'\,\mathrm{d}\Omega' - \sigma_t I(\mathbf{x},\boldsymbol{\Omega},\nu,t)\,, \tag{2.21}$$

where $\sigma_s(\mathbf{\Omega}' \to \mathbf{\Omega}, \nu' \to \nu)$ is the differential scattering opacity. We define

$$\sigma_s(\nu) = \int_{\mathbb{S}^2} \int_0^\infty \sigma_s(\mathbf{\Omega}' \to \mathbf{\Omega}, \nu' \to \nu) \,\mathrm{d}\nu' \,\mathrm{d}\Omega' \tag{2.22}$$

as the scattering opacity. The total opacity is given by $\sigma_t(\nu) = \sigma_a(\nu) + \sigma_s(\nu)$ where $\sigma_a(\nu)$ is the absorption opacity. In general, all of these opacities also depend on the material's temperature. The source term is used to model the emission of thermal photons by the material. This is modeled by Planck's black body source:

$$\frac{1}{c}\left(\frac{\partial I}{\partial t}\right)_{\text{source}} = \frac{\sigma_a(\nu) B(\nu, T)}{4\pi}\,, \tag{2.23}$$

where

$$B(\nu, T) = \frac{2h\nu^3}{c^2} \frac{1}{e^{h\nu/kT} - 1} \tag{2.24}$$

is the Planck emission function with k Boltzmann's constant and T the temperature of the material. Note that the above source is only valid under the assumption of local thermodynamic equilibrium. Thus, the Boltzmann transport equation for thermal photons is given by

$$\frac{1}{c}\frac{\partial I}{\partial t} + \mathbf{\Omega} \cdot \nabla I + \sigma_t I = \int_{\mathbb{S}^2} \int_0^\infty \sigma_s(\mathbf{\Omega}' \to \mathbf{\Omega}, \nu' \to \nu) I(\cdot, \mathbf{\Omega}', \nu', \cdot) \,\mathrm{d}\nu' \,\mathrm{d}\Omega' + \frac{\sigma_a B}{4\pi}\,. \tag{2.25}$$

Converting the boundary condition in Eq. 2.6 to apply to the intensity by multiplying by $ch\nu$ and swapping the velocity for the direction of flight, the boundary conditions for the transport equation for thermal photons are given by:

$$I(\mathbf{x}, \mathbf{\Omega}, \nu, t) = \bar{I}(\mathbf{x}, \mathbf{\Omega}, \nu, t)\,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \mathbf{\Omega} \cdot \mathbf{n} < 0\,, \tag{2.26}$$

where $\bar{I}$ is the inflow boundary condition for the intensity. The initial condition supplies $I(\mathbf{x}, \mathbf{\Omega}, \nu, 0)$.

We now wish to derive an equation for the evolution of the material temperature. We model the material gaining energy by absorbing radiation and losing energy by emitting thermal radiation. Here, we are interested in the energy exchanged between matter and radiation traveling in all directions and at any frequency. The energy absorption rate is

$$c \int_0^\infty \sigma_a(\nu) E(\mathbf{x}, \nu, t) \,\mathrm{d}\nu\,. \tag{2.27}$$

The emission rate is given by the Planck source, $B(\nu, T)$. The balance of energy between radiation and matter is then written:

$$\begin{aligned} C_v(T)\frac{\partial T}{\partial t} &= \int_{\mathbb{S}^2} \int_0^\infty \sigma_a(\nu) \left(I(\mathbf{x}, \mathbf{\Omega}, \nu, t) - \frac{B(\nu, T)}{4\pi} \right) \mathrm{d}\nu \,\mathrm{d}\Omega \\ &= \int_0^\infty \sigma_a(\nu) (cE(\mathbf{x}, \nu, t) - B(\nu, T)) \,\mathrm{d}\nu\,, \end{aligned} \tag{2.28}$$

where $C_v(T)$ is the material's heat capacity. An initial temperature field must be provided. Boundary conditions for the temperature field are not required since the energy balance equation does not have spatial derivatives. The coupled equations given by Eqs. 2.25 and 2.28 are called the equations of thermal radiative transfer. This system is nonlinear due to the Planck emission term.

2.5 Steady-State, Linear Transport: A Proxy for TRT

In this section, the model problem of the steady-state, frequency-independent, linear Boltzmann equation with isotropic scattering is derived from the TRT equations.

2.5.1 Isotropic Scattering

First, we assume that scattering is isotropic. In other words, any outgoing direction of flight is equally likely for a photon leaving a scattering event. The differential scattering opacity then simplifies to

$$\sigma_s(\mathbf{\Omega}' \to \mathbf{\Omega}, \nu' \to \nu) \to \frac{1}{4\pi}\sigma_s(\nu' \to \nu)\,. \tag{2.29}$$

Due to this, the scattering source becomes:

$$\int_{\mathbb{S}^2}\int_0^\infty \sigma_s(\mathbf{\Omega}' \to \mathbf{\Omega}, \nu' \to \nu) I(\cdot, \mathbf{\Omega}', \nu', \cdot)\,\mathrm{d}\nu'\,\mathrm{d}\Omega' \to \frac{c}{4\pi}\int_0^\infty \sigma_s(\nu' \to \nu) E(\mathbf{x}, \nu', t)\,\mathrm{d}\nu'\,, \tag{2.30}$$

where the definition of the frequency-dependent energy density given in Eq. 2.18a was used. Isotropic scattering is rarely the correct model for scattering. However, it shares many of the same implementational aspects of more complicated scattering models that have been expanded in spherical harmonics.

2.5.2 Gray Transport

The gray TRT system is derived by defining suitable frequency-averaged opacities and integrating the transport equation over all angles. The gray opacities are defined as weighted averages of the form:

$$\underline{\sigma}_x = \frac{\int_0^\infty \sigma_x\, w(\mathbf{x}, \mathbf{\Omega}, \nu, t)\,\mathrm{d}\nu}{\int w(\mathbf{x}, \mathbf{\Omega}, \nu, t)\,\mathrm{d}\nu}\,, \tag{2.31}$$

where x is the total, scattering, and absorption events and w is a weight function that approximates the true frequency-dependent intensity. In practice, w could be an approximation to the intensity computed from a larger algorithm with multiple levels in frequency, an analytical approximation to the intensity (e.g. Rosseland opacities) [8], or through another numerical approximation such as with a stochastic method.

The gray emission rate is the average of the Planck emission source over all directions and frequencies:

$$\underline{B}(T) = \int_{\mathbb{S}^2}\int_0^\infty B(\nu, T)\,\mathrm{d}\nu\,\mathrm{d}\Omega = acT^4\,, \tag{2.32}$$

where $a = 8\pi^5 k^4/15h^3c^3$. Letting $\underline{I}(\mathbf{x}, \boldsymbol{\Omega}, t) = \int_0^\infty I(\mathbf{x}, \boldsymbol{\Omega}, \nu, t)\,\mathrm{d}\nu$ and $\underline{E}(\mathbf{x}, t) = \int_0^\infty E(\mathbf{x}, \nu, t)\,\mathrm{d}\nu$ be the frequency-averaged intensity and energy density, respectively, the gray TRT system is:

$$\frac{1}{c}\frac{\partial \underline{I}}{\partial t} + \boldsymbol{\Omega}\cdot\nabla\underline{I} + \underline{\sigma}_t\underline{I} = \frac{c\underline{\sigma}_s}{4\pi}\underline{E} + \frac{\underline{\sigma}_a cT^4}{4\pi}\,, \tag{2.33a}$$

$$\underline{I}(\mathbf{x}, \boldsymbol{\Omega}, t) = \bar{\underline{I}}(\mathbf{x}, \boldsymbol{\Omega}, t)\,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega}\cdot\mathbf{n} < 0\,, \tag{2.33b}$$

$$C_v(T)\frac{\partial T}{\partial t} = c\underline{\sigma}_a(\underline{E} - aT^4)\,, \tag{2.33c}$$

where $\bar{\underline{I}}(\mathbf{x}, \boldsymbol{\Omega}, t) = \int_0^\infty \bar{I}(\mathbf{x}, \boldsymbol{\Omega}, \nu, t)\,\mathrm{d}\nu$ is the gray inflow boundary function. The gray TRT equations are often used to iteratively accelerate the convergence of frequency-dependent simulations. In this way, algorithms for the frequency-dependent TRT system must also be effective for the gray TRT system. The gray TRT system then serves as an important first step in the development of frequency-dependent TRT methods.

2.5.3 Steady State

Time derivatives are neglected under the assumption of steady state. The gray TRT system with isotropic scattering simplifies to

$$\boldsymbol{\Omega}\cdot\nabla\underline{I} + \underline{\sigma}_t\underline{I} = \frac{c\underline{\sigma}_s}{4\pi}\underline{E} + \frac{\underline{\sigma}_a acT^4}{4\pi}\,, \tag{2.34a}$$

$$\underline{I}(\mathbf{x}, \boldsymbol{\Omega}) = \bar{\underline{I}}(\mathbf{x}, \boldsymbol{\Omega})\,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega}\cdot\mathbf{n} < 0\,, \tag{2.34b}$$

$$0 = c\underline{\sigma}_a(\underline{E} - aT^4)\,. \tag{2.34c}$$

In this case, the material temperature is determined by the radiation field such that:

$$T^4 = \frac{\underline{E}}{a}\,. \tag{2.35}$$

We are then left with:

$$\boldsymbol{\Omega}\cdot\nabla\underline{I} + \underline{\sigma}_t\underline{I} = \frac{c\underline{\sigma}_s + c\underline{\sigma}_a}{4\pi}\underline{E}\,, \tag{2.36}$$

along with the gray boundary condition in Eq. 2.34b. This equation models the radiation field at long time scales where the radiation and material have achieved equilibrium. When implicit time integration schemes are applied to the time-dependent TRT system, a steady-state system analogous to Eq. 2.36 with additional sources must be solved at each time step. Thus, the steady-state, frequency-independent TRT system with isotropic scattering represents the core kernel of the full TRT system.

2.5.4 Definition of the Model Problem

We now define the transport problem that serves as the model for the development of the moment methods presented in this dissertation. The properties of the steady-state, gray TRT system with isotropic scattering can be emulated with the following neutron transport equation:

$$\mathbf{\Omega} \cdot \nabla \psi + \sigma_t \psi = \frac{\sigma_s}{4\pi} \int \psi \, \mathrm{d}\Omega' + q \,, \tag{2.37a}$$

$$\psi(\mathbf{x}, \mathbf{\Omega}) = \bar{\psi}(\mathbf{x}, \mathbf{\Omega}) \,, \quad \mathbf{x} \in \partial \mathcal{D} \text{ and } \mathbf{\Omega} \cdot \mathbf{n} < 0 \,, \tag{2.37b}$$

where $\psi(\mathbf{x}, \mathbf{\Omega})$ represents the energy and time-independent angular flux, $\sigma_t(\mathbf{x})$ and $\sigma_s(\mathbf{x})$ energy-independent cross sections, $q(\mathbf{x}, \mathbf{\Omega})$ a fixed-source of particles, and $\bar{\psi}(\mathbf{x}, \mathbf{\Omega})$ the inflow boundary function for the angular flux. In this document, we solve Eq. 2.37 and use the nuclear engineering notation for the moments. That is, we use

$$\phi(\mathbf{x}) = \int_{\mathbb{S}^2} \psi(\mathbf{x}, \mathbf{\Omega}) \, \mathrm{d}\Omega \,, \tag{2.38}$$

$$\boldsymbol{J}(\mathbf{x}) = \int_{\mathbb{S}^2} \mathbf{\Omega} \, \psi(\mathbf{x}, \mathbf{\Omega}) \, \mathrm{d}\Omega \,, \tag{2.39}$$

$$\mathbf{P}(\mathbf{x}) = \int_{\mathbb{S}^2} \mathbf{\Omega} \otimes \mathbf{\Omega} \, \psi(\mathbf{x}, \mathbf{\Omega}) \, \mathrm{d}\Omega \,. \tag{2.40}$$

2.6 The Radiation Diffusion Approximation

Here, we apply a simplification to the angular dependence in our model transport problem to derive the radiation diffusion approximation. We wish to solve

$$\mathbf{\Omega} \cdot \nabla \psi + \sigma_t \psi = \frac{\sigma_s}{4\pi} \int \psi \, \mathrm{d}\Omega' + \frac{Q}{4\pi} \,, \tag{2.41a}$$

$$\psi(\mathbf{x}, \mathbf{\Omega}) = \bar{\psi}(\mathbf{x}, \mathbf{\Omega}) \,, \quad \mathbf{x} \in \partial \mathcal{D} \text{ and } \mathbf{\Omega} \cdot \mathbf{n} < 0 \,, \tag{2.41b}$$

under the assumption that the angular flux is linearly anisotropic in angle. In other words, the angular flux can be written:

$$\psi(\mathbf{x}, \mathbf{\Omega}) = \frac{1}{4\pi} (\phi(\mathbf{x}) + 3 \mathbf{\Omega} \cdot \boldsymbol{J}(\mathbf{x}) \,. \tag{2.42}$$

We will use the following angular identities:

$$\int_{\mathbb{S}^2} \mathrm{d}\Omega = 4\pi \,, \tag{2.43a}$$

$$\int_{\mathbb{S}^2} \mathbf{\Omega} \otimes \mathbf{\Omega} \, \mathrm{d}\Omega = \frac{4\pi}{3} \mathbf{I} \,, \tag{2.43b}$$

and that integrating any function that is odd in angle is zero. Integrating Eq. 2.41 over all angles yields the zeroth moment equation:

$$\nabla \cdot \boldsymbol{J} + \sigma_t \phi = \sigma_s \phi + Q \,. \tag{2.44}$$

Multiplying by $\mathbf{\Omega}$ and integrating over all angles yields the first moment equation:

$$\int_{\mathbb{S}^2} \mathbf{\Omega} \otimes \mathbf{\Omega} \cdot \nabla \psi \, \mathrm{d}\Omega + \sigma_t \boldsymbol{J} = 0 \,, \tag{2.45}$$

where the source Q has been assumed to be isotropic. We now use that ψ is linearly anisotropic to simplify the first term:

$$\begin{aligned} \int_{\mathbb{S}^2} \mathbf{\Omega} \otimes \mathbf{\Omega} \cdot \nabla \psi \, \mathrm{d}\Omega &= \nabla \cdot \int_{\mathbb{S}^2} \mathbf{\Omega} \otimes \mathbf{\Omega} \, \psi \, \mathrm{d}\Omega \\ &= \nabla \cdot \int_{\mathbb{S}^2} \mathbf{\Omega} \otimes \mathbf{\Omega} \frac{1}{4\pi} \left(\phi + 3 \mathbf{\Omega} \cdot \boldsymbol{J} \right) \mathrm{d}\Omega \\ &= \nabla \cdot \frac{\phi}{3} \mathbf{I} \\ &= \frac{1}{3} \nabla \phi \,. \end{aligned} \tag{2.46}$$

All together, the radiation diffusion system is

$$\nabla \cdot \boldsymbol{J} + \sigma_a \phi = Q \,, \tag{2.47a}$$

$$\frac{1}{3} \nabla \phi + \sigma_t \boldsymbol{J} = 0 \,, \tag{2.47b}$$

where $\sigma_a = \sigma_t - \sigma_s$ was used. By eliminating the current, the second-order form of radiation diffusion is:

$$- \nabla \cdot \frac{1}{3\sigma_t} \nabla \phi + \sigma_a \phi = Q \,. \tag{2.48}$$

We refer to $D = \frac{1}{3\sigma_t}$ as the diffusion coefficient.

Boundary conditions are derived by manipulating the so-called partial currents. Let

$$J_n^{\pm} = \int_{\mathbf{\Omega} \cdot \mathbf{n} \gtrless 0} \mathbf{\Omega} \cdot \mathbf{n} \, \psi \, \mathrm{d}\Omega \,, \tag{2.49}$$

so that

$$\boldsymbol{J} \cdot \mathbf{n} = \int \mathbf{\Omega} \cdot \mathbf{n} \, \psi \, \mathrm{d}\Omega = J_n^+ + J_n^- \,. \tag{2.50}$$

We add and subtract the incoming partial current, J_n^-, to arrive at

$$\begin{aligned} \boldsymbol{J} \cdot \mathbf{n} &= (J_n^+ - J_n^-) + 2 J_n^- \\ &= \int |\mathbf{\Omega} \cdot \mathbf{n}| \, \psi \, \mathrm{d}\Omega + 2 J_n^- \,. \end{aligned} \tag{2.51}$$

Using the assumption that the angular flux is linearly anisotropic the first term on the right hand side becomes:

$$\int |\boldsymbol{\Omega} \cdot \mathbf{n}|\, \psi \, \mathrm{d}\Omega = \frac{1}{4\pi} \int |\boldsymbol{\Omega} \cdot \mathbf{n}| (\phi + 3\boldsymbol{\Omega} \cdot \boldsymbol{J}) \, \mathrm{d}\Omega = \frac{\phi}{2} \,, \tag{2.52}$$

where we have used the angular identity:

$$\int |\boldsymbol{\Omega} \cdot \mathbf{n}| \, \mathrm{d}\Omega = 2\pi \,, \tag{2.53}$$

and that $\int |\boldsymbol{\Omega} \cdot \mathbf{n}| \, \boldsymbol{\Omega} \, \mathrm{d}\Omega = 0$ since its integrand is an odd function in angle. On the boundary of the domain, the inflow partial current is determined by the inflow angular flux function, $\bar{\psi}$. Defining

$$J_{\mathrm{in}} = \int_{\boldsymbol{\Omega} \cdot \mathbf{n} < 0} \boldsymbol{\Omega} \cdot \mathbf{n} \, \bar{\psi} \, \mathrm{d}\Omega \tag{2.54}$$

the radiation diffusion boundary conditions are:

$$\boldsymbol{J} \cdot \mathbf{n} = \frac{\phi}{2} + 2J_{\mathrm{in}} \,. \tag{2.55}$$

This boundary condition is referred to as the Marshak boundary condition.

Chapter 3

Moment Methods for Radiation Transport

This chapter discusses the VEF and SMM algorithms as applied to the continuous model transport problem:

$$\mathbf{\Omega} \cdot \nabla\psi + \sigma_t\psi = \frac{\sigma_s}{4\pi}\int \psi \, \mathrm{d}\Omega' + q \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{3.1a}$$

$$\psi(\mathbf{x}, \mathbf{\Omega}) = \bar{\psi}(\mathbf{x}, \mathbf{\Omega}) \,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \mathbf{\Omega} \cdot \mathbf{n} < 0 \,, \tag{3.1b}$$

where $\psi(\mathbf{x}, \mathbf{\Omega})$ is the angular flux, $\mathcal{D}$ the domain of the problem with $\partial\mathcal{D}$ its boundary, $\sigma_t(\mathbf{x})$ and $\sigma_s(\mathbf{x})$ the total and scattering macroscopic cross sections, respectively, $q(\mathbf{x}, \mathbf{\Omega})$ the fixed-source, and $\bar{\psi}(\mathbf{x}, \mathbf{\Omega})$ the inflow boundary function. The moment system is formed by taking the zeroth and first angular moments of the transport equation. Due to the streaming term, $\mathbf{\Omega} \cdot \nabla\psi$, angular moments always produce more unknowns than equations. The VEF and SMM moment systems are formulated by defining additional algebraic equations, called closures, which define the second moment of the transport solution in terms of the zeroth moment, closing the moment system. The SMM and VEF methods are differentiated by their choice of closure: VEF uses a multiplicative, nonlinear closure while SMM uses an additive, linear closure. In both cases, the closures are exact such that the moment system is an equivalent reformulation of the transport equation and trivial in that the transport solution must already be known in order to define the closures.

Moment methods use iterative schemes to solve the coupled transport-moment system simultaneously. An efficient algorithm is found by using the moment system with lagged closures to compute the expensive and slow to converge scattering physics. Rapid convergence is achieved due to the fact that the VEF and SMM closures are weak functions of the transport solution.

Here, we derive the moment systems for the VEF and second moment methods and define the iterative schemes used to solve the coupled transport-moment system. We discuss the mathematical properties of the VEF and SMM closures and their moment systems. The connection between the SMM and VEF closures is established. The chapter concludes with a discussion of the two primary philosophies used to design discrete moment methods.

3.1 Derivation of Moment Systems

3.1.1 The Moment Equations

Integrating the transport equation over all angles yields the zeroth moment:

$$\nabla \cdot \boldsymbol{J} + \sigma_a \varphi = Q_0 \,, \tag{3.2}$$

where φ and $\boldsymbol{J}$ are the zeroth and first angular moments of the angular flux, respectively, Q_0 the zeroth moment of the fixed-source, q, and $\sigma_a(\mathbf{x}) = \sigma_t(\mathbf{x}) - \sigma_s(\mathbf{x})$ the absorption cross section. We have assumed Cartesian geometry so that

$$\int \boldsymbol{\Omega} \cdot \nabla \psi \, \mathrm{d}\Omega = \int \nabla \cdot \boldsymbol{\Omega} \, \psi \, \mathrm{d}\Omega = \nabla \cdot \boldsymbol{J} \,. \tag{3.3}$$

The first moment is found by multiplying the transport equation by $\boldsymbol{\Omega}$ and integrating over angle:

$$\nabla \cdot \mathbf{P} + \sigma_t \boldsymbol{J} = \boldsymbol{Q}_1 \,, \tag{3.4}$$

where $\mathbf{P} = \int \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \, \psi \, \mathrm{d}\Omega$ is the second moment of the angular flux and $\boldsymbol{Q}_1$ is the first angular moment of q. We refer to the first three moments of the angular flux as the scalar flux, current, and pressure, respectively.

Boundary conditions for the moment system are derived by manipulating partial currents. Letting $J_n^\pm = \int_{\boldsymbol{\Omega}\cdot\mathbf{n}\gtrless 0} \boldsymbol{\Omega} \cdot \mathbf{n} \, \psi \, \mathrm{d}\Omega$ where $\mathbf{n}$ is the outward unit normal on the boundary of the domain, consider

$$\begin{aligned} \boldsymbol{J} \cdot \mathbf{n} &= J_n^- + J_n^+ \\ &= 2J_n^- + (J_n^+ - J_n^-) \\ &= 2J_n^- + \int |\boldsymbol{\Omega} \cdot \mathbf{n}| \, \psi \, \mathrm{d}\Omega \,. \end{aligned} \tag{3.5}$$

Defining

$$B(\psi) = \int |\boldsymbol{\Omega} \cdot \mathbf{n}| \, \psi \, \mathrm{d}\Omega \,, \tag{3.6}$$

the boundary conditions for the moment system are:

$$\boldsymbol{J} \cdot \mathbf{n} = B(\psi) + 2J_{\mathrm{in}} \,, \tag{3.7}$$

where $J_{\mathrm{in}} = \int_{\boldsymbol{\Omega}\cdot\mathbf{n}<0} \boldsymbol{\Omega} \cdot \mathbf{n} \, \bar{\psi} \, \mathrm{d}\Omega$ is the incoming partial current computed from the inflow boundary function, $\bar{\psi}$.

The moment equations are given by:

$$\nabla \cdot \boldsymbol{J} + \sigma_a \varphi = Q_0 \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{3.8a}$$

$$\nabla \cdot \mathbf{P} + \sigma_t \boldsymbol{J} = \boldsymbol{Q}_1 \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{3.8b}$$

$$\boldsymbol{J} \cdot \mathbf{n} = B + J_{\text{in}} \,, \quad \mathbf{x} \in \partial\mathcal{D} \,. \tag{3.8c}$$

In three dimensions, the moment system has 10 unknowns corresponding to the scalar flux, three components of the current, and the six unique components of the symmetric pressure tensor but only four equations arising from the scalar zeroth moment and vector first moment equations. On the boundary of the domain, we have one equation but two unknowns corresponding to the normal component of the current and the boundary functional, $B(\psi)$. Closures provide additional algebraic equations which define $\mathbf{P}$ and B in terms of the lower moments. For both VEF and SMM, the closures are formulated in terms of the scalar flux and angular flux-dependent functionals. If the angular flux were known, the closed moment system defines the zeroth and first moment of the angular flux. In other words, the moment system is an equivalent reformulation of the transport equation.

Note that for an independent moment method the discretized solution of the moment system will not be equivalent to the moments of the discrete angular flux; they will differ on the order of the spatial discretization error. To notationally separate the two scalar flux solutions, we use φ to denote the moment system's scalar flux and ϕ for the zeroth moment of the angular flux.

3.1.2 VEF Closures

VEF uses multiplicative, nonlinear closures derived by multiplying and dividing by the scalar flux. For the pressure, VEF uses

$$\mathbf{P} = \mathbf{E}\varphi \,, \tag{3.9}$$

where

$$\mathbf{E} = \frac{\int \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \, \psi \, \mathrm{d}\Omega}{\int \psi \, \mathrm{d}\Omega} \tag{3.10}$$

is the Eddington tensor. The boundary functional is closed in an analogous manner:

$$B = E_b \varphi \,, \tag{3.11}$$

with

$$E_b = \frac{\int |\boldsymbol{\Omega} \cdot \mathbf{n}| \, \psi \, \mathrm{d}\Omega}{\int \psi \, \mathrm{d}\Omega} \tag{3.12}$$

the Eddington boundary factor. Since, for the continuous equations, $\varphi / \int \psi \, \mathrm{d}\Omega = 1$, these closures are simply algebraic reformulations of the pressure tensor and boundary functional. Note that due to the normalization of the Eddington tensor and boundary factor, both $\mathbf{E}$ and E_b are nonlinear functions of the angular flux.

With these closures, the VEF equations are

$$\nabla \cdot \boldsymbol{J} + \sigma_a \varphi = Q_0 \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{3.13a}$$

$$\nabla \cdot (\mathbf{E}\varphi) + \sigma_t \boldsymbol{J} = \boldsymbol{Q}_1 \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{3.13b}$$

$$\boldsymbol{J} \cdot \mathbf{n} = E_b \varphi + 2J_{\text{in}} \,, \quad \mathbf{x} \in \partial\mathcal{D} \,. \tag{3.13c}$$

By eliminating the current, the VEF equations can be cast as a drift-diffusion equation:

$$-\nabla \cdot \frac{1}{\sigma_t} \nabla \cdot (\mathbf{E}\varphi) + \sigma_a \varphi = Q_0 - \nabla \cdot \frac{\boldsymbol{Q}_1}{\sigma_t} \,. \tag{3.14}$$

In both the first-order form (Eq. 3.13) and the second-order form (Eq. 3.14), the presence of the Eddington tensor inside the divergence leads to diffusion, advection, and reaction-like terms that make applying existing discretization techniques difficult.

3.1.3 SMM Closures

The SMM moment system is formed using additive closures. The pressure is closed with:

$$\mathbf{P} = \mathbf{T}(\psi) + \frac{1}{3}\mathbf{I}\varphi \tag{3.15}$$

where $\mathbf{T}(\psi)$ is a *correction tensor* defined as:

$$\mathbf{T}(\psi) = \int \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \, \psi \, \mathrm{d}\Omega - \frac{1}{3}\mathbf{I} \int \psi \, \mathrm{d}\Omega \,. \tag{3.16}$$

Note that this is simply an algebraic reformulation of the second moment $\mathbf{P} = \int \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} \, \psi \, \mathrm{d}\Omega$ where an isotropic pressure tensor proportional to the zeroth moment is added and subtracted. That is, in the same way that VEF multiplies and divides by the zeroth moment, SMM adds and subtracts. Like the VEF closure, the SMM closure is trivial in that the solution to the transport equation must already be known in order to define the correction tensor.

For the boundary conditions, let

$$\beta(\psi) = \int |\boldsymbol{\Omega} \cdot \mathbf{n}| \, \psi \, \mathrm{d}\Omega - \frac{1}{2} \int \psi \, \mathrm{d}\Omega \tag{3.17}$$

be the boundary correction factor. The boundary functional is closed using $B = \beta + \frac{1}{2}\varphi$ so that the SMM boundary conditions are:

$$\boldsymbol{J} \cdot \mathbf{n} = \frac{1}{2}\varphi + 2J_{\text{in}} + \beta(\psi) \,. \tag{3.18}$$

The factors of one third and one half used in the closures of $\mathbf{P}$ and B, respectively, are chosen so that the SMM moment system is equivalent to radiation diffusion when the angular flux is linearly anisotropic.

With these closures, the SMM equations are

$$\nabla \cdot \boldsymbol{J} + \sigma_a \varphi = Q_0 \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{3.19a}$$

$$\frac{1}{3}\nabla\varphi + \sigma_t \boldsymbol{J} = \boldsymbol{Q}_1 - \nabla\cdot\mathbf{T}\,, \quad \mathbf{x}\in\mathcal{D}\,, \tag{3.19b}$$

$$\boldsymbol{J}\cdot\mathbf{n} = \frac{1}{2}\varphi + 2J_{\text{in}} + \beta\,, \quad \mathbf{x}\in\partial\mathcal{D}\,. \tag{3.19c}$$

The second-order form is found by eliminating the current:

$$-\nabla\cdot\frac{1}{3\sigma_t}\nabla\varphi + \sigma_a\varphi = Q_0 - \nabla\cdot\frac{\boldsymbol{Q}_1}{\sigma_t} + \nabla\cdot\frac{1}{\sigma_t}\nabla\cdot\mathbf{T}\,. \tag{3.20}$$

Observe that the SMM moment system is the radiation diffusion system with additional transport-dependent source terms. Likewise, the boundary condition is the Marshak boundary condition with an additional transport-dependent boundary source.

3.2 The Moment Algorithm

Moment methods solve the coupled transport-moment system simultaneously. The transport equation is used to provide the VEF or SMM closures while the moment system is used to compute the moment-dependent physics. In our case, the moment system's scalar flux solution is used to compute the isotropic scattering source. In this way, the coupling of the angular phase space induced by integrating over all angles is avoided. This allows use of the efficient solution procedure known as the transport sweep discussed in Chapter 5 for the discrete transport equations. Since the closures are weak functions of the transport solution, simple iterative schemes can converge rapidly and robustly.

We first introduce notation that abstracts away the choice of the closures and casting the moment system in first or second-order form. Let $\mathcal{M}(\psi,\mathbf{X}) = 0$ denote one of the moment systems derived in the previous section with $\mathbf{X}$ the moment system's unknowns. For example, $\mathcal{M}(\psi,\mathbf{X})$ could represent the VEF moment system in first-order form given by Eq. 3.13 where $\mathbf{X}$ would include both the scalar flux and current. In the case of the second-order form, we would set $\mathbf{X} = \varphi$ since the scalar flux is the only unknown. For VEF, $\mathcal{M}(\psi,\mathbf{X})$ is nonlinear in ψ and linear in the moments, $\mathbf{X}$. For SMM, $\mathcal{M}(\psi,\mathbf{X})$ is linear in both arguments.

The moment algorithm solves the coupled system given by:

$$\boldsymbol{\Omega}\cdot\nabla\psi + \sigma_t\psi = \frac{\sigma_s}{4\pi}\varphi + q\,, \tag{3.21a}$$

$$\mathcal{M}(\psi,\mathbf{X}) = 0\,, \tag{3.21b}$$

where transport boundary conditions are specified in Eq. 3.1b. The moment system's boundary conditions are given by Eq. 3.13c for a VEF method and Eq. 3.19c for SMM. Here, the moment system is coupled to the transport equation through the closures and the transport equation's scattering source is coupled to the moment system through the moment system's scalar flux. We have increased the complexity of the problem by adding the moment system's unknowns. In the case of VEF, the coupled system in Eq. 3.21 is also nonlinear due to the use of nonlinear closures. However, solving the coupled system is still advantageous due to the ability to use the transport sweep and the rapid convergence of the closures.

Let

$$\mathbf{L}\psi = \mathbf{\Omega} \cdot \nabla\psi + \sigma_t\psi \tag{3.22}$$

be the streaming and collision operator. The coupled transport-moment system can then be rewritten

$$\mathbf{L}\psi = \frac{\sigma_s}{4\pi}\varphi + q \,, \tag{3.23a}$$

$$\mathcal{M}(\psi, \mathbf{X}) = 0 \,. \tag{3.23b}$$

By linearly eliminating the angular flux, the coupled system is equivalent to:

$$\mathcal{M}\Big(\mathbf{L}^{-1}\Big(\frac{\sigma_s}{4\pi}\varphi + q\Big) \,, \mathbf{X}\Big) = 0 \,. \tag{3.24}$$

Observe that Eq. 3.24 is now a function of the moment solution only. That is, we can define

$$\mathbf{F}(\mathbf{X}) = \mathcal{M}\Big(\mathbf{L}^{-1}\Big(\frac{\sigma_s}{4\pi}\varphi + q\Big) \,, \mathbf{X}\Big) \tag{3.25}$$

and equivalently solve $\mathbf{F}(\mathbf{X}) = 0$. In this reduced problem, the angular flux appears only as an auxiliary variable used to compute the residual $\mathbf{F}(\mathbf{X})$ and we say that the angular flux is enslaved to the moment system. This reduced formulation $\mathbf{F}(\mathbf{X}) = 0$ has much lower dimension than the original coupled system given in Eq. 3.21 but has the same solution. Due to this, advanced solvers for $\mathbf{F}(\mathbf{X})$ can be applied that would otherwise be impractical for Eq. 3.21 due to the storage and computation costs associated with the high-dimensional angular flux.

We now leverage the structure of the VEF and SMM moment systems to further simplify the above algorithm. Let

$$\mathbf{V}(\psi)\mathbf{X} = f \,, \tag{3.26}$$

represent the VEF moment system such that $\mathcal{M}(\psi, \mathbf{X}) = \mathbf{V}(\psi)\mathbf{X} - f$. We then have that

$$\mathbf{F}(\mathbf{X}) = \mathbf{V}\Big(\mathbf{L}^{-1}\Big(\frac{\sigma_s}{4\pi}\varphi + q\Big)\Big)\,\mathbf{X} - f = 0 \,. \tag{3.27}$$

Operating by the inverse of the VEF moment system, the coupled transport-VEF system is equivalent to:

$$\mathbf{X} = \mathbf{V}\Big(\mathbf{L}^{-1}\Big(\frac{\sigma_s}{4\pi}\varphi + q\Big)\Big)^{-1} f \,. \tag{3.28}$$

For SMM, the moment system is of the form

$$\mathbf{D}\mathbf{X} = b(\psi) \,, \tag{3.29}$$

where $\mathbf{D}$ is a diffusion operator and $b(\psi)$ includes the moments of the fixed-source and the transport-dependent correction sources. The root-finding problem $\mathbf{F}(\mathbf{X}) = 0$ is equivalent to

$$\mathbf{X} = \mathbf{D}^{-1} b\Big(\mathbf{L}^{-1}\Big(\frac{\sigma_s}{4\pi}\varphi + q\Big)\Big) \,. \tag{3.30}$$

$$\begin{tikzcd}
\boldsymbol{\Omega}\cdot\nabla\psi + \sigma_t\psi = \frac{\sigma_s}{4\pi}\varphi + q \arrow[r, bend right, "\mathbf{E}(\cdot) = \frac{\int \boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\,(\cdot)\,\mathrm{d}\Omega}{\int(\cdot)\,\mathrm{d}\Omega}"'] & \begin{gathered} \nabla\cdot\boldsymbol{J} + \sigma_a\varphi = Q_0 \\ \nabla\cdot(\mathbf{E}\varphi) + \sigma_t\boldsymbol{J} = \boldsymbol{Q}_1 \end{gathered} \arrow[l, bend right, "\varphi"']
\end{tikzcd}$$

(a)

$$\begin{tikzcd}
\boldsymbol{\Omega}\cdot\nabla\psi + \sigma_t\psi = \frac{\sigma_s}{4\pi}\varphi + q \arrow[r, bend right, "\mathbf{T}(\cdot) = \int \boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\,(\cdot)\,\mathrm{d}\Omega - \frac{1}{3}\int(\cdot)\,\mathrm{d}\Omega"'] & \begin{gathered} \nabla\cdot\boldsymbol{J} + \sigma_a\varphi = Q_0 \\ \tfrac{1}{3}\nabla\varphi + \sigma_t\boldsymbol{J} = \boldsymbol{Q}_1 - \nabla\cdot\mathbf{T} \end{gathered} \arrow[l, bend right, "\varphi"']
\end{tikzcd}$$

(b)

Figure 3.1: A depiction of the iteration scheme used in (a) VEF and (b) SMM algorithms. The transport equation informs the moment system through the closures while the moment system drives the transport equation through computation of the scattering source. By lagging the scattering term, the transport equation can be efficiently inverted. Rapid convergence occurs because the closures are weak functions of the solution.

Thus, for both VEF and SMM, the solution of the coupled transport-moment system is the fixed-point:

$$\mathbf{X} = \mathbf{G}(\mathbf{X})\,, \tag{3.31}$$

where $\mathbf{G}(\mathbf{X})$ is given by

$$\mathbf{G}(\mathbf{X}) = \mathbf{V}\Big(\mathbf{L}^{-1}\Big(\frac{\sigma_s}{4\pi}\varphi + q\Big)\Big)^{-1} f \tag{3.32}$$

for VEF and

$$\mathbf{G}(\mathbf{X}) = \mathbf{D}^{-1} b\Big(\mathbf{L}^{-1}\Big(\frac{\sigma_s}{4\pi}\varphi + q\Big)\Big) \tag{3.33}$$

for SMM. The fixed-point operator $\mathbf{G}$ is applied in two stages: 1) solve the transport equation using a scattering source formed from the moment system's scalar flux and 2) solve the moment system using the closures computed with the angular flux from stage 1). The definitions of the fixed-point operator, $\mathbf{G}$, show the key differences between the VEF and SMM algorithms. VEF has a transport-dependent left hand side operator while the right hand side sources are fixed. On the other hand, SMM has transport-dependent sources but a fixed left hand side operator corresponding to radiation diffusion.

The simplest algorithm to solve $\mathbf{X} = \mathbf{G}(\mathbf{X})$ is fixed-point iteration:

$$\mathbf{X}^{k+1} = \mathbf{G}(\mathbf{X}^k) \tag{3.34}$$

where $\mathbf{X}^0$ is an initial guess. This process is repeated until the difference between successive iterates is small enough. Again, we note that the angular flux has been eliminated and thus appears only implicitly in computing the fixed-point operator, $\mathbf{G}$. The coupling between the transport and moment system is depicted in Fig. 3.1 for the VEF and SMM algorithms. Convergence of the fixed-point iteration is expected to be rapid since the closures are weak functions of the transport solution [19].

3.3 Anderson Acceleration

Iterative efficiency for solving $\mathbf{X} = \mathbf{G}(\mathbf{X})$ can be improved with the use of Anderson acceleration. Anderson acceleration defines the next iterate as the linear combination of the previous m iterates that minimizes the residual $\mathbf{X} - \mathbf{G}(\mathbf{X})$. For the storage cost of m previous iterates, Anderson acceleration increases the convergence rate and improves robustness. While it is not practical to store multiple copies of the angular flux, it is reasonable to expect that a small set of moment-sized vectors can be stored. The process of linearly eliminating the transport equation, codified in Eq. 3.24, allows the Anderson space to be built from the much smaller moment-sized vectors only. In the case where a subset of the angular flux unknowns are not eliminated, such as when a parallel block Jacobi sweep is used to avoid communication costs or when mesh cycles or reentrant faces are present, the solution vector can be augmented with these un-eliminated unknowns so that they are included in the Anderson space. This is the nonlinear analog to the ideas used for Krylov-accelerated source iteration [62].

In addition, the problem $\mathbf{F}(\mathbf{X}) = 0$, with $\mathbf{F}$ defined in Eq. 3.25, can be solved directly with root-finding methods such as Jacobian-free Newton Krylov (JFNK). Such an approach appears advantageous since only the nonlinear residual $\mathbf{F}(\mathbf{X})$ must be computed. This avoids the need to invert the moment system. However, such an approach would require additional preconditioning in order to form a scalable solution method.

Root-finding methods could also be applied to the problem $\mathbf{f}(\mathbf{X}) = \mathbf{X} - \mathbf{G}(\mathbf{X}) = 0$. We observed that JFNK applied to this problem typically required significantly more evaluations of $\mathbf{G}$ than Anderson-accelerated fixed-point iteration. This is because JFNK builds a new Krylov space to approximate the gradient of F *at each iteration* meaning information across iterations is not kept. Since evaluating $\mathbf{G}$ involves inverting the transport equation, this significantly increases the expense of the algorithm. Thus, we present results using fixed-point iteration and Anderson-accelerated fixed-point iteration only.

3.4 Bounds and Asymptotic Limits of the Closures

The Eddington tensor and boundary factor are defined as

$$\mathbf{E} = \frac{\int \mathbf{\Omega} \otimes \mathbf{\Omega}\, \psi \,\mathrm{d}\Omega}{\int \psi \,\mathrm{d}\Omega} \,, \tag{3.35}$$

$$E_b = \frac{\int |\mathbf{\Omega} \cdot \mathbf{n}|\, \psi \,\mathrm{d}\Omega}{\int \psi \,\mathrm{d}\Omega} \,, \tag{3.36}$$

respectively. Observe that the Eddington tensor and boundary factor are ψ-weighted averages of $\mathbf{\Omega}\otimes\mathbf{\Omega}$ and $|\mathbf{\Omega}\cdot\mathbf{n}|$, respectively. This means the VEF data are bounded functions of ψ. Using this property, the Eddington tensor's maximum and minimum occur at the maximum and minimum of $\mathbf{\Omega}\otimes\mathbf{\Omega}$. This can be seen by setting ψ to be a Dirac delta function centered at an extreme value of $\boldsymbol{\Omega}\otimes\boldsymbol{\Omega}$. Due to to this, the Eddington tensor obeys the bounds

$$\mathbf{E}_{ij} \in \begin{cases} [0,1]\,, & i = j \\ [-1/2, 1/2]\,, & i \neq j \end{cases} . \tag{3.37}$$

Likewise, the boundary factor has the extreme values of $|\boldsymbol{\Omega}\cdot\mathbf{n}|$. Thus,

$$E_b \in [0,1]\,. \tag{3.38}$$

In the thick diffusion limit, the angular flux is a linearly anisotropic function in angle. In other words, for some spatially-dependent functions $f(\mathbf{x})$ and $\boldsymbol{g}(\mathbf{x})$, the angular flux is of the form:

$$\psi(\mathbf{x},\boldsymbol{\Omega}) = \frac{1}{4\pi}\left(f(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{g}(\mathbf{x})\right)\,. \tag{3.39}$$

The zeroth and second moments of this linearly anisotropic solution are

$$\int \psi \,\mathrm{d}\Omega = \frac{1}{4\pi}\int f(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{g}(\mathbf{x})\,\mathrm{d}\Omega = f(\mathbf{x})\,, \tag{3.40}$$

$$\int \boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\,\psi\,\mathrm{d}\Omega = \frac{1}{4\pi}\int \boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\left(f(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{g}(\mathbf{x})\right)\mathrm{d}\Omega = \frac{f(\mathbf{x})}{3}\mathbf{I}\,, \tag{3.41}$$

since integrals of odd functions in angle over the unit sphere are zero. Thus, in the thick diffusion limit, the Eddington tensor is

$$\mathbf{E} = \frac{f(\mathbf{x})/3\mathbf{I}}{f(\mathbf{x})} = \frac{1}{3}\mathbf{I}\,. \tag{3.42}$$

For the boundary factor,

$$\int |\boldsymbol{\Omega}\cdot\mathbf{n}|\,\psi\,\mathrm{d}\Omega = \frac{1}{4\pi}\int |\boldsymbol{\Omega}\cdot\mathbf{n}|\left(f(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{g}(\mathbf{x})\right)\mathrm{d}\Omega = \frac{f(\mathbf{x})}{2}\,, \tag{3.43}$$

and thus

$$E_b = \frac{f(\mathbf{x})/2}{f(\mathbf{x})} = \frac{1}{2} \tag{3.44}$$

in the thick diffusion limit. With these asymptotic values, the VEF drift-diffusion equation, given by Eq. 3.14, simplifies to

$$-\nabla\cdot\frac{1}{3\sigma_t}\nabla\varphi + \sigma_a\varphi = Q_0 - \nabla\cdot\frac{\boldsymbol{Q}_1}{\sigma_t}\,, \quad \mathbf{x}\in\mathcal{D}\,, \tag{3.45a}$$

$$\boldsymbol{J}\cdot\mathbf{n} = \frac{1}{2}\varphi + 2J_{\mathrm{in}}\,, \quad \mathbf{x}\in\partial\mathcal{D}\,. \tag{3.45b}$$

If we also assume that the transport equation's fixed-source, q, is isotropic, then the VEF moment system with Miften-Larsen boundary conditions is equivalent to radiation diffusion with Marshak boundary conditions in the thick diffusion limit where the angular flux is linearly anisotropic.

For SMM, the closures are not normalized. This means they are not guaranteed to be bounded functions of the angular flux. However, the SMM closures have the same asymptotic limit as the VEF closures. To see this, let $\psi(\mathbf{x}, \mathbf{\Omega}) = f(\mathbf{x}) + \mathbf{\Omega} \cdot \boldsymbol{g}(\mathbf{x})$ then

$$\mathbf{T}(\psi) = \int \mathbf{\Omega} \otimes \mathbf{\Omega} \, (f + \mathbf{\Omega} \cdot \boldsymbol{g}) \, \mathrm{d}\Omega - \frac{1}{3}\mathbf{I} \int f + \mathbf{\Omega} \cdot \boldsymbol{g} \, \mathrm{d}\Omega = \frac{4\pi f}{3}\mathbf{I} - \frac{4\pi f}{3}\mathbf{I} = 0 \,, \tag{3.46}$$

$$\beta(\psi) = \int |\mathbf{\Omega} \cdot \mathbf{n}| \, (f + \mathbf{\Omega} \cdot \boldsymbol{g}) \, \mathrm{d}\Omega - \frac{1}{2} \int f + \mathbf{\Omega} \cdot \boldsymbol{g} \, \mathrm{d}\Omega = 2\pi f - \frac{4\pi f}{2} = 0 \,. \tag{3.47}$$

Thus, the closures simplify to

$$\mathbf{P} = \frac{1}{3}\mathbf{I}\varphi \,, \quad B = \frac{1}{2}\varphi \,, \tag{3.48}$$

in the thick diffusion limit. In other words, the moment equations with SMM closures are equivalent to radiation diffusion with Marshak boundary conditions when the angular flux is linearly anisotropic.

3.5 Functional Derivatives of the VEF Data

Here, we compute the functional derivatives of the VEF data in order to understand some of the convergence properties of the VEF algorithm. These functional derivatives are also used in Section 3.6 to establish the connection between the VEF and SMM closures. An introduction to the Gateaux derivative is provided before deriving the functional derivatives of the VEF data and applying them to investigate the convergence properties of the VEF algorithm.

3.5.1 The Gateaux Derivative

The Gateaux derivative is a generalization of the directional derivative that supports derivatives of functionals (i.e. a function whose argument is a function) as well as complicated mathematical objects such as second-order tensors [63]. Let $f : X \to Y$ be a (possibly nonlinear) mapping from a space X to a space Y. For example, a simple scalar function $f(\mathbf{x})$ would set $X = \mathbb{R}^{\mathrm{dim}}$ and $Y = \mathbb{R}$. The Gateaux derivative of f evaluated at $u \in X$ in the direction $v \in X$ is given by

$$D[f](u, v) = \lim_{\omega \to 0} \frac{f(u + \omega v) - f(u)}{\omega} \,, \tag{3.49}$$

where $\omega \in \mathbb{R}$. Since

$$\begin{aligned}\left[\frac{\partial}{\partial \omega} f(u+\omega v)\right]_{\omega=0} &= \left[\lim_{\Delta\omega \to 0} \frac{f(u+(\omega+\Delta\omega)v) - f(u+\omega v)}{\Delta\omega}\right]_{\omega=0} \\ &= \lim_{\Delta\omega \to 0} \left[\frac{f(u+\omega v+\Delta\omega v) - f(u+\omega v)}{\Delta\omega}\right]_{\omega=0} \\ &= \lim_{\Delta\omega \to 0} \frac{f(u+\Delta\omega v) - f(u)}{\Delta\omega}\,,\end{aligned} \tag{3.50}$$

where continuity of f is used to move the limit outside of the brackets, the Gateaux derivative can also equivalently be written:

$$D[f](u,v) = \left[\frac{\partial}{\partial \omega} f(u+\omega v)\right]_{\omega=0}\,. \tag{3.51}$$

We favor the definition in Eq. 3.51 over Eq. 3.49 as it leads to simpler calculations by leveraging the familiar machinery of the partial derivative.

As an example, if $u : \mathbb{R}^2 \to \mathbb{R}$ and $\boldsymbol{v} : \mathbb{R}^2 \to \mathbb{R}^2$, we can compute $\boldsymbol{v} \cdot \nabla u|_{\mathbf{x}}$ using the above definition as

$$D[u](\mathbf{x},\boldsymbol{v}) = \frac{\partial}{\partial \omega}\left[u(\mathbf{x}+\omega\boldsymbol{v})\right]_{\omega=0}\,. \tag{3.52}$$

To particularize, let $u(\mathbf{x}) = xy$ and $\boldsymbol{v} = \begin{bmatrix} v_1 & v_2 \end{bmatrix}^T$, then

$$\begin{aligned}D[u](\mathbf{x},\boldsymbol{v}) &= \frac{\partial}{\partial \omega}\left[(x+\omega v_1)(y+\omega v_2)\right]_{\omega=0} \\ &= \frac{\partial}{\partial \omega}\left[xy + \omega(xv_2 + yv_1) + \omega^2 v_1 v_2\right]_{\omega=0} \\ &= \left[xv_2 + yv_1 + 2\omega v_1 v_2\right]_{\omega=0} \\ &= xv_2 + yv_1 \\ &= \boldsymbol{v} \cdot \nabla(xy)\,.\end{aligned} \tag{3.53}$$

This establishes the connection between the directional derivative and the Gateaux derivative.

In the context of a Newton method, the Gateaux derivative defines a systematic process for computing the action of the Jacobian. Consider the first-order Taylor series expansion of a function f about $\mathbf{x}_0$:

$$f(\mathbf{x}) \xrightarrow{\text{TSE}} f(\mathbf{x}_0) + \left.\frac{\partial f}{\partial \mathbf{x}}\right|_{\mathbf{x}_0} (\Delta\mathbf{x})\,, \tag{3.54}$$

where $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_0$. That is, the function f is approximated by its value at $\mathbf{x}_0$ and its gradient evaluated at $\mathbf{x}_0$ in the direction of $\Delta\mathbf{x}$. Thus, we can alternatively write

$$f(\mathbf{x}) \xrightarrow{\text{TSE}} f(\mathbf{x}_0) + D[f](\mathbf{x}_0, \Delta\mathbf{x})\,. \tag{3.55}$$

In this way, the Gateaux derivative provides a process for linearizing any f even when f is tensor-valued and the argument $\mathbf{x}$ is itself a function. For example, we can linearize the Eddington tensor about some angular flux ψ_0 using:

$$\mathbf{E}(\psi) \xrightarrow{\text{TSE}} \mathbf{E}(\psi_0) + D[\mathbf{E}](\psi_0, \psi') \, . \tag{3.56}$$

This linearization is used in this section to investigate the properties of the VEF data and is also used in Section 3.6 to establish the connection between the VEF and SMM closures.

3.5.2 Derivation of Functional Derivatives

Applying the definition in Eq. 3.51 to the Eddington tensor, the derivative of the Eddington tensor evaluated at ψ_0 in the direction ψ' is

$$\begin{aligned} D[\mathbf{E}](\psi_0, \psi') &= \frac{\partial}{\partial\omega}[\mathbf{E}(\psi_0 + \omega\psi')]_{\omega=0} \\ &= \frac{\partial}{\partial\omega}\left[\frac{\int \mathbf{\Omega}\otimes\mathbf{\Omega}\,(\psi_0 + \omega\psi')}{\int \psi_0 + \omega\psi' \, \mathrm{d}\Omega}\right]_{\omega=0} \\ &= \frac{\partial}{\partial\omega}\left[\frac{\mathbf{P}_0 + \omega\mathbf{P}'}{\phi_0 + \omega\phi'}\right]_{\omega=0} , \end{aligned} \tag{3.57}$$

where ϕ_0 and $\mathbf{P}_0$ are the zeroth and second moments of ψ_0 and ϕ' and $\mathbf{P}'$ the zeroth and second moments of ψ'. Applying the quotient rule,

$$\begin{aligned} \frac{\partial}{\partial\omega}\left[\frac{\mathbf{P}_0 + \omega\mathbf{P}'}{\phi_0 + \omega\phi'}\right]_{\omega=0} &= \left.\frac{\mathbf{P}'(\phi_0 + \omega\phi') - (\mathbf{P}_0 + \omega\mathbf{P}')\phi'}{(\phi_0 + \omega\phi')^2}\right|_{\omega=0} \\ &= \frac{\mathbf{P}'\phi_0 - \mathbf{P}_0\phi}{\phi_0^2} \\ &= \frac{1}{\phi_0}\left(\mathbf{P}' - \frac{\mathbf{P}_0}{\phi_0}\phi'\right) \\ &= \frac{1}{\phi_0}\left(\mathbf{P}' - \mathbf{E}_0\phi'\right) , \end{aligned} \tag{3.58}$$

where $\mathbf{E}_0 = \mathbf{P}_0/\phi_0$ is the Eddington tensor evaluated at $\psi = \psi_0$. Thus, the derivative of the Eddington tensor evaluated at ψ_0 in the direction ψ' is:

$$D[\mathbf{E}](\psi_0, \psi') = \frac{1}{\phi_0}\left(\int \mathbf{\Omega}\otimes\mathbf{\Omega}\,\psi' \, \mathrm{d}\Omega - \mathbf{E}_0 \int \psi' \, \mathrm{d}\Omega\right) . \tag{3.59}$$

Note that $D[\mathbf{E}](\psi_0, \psi')$ is also a second-order tensor. The above process applies analogously to the boundary factor. The Gateaux derivative of the boundary factor at ψ_0 in the direction

ψ' is

$$\begin{aligned} D[E_b](\psi_0, \psi') &= \frac{\partial}{\partial \omega}[E_b(\psi + \omega\psi')]_{\omega=0} \\ &= \frac{\partial}{\partial \omega}\left[\frac{\int |\mathbf{\Omega}\cdot\mathbf{n}|\,(\psi_0 + \omega\psi')\,\mathrm{d}\Omega}{\int \psi_0 + \omega\psi'\,\mathrm{d}\Omega}\right]_{\omega=0} \\ &= \left.\frac{\left(\int \psi_0 + \omega\psi'\,\mathrm{d}\Omega\right)\left(\int |\mathbf{\Omega}\cdot\mathbf{n}|\,\psi'\,\mathrm{d}\Omega\right) - \left(\int |\mathbf{\Omega}\cdot\mathbf{n}|\,(\psi_0 + \omega\psi')\,\mathrm{d}\Omega\right)\left(\int \psi'\,\mathrm{d}\Omega\right)}{\left(\int \psi_0 + \omega\psi'\,\mathrm{d}\Omega\right)^2}\right|_{\omega=0} \\ &= \frac{1}{\phi_0}\left[\int |\mathbf{\Omega}\cdot\mathbf{n}|\,\psi'\,\mathrm{d}\Omega - E_{b0}\int \psi'\,\mathrm{d}\Omega\right] , \end{aligned} \tag{3.60}$$

where $E_{b0} = \int |\mathbf{\Omega}\cdot\mathbf{n}|\,\psi_0\,\mathrm{d}\Omega / \int \psi_0\,\mathrm{d}\Omega$.

3.5.3 Intuition for the Rapid Convergence of the VEF Algorithm

The rapid convergence of VEF algorithms is due to the VEF data having weak dependence on the angular flux as characterized by having small functional derivatives with respect to the solution [19]. Too see this, consider the linearization of the Eddington tensor about the previous iteration's angular flux, ψ^ℓ:

$$\mathbf{E}(\psi) \approx \mathbf{E}(\psi^\ell) + D[\mathbf{E}](\psi^\ell, \psi') \,. \tag{3.61}$$

If we set ψ' to be the error at iteration ℓ such that $\psi' = \psi - \psi^\ell$ with ψ the solution of the transport problem, the above linearization provides an approximation for how the Eddington tensor will change as the algorithm proceeds to the next iteration. Note that the VEF algorithm converges when the VEF data converge. Thus, if the size of $D[E](\psi^\ell, \psi - \psi^\ell)$ is small enough relative to the iteration's stopping tolerance, the Eddington tensor will change by an amount small enough to allow the iteration to terminate.

We now show three examples of pairs of evaluation points, ψ_0, and directions, ψ', where the Gateaux derivative is zero. In these cases, the Eddington tensor evaluated at ψ_0 is an approximation to the true Eddington tensor to at least the accuracy of the linearization process (e.g. $\mathcal{O}(\psi')^2$). First, consider the direction being a scalar multiple of ψ_0 such that $\psi' = \alpha\psi_0$ for some $\alpha \in \mathbb{R}$. The Gateaux derivative for this case is:

$$D[\mathbf{E}](\psi_0, \alpha\psi_0) = \frac{1}{\phi_0}\left(\int \mathbf{\Omega}\otimes\mathbf{\Omega}\,\alpha\psi_0\,\mathrm{d}\Omega - \mathbf{E}_0\int \alpha\psi_0\,\mathrm{d}\Omega\right) = \frac{\alpha}{\phi_0}\left(\mathbf{P}_0 - \mathbf{P}_0\right) = 0 \,, \tag{3.62}$$

since $\mathbf{E}_0 \int \psi_0 \, \mathrm{d}\Omega = \mathbf{P}_0$. Additionally, if we add a perturbation to ψ_0 that is linear in angle,

$$\begin{aligned} D[\mathbf{E}](\psi_0, \psi_0 + \boldsymbol{\Omega}\cdot\boldsymbol{g}(\mathbf{x})) &= \frac{1}{\phi_0}\left(\int \boldsymbol{\Omega}\otimes\boldsymbol{\Omega}\,(\psi_0 + \boldsymbol{\Omega}\cdot\boldsymbol{g}(\mathbf{x}))\,\mathrm{d}\Omega - \mathbf{E}_0 \int \psi_0 + \boldsymbol{\Omega}\cdot\boldsymbol{g}(\mathbf{x})\,\mathrm{d}\Omega\right) \\ &= \frac{1}{\phi_0}\left(\mathbf{P}_0 - \mathbf{P}_0\right) \\ &= 0\,. \end{aligned} \tag{3.63}$$

The above holds for any spatially-dependent function, $\boldsymbol{g}(\mathbf{x})$, and also for any perturbation that is odd in angle. Finally, let ψ_0 and ψ' be linearly anisotropic in angle such that $\psi_0 = f_0(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{g}_0(\mathbf{x})$ and $\psi' = f'(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{g}'(\mathbf{x})$, then

$$\begin{aligned} D[\mathbf{E}](f_0(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{g}_0(\mathbf{x}), f'(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{g}'(\mathbf{x})) &= \frac{1}{4\pi f_0}\left(\frac{4\pi f'}{3}\mathbf{I} - \frac{1}{3}\mathbf{I}\cdot 4\pi f'\right) \\ &= \frac{f'}{f}\left(\frac{1}{3}\mathbf{I} - \frac{1}{3}\mathbf{I}\right) \\ &= 0\,. \end{aligned} \tag{3.64}$$

Note that the $D[E_b](\psi_0, \psi') = 0$ for each of the pairs (ψ_0, ψ') discussed above as well. Thus, if the error at any iteration is a scalar multiple or an odd-in-angle perturbation of the current iteration's solution, the functional derivatives of the VEF data are zero. This is also true if the current iteration's solution is linearly anisotropic and the true solution is linearly anisotropic.

3.6 SMM as a Linearized VEF Algorithm

In the process of performing a Fourier stability analysis of the VEF algorithm, Cefus and Larsen [54] showed that SMM is equivalent to the VEF algorithm linearized about a linearly anisotropic solution. Let

$$\mathbf{V}(\psi, \varphi) = -\nabla\cdot\frac{1}{\sigma_t}\nabla\cdot(\mathbf{E}(\psi)\varphi) + \sigma_a\varphi - Q = 0\,, \quad \mathbf{x}\in\mathcal{D}\,, \tag{3.65}$$

$$\mathbf{B}(\psi, \varphi) = \boldsymbol{J}\cdot\mathbf{n} - E_b\varphi - 2J_{\text{in}} = 0\,, \quad \mathbf{x}\in\partial\mathcal{D}\,, \tag{3.66}$$

represent the VEF drift-diffusion equation and Miften-Larsen boundary conditions, respectively. Here, $Q = Q_0 - \nabla\cdot\frac{\boldsymbol{Q}_1}{\sigma_t}$ is used for brevity. The coupled transport-VEF system can be written as the root-finding problem:

$$\mathbf{F}(\mathbf{y}) = \begin{bmatrix} \mathbf{L}\psi - \frac{\sigma_s}{4\pi}\varphi - q \\ \mathbf{V}(\psi, \varphi) \\ \mathbf{B}(\psi, \varphi) \end{bmatrix} = 0\,, \tag{3.67}$$

where $\mathbf{y} = \begin{bmatrix}\psi & \varphi\end{bmatrix}^T$ and $\mathbf{L}$ is the streaming and collision operator defined in Eq. 3.22. In this section, we show that the SMM algorithm is equivalent to a first-order Taylor series of $\mathbf{F}$ expanded about $\mathbf{y}_0 = \begin{bmatrix}\psi_0 & \varphi_0\end{bmatrix}^T$ where ψ_0 is a linearly anisotropic solution of the transport equation and $\varphi_0 = \int \psi_0 \,\mathrm{d}\Omega$. In other words, ψ_0 is the diffusion approximation to the transport problem at hand. We assume that ψ_0 and φ_0 satisfy the transport and Marshak diffusion boundary conditions, respectively.

The first-order Taylor series approximation to the root finding problem $\mathbf{F}(\mathbf{y}) = 0$ is:

$$0 = \mathbf{F}(\mathbf{y}) \xrightarrow{\text{TSE}} \mathbf{F}(\mathbf{y}_0) + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{y}}\right|_{\mathbf{y}_0} (\mathbf{y} - \mathbf{y}_0) \,. \tag{3.68}$$

The Jacobian is given by:

$$\frac{\partial \mathbf{F}}{\partial \mathbf{y}} = \begin{bmatrix} \frac{\partial F_1}{\partial \psi} & \frac{\partial F_1}{\partial \varphi} \\ \frac{\partial F_2}{\partial \psi} & \frac{\partial F_2}{\partial \varphi} \\ \frac{\partial F_3}{\partial \psi} & \frac{\partial F_3}{\partial \varphi} \end{bmatrix} , \tag{3.69}$$

where F_i are the rows of $\mathbf{F}$. The transport equation is linear in both ψ and φ so the first row of the Jacobian is simply:

$$\frac{\partial F_1}{\partial \mathbf{y}} = \begin{bmatrix}\mathbf{L} & -\frac{\sigma_s}{4\pi}\end{bmatrix} \,. \tag{3.70}$$

The second and third rows are complicated by the nonlinear dependence on ψ in the operators $\mathbf{V}$ and $\mathbf{B}$. The second row of the Jacobian is:

$$\begin{aligned} \left.\frac{\partial F_2}{\partial \mathbf{y}}\right|_{\mathbf{y}_0} &= \left.\begin{bmatrix}\frac{\partial \mathbf{V}}{\partial \psi} & \frac{\partial \mathbf{V}}{\partial \varphi}\end{bmatrix}\right|_{\mathbf{y}_0} \\ &= \begin{bmatrix} -\nabla \cdot \frac{1}{\sigma_t} \nabla \cdot \left(\left.\frac{\partial \mathbf{E}}{\partial \psi}\right|_{\psi_0} \varphi_0 \right) & -\nabla \cdot \frac{1}{\sigma_t} \nabla \cdot \mathbf{E}(\psi_0) + \sigma_a \end{bmatrix} \end{aligned} \tag{3.71}$$

where

$$\left.\frac{\partial \mathbf{E}}{\partial \psi}\right|_{\psi_0} = \frac{1}{\phi_0} \left(\int \mathbf{\Omega} \otimes \mathbf{\Omega} \,(\cdot) \,\mathrm{d}\Omega - \mathbf{E}_0 \int (\cdot) \,\mathrm{d}\Omega \right) \tag{3.72}$$

is derived in Eq. 3.59. Here, $\phi_0 = \int \psi_0 \,\mathrm{d}\Omega$ and $\mathbf{E}_0 = \mathbf{E}(\psi_0)$. Since ψ_0 is defined to be a linearly anisotropic function in angle, $\mathbf{E}_0 = \frac{1}{3}\mathbf{I}$. In addition, since $\varphi_0 = \int \psi_0 \,\mathrm{d}\Omega$, $\varphi_0/\phi_0 = 1$ and thus

$$\left.\frac{\partial \mathbf{E}}{\partial \psi}\right|_{\psi_0} \varphi_0 = \int \mathbf{\Omega} \otimes \mathbf{\Omega} \;(\cdot) \,\mathrm{d}\Omega - \frac{1}{3}\mathbf{I} \int (\cdot) \,\mathrm{d}\Omega \equiv \mathbf{T}(\cdot) \,. \tag{3.73}$$

In other words, the product $\frac{\partial \mathbf{E}}{\partial \psi}|_{\psi_0} \varphi_0$ is equivalent to the correction tensor used in SMM (see Eq. 3.16). Thus, the second row of the Jacobian becomes

$$\frac{\partial F_2}{\partial \mathbf{y}} = \begin{bmatrix} -\nabla \cdot \frac{1}{\sigma_t} \nabla \cdot \mathbf{T} & \mathbf{D} \end{bmatrix} \tag{3.74}$$

where

$$\mathbf{D} = -\nabla \cdot \frac{1}{3\sigma_t}\nabla + \sigma_a \tag{3.75}$$

is the diffusion operator.

For the boundary conditions, an analogous process yields

$$\begin{aligned}\frac{\partial F_3}{\partial \mathbf{y}} &= \begin{bmatrix} -\left.\dfrac{\partial E_b}{\partial \psi}\right|_{\psi_0} \varphi_0 & -E_b(\psi_0) \end{bmatrix} \\ &= \begin{bmatrix} -\beta & -\frac{1}{2} \end{bmatrix} ,\end{aligned} \tag{3.76}$$

where we have used the form of the derivative of the Eddington boundary factor derived in Eq. 3.60 and β is the correction factor given in Eq. 3.17.

The linear part of $\mathbf{F}$ is then:

$$\begin{aligned}\mathbf{F}(\mathbf{y}) &\approx \mathbf{F}(\mathbf{y}_0) + \left.\frac{\partial \mathbf{F}}{\partial \mathbf{y}}\right|_{\mathbf{y}_0} (\mathbf{y} - \mathbf{y}_0) \\ &= \begin{bmatrix} \mathbf{L}\psi_0 - \frac{\sigma_s}{4\pi}\varphi_0 - q \\ -\mathbf{D}\varphi_0 - Q \\ \boldsymbol{J} \cdot \mathbf{n} - \frac{1}{2}\varphi_0 - 2J_{\text{in}} \end{bmatrix} + \begin{bmatrix} \mathbf{L} & -\frac{\sigma_s}{4\pi} \\ -\nabla \cdot \frac{1}{\sigma_t}\nabla \cdot \mathbf{T} & \mathbf{D} \\ -\beta & -\frac{1}{2} \end{bmatrix} \begin{bmatrix} \psi - \psi_0 \\ \varphi - \varphi_0 \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{L}\psi - \frac{\sigma_s}{4\pi}\varphi - q \\ -\nabla \cdot \frac{1}{\sigma_t}\nabla \cdot \mathbf{T}(\psi - \psi_0) + \mathbf{D}\varphi - Q \\ \boldsymbol{J} \cdot \mathbf{n} - \frac{1}{2}\varphi - 2J_{\text{in}} - \beta(\psi - \psi_0) \end{bmatrix} \\ &= \begin{bmatrix} \mathbf{L}\psi - \frac{\sigma_s}{4\pi}\varphi - q \\ -\nabla \cdot \frac{1}{\sigma_t}\nabla \cdot \mathbf{T}(\psi) + \mathbf{D}\varphi - Q \\ \boldsymbol{J} \cdot \mathbf{n} - \frac{1}{2}\varphi - 2J_{\text{in}} - \beta(\psi) \end{bmatrix}\end{aligned} \tag{3.77}$$

where the last equivalence is due to the fact that $\mathbf{T}(\psi_0) = 0$ and $\beta(\psi_0) = 0$ since

$$\left.\frac{\partial \mathbf{E}}{\partial \psi}\right|_{\psi_0} (\psi_0) = 0\,, \quad \left.\frac{\partial E_b}{\partial \psi}\right|_{\psi_0} (\psi_0) = 0\,, \tag{3.78}$$

as discussed in Section 3.5.3. Converting the operator notation back to equations, this is equivalent to:

$$\boldsymbol{\Omega} \cdot \nabla\psi + \sigma_t\psi = \frac{\sigma_s}{4\pi}\varphi + q\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{3.79a}$$

$$\psi(\mathbf{x}, \boldsymbol{\Omega}) = \bar{\psi}(\mathbf{x}, \boldsymbol{\Omega})\,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0\,, \tag{3.79b}$$

$$-\nabla \cdot \frac{1}{3\sigma_t}\nabla\varphi + \sigma_a\varphi = Q_0 - \nabla \cdot \frac{\boldsymbol{Q}_1}{\sigma_t} + \nabla \cdot \frac{1}{\sigma_t}\nabla \cdot \mathbf{T}\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{3.80a}$$

$$\boldsymbol{J} \cdot \mathbf{n} = \frac{1}{2}\varphi + J_{\text{in}} + \beta\,, \quad \mathbf{x} \in \partial\mathcal{D}\,. \tag{3.80b}$$

Observe that these equations are equivalent to the transport equation and the SMM diffusion equation given in Eq. 3.20 with the corrected Marshak boundary condition from Eq. 3.19c. An equivalent fixed-point operator can be derived by eliminating the angular flux and operating by the inverse of the diffusion operator. Thus, SMMs are both 1) an algorithm based on a reformulation of the transport equation using additive closures and 2) VEF algorithms linearized about a linearly anisotropic solution.

3.7 Mathematical Properties of the Moment Systems

The presence of the Eddington tensor inside the divergence in the VEF first moment equation leads to cross derivative terms not present in the standard form of the radiation diffusion equation. This can be seen in the differential term in the drift-diffusion form of the VEF equations given by:

$$\nabla \cdot \frac{1}{\sigma_t} \nabla \cdot (\mathbf{E}\varphi) \, . \tag{3.81}$$

Assuming the Eddington tensor and total cross section have the required differentiability, we can use the product rule to write:

$$\begin{aligned} \nabla \cdot \frac{1}{\sigma_t} \nabla \cdot (\mathbf{E}\varphi) &= \nabla \cdot \frac{\mathbf{E}}{\sigma_t} \nabla \varphi + \nabla \cdot \left(\frac{\nabla \cdot \mathbf{E}}{\sigma_t} \varphi \right) \\ &= \nabla \cdot \frac{\mathbf{E}}{\sigma_t} \nabla \varphi + \frac{\nabla \cdot \mathbf{E}}{\sigma_t} \cdot \nabla \varphi + \left(\nabla \cdot \frac{\nabla \cdot \mathbf{E}}{\sigma_t} \right) \varphi \, . \end{aligned} \tag{3.82}$$

Defining

$$\mathbf{D} = \frac{\mathbf{E}}{\sigma_t} \, , \quad \boldsymbol{c} = -\frac{\nabla \cdot \mathbf{E}}{\sigma_t} \, , \quad \gamma = \nabla \cdot \frac{\nabla \cdot \mathbf{E}}{\sigma_t} \, , \tag{3.83}$$

the VEF drift-diffusion equation can be written as the diffusion-advection-reaction equation:

$$- \nabla \cdot \mathbf{D} \nabla \varphi + \boldsymbol{c} \cdot \nabla \varphi + (\sigma_a - \gamma)\varphi = Q \, . \tag{3.84}$$

Here, it is clear that the VEF drift-diffusion equation is not symmetric due to the presence of the advective term, $\boldsymbol{c} \cdot \nabla \varphi$. In addition, since $\mathbf{D}$ is symmetric positive definite (since the Eddington tensor is symmetric positive definite), the VEF drift-diffusion equation is an elliptic partial differential equation.

However, the transport equation allows discontinuous solutions in space and angle. This means the Eddington tensor is generally not differentiable in space. Numerically, it is common to use a DG spatial discretization for the transport equation. In such case, the solution is generally discontinuous across interior mesh interfaces. Thus, the VEF drift-diffusion equation cannot be written in the standard elliptic form of Eq. 3.84 since the Eddington tensor does not have the required regularity to have $\nabla \cdot \mathbf{E}$ be well defined. This means discretization techniques must be extended to handle the non-standard form of the VEF drift-diffusion equation.

These same difficulties also apply to the SMM correction sources. For SMM, the second-order form has the correction source:

$$\nabla \cdot \frac{1}{\sigma_t} \nabla \cdot \mathbf{T} \,, \tag{3.85}$$

which also requires derivatives of the angular flux through the correction tensor, $\mathbf{T}$. Thus, discretizations for the SMM correction source must also be developed to handle the low regularity of the angular flux.

3.8 Discrete Moment Methods

The goal of this dissertation is to evaluate the fixed-point operator $\mathbf{G}(\mathbf{X})$, defined in Eq. 3.32 for VEF and Eq. 3.33 for SMM, numerically in a computationally efficient manner. Numerically approximating this operator requires defining 1) a discretization of the transport equation, 2) a representation for VEF and SMM closures, and 3) a discretization for the moment system. In the VEF literature, discrete VEF methods are generally classified as consistent and independent based on their approach for defining the algorithmic choices corresponding to 2) and 3). Consistent methods are characterized by having discrete transport and moment equations that are algebraically consistent. These methods produce solutions for the moment system that match the moments of the discrete transport equation to machine precision (or to the maximum value of the solver tolerances). In other words, the difference between the moments of the transport solution and the solution of the moment system differ in a manner that is independent of the mesh size. Consistent methods are derived by forming moment equations from the *discrete* transport equation and applying discrete closures. This leads to a moment discretization that is an equivalent reformulation of the discrete transport equation leading to the discrete equivalence that characterizes these methods.

On the other hand, independent moment methods are characterized as having discrete transport and moment equations that are not algebraically consistent. Due to this, the moments of the discrete transport solution and the solution of the moment system differ on the order of the spatial discretization error and are thus only equivalent in the limit as the spatial mesh is refined. Independent methods are derived by discretizing the continuous moment system (with closures already applied) without regard for the discretization used for the transport equation. Rapid convergence is maintained when the discrete closures are represented in a sufficiently consistent manner [22].

Figure 3.2 depicts a commuting diagram for approximating the continuous transport problem with a discrete moment system. The consistent approach moves down and to the right from the continuous transport equation corresponding to discretizing and using discrete moments and closures to form the moment system. Independent methods move right and then down corresponding to forming the continuous moment equations and then discretizing. It is important to note that both approaches lead to discrete moment systems that approximate the continuous transport problem. However, only the consistent approach produces a method with an equivalence in the bottom row of the commuting diagram.

$$\begin{tikzcd}
\text{Continuous Transport} \arrow[r, "\int \boldsymbol{\Omega}^i\,(\cdot)\,\mathrm{d}\Omega"] \arrow[d] & \text{Continuous Moment System} \arrow[d] \\
\text{Discrete Transport} \arrow[r, "\sum \boldsymbol{\Omega}_d^i\,(\cdot)\,w_d"] & \text{Discrete Moment System}
\end{tikzcd}$$

Figure 3.2: The commuting diagram for moment-based approximations of the transport equation. Consistent moment methods apply discrete closures to a discretized transport equation whereas the independent methods discretize the continuous moment system formed through closures of the continuous transport equation. In both cases, the discrete moment system is an approximation to the continuous transport equation. However, independent methods generally do not have equivalence of the bottom row. That is, the discrete moment and transport equations are in general not equivalent.

Consistent methods are attractive in that they can be used in place of an existing transport method without changing the solution. This is particularly important in reactor physics applications where licensing restrictions may require that new methods exactly reproduce solutions produced by older methods. In addition, forming the discrete moment system from the discrete transport equation provides a systematic process for steps 2) and 3) of the discrete moment method. However, for VEF, consistent discretization of the moment system often makes the resulting algebraic system difficult to precondition effectively with existing linear solver technology. Furthermore, use of negative flux fixups to ensure positivity of the transport equation will render an otherwise consistent VEF method inconsistent.

By contrast, the independent approach allows significant algorithmic flexibility. Warsa and Anistratov [22] compared the iterative efficiency of consistent and independent VEF methods and saw equivalent convergence as long as the independent method properly represented the VEF data. In particular, using S_N angular quadrature and finite element interpolation produced independent methods that converged as rapidly as a consistent method. This flexibility allows the design of efficient and robust moment methods that can leverage existing discretization and linear solver technology and have multiphysics compatibility. We pursue independent methods in this dissertation to have the flexibility to discretize the moment system such that it can be efficiently solved and to match the hydrodynamics framework of [12].

Chapter 4

Finite Element Preliminaries

The finite element method (FEM) is a popular technique for numerically approximating a wide variety of problems from mathematical physics. In FEM, partial differential equations are numerically solved using Galerkin's method where finite-dimensional functional spaces are used to approximate the weak, or variational, form of the problem. The basic finite element method embellishes and extends Galerkin's method by defining a systematic process for building these approximation spaces such that they mimic the properties of the infinite-dimensional functional spaces associated with the continuous problem, lead to numerical solutions that increase in accuracy exponentially as the mesh is refined, and are amenable to efficient computer implementation.

Finite element methods are characterized by three basis aspects. First, the domain is divided into a finite set of smaller domains called elements. The union of these elements forms a computational approximation of the domain, known as a mesh, that enables the simulation of problems defined on complex and irregular geometries. Second, the approximation spaces are formed in terms of a finite number of parameters, known as degrees of freedom, corresponding to piecewise polynomial functions. Suitable matching conditions are enforced so that these piecewise polynomial spaces are subsets of the desired infinite-dimensional function spaces. Finally, the approximation spaces are designed to be easily described using a canonical basis that has small support. Use of this canonical basis in Galerkin's method yields algebraic systems of equations that can be formed through computations associated with a single element and are sparse in that their entries are primarily zero. This means that, through a process known as finite element assembly, the entries of the algebraic system can be computed efficiently. In addition, the sparsity of the matrices allows use of specialized data structures that reduce the memory and computational costs of storing and solving the linear system.

This chapter introduces the finite element method and the required notation and machinery needed to understand and implement the finite element discretizations of the transport and moment systems discussed in subsequent chapters. The content of this chapter stems from a variety of sources. For mathematical analyses, Ciarlet [64] and Brenner and Scott [65] are excellent for the standard conforming finite element method, Quarteroni and Valli [66] and Boffi *et al.* [67] for mixed finite elements, and Ern and Guermond [68] for discontinuous finite elements. In addition, Zhodi [69] provides an introduction of the finite element method geared toward engineers.

We begin with a high-level overview of weak forms and Galerkin's method. We present the element-local polynomial spaces used in this document to describe the mesh and solution spaces. The mesh is then defined paying particular attention to the construction of the reference to physical space transformations that allow integration over high-order elements along with the numerical integration of functions defined through these transformations. The finite element spaces used heavily in subsequent chapters are defined. The chapter concludes with implementation details associated with finite element assembly and an introduction to preconditioned iterative solvers.

4.1 The Weak Form and Galerkin's Method

Finite element methods discretize and solve the weak, or variational, form of a partial differential equation. Consider the abstract problem: find $u \in X$ such that

$$A(u) = q\,, \tag{4.1}$$

where, for example, the operator A could be the Poisson operator,

$$A(u) = -\nabla^2 u \tag{4.2}$$

with q representing a source term and the solution space $X = \mathcal{C}^1$, the space of twice differentiable functions. The problem in Eq. 4.1 is referred to as the strong form and its solutions are often plagued by needlessly restrictive differentiability requirements. For example, consider the radiation diffusion equation given by

$$-\nabla \cdot D\nabla\varphi + \sigma_a\varphi = Q\,. \tag{4.3}$$

Here, the diffusion coefficient, D, must be once differentiable and the solution, φ, must be twice differentiable. In problems with multiple materials, the diffusion coefficient can be discontinuous. Solving radiation diffusion in strong form then necessitates the use of domain decomposition methods where the problem is solved with a set of subproblems corresponding to each material that are coupled through interface conditions. Furthermore, in many physical problems discontinuities in the solution, known as shocks, are possible. In such cases, the solution itself is not differentiable. The location of the discontinuities are not known a priori making the use of domain decomposition methods much more difficult.

This motivates the use of reformulations of the strong form that relax, or weaken, the requirements on the differentiability of the solution and its coefficients. These weak forms are derived by multiplying the strong form by a suitably smooth (i.e. differentiable) function and integrating over the domain. By applying integration by parts formulae, derivatives can be offloaded to the test function, leading to a weakening of the differentiability requirements.

In this section, we provide insight into the use of test functions and weak derivatives, derive the weak form for the abstract problem $A(u) = q$ along with an example weak form for the radiation diffusion problem, discuss the mathematical spaces that are required to make the weak form well defined, and define Galerkin's method for numerically approximating the weak form of a partial differential equation.

4.1.1 Test Functions and Weak Derivatives

Consider a one-dimensional function f defined on the interval $[a, b]$. We "test" f by multiplying it by an arbitrary function v and integrating over the interval $[a, b]$. If, for example, $v = 1$ or $v = x$, we can glean information about the mean and variance of f, respectively. In addition, by setting the test function to the Dirac delta function centered at an arbitrary position x we have that

$$\int_a^b \delta(x' - x)\, f(x')\, \mathrm{d}x' = f(x)\,, \tag{4.4}$$

meaning integrating f against a test function v can be viewed as a generalization of inspecting a function with the familiar means of point-wise evaluation. Testing f with v is simply an alternative method for investigating the properties of f.

Integrating against a test function is particularly useful for inspecting the properties of derivatives. Consider $f = \frac{\mathrm{d}g}{\mathrm{d}x}$ for some function g. Let v be a differentiable function that satisfies $v = 0$ at $x = a$ and $x = b$. Testing f with v and integrating by parts yields

$$\int_a^b v \frac{\mathrm{d}g}{\mathrm{d}x}\, \mathrm{d}x = -\int_a^b \frac{\mathrm{d}v}{\mathrm{d}x} g\, \mathrm{d}x\,. \tag{4.5}$$

Thus, we can inspect the properties of $f = \frac{\mathrm{d}g}{\mathrm{d}x}$ without requiring that g is differentiable! Leveraging these so-called "weak derivatives" is a key aspect of the success of the finite element method in modeling phenomena with discontinuous data or shocks in the solution.

4.1.2 An Abstract Weak Form

For the abstract problem, multiplying by a test function v and integrating over the domain yields

$$\int vA(u)\, \mathrm{d}\mathbf{x} = \int v\, q\, \mathrm{d}\mathbf{x}\,. \tag{4.6}$$

Let the bilinear form $\mathcal{A}(v, u)$ represent the operator derived by applying an integration by parts formula to $\int v\, A(u)\, \mathrm{d}\mathbf{x}$ and the linear form $b(v) = \int v\, q\, \mathrm{d}\mathbf{x}$. Furthermore, let $u, v \in V$ be a space of functions such that $\mathcal{A}(v, u)$ is well defined. The statement $\mathcal{A}(v, u) = b(v)$ is then an equivalent reformulation of Eq. 4.6. If we add a condition that this holds for all $(\equiv \forall)$ test functions $v \in V$, then

$$\mathcal{A}(v, u) = b(v)\,, \quad \forall v \in V\,, \tag{4.7}$$

implies that the strong form, $A(u) = q$, is satisfied provided that there exists a solution to the strong form. That is, when there exists $u \in X$ such that $A(u) = q$, the weak form in Eq. 4.7 is equivalent to the strong form. Note that if there does not exist a $u \in X$ that satisfies $A(u) = q$, there may still be a $u \in V$ that satisfies $\mathcal{A}(v, u) = b(v)\,, \forall v \in V$. In general, we have that $X \subset V$ meaning the weak form allows a broader class of solutions. A sufficiently weak form defined over V allows the solution of problems with discontinuous data or shocks in the solution that would not be possible using the strong form defined over X.

In problems where it is possible, the solution space is typically restricted to functions that satisfy the boundary conditions. For example, if the problem contains the Dirichlet boundary condition $u = u^*$ for $\mathbf{x} \in \partial\mathcal{D}$, the solution space would be restricted to

$$V_{u^*} = \{u \in V : u|_{\partial\mathcal{D}} = u^*\}\,, \tag{4.8}$$

the space of functions $u \in V$ that attain $u = u^*$ for $\mathbf{x} \in \partial\mathcal{D}$. This restricted solution space is supported by correspondingly restricting the test space to be zero on the boundary. That is, the test function $v \in V_0$. This modified weak form then reads: find $u \in V_{u^*}$ such that

$$\mathcal{A}(v, u) = b(v)\,, \quad \forall v \in V_0\,. \tag{4.9}$$

4.1.3 Radiation Diffusion Example

Returning to the radiation diffusion example in Eq. 4.3 with the additional requirement of a Dirichlet boundary condition $\varphi = 0$ on $\mathbf{x} \in \partial\mathcal{D}$, the weak form is: find $\varphi \in V$ such that

$$\int \nabla u \cdot D\nabla\varphi \,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi \,\mathrm{d}\mathbf{x} = \int u\, Q \,\mathrm{d}\mathbf{x}\,, \quad \forall u \in V\,. \tag{4.10}$$

Here, we set $V = \{u \in C^0 : u|_{\partial\mathcal{D}} = 0\}$ such that $V$is the space of continuous functions that are zero on the boundary of the domain. We have used the integration by parts formula:

$$\int u\,\nabla \cdot \boldsymbol{v} \,\mathrm{d}\mathbf{x} = \oint_{\partial\mathcal{D}} u\,\boldsymbol{v} \cdot \mathbf{n} \,\mathrm{d}s - \int \nabla u \cdot \boldsymbol{v} \,\mathrm{d}\mathbf{x}\,, \tag{4.11}$$

along with the fact that $u = 0$ for $\mathbf{x} \in \partial\mathcal{D}$ since $u \in V$. Observe that the weak form in Eq. 4.10 has reduced the differentiability requirements of both the solution and the diffusion coefficient. In particular, the test function and solution need only be once differentiable. In addition, there are no longer any differentiability requirements on the diffusion coefficient. Thus, the weak form expands the space of possible solutions from the space of twice differentiable functions to the space of once differentiable functions and allows discontinuous data.

4.1.4 Sobolev Spaces

A key question is the choice of the space V the weak form is defined over. For the strong form, the spaces $\mathcal{C}^i$ corresponding to functions having i continuous derivatives for $i = 0, 1, \dots$ are a natural choice. However, the integrations used in the weak form require a more nuanced approach (see "The Lebesgue integral" from Reed and Simon [70]), the goal being to define the largest possible spaces such that the integrals in the weak form remain finite. That is, in the radiation diffusion example, we seek to define the space for u and φ such that

$$\int \nabla u \cdot D \nabla \varphi \, \mathrm{d}\mathbf{x} < \infty \,, \qquad \int \sigma_a \, u \varphi \, \mathrm{d}\mathbf{x} < \infty \,, \qquad \int u \, q \, \mathrm{d}\mathbf{x} < \infty \,. \tag{4.12}$$

Such properties are achieved by using Hilbertian Sobolev spaces. These spaces are used frequently throughout mathematical physics and are a natural choice for the bilinear and linear forms arising in finite element methods. Following standard notation, we define

$$L^2(\mathcal{D}) = \{u : \mathcal{D} \to \mathbb{R} : \int_{\mathcal{D}} u^2 \, \mathrm{d}\mathbf{x} < \infty\} \,, \tag{4.13}$$

as the space of square-integrable functions. Note that this space does not place any requirements on the differentiability of its elements; functions in $L^2(\mathcal{D})$ need only be square integrable. We will also use the space of functions with square-integrable gradient defined as:

$$H^1(\mathcal{D}) = \{u \in L^2(\mathcal{D}) : \int_{\mathcal{D}} \nabla u \cdot \nabla u \, \mathrm{d}\mathbf{x} < \infty\} \,, \tag{4.14}$$

and the space of vector-valued functions with square-integrable divergence:

$$H(\mathrm{div}; \mathcal{D}) = \{\boldsymbol{v} \in [L^2(\mathcal{D})]^{\mathrm{dim}} : \nabla \cdot \boldsymbol{v} \in L^2(\mathcal{D})\} \,. \tag{4.15}$$

Square integrability is a desired property due to the following result.

Proposition 4.1. *The product of two square-integrable functions is integrable.*

Proof. We must show that given $u, v \in L^2(\mathcal{D})$, $\int uv \, \mathrm{d}\mathbf{x} < \infty$. Observe that

$$(u+v)^2 = u^2 + 2uv + v^2 \geq 0 \iff uv \leq \frac{1}{2}\left(u^2 + v^2\right) \,.$$

Thus,

$$\int uv \, \mathrm{d}\mathbf{x} \leq \int \frac{1}{2}\left(u^2 + v^2\right) \mathrm{d}\mathbf{x} = \frac{1}{2}\int u^2 \, \mathrm{d}\mathbf{x} + \frac{1}{2}\int v^2 \, \mathrm{d}\mathbf{x} < \infty \,,$$

since $u, v \in L^2(\mathcal{D})$ are square-integrable. □

Thus, under mild assumptions on the data D, σ_a, and q, we have that

$$\int \sigma_a \, u \varphi \, \mathrm{d}\mathbf{x} < \infty \,, \quad \forall u, \varphi \in L^2(\mathcal{D}) \,, \tag{4.16a}$$

$$\int u\, q \,\mathrm{d}\mathbf{x} < \infty\,, \quad \forall u \in L^2(\mathcal{D})\,, \tag{4.16b}$$

$$\int \nabla u \cdot D \nabla \varphi \,\mathrm{d}\mathbf{x} < \infty\,, \quad \forall u, \varphi \in H^1(\mathcal{D})\,. \tag{4.16c}$$

Since $H^1(\mathcal{D}) \subset L^2(\mathcal{D})$, taking $u, \varphi \in H^1(\mathcal{D})$ makes all terms in the radiation diffusion weak form well defined. In other words, the proper choice for the test and solution space is $V = H^1(\mathcal{D})$.

4.1.5 Galerkin's Method

We now construct a finite-dimensional approximation for the abstract problem in weak form given by: find $u \in V$ such that

$$\mathcal{A}(v, u) = b(v)\,, \quad \forall v \in V\,. \tag{4.17}$$

The Lax-Milgram theorem (Evans [71, §6.2.1]) states that Eq. 4.17 has a unique solution when $\mathcal{A}(\cdot,\cdot)$ is continuous, i.e., there exists $\alpha > 0$ such that

$$|\mathcal{A}(v, u)| \leq \alpha \|u\| \|v\|\,, \quad \forall u, v \in V\,, \tag{4.18}$$

and coercive, i.e., there exists $\beta > 0$ such that

$$\mathcal{A}(u, u) \geq \beta \|u\|^2\,, \quad \forall u \in V\,. \tag{4.19}$$

Note that the condition of coercivity is a stronger form of positive definiteness. Kirby [72] connects the constants α, β to the condition number of the bilinear form $\mathcal{A}(\cdot,\cdot)$, denoted $\kappa(\mathcal{A})$, as

$$\kappa(\mathcal{A}) \leq \alpha/\beta\,. \tag{4.20}$$

The condition number is commonly used as a proxy for the "difficulty" of solving a problem with iterative methods. The constants α and β depend on the bilinear form $\mathcal{A}(\cdot,\cdot)$ and on the choice for the space V. Thus, the choice of V impacts both the existence and uniqueness of the solution to Eq. 4.17 as well as the ease with which it can be solved.

Galerkin's method for approximating the problem in Eq. 4.17 consists of defining similar problems in finite-dimensional spaces V_h. The conforming finite element method restricts the finite-dimensional approximation space to be $V_h \subset V$. To be specific, for any finite-dimensional subspace $V_h \subset V$ we define the discrete problem to be: find $u_h \in V_h$ such that

$$\mathcal{A}(v_h, u_h) = b(v_h)\,, \quad \forall v_h \in V_h\,. \tag{4.21}$$

Since $V_h \subset V$, we can directly apply the Lax-Milgram theorem to show that the discrete problem in 4.21 has a unique solution. Furthermore, we have that

$$\kappa(\mathcal{A}_h) \leq \alpha/\beta\,, \tag{4.22}$$

where $\mathcal{A}_h(\cdot,\cdot)$ is $\mathcal{A}(\cdot,\cdot)$ restricted to the space V_h [72, Corollary 2.4]. Thus, by defining the approximation space as a subset of the infinite-dimensional space, the discrete operator inherits the analytic structure of the underlying weak problem. In other words, conforming methods allow well defined analytic problems to map to well defined discrete problems that can be solved efficiently. Note that in this document, we also consider non-conforming methods where the approximation space is not a subspace of V. In the case $V_h \not\subset V$, additional requirements are placed on the discrete problem in order to guarantee solvability and stability. This is an important aspect of the DG methods presented in Chapter 6.

Such a general framework as Galerkin's method, however, does not provide a way to define the approximation space V_h. The finite element method fills this gap by providing a systematic process for constructing the finite-dimensional spaces V_h in ways that are amenable to efficient computer implementation.

4.2 Local Polynomial Spaces

Both the mesh and the solution are represented with a polynomial space defined locally on each element. Let $\hat{K}^{\mathrm{dim}}$ denote the dim-dimensional reference element which we set to the unit dim-dimensional cube, $\hat{K} = [0,1]^{\mathrm{dim}}$. We omit the superscript denoting the dimensionality when the dimensionality of the reference element can be inferred from context. We define the space of univariate polynomials of degree less than or equal to k as:

$$\mathcal{P}_k(\hat{K}^1) = \{p : \hat{K}^1 \to \mathbb{R} : p = \sum_{i=0}^{k} \alpha_i x^i \,,\ \alpha_i \in \mathbb{R}\} = \operatorname{span}\{1, x, x^2, \dots, x^k\}\,. \tag{4.23}$$

Multi-dimensional polynomial spaces are built through tensor products of the one-dimensional space. The tensor product polynomial spaces are:

$$\mathcal{Q}_{m,n}(\hat{K}^2) = \{p(x)q(y) : p \in \mathcal{P}_m(\hat{K}^1)\,, q \in \mathcal{P}_n(\hat{K}^1)\} \tag{4.24}$$

in two dimensions and

$$\mathcal{Q}_{\ell,m,n}(\hat{K}^3) = \{p(x)q(y)r(z) : p \in \mathcal{P}_\ell(\hat{K}^1)\,, q \in \mathcal{P}_m(\hat{K}^1)\,, r \in \mathcal{P}_n(\hat{K}^1)\} \tag{4.25}$$

in three dimensions. The tensor product polynomial space of equal degree in each variable is denoted by

$$\mathcal{Q}_p(\hat{K}) = \begin{cases} \mathcal{Q}_{p,p}(\hat{K})\,, & \dim = 2 \\ \mathcal{Q}_{p,p,p}(\hat{K})\,, & \dim = 3 \end{cases}\,. \tag{4.26}$$

Nodal bases for the space $\mathcal{P}_p(\hat{K})$ are constructed using Lagrange interpolating polynomials. We consider interpolation through the Gauss-Lobatto and Gauss-Legendre points. Let $\{\xi_i\}$ represent the $p+1$, one-dimensional Gauss-Lobatto or Gauss-Legendre points in the interval $[0,1]$. Let ℓ_i denote the Lagrange interpolating polynomial satisfying $\ell_i(\xi_j) = \delta_{ij}$

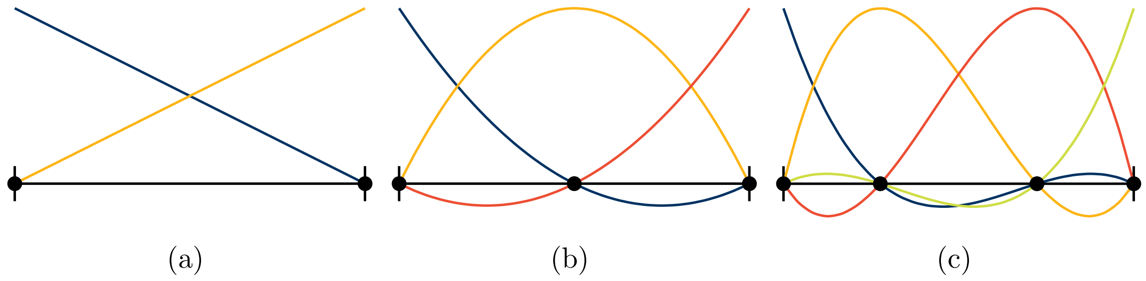


Figure 4.1: Plots of the one-dimensional nodal basis functions through the Gauss-Lobatto nodes for (a) linear, (b) quadratic, and (c) cubic polynomial orders.

where δ_{ij} is the Kronecker delta. The set of functions $\{\ell_i\}$ form a basis for $\mathcal{P}_p(\hat{K})$. The basis functions can be written as:

$$\ell_i(\xi) = \prod_{\substack{0 \le j \le k \\ i \ne j}} \frac{\xi - \xi_j}{\xi_i - \xi_j}, \quad i \in [0, k]. \tag{4.27}$$

Alternatively, writing $\ell_i(\xi) = \sum_{j=0}^{k} c_{ij} \xi^j$, the coefficients c_{ij} that interpolate through the ξ_i can be found by solving the Vandermonde system

$$\begin{bmatrix} 1 & \xi_0 & \xi_0^2 & \dots & \xi_0^k \\ 1 & \xi_1 & \xi_1^2 & \dots & \xi_1^k \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & \xi_k & \xi_k^2 & \dots & \xi_k^k \end{bmatrix} \mathbf{C} = \mathbf{I}, \tag{4.28}$$

where $\mathbf{C} \in \mathbb{R}^{(k+1)\times(k+1)}$ is the matrix of coefficients with entries c_{ij}. The Vandermonde system is simply a change of basis from $\{1, x, x^2, \dots\}$ to the nodal basis. Note that this system becomes increasingly ill-conditioned as the polynomial order increases and is not suitable for computing nodal bases when p is large. Nodal bases for the spaces $\mathcal{Q}_{m,n}(\hat{K})$ and $\mathcal{Q}_{\ell,m,n}(\hat{K})$ are formed through tensor products of the corresponding one-dimensional nodal bases. The basis functions for $\mathcal{P}_1(\hat{K})$, $\mathcal{P}_2(\hat{K})$, and $\mathcal{P}_3(\hat{K})$ through the Gauss-Lobatto and Gauss-Legendre points are shown in Fig. 4.1 and 4.2, respectively. Figure 4.3 shows a selection of the two-dimensional nodal basis functions through the Gauss-Lobatto points for the spaces $\mathcal{Q}_1(\hat{K})$, $\mathcal{Q}_2(\hat{K})$, and $\mathcal{Q}_3(\hat{K})$.

Interpolation through the Gauss-Lobatto and Gauss-Legendre points both have the required properties to be accurate in the limit as $p \to \infty$ [73, 74]. Thus, the choice of interpolating points is typically dictated by other aspects of the overall algorithm. Note that the Gauss-Lobatto points include the interval end points 0 and 1 while the Gauss-Legendre points do not. The bases resulting from Lagrange interpolation through the Gauss-Lobatto and Gauss-Legendre points are referred to as closed and open, respectively, due to this. The Gauss-Legendre basis has the beneficial property of diagonal mass matrices on affine meshes

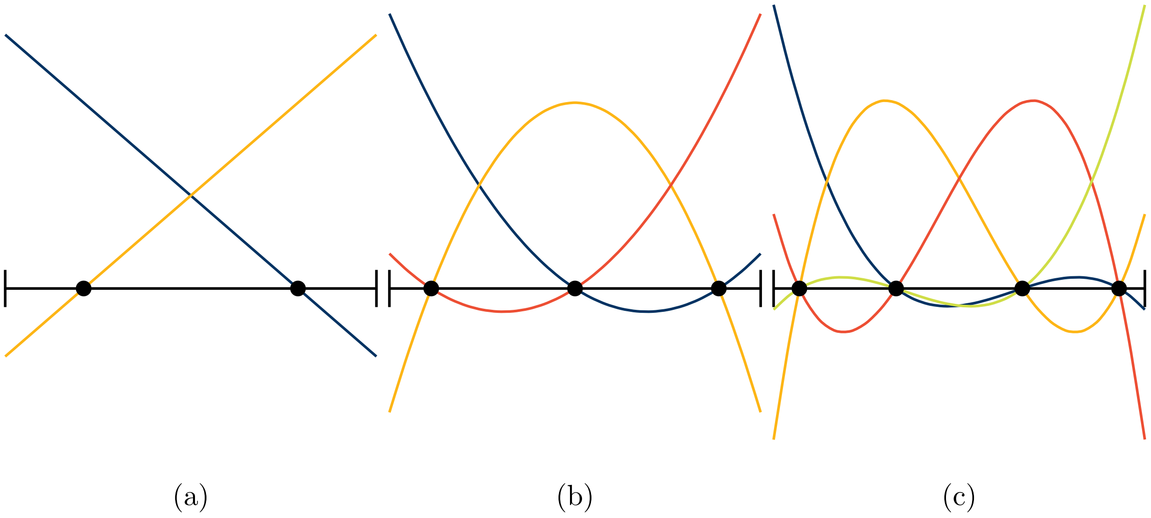


Figure 4.2: Plots of the one-dimensional nodal basis functions through the Gauss-Legendre nodes for (a) linear, (b) quadratic, and (c) cubic polynomial orders.

while the closed Gauss-Lobatto basis typically leads to sparser globally coupled systems since closed bases couple fewer degrees of freedom on interior faces.

4.3 Description of the Mesh

4.3.1 Admissible Tessellations

Let $\mathcal{D} \subset \mathbb{R}^{\dim}$ with $\dim = 2, 3$ be the domain of the problem. The domain is subdivided into a finite number of subsets K, called finite elements, such that

$$\mathcal{D} = \bigcup_{K \in \mathcal{T}} K \,, \tag{4.29}$$

where $\mathcal{T}$ denotes a tessellation of the domain called the mesh. In this document, we only consider the use of tensor product elements. That is, in two dimensions each K is a quadrilateral and in three dimensions each K is a hexahedron. The tessellation of the domain satisfies the following properties:

1. each K is a closed set with an interior, denoted $\mathring{K}$, that is non-empty,

2. the elements do not overlap i.e. for each distinct $K_1, K_2 \in \mathcal{T}$, $\mathring{K}_1 \cap \mathring{K}_2 = \emptyset$,

3. any face of $K_1 \in \mathcal{T}$ is either a subset of the boundary of the domain or a face of another $K_2 \in \mathcal{T}$.

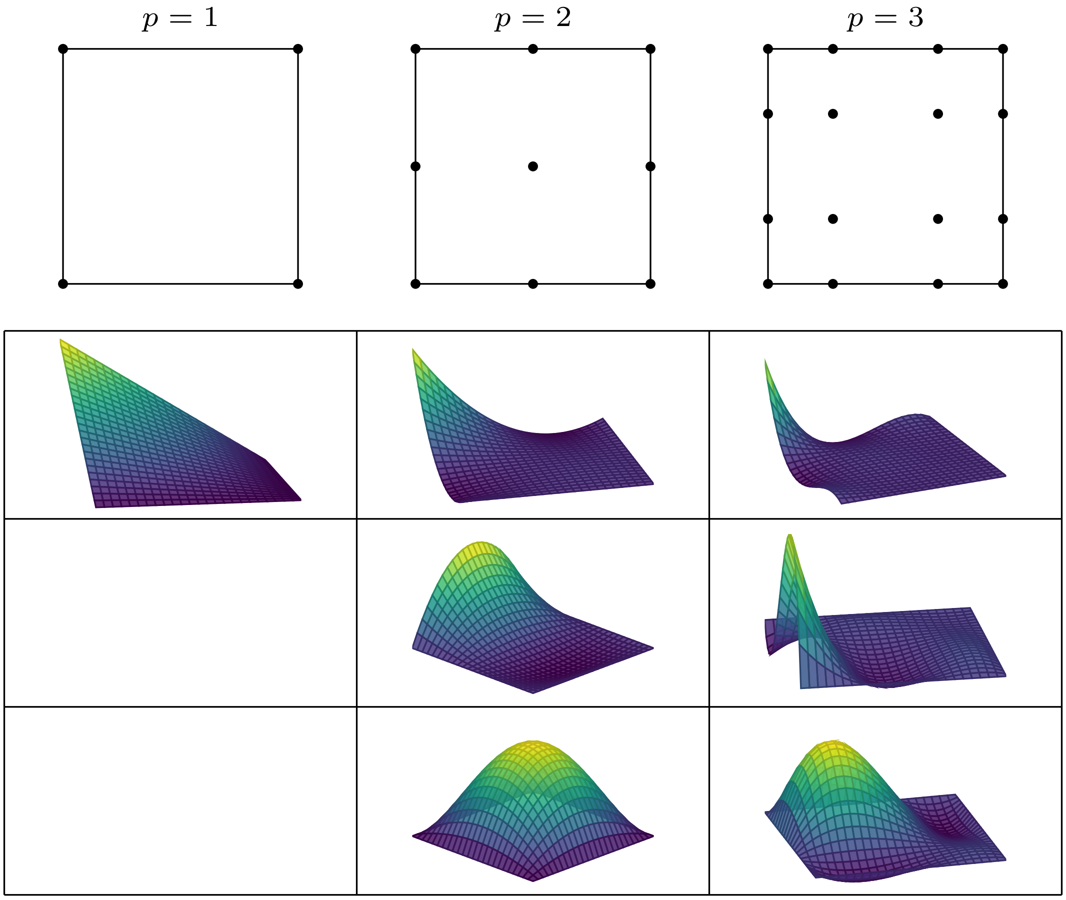
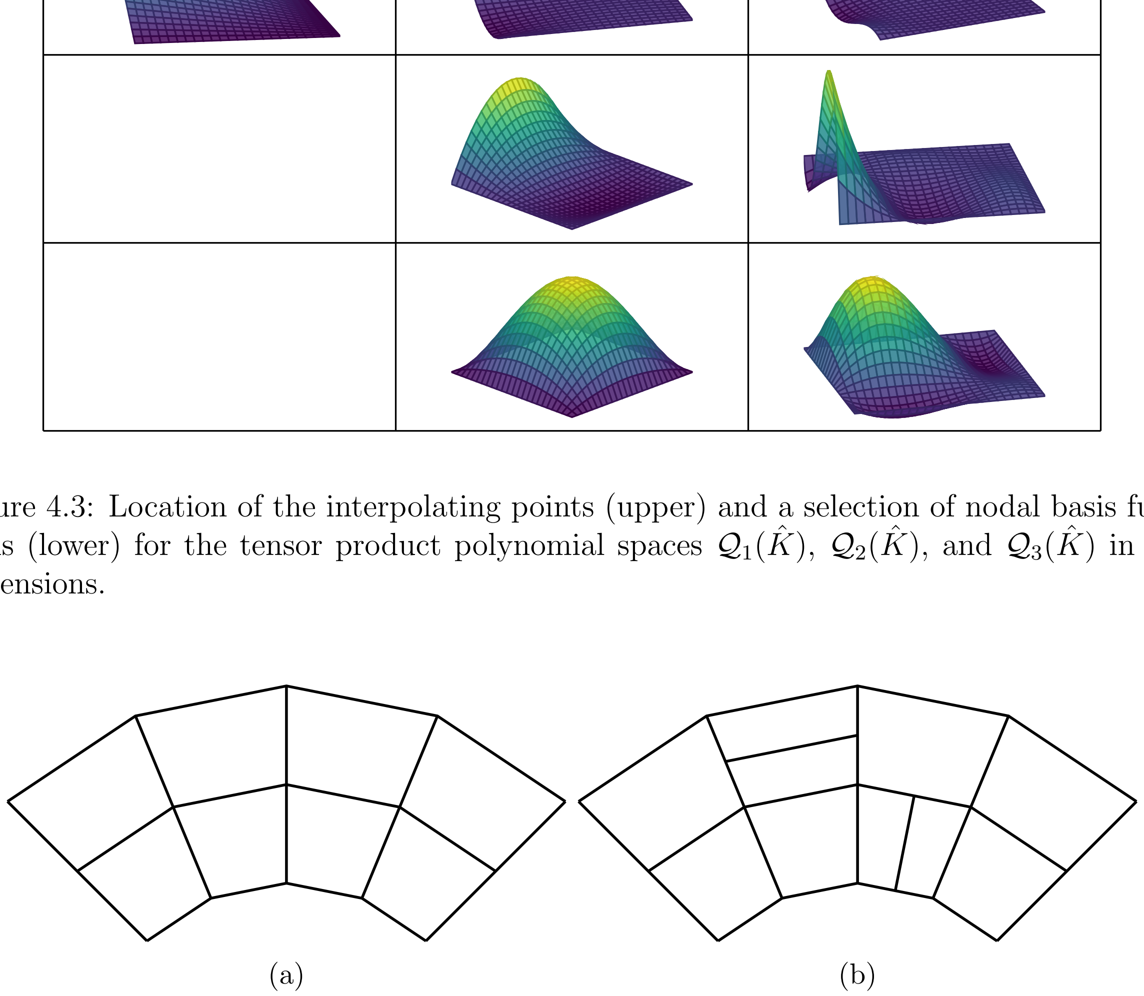


Figure 4.3: Location of the interpolating points (upper) and a selection of nodal basis functions (lower) for the tensor product polynomial spaces $\mathcal{Q}_1(\hat{K})$, $\mathcal{Q}_2(\hat{K})$, and $\mathcal{Q}_3(\hat{K})$ in two dimensions.


(a)

(b)

Figure 4.4: Examples of (a) an admissible tessellation and (b) an inadmissible tessellation.

An example of an admissible mesh is shown in Fig. 4.4a. The elements have non-empty interior, do not overlap, and all faces are either a subset of the boundary or a face of another element in the mesh. Figure 4.4b shows an inadmissible mesh that violates requirement 3 since there are faces in the mesh that correspond to only a subset of another element's boundary. Such a mesh is said to have "hanging nodes" which require a special treatment that we do not consider here.

4.3.2 Mathematical Notation

We define Γ as the set of unique faces in the mesh with $\Gamma_0 = \Gamma \setminus \partial\mathcal{D}$ the set of interior faces and $\Gamma_b = \Gamma \cap \partial\mathcal{D}$ the set of boundary faces so that $\Gamma = \Gamma_0 \cup \Gamma_b$. We denote the outward unit normal to element K as $\mathbf{n}_K$. On an interior face $\mathcal{F} \in \Gamma_0$ between elements K_1 and K_2, we use the convention that $\mathbf{n}$ is the unit vector perpendicular to the shared face $K_1 \cap K_2$ pointing from K_1 to K_2 (see Fig. 4.5). On such an interior face, the jump, $[\![\cdot]\!]$, and average, $\{\!\{\cdot\}\!\}$, are defined as:

$$[\![u]\!] = u_1 - u_2 \,, \quad \{\!\{u\}\!\} = \frac{1}{2}(u_1 + u_2) \,, \quad \text{on } \mathcal{F} \in \Gamma_0 \,, \tag{4.30}$$

where $u_i = u|_{K_i}$ with analogous definitions for vectors. Note that a continuous function u satisfies $[\![u]\!] = 0$ on each interior face. On boundary faces, the jump and average are set to

$$[\![u]\!] = u \,, \quad \{\!\{u\}\!\} = u \,, \quad \text{on } \mathcal{F} \in \Gamma_b \,, \tag{4.31}$$

and likewise for vector-valued functions on the boundary. A straightforward computation shows that

$$\sum_{K\in\mathcal{T}} \int_{\partial K} u \,\boldsymbol{v} \cdot \mathbf{n}_K \,\mathrm{d}s = \int_\Gamma [\![u]\!] \, \{\!\{\boldsymbol{v} \cdot \mathbf{n}\}\!\} \,\mathrm{d}s + \int_{\Gamma_0} \{\!\{u\}\!\} \, [\![\boldsymbol{v} \cdot \mathbf{n}]\!] \,\mathrm{d}s \,. \tag{4.32}$$

We refer to this as the "jumps and averages identity." The restriction of the integration to the interior faces for the second term on the right side of Eq. 4.32 is used so that only one term contributes on the boundary of the mesh and is supported by our definition of the jump and average on the boundary of the domain. Finally, we define the "broken" gradient, denoted by ∇_h, obtained by applying the gradient locally on each element. That is,

$$(\nabla_h u)|_K = \nabla(u|_K) \,, \quad \forall K \in \mathcal{T} \,. \tag{4.33}$$

This distinction is important for the piecewise polynomial spaces discussed in Section 4.5.

4.3.3 Reference Transformations

Each element K is obtained as the image of the reference element $\hat{K}$ under an invertible, polynomial mapping $\mathbf{T} : \hat{K} \to K$ with $\mathbf{T} \in [\mathcal{Q}_m(\hat{K})]^{\mathrm{dim}}$. The mapping $\mathbf{T}$ is derived from

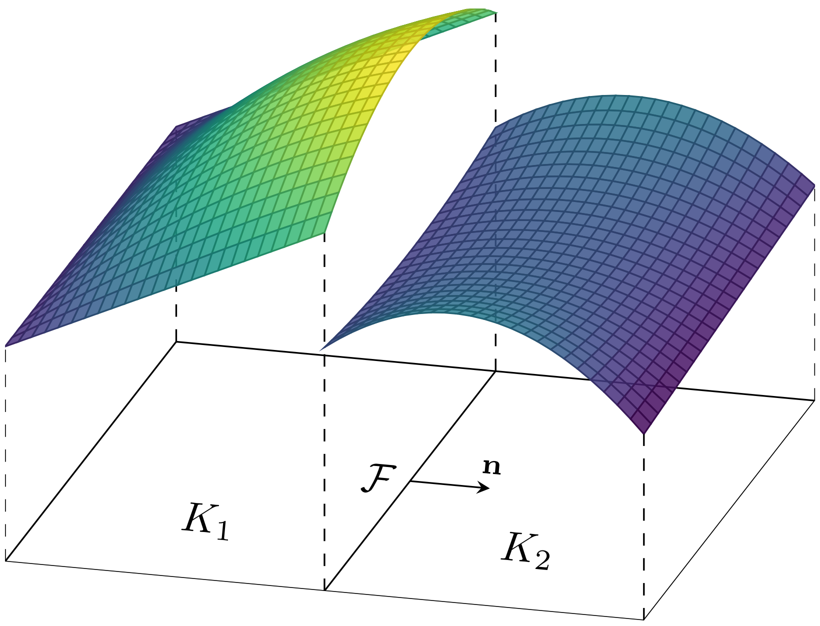


Figure 4.5: A depiction of a discontinuous, piecewise quadratic solution across two quadrilateral elements. The normal vector, $\mathbf{n}$, is defined as pointing from K_1 to K_2 along the face between K_1 and K_2.

a set of global control points and the element-local nodal basis for $\mathcal{Q}_m(\hat{K})$, denoted $\{\ell_i\}$. Figure 4.6a shows an example mesh where the control points labeled 2, 7, and 12 are shared so that the mesh coordinates are continuous across the interface between the two elements. On each element, the mapping is

$$\mathbf{x}(\boldsymbol{\xi}) = \mathbf{T}(\boldsymbol{\xi}) = \sum_i \mathbf{x}_i \ell_i(\boldsymbol{\xi}) \tag{4.34}$$

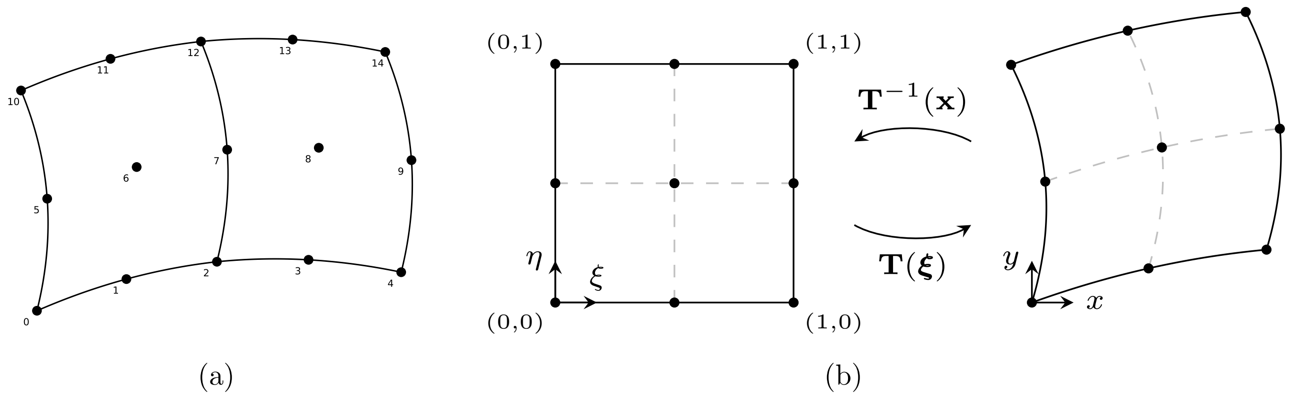
where $\mathbf{x} \in K$, $\boldsymbol{\xi} \in \hat{K}$, and the $\mathbf{x}_i$ are the control points corresponding to element K. Figure 4.6b depicts the mesh transformation used for the left element of Fig. 4.6a.

Let $\boldsymbol{\xi} \in \hat{K}$ denote the reference coordinates and $\mathbf{x} \in \mathcal{D}$ the physical coordinates such that $\mathbf{x} = \mathbf{T}(\boldsymbol{\xi})$. The Jacobian matrix of the mapping is

$$\mathbf{F} = \frac{\partial \mathbf{T}}{\partial \boldsymbol{\xi}} \in \mathbb{R}^{\dim \times \dim}, \tag{4.35}$$

with $J = |\mathbf{F}|$ its determinant. In two dimensions, with $\mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T$ and $\boldsymbol{\xi} = \begin{bmatrix} \xi & \eta \end{bmatrix}^T$, the Jacobian matrix is:

$$\mathbf{F} = \begin{bmatrix} \frac{\partial x}{\partial \xi} & \frac{\partial x}{\partial \eta} \\ \frac{\partial y}{\partial \xi} & \frac{\partial y}{\partial \eta} \end{bmatrix}. \tag{4.36}$$



(a)

(b)

Figure 4.6: Depictions of (a) the mesh control points in a quadratic quadrilateral mesh and (b) the reference transformation used to describe the left element of (a).

The partial derivatives of the mesh transformation are computed by taking derivatives of the nodal basis functions. In other words,

$$\mathbf{F} = \sum_i \mathbf{x}_i \otimes \hat{\nabla} \ell_i \, , \tag{4.37}$$

where $\hat{\nabla}$ denotes the gradient with respect to $\boldsymbol{\xi}$. In two dimensions, this is equivalent to

$$\mathbf{F} = \sum_i \begin{bmatrix} x_i \frac{\partial \ell_i}{\partial \xi} & x_i \frac{\partial \ell_i}{\partial \eta} \\ y_i \frac{\partial \ell_i}{\partial \xi} & y_i \frac{\partial \ell_i}{\partial \eta} \end{bmatrix} \, , \tag{4.38}$$

where $\mathbf{x}_i = \begin{bmatrix} x_i & y_i \end{bmatrix}^T$. The characteristic mesh length, h, is computed using the Jacobian of the transformation as:

$$h = \max_{K \in \mathcal{T}} \left(\int_{\hat{K}} J \, \mathrm{d}\boldsymbol{\xi} \right)^{1/\dim} \, . \tag{4.39}$$

A mesh transformation is called affine when it can be written as

$$\mathbf{T} = \mathbf{A}\boldsymbol{\xi} + \boldsymbol{b} \tag{4.40}$$

where $\mathbf{A} \in \mathbb{R}^{\dim \times \dim}$ and $\boldsymbol{b} \in \mathbb{R}^{\dim}$ are constant with respect to $\boldsymbol{\xi}$. In such case, the Jacobian matrix is $\mathbf{F} = \mathbf{A}$ and the Hessian of the transformation, defined as $\frac{\partial^2 \mathbf{x}}{\partial^2 \boldsymbol{\xi}}$, is identically zero. Quadrilateral elements obtained by scaling, stretching along the ξ or η axes, or rotating the reference element are all affine while general quadrilateral elements, such as trapezoidal elements, and curved elements are not affine. Note that for affine elements, the inverse transformation, denoted $\mathbf{T}^{-1} : K \to \hat{K}$, is given by

$$\mathbf{T}^{-1}(\mathbf{x}) = \mathbf{A}^{-1}(\mathbf{x} - \boldsymbol{b}) \tag{4.41}$$

and is still polynomial. However, for non-affine elements, the inverse transformation is generally not polynomial.

4.3.4 Reference Transformations for Embedded Surfaces

The faces $\mathcal{F} \in \Gamma$ are represented as $(\dim -1)$-dimensional surfaces embedded in the dim-dimensional geometry. That is, a face in a two-dimensional problem is a one-dimensional line and a face in a three-dimensional problem is a two-dimensional plane. If $[\mathcal{Q}_m(\hat{K}^{\dim})]^{\dim}$ is used to define the volumetric reference transformation, the face is described by the polynomial $[\mathcal{Q}_m(\hat{K}^{\dim -1})]^{\dim -1}$ that interpolates through the nodal values corresponding to the face. Let $\mathbf{x}_i$ denote the control points corresponding to a single face in the mesh. For the face between the two elements depicted in the mesh shown in Fig. 4.6a, the $\mathbf{x}_i$ would consist of the positions in two-dimensional space of the nodes labeled 2, 7, and 12. Letting $\{\ell_i\}$ denote the nodal basis for $\mathcal{Q}_m(\hat{K}^{\dim -1})$, the transformation for the face, denoted $\mathbf{T}_{\mathcal{F}} : \hat{K}^{\dim -1} \to \mathcal{F}$, is computed analogously to the volumetric transformation with:

$$\mathbf{T}_{\mathcal{F}}(\boldsymbol{\xi}) = \sum_i \mathbf{x}_i \ell_i(\boldsymbol{\xi})\,. \tag{4.42}$$

However, for a face transformation, the referential coordinate, $\boldsymbol{\xi} \in \hat{K}^{\dim -1} \subset \mathbb{R}^{\dim -1}$, is of dimension one lower than that of the dimension of the mesh. Analogously to the volumetric transformation, the Jacobian matrix is:

$$\mathbf{F}_{\mathcal{F}} = \frac{\partial \mathbf{T}_{\mathcal{F}}}{\partial \boldsymbol{\xi}} \in \mathbb{R}^{\dim \times (\dim -1)}\,. \tag{4.43}$$

Since $\mathcal{F}$ is an embedded surface, the Jacobian matrix is no longer square and thus $|\mathbf{F}_{\mathcal{F}}|$ is no longer well defined. To that end, we define $J_{\mathcal{F}}$ using the Gram determinant:

$$J_{\mathcal{F}} = \sqrt{|\mathbf{F}^T\mathbf{F}|}\,. \tag{4.44}$$

We then have that the characteristic length of the face is

$$h_{\mathcal{F}} = \left(\int_{\hat{K}^{\dim -1}} J_{\mathcal{F}} \, \mathrm{d}\hat{s} \right)^{1/(\dim -1)}\,. \tag{4.45}$$

The normal vector to $\mathcal{F}$ can be be computed by leveraging the fact that the gradient of the components of the embedded transformation points *tangent* to the face. The normal is then the vector that is perpendicular to the tangent vectors. Note that since the Jacobian matrix of the transformation, $\mathbf{F}_{\mathcal{F}}$, represents the gradient of the transformation with respect to the reference coordinates, its columns represent tangent vectors. Since $\mathbf{F}_{\mathcal{F}} \in \mathbb{R}^{\dim \times (\dim -1)}$, the Jacobian matrix will have one tangent vector when $\dim = 2$ and two tangent vectors when $\dim = 3$. Let $\mathbf{t}_i$ be the columns of $\mathbf{F}_{\mathcal{F}}$. We seek the vector $\mathbf{n}$ such that $\mathbf{n} \cdot \mathbf{t}_i = 0$ for $1 \leq i \leq \dim -1$. This can be achieved with a suitable cross product. In two dimensions, we use

$$\mathbf{n} = \frac{\mathbf{t}_1 \times \mathbf{e}_3}{\|\mathbf{t}_1 \times \mathbf{e}_3\|} \tag{4.46}$$

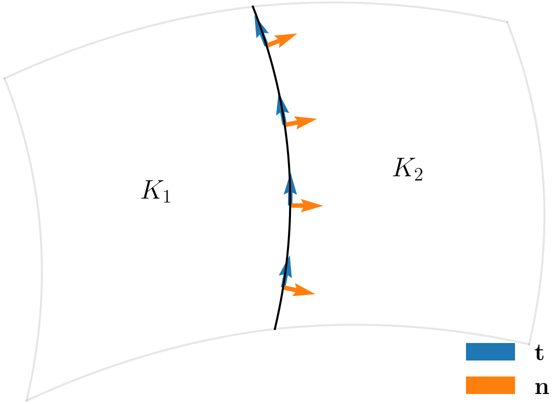


Figure 4.7: The tangent and normal vectors for the face between the two elements of the mesh depicted in Fig. 4.6a. The transformation for the embedded surface between the two elements is defined by the quadratic polynomial in the space $\mathcal{Q}_2(\hat{K}^1)$ that interpolates between the nodes labeled 2, 7, and 12. The tangent vectors are computed from the Jacobian matrix of the embedded transformation. The normals are computed by rotating the tangent vector by 90° in the clockwise direction.

so that the normal vector is a 90° clockwise rotation of the tangent vector. In three dimensions, the normal is computed as the cross product of the two tangent vectors:

$$\mathbf{n} = \frac{\mathbf{t}_1 \times \mathbf{t}_2}{\|\mathbf{t}_1 \times \mathbf{t}_2\|} \,. \tag{4.47}$$

This ensures the normal will be perpendicular to both $\mathbf{t}_1$ and $\mathbf{t}_2$. This process is depicted in Fig. 4.7 for the face between the two quadratic elements shown in Fig. 4.6a. A three-dimensional example is shown in Fig. 4.8. Note that it is important to choose a consistent orientation of the embedded surface so that the normal always points outward. Due to its close relation to the gradient of the embedded transformation, the normal vector will vary as the gradient of the space used for the volumetric transformation. In other words, a mesh with linear faces will have constant normal vectors, quadratic faces will have linear normal vectors, etc.

Alternatively, Nanson's formula [75, 76]:

$$\mathbf{n}\,\mathrm{d}s = J\mathbf{F}^{-T}\hat{\mathbf{n}}\,\mathrm{d}\hat{s} \tag{4.48}$$

can be used to compute the normal vector. Here, $\hat{\mathbf{n}}$ and $\mathrm{d}\hat{s}$ are the normal and magnitude of the face in reference space, respectively. Note that the *volumetric* Jacobian matrix and determinant are used. The normal vector is then

$$\mathbf{n} = \frac{\mathbf{F}^{-T}\hat{\mathbf{n}}}{\|\mathbf{F}^{-T}\hat{\mathbf{n}}\|} \,. \tag{4.49}$$

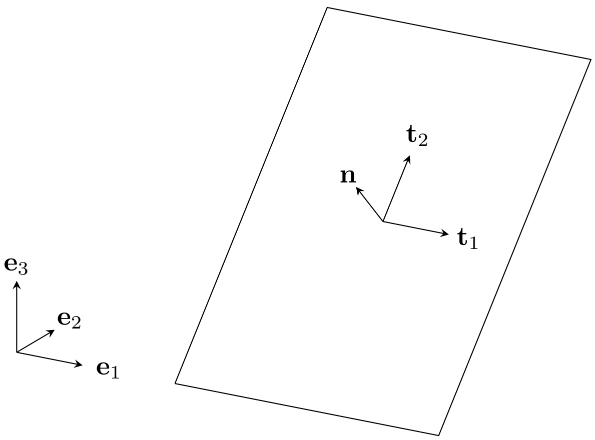


Figure 4.8: A depiction of a two-dimensional, linear quadrilateral element embedded in three dimensions. Here, the Jacobian matrix of the embedded transformation has two columns corresponding to the directions labeled $\mathbf{t}_1$ and $\mathbf{t}_2$. The normal vector points in the direction $\mathbf{t}_1 \times \mathbf{t}_2$, ensuring it is perpendicular to both $\mathbf{t}_1$ and $\mathbf{t}_2$.

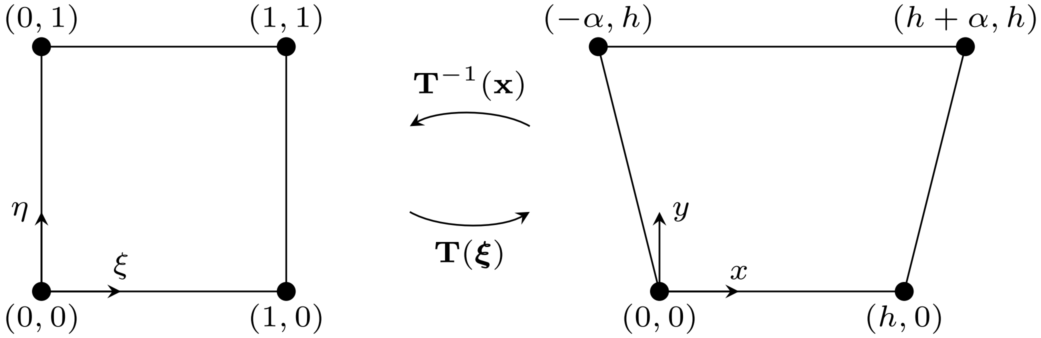


Figure 4.9: An example of the mesh transformation associated with a linear quadrilateral element.

This formulation is advantageous when only the volumetric transformations are accessible. However, it requires knowledge of the location of the desired face in reference space (in order to find the reference coordinate to evaluate $\mathbf{F}$ at) as well as the corresponding referential normal vector.

4.3.5 An Example Transformation: A Scaled Trapezoidal Element

Consider the linear, quadrilateral element depicted in Fig. 4.9 where $\alpha, h \in \mathbb{R}$ with $h > 0$. The case $\alpha = 0$ corresponds to a scaling of the reference element by h. Starting from the point $(0,0)$ in reference space and traversing the nodes counter-clockwise, the nodal basis for $\mathcal{Q}_1(\hat{K})$ is:

$$\begin{aligned} \ell_1(\xi,\eta) &= (1-\xi)(1-\eta)\,, & \ell_2(\xi,\eta) &= \xi(1-\eta)\,, \\ \ell_3(\xi,\eta) &= \xi\eta\,, & \ell_4(\xi,\eta) &= (1-\xi)\eta\,. \end{aligned} \tag{4.50}$$

Let $\mathbf{N} = \begin{bmatrix}\ell_1 & \dots & \ell_4\end{bmatrix}^T$ and $\mathbf{X}$ the matrix of nodal positions corresponding to the nodal basis such that

$$\mathbf{X} = \begin{bmatrix} 0 & h & h+\alpha & -\alpha \\ 0 & 0 & h & h \end{bmatrix}. \tag{4.51}$$

With this notation, the transformation $\mathbf{T} = \sum_i \mathbf{x}_i \ell_i$ can be written as the matrix vector product:

$$\mathbf{T} = \mathbf{X}\mathbf{N}\,. \tag{4.52}$$

Simplifying terms yields:

$$\mathbf{T} = \begin{bmatrix} h\xi + \alpha\eta(2\xi-1) \\ h\eta \end{bmatrix}, \tag{4.53}$$

where $\boldsymbol{\xi} = \begin{bmatrix}\xi & \eta\end{bmatrix}^T$. Letting

$$\mathbf{G} = \hat{\nabla}\mathbf{N} \in \mathbb{R}^{4\times 2} \tag{4.54}$$

represent the matrix of gradients of each of the nodal basis functions, the Jacobian matrix can be computed with

$$\mathbf{F} = \mathbf{X}\mathbf{G}\,, \tag{4.55}$$

so that

$$\mathbf{F} = \begin{bmatrix} 2\alpha\eta + h & \alpha(2\xi-1) \\ & h \end{bmatrix}. \tag{4.56}$$

Note that since $\mathbf{F}$ is not constant with respect to $\boldsymbol{\xi}$, this transformation is not affine. Taking the determinant of $\mathbf{F}$, we have that

$$J = 2\alpha\eta h + h^2\,. \tag{4.57}$$

The area of the element can be computed in reference space with:

$$\int_0^1 \int_0^1 J \,\mathrm{d}\xi \,\mathrm{d}\eta = h^2 + \alpha h\,. \tag{4.58}$$

Observe that this formula matches geometrically computing the area of the scaled trapezoid as the sum of an $h \times h$ square and two triangles with base α and height h. The inverse transformation is found by solving $y = h\eta$ for η and substituting this into the first coordinate of $\mathbf{T}$. The inverse transformation is then:

$$\mathbf{T}^{-1}(\mathbf{x}) = \begin{bmatrix} \frac{hx+\alpha y}{h^2+2\alpha y} \\ y/h \end{bmatrix}. \tag{4.59}$$

Note that the inverse transformation is not polynomial in the first coordinate.

We conclude with a computation of the embedded transformation associated with the right face of the transformation described by $\mathbf{T}$. This face corresponds to fixing $\xi = 1$ and letting $\eta \in [0,1]$ vary. Let

$$b_1(\chi) = 1 - \chi\,, \quad b_2(\chi) = \chi\,, \tag{4.60}$$

be the nodal basis for $\mathcal{Q}_1([0,1])$ where $\chi \in [0,1]$ denotes the reference coordinate for $\mathcal{F}$. The embedded transformation is then

$$\mathbf{T}_{\mathcal{F}} = \begin{bmatrix} h \\ 0 \end{bmatrix} (1-\chi) + \begin{bmatrix} h+\alpha \\ h \end{bmatrix} \chi = \begin{bmatrix} h+\alpha\chi \\ h\chi \end{bmatrix}\,. \tag{4.61}$$

The Jacobian is:

$$\mathbf{F} = \frac{\mathrm{d}\mathbf{T}_{\mathcal{F}}}{\mathrm{d}\chi} = \begin{bmatrix} \alpha \\ h \end{bmatrix}\,. \tag{4.62}$$

The Gram determinant is given by

$$J_{\mathcal{F}} = \sqrt{|\mathbf{F}^T\mathbf{F}|} = \sqrt{\alpha^2 + h^2}\,. \tag{4.63}$$

The length of the face is computed in reference space with

$$\int_0^1 J_{\mathcal{F}}\,\mathrm{d}\chi = \sqrt{\alpha^2 + h^2} \tag{4.64}$$

which exactly matches the computation of the length using the distance formula applied to the points $(h, 0)$ and $(h+\alpha, h)$. Finally, the normal vector to the face is computed by rotating the sole column of $\mathbf{F}$ by $-90°$ and normalizing it:

$$\mathbf{n} = \frac{\begin{bmatrix} h & -\alpha \end{bmatrix}^T}{\|\begin{bmatrix} h & -\alpha \end{bmatrix}^T\|}\,. \tag{4.65}$$

On this linear face, the normal vector is constant.

Alternatively, Nanson's formula can be used to compute the normal. The inverse transpose of the Jacobian for $\mathbf{T}$ from Eq. 4.56 is

$$\mathbf{F}^{-T} = \frac{1}{(2\alpha\eta + h)h} \begin{bmatrix} h & 0 \\ -\alpha(2\xi - 1) & 2\alpha\eta + h \end{bmatrix}\,. \tag{4.66}$$

The normal is computed by applying $\mathbf{F}^{-T}$ to the referential normal, $\hat{\mathbf{n}} = \begin{bmatrix} 1 & 0 \end{bmatrix}^T$, at a point $\boldsymbol{\xi}_{\mathcal{F}} = \begin{bmatrix} 1 & \eta \end{bmatrix}^T$along the face and normalizing:

$$\mathbf{n} = \frac{\mathbf{F}^{-T}|_{\boldsymbol{\xi}_{\mathcal{F}}}\hat{\mathbf{n}}}{\|\mathbf{F}^{-T}|_{\boldsymbol{\xi}_{\mathcal{F}}}\hat{\mathbf{n}}\|} = \frac{\begin{bmatrix} h & -\alpha \end{bmatrix}^T}{\|\begin{bmatrix} h & -\alpha \end{bmatrix}^T\|}\,. \tag{4.67}$$

4.4 Integration Transformations

In this section, we present the machinery used to transform integrands involving scalar and vector-valued functions and their derivatives between reference and physical space. These transformations allow integration over the arbitrary geometries defined by the map **T** using numerical quadrature rules defined on the reference element. These transformations and numerical quadrature are implicitly used to compute all of the bilinear and linear forms discussed in this document.

4.4.1 The Scalar Case

For a scalar function $u : K \to \mathbb{R}$, denote by $\hat{u} : \hat{K} \to \mathbb{R}$ its representation in reference space. The functions u and $\hat{u}$ are related by

$$u(\mathbf{x}) = \hat{u}(\mathbf{T}^{-1}(\mathbf{x}))\,. \tag{4.68}$$

Integration over the physical element is then equivalent to

$$\int_K u \, \mathrm{d}\mathbf{x} = \int_{\hat{K}} \hat{u}\, J \mathrm{d}\boldsymbol{\xi}\,. \tag{4.69}$$

The gradient of a scalar function transforms as

$$\nabla u = \mathbf{F}^{-T} \hat{\nabla} \hat{u}\,, \tag{4.70}$$

where $\mathbf{F}^{-T}$ is the inverse transpose of the Jacobian matrix. Note that the inverse of the Jacobian matrix represents

$$\mathbf{F}^{-1} = \frac{\partial \boldsymbol{\xi}}{\partial \mathbf{x}}\,, \tag{4.71}$$

and in two dimensions is given by

$$\mathbf{F}^{-1} = \begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \xi}{\partial y} \\ \frac{\partial \eta}{\partial x} & \frac{\partial \eta}{\partial y} \end{bmatrix}\,. \tag{4.72}$$

The transformation of the gradient is derived using the chain rule. Observe that in two dimensions,

$$\begin{aligned} \nabla \hat{u} &= \begin{bmatrix} \frac{\partial \hat{u}}{\partial \xi}\frac{\partial \xi}{\partial x} + \frac{\partial \hat{u}}{\partial \eta}\frac{\partial \eta}{\partial x} \\ \frac{\partial \hat{u}}{\partial \xi}\frac{\partial \xi}{\partial y} + \frac{\partial \hat{u}}{\partial \eta}\frac{\partial \eta}{\partial y} \end{bmatrix} \\ &= \underbrace{\begin{bmatrix} \frac{\partial \xi}{\partial x} & \frac{\partial \eta}{\partial x} \\ \frac{\partial \xi}{\partial y} & \frac{\partial \eta}{\partial y} \end{bmatrix}}_{\mathbf{F}^{-T}} \underbrace{\begin{bmatrix} \frac{\partial \hat{u}}{\partial \xi} \\ \frac{\partial \hat{u}}{\partial \eta} \end{bmatrix}}_{\hat{\nabla}\hat{u}}\,. \end{aligned} \tag{4.73}$$

This derivation extends analogously to three dimensions.

4.4.2 The Vector Case

Here, we present the transformations for vector-valued functions assuming dim = 2 for the sake of brevity and note that the ideas presented here extend analogously to three dimensions. For vector-valued functions, the basis the vector is defined on must also be considered. The simplest basis is the canonical basis, $\mathbf{e}_i$, corresponding to the x and y axes. In this case, a vector $\boldsymbol{v} : K \to \mathbb{R}^2$ is

$$\boldsymbol{v} = v_1\mathbf{e}_1 + v_2\mathbf{e}_2 \tag{4.74}$$

and each component transforms independently as $v_i = \hat{v}_i(\mathbf{T}^{-1}(\mathbf{x}))$. Writing

$$\nabla\boldsymbol{v} = \begin{bmatrix} \frac{\partial v_1}{\partial x} & \frac{\partial v_1}{\partial y} \\ \frac{\partial v_2}{\partial x} & \frac{\partial v_2}{\partial y} \end{bmatrix} = \begin{bmatrix} (\nabla v_1)^T \\ (\nabla v_2)^T \end{bmatrix} \tag{4.75}$$

then the gradient of a vector defined as in Eq. 4.74 transforms as

$$\nabla\boldsymbol{v} = \begin{bmatrix} (\mathbf{F}^{-T}\hat{\nabla}\hat{v}_1)^T \\ (\mathbf{F}^{-T}\hat{\nabla}\hat{v}_2)^T \end{bmatrix} . \tag{4.76}$$

Note that defining a vector in this way does not preserve the normal or tangential components under a rotation. That is, $\boldsymbol{v} \cdot \mathbf{n}$ and $\boldsymbol{v} \cdot \mathbf{t}$ are linear combinations of the v_i instead of a single component representing the normal or tangential components, respectively.

Alternatively, the contravariant Piola transform represents vectors on the so-called tangent basis so that the normal component can be preserved [75, 77]. Such a transformation is required by the Raviart Thomas space introduced in Section 4.5.3 in order to strongly enforce continuity in the normal component of the current. The contravariant Piola transform is:

$$\boldsymbol{v} = \frac{1}{J}\mathbf{F}\hat{\boldsymbol{v}} \circ \mathbf{T}^{-1} . \tag{4.77}$$

Here, $\hat{\boldsymbol{v}} : \hat{K} \to \mathbb{R}^2$ is a vector in reference space. Writing the columns of the Jacobian matrix as:

$$\mathbf{F} = \begin{bmatrix} \mathbf{t}_1 & \mathbf{t}_2 \end{bmatrix} , \tag{4.78}$$

the contravariant Piola transformation is equivalent to

$$\boldsymbol{v} = \frac{1}{J}(\hat{v}_1\mathbf{t}_1 + \hat{v}_2\mathbf{t}_2) . \tag{4.79}$$

Observe that, on the reference canonical basis $\hat{\mathbf{e}}_i$, $\boldsymbol{v} = \hat{v}_1\hat{\mathbf{e}}_1 + \hat{v}_2\hat{\mathbf{e}}_2$, and thus the contravariant Piola transform maps the canonical reference basis to the tangent space spanned by $\{\mathbf{t}_1, \mathbf{t}_2\}$ and scales by $1/J$.

When the mesh transformation $\mathbf{T}_e$ is not affine, the tangent basis is not orthogonal. In this case, the usual method of selecting components of a vector through the dot product (e.g. $v_i = \mathbf{t}_i \cdot \boldsymbol{v}$) is inappropriate since $\mathbf{t}_i \cdot \mathbf{t}_j \neq \delta_{ij}$. Instead, a dual basis, referred to as the cotangent basis, is constructed such that

$$\mathbf{n}_i \cdot \mathbf{t}_j = \delta_{ij} . \tag{4.80}$$

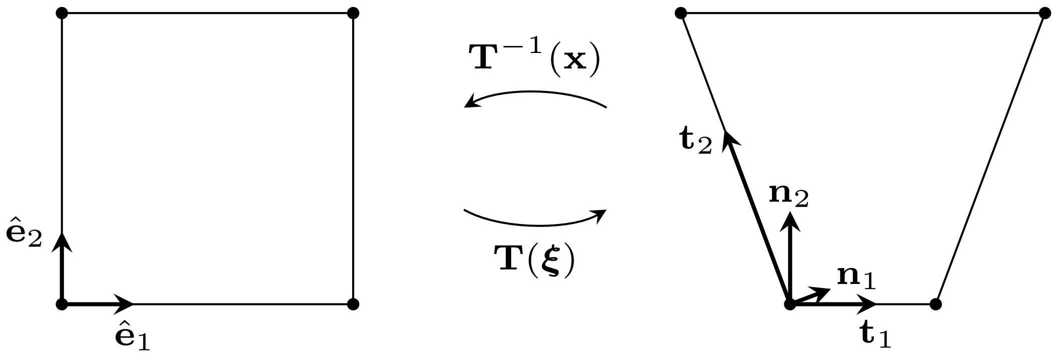


Figure 4.10: A depiction of the tangent and cotangent bases at the point $\boldsymbol{\xi} = (0, 0)$ under a non-affine mesh transformation.

Since the $\mathbf{t}_i$ are the columns of the Jacobian matrix, defining the cotangent basis as the rows of the inverse of the Jacobian matrix satisfies $\mathbf{n}_i \cdot \mathbf{t}_j = \delta_{ij}$ since $\mathbf{F}^{-1}\mathbf{F} = \mathbf{I}$. In other words, the cotangent basis is defined such that

$$\mathbf{F}^{-1} = \begin{bmatrix} \mathbf{n}_1^T \\ \mathbf{n}_2^T \end{bmatrix} . \tag{4.81}$$

For a contravariant vector, the usual method of selecting a component is now replaced with $v_i = \mathbf{n}_i \cdot \boldsymbol{v}$. The cotangent space is associated with vectors normal to the faces. By representing the vector on the tangent space, the contravariant Piola transform allows selection of the component representing the normal component through $\mathbf{n} \cdot \boldsymbol{v}$. Note that for non-affine meshes, $\mathbf{F}$ depends on $\boldsymbol{\xi}$ and thus the tangent and cotangent bases also depend on $\boldsymbol{\xi}$.

Figure 4.10 depicts an example non-affine mesh transformation and the tangent and cotangent bases evaluated at the point $\boldsymbol{\xi} = (0, 0)$. Observe that the pairs $(\mathbf{t}_1, \mathbf{n}_2)$ and $(\mathbf{t}_2, \mathbf{n}_1)$ are perpendicular. The pairs $(\mathbf{t}_1, \mathbf{n}_1)$ and $(\mathbf{t}_2, \mathbf{n}_2)$ do not point in the same direction but their magnitudes and directions balance so that $\mathbf{t}_i \cdot \mathbf{n}_i = 1$. Thus, the bi-orthogonality condition $\mathbf{n}_i \cdot \mathbf{t}_j = \delta_{ij}$ is satisfied. In addition, the tangent vectors and cotangent vectors are tangential and normal, respectively, to one of the faces connecting at the point $\boldsymbol{\xi} = (0, 0)$.

For a contravariant vector,

$$\int_K \nabla u \cdot \boldsymbol{v} \, \mathrm{d}\mathbf{x} = \int_{\hat{K}} \mathbf{F}^{-T} \hat{\nabla} \hat{u} \cdot \frac{1}{J} \mathbf{F} \hat{v} \, J \mathrm{d}\boldsymbol{\xi} = \int_{\hat{K}} \hat{\nabla} \hat{u} \cdot \hat{\boldsymbol{v}} \, \mathrm{d}\boldsymbol{\xi} \, . \tag{4.82}$$

The gradient transforms as

$$\nabla \boldsymbol{v} = \nabla \left(\frac{1}{J} \mathbf{F} \hat{v} \circ \mathbf{T}^{-1} \right) = \frac{1}{J} \mathbf{F} \Big(\hat{\nabla} \hat{\boldsymbol{v}} - \hat{\mathbf{B}} \Big) \mathbf{F}^{-1} \tag{4.83}$$

where

$$\hat{\mathbf{B}} = \frac{1}{J} \hat{\nabla} \big(J \mathbf{F}^{-1} \big) \mathbf{F} \hat{\boldsymbol{v}} \, . \tag{4.84}$$

This result is derived by direct computation in Section 4.4.3 along with the details required to implement this transformation using the machinery commonly provided in finite element libraries. It is also shown that $\hat{\mathbf{B}} = 0$ when the mesh transformation is affine and that $\mathrm{trace}(\hat{\mathbf{B}}) = 0$. This last result is known as the Piola identity [75]. Using the Piola identity, the linearity of the trace, and the invariance of the trace under similarity transformations, the divergence transforms as

$$\nabla \cdot \boldsymbol{v} = \mathrm{trace}\left(\nabla \boldsymbol{v}\right) = \frac{1}{J}\,\mathrm{trace}\Big(\mathbf{F}\Big(\hat{\nabla}\hat{\boldsymbol{v}} - \hat{\mathbf{B}}\Big)\mathbf{F}^{-1}\Big) = \frac{1}{J}\hat{\nabla} \cdot \hat{\boldsymbol{v}}\,. \tag{4.85}$$

Thus,

$$\int_K u\,\nabla \cdot \boldsymbol{v}\,\mathrm{d}\mathbf{x} = \int_{\hat{K}} \hat{u}\,\hat{\nabla} \cdot \hat{\boldsymbol{v}}\,\mathrm{d}\boldsymbol{\xi}\,. \tag{4.86}$$

Combining the results from Eqs. 4.82 and 4.86 yields:

$$\int_{\partial K} u\,\boldsymbol{v} \cdot \mathbf{n}\,\mathrm{d}s = \int_{\partial \hat{K}} \hat{u}\,\hat{\boldsymbol{v}} \cdot \hat{\mathbf{n}}\,\mathrm{d}\hat{s}\,, \tag{4.87}$$

where $\hat{\mathbf{n}}$ is the normal vector in reference space corresponding to the physical space normal $\mathbf{n}$. In other words, the contravariant Piola transformation preserves the normal component.

4.4.3 The Gradient of the Piola Transformation

Here we derive a formula for the transformation of the gradient of a vector defined under the contravariant Piola transformation. This result is crucial for implementing the Raviart Thomas (RT) discretization of the VEF moment system. We present the derivation in its entirety since this result is not commonly included in finite element textbooks. For the contravariant Piola transform $\boldsymbol{v} = \frac{1}{J}\mathbf{F}\hat{\boldsymbol{v}} \circ \mathbf{T}^{-1}$ the inverse transform is:

$$\hat{\boldsymbol{v}} = J\mathbf{F}^{-1}\boldsymbol{v} \circ \mathbf{T}\,. \tag{4.88}$$

Here, we seek to derive

$$\hat{\nabla}\hat{\boldsymbol{v}} = \hat{\nabla}\big(J\mathbf{F}^{-1}\boldsymbol{v}\big)\ , \tag{4.89}$$

so that we can solve for $\nabla\boldsymbol{v}$. The goal is to derive the functional form of the transformation in terms of functionality commonly implemented in finite element codes. That is, we cast the computation in terms of the Jacobian matrix and Hessian of the transformation.

Through their connection to the Jacobian matrix and the inverse of the Jacobian matrix, the tangent and cotangent spaces are related by

$$\mathbf{n}_1 = \mathbf{t}_2 \times \hat{\mathbf{e}}_3\,, \quad \mathbf{n}_2 = \hat{\mathbf{e}}_3 \times \mathbf{t}_1\,, \tag{4.90}$$

where $\hat{\mathbf{e}}_3$ points out of the page. In other words, $\mathbf{n}_1$ is a 90 degree clockwise rotation of $\mathbf{t}_2$ and $\mathbf{n}_2$ is a 90 degree counterclockwise rotation of $\mathbf{t}_1$ (see Fig. 4.10). Thus, we can write

$$\begin{aligned}
\hat{\nabla}\hat{\boldsymbol{v}} &= \hat{\nabla}\begin{bmatrix} J\mathbf{n}_1\cdot\boldsymbol{v} \\ J\mathbf{n}_2\cdot\boldsymbol{v}\end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial}{\partial\xi}(J\mathbf{n}_1\cdot\boldsymbol{v}) & \frac{\partial}{\partial\eta}(J\mathbf{n}_1\cdot\boldsymbol{v}) \\ \frac{\partial}{\partial\xi}(J\mathbf{n}_2\cdot\boldsymbol{v}) & \frac{\partial}{\partial\eta}(J\mathbf{n}_2\cdot\boldsymbol{v})\end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial}{\partial\xi}(J\mathbf{n}_1)\cdot\boldsymbol{v} & \frac{\partial}{\partial\eta}(J\mathbf{n}_1)\cdot\boldsymbol{v} \\ \frac{\partial}{\partial\xi}(J\mathbf{n}_2)\cdot\boldsymbol{v} & \frac{\partial}{\partial\eta}(J\mathbf{n}_2)\cdot\boldsymbol{v}\end{bmatrix} + \begin{bmatrix} J\mathbf{n}_1\cdot\frac{\partial\boldsymbol{v}}{\partial\xi} & J\mathbf{n}_1\cdot\frac{\partial\boldsymbol{v}}{\partial\eta} \\ J\mathbf{n}_2\cdot\frac{\partial\boldsymbol{v}}{\partial\xi} & J\mathbf{n}_2\cdot\frac{\partial\boldsymbol{v}}{\partial\eta}\end{bmatrix}\,.
\end{aligned} \tag{4.91}$$

The second term can be written as

$$\begin{bmatrix} J\mathbf{n}_1\cdot\frac{\partial\boldsymbol{v}}{\partial\xi} & J\mathbf{n}_1\cdot\frac{\partial\boldsymbol{v}}{\partial\eta} \\ J\mathbf{n}_2\cdot\frac{\partial\boldsymbol{v}}{\partial\xi} & J\mathbf{n}_2\cdot\frac{\partial\boldsymbol{v}}{\partial\eta}\end{bmatrix} = J\mathbf{F}^{-1}\hat{\nabla}\boldsymbol{v} = J\mathbf{F}^{-1}\nabla\boldsymbol{v}\mathbf{F}\,, \tag{4.92}$$

where $\hat{\nabla}\boldsymbol{v} = \nabla\boldsymbol{v}\mathbf{F}$ transforms the reference gradient to the physical gradient. The first term is a third-order tensor contracted with a vector to yield a second-order tensor. By expanding the dot products, we can emulate this contraction as a sum of two second-order tensors:

$$\begin{aligned}
\begin{bmatrix} \frac{\partial}{\partial\xi}(J\mathbf{n}_1)\cdot\boldsymbol{v} & \frac{\partial}{\partial\eta}(J\mathbf{n}_1)\cdot\boldsymbol{v} \\ \frac{\partial}{\partial\xi}(J\mathbf{n}_2)\cdot\boldsymbol{v} & \frac{\partial}{\partial\eta}(J\mathbf{n}_2)\cdot\boldsymbol{v}\end{bmatrix} &= \begin{bmatrix} \frac{\partial}{\partial\xi}(Jn_{11})v_1 + \frac{\partial}{\partial\xi}(Jn_{12})v_2 & \frac{\partial}{\partial\eta}(Jn_{11})v_1 + \frac{\partial}{\partial\eta}(Jn_{12})v_2 \\ \frac{\partial}{\partial\xi}(Jn_{21})v_1 + \frac{\partial}{\partial\xi}(Jn_{22})v_2 & \frac{\partial}{\partial\eta}(Jn_{21})v_1 + \frac{\partial}{\partial\eta}(Jn_{22})v_2\end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial}{\partial\xi}(Jn_{11}) & \frac{\partial}{\partial\eta}(Jn_{11}) \\ \frac{\partial}{\partial\xi}(Jn_{21}) & \frac{\partial}{\partial\eta}(Jn_{21})\end{bmatrix} v_1 + \begin{bmatrix} \frac{\partial}{\partial\xi}(Jn_{12}) & \frac{\partial}{\partial\eta}(Jn_{12}) \\ \frac{\partial}{\partial\xi}(Jn_{22}) & \frac{\partial}{\partial\eta}(Jn_{22})\end{bmatrix} v_2 \\
&= \hat{\nabla}(J\mathbf{F}_1^{-1})v_1 + \hat{\nabla}(J\mathbf{F}_2^{-1})v_2
\end{aligned} \tag{4.93}$$

where $\mathbf{F}_i^{-1}$ are the columns of $\mathbf{F}^{-1}$. Typically, finite element codes provide the Hessian matrix of the forward map but not the inverse map. Thus, to leverage existing functionality, we must write the above matrices in terms of $\mathbf{H} = \hat{\nabla}\mathbf{F}$ instead of $\hat{\nabla}\mathbf{F}^{-1}$. Assume that the code computes the Hessian matrix in *flattened* and symmetric form as:

$$\langle\mathbf{H}\rangle = \begin{bmatrix} \frac{\partial^2 x}{\partial\xi^2} & \frac{\partial^2 x}{\partial\xi\partial\eta} & \frac{\partial^2 x}{\partial\eta^2} \\ \frac{\partial^2 y}{\partial\xi^2} & \frac{\partial^2 y}{\partial\xi\partial\eta} & \frac{\partial^2 y}{\partial\eta^2}\end{bmatrix}\,. \tag{4.94}$$

Then the above can be rewritten as

$$\begin{aligned}
\hat{\nabla}(J\mathbf{F}_1^{-1}) &= \hat{\nabla}\begin{bmatrix} F_{22} \\ -F_{21}\end{bmatrix} \\
&= \hat{\nabla}\begin{bmatrix} \frac{\partial y}{\partial\eta} \\ -\frac{\partial y}{\partial\xi}\end{bmatrix} \\
&= \begin{bmatrix} \frac{\partial^2 y}{\partial\xi\partial\eta} & \frac{\partial^2 y}{\partial\eta^2} \\ -\frac{\partial^2 y}{\partial\xi^2} & -\frac{\partial^2 y}{\partial\xi\partial\eta}\end{bmatrix} \\
&= \begin{bmatrix} H_{22} & H_{23} \\ -H_{21} & -H_{22}\end{bmatrix}\,,
\end{aligned} \tag{4.95}$$

$$\begin{aligned}\hat{\nabla}(J\mathbf{F}_2^{-1}) &= \hat{\nabla}\begin{bmatrix}-F_{12}\\ F_{11}\end{bmatrix}\\ &= \hat{\nabla}\begin{bmatrix}-\frac{\partial x}{\partial \eta}\\ \frac{\partial x}{\partial \xi}\end{bmatrix}\\ &= \begin{bmatrix}-\frac{\partial^2 x}{\partial\xi\partial\eta} & -\frac{\partial^2 x}{\partial\eta^2}\\ \frac{\partial^2 x}{\partial\xi^2} & \frac{\partial^2 x}{\partial\xi\partial\eta}\end{bmatrix}\\ &= \begin{bmatrix}-H_{12} & -H_{13}\\ H_{11} & H_{12}\end{bmatrix}.\end{aligned} \tag{4.96}$$

We can define the matrix

$$\hat{\mathbf{B}} = \hat{\nabla}(J\mathbf{F}^{-1})\boldsymbol{v} = \begin{bmatrix}H_{22} & H_{23}\\ -H_{21} & -H_{22}\end{bmatrix} v_1 + \begin{bmatrix}-H_{12} & -H_{13}\\ H_{11} & H_{12}\end{bmatrix} v_2\,. \tag{4.97}$$

This is computed in flattened form as

$$\begin{aligned}\langle\hat{\mathbf{B}}\rangle &= \begin{bmatrix}\langle\hat{\nabla}(J\mathbf{F}_1^{-1})\rangle & \langle\hat{\nabla}(J\mathbf{F}_2^{-1})\rangle\end{bmatrix}\boldsymbol{v}\\ &= \begin{bmatrix}H_{22} & -H_{12}\\ H_{23} & -H_{13}\\ -H_{21} & H_{11}\\ -H_{22} & H_{12}\end{bmatrix}\frac{1}{J}\mathbf{F}\hat{\boldsymbol{v}}\end{aligned} \tag{4.98}$$

where $\boldsymbol{v} = \frac{1}{J}\mathbf{F}\hat{\boldsymbol{v}}$ was used. Finally, we have that

$$\hat{\nabla}\hat{\boldsymbol{v}} = \hat{\mathbf{B}} + J\mathbf{F}^{-1}\nabla\boldsymbol{v}\mathbf{F} \iff \nabla\boldsymbol{v} = \frac{1}{J}\mathbf{F}\left(\hat{\nabla}\hat{\boldsymbol{v}} - \hat{\mathbf{B}}\right)\mathbf{F}^{-1}\,. \tag{4.99}$$

We can then say that:

$$\begin{aligned}\nabla\boldsymbol{v} : \mathbf{E}\,\mathrm{d}\mathbf{x} &= \frac{1}{J}\mathbf{F}\left(\hat{\nabla}\hat{\boldsymbol{v}} - \hat{\mathbf{B}}\right)\mathbf{F}^{-1} : \mathbf{E}\,J\mathrm{d}\boldsymbol{\xi}\\ &= \left(\hat{\nabla}\hat{\boldsymbol{v}} - \hat{\mathbf{B}}\right) : \mathbf{F}^T\mathbf{E}\mathbf{F}^{-T}\,\mathrm{d}\boldsymbol{\xi}\,.\end{aligned} \tag{4.100}$$

Here, we used the fact that $\mathbf{A} : \mathbf{B} = \mathrm{trace}(\mathbf{A}\mathbf{B}^T)$ and apply the cyclic property of the trace to permute $\mathbf{F}$ and $\mathbf{F}^{-1}$. In this form, we can implement the gradient calculation as a matrix-vector product of the flattened referential gradient and the coefficients of $\hat{\boldsymbol{v}}$.

When the mesh transformation is affine, $\hat{\mathbf{B}} = 0$ since the Hessian of an affine transformation is zero. In addition, the Piola identity states that $\mathrm{trace}\,\hat{\mathbf{B}} = 0$. This can be most easily seen in Eq. 4.97 where

$$\mathrm{trace}\,\hat{\mathbf{B}} = (H_{22} - H_{22})v_1 + (-H_{12} + H_{12})v_2 = 0\,. \tag{4.101}$$

Using the Piola identity and Eq. 4.100, we have that

$$\begin{aligned}\nabla\cdot\boldsymbol{v}\,\mathrm{d}\mathbf{x} &= \nabla\boldsymbol{v} : \mathbf{I}\,\mathrm{d}\mathbf{x} \\ &= \left(\hat{\nabla}\hat{\boldsymbol{v}} - \hat{\mathbf{B}}\right) : \mathbf{F}^T\mathbf{I}\mathbf{F}^{-T}\,\mathrm{d}\boldsymbol{\xi} \\ &= \mathrm{trace}\left(\hat{\nabla}\hat{\boldsymbol{v}} - \hat{\mathbf{B}}\right)\mathrm{d}\boldsymbol{\xi} \\ &= \hat{\nabla}\cdot\hat{\boldsymbol{v}}\,\mathrm{d}\boldsymbol{\xi}\,,\end{aligned} \tag{4.102}$$

so that $\nabla\boldsymbol{v} : \mathbf{I}$ reduces to the standard transformation for the divergence, as expected.

4.4.4 Integration on Embedded Surfaces

We now discuss the machinery needed to integrate a function defined on the mesh over an embedded surface in the mesh. The primary difficulty is that the domain of integration is described by a dim −1-dimensional embedded surface while the integrand is defined as a grid function on the dim-dimensional mesh. We thus need a way to convert from a reference point $\boldsymbol{\xi}_{\mathcal{F}} \in \hat{K}^{\dim -1}$ to a reference point $\boldsymbol{\xi}_e \in \hat{K}^{\dim}$ that corresponds to an adjacent element K_e. This is achieved through *integration point transformations*, $\hat{\mathbf{T}}_e : \hat{K}^{\dim -1} \to \hat{K}^{\dim}$, which map the reference point for the embedded surface to a reference coordinate in an adjacent element K_e such that

$$\mathbf{T}_{\mathcal{F}}(\boldsymbol{\xi}_{\mathcal{F}}) = \mathbf{T}_e(\hat{\mathbf{T}}_e(\boldsymbol{\xi}_{\mathcal{F}}))\,. \tag{4.103}$$

In other words, the points $\boldsymbol{\xi}_{\mathcal{F}} \in \hat{K}^{\dim -1}$ and $\boldsymbol{\xi}_e = \hat{\mathbf{T}}_e(\boldsymbol{\xi}_{\mathcal{F}}) \in \hat{K}^{\dim}$ correspond to the same point in physical space under the action of their associated transformations. An example of an integration point transformation that converts a point on $\hat{K}^2$ to a point on the right face of $\hat{K}^3$ is shown in Fig. 4.11. In this case, points $\boldsymbol{\xi}_{\mathcal{F}} = \begin{bmatrix}\xi_{\mathcal{F}} & \eta_{\mathcal{F}}\end{bmatrix}^T$ are mapped to the plane $\xi = 1$ such that $\boldsymbol{\xi} = \begin{bmatrix}1 & \xi_{\mathcal{F}} & \eta_{\mathcal{F}}\end{bmatrix}^T$.

If $\mathcal{F} = K_1 \cap K_2$ and $f(u_1, u_2)$ is a function that depends on the functions $u_e : K_e \to \mathbb{R}$ defined on the elements that share the face $\mathcal{F}$, the integral

$$\int_{\mathcal{F}} f(u_1, u_2)\,\mathrm{d}s \tag{4.104}$$

can be computed in reference space using

$$\int_{\hat{K}^{\dim -1}} f(\bar{u}_1, \bar{u}_2)\,J_{\mathcal{F}}\mathrm{d}\boldsymbol{\xi}_{\mathcal{F}}\,, \tag{4.105}$$

where $\bar{u}_e(\boldsymbol{\xi}_{\mathcal{F}}) = u_e(\mathbf{T}_e(\hat{\mathbf{T}}_e(\boldsymbol{\xi}_{\mathcal{F}})))$ is a function of the reference coordinate of the dim −1-dimensional face through the composition with the integration point transformation, $\hat{\mathbf{T}}_e$, and the volumetric transformation, $\mathbf{T}_e$, associated with element K_e. In this way, any function can be evaluated using the dim-dimensional grid function by first transforming the dim −1-dimensional location $\boldsymbol{\xi}_{\mathcal{F}}$ to the dim-dimensional point associated with an adjacent element K_e through $\boldsymbol{\xi}_e = \hat{\mathbf{T}}_e(\boldsymbol{\xi}_{\mathcal{F}})$. Note that if $f = f(u_1, u_2, \nabla u_1, \nabla u_2)$, the volumetric transformations discussed in this section can be applied to compute ∇u_1 and ∇u_2 by first converting the reference point on the face to a reference point on the volume.

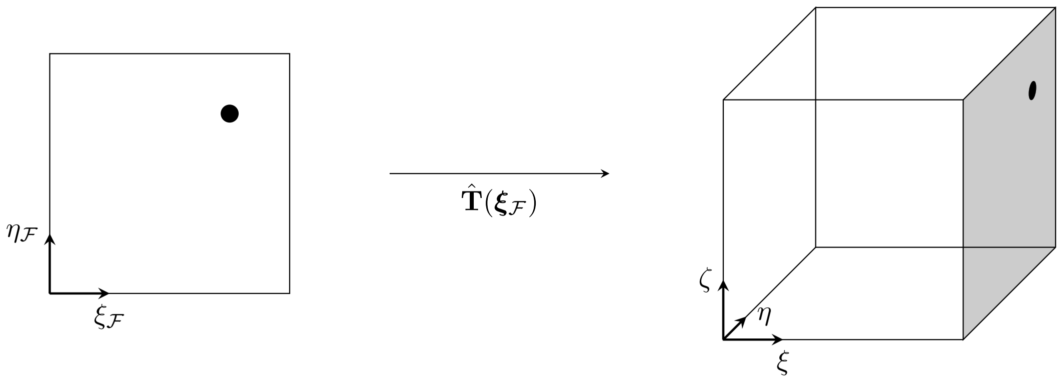


Figure 4.11: An example of an integration point transformation that maps $\hat{K}^2$ to the right face of $\hat{K}^3$. A point $\boldsymbol{\xi}_\mathcal{F} = \begin{bmatrix}\xi_\mathcal{F} & \eta_\mathcal{F}\end{bmatrix}^T$ is mapped to the plane $\xi = 1$ of the three-dimensional reference element such that $\boldsymbol{\xi} = \begin{bmatrix}1 & \xi_\mathcal{F} & \eta_\mathcal{F}\end{bmatrix}^T$. The points $\boldsymbol{\xi}_\mathcal{F}$ and $\boldsymbol{\xi} = \hat{\mathbf{T}}(\boldsymbol{\xi}_\mathcal{F})$ satisfy $\mathbf{T}_\mathcal{F}(\boldsymbol{\xi}_\mathcal{F}) = \mathbf{T}(\boldsymbol{\xi})$ where $\mathbf{T}_\mathcal{F}$ and $\mathbf{T}$ are the reference to physical space transformations for the embedded face $\mathcal{F}$ and the volumetric element K, respectively.

4.5 Finite Element Spaces

Finite element spaces are defined on the mesh $\mathcal{T}$ or the interior skeleton of the mesh Γ_0 and consist of an element-local function space and a set of inter-element matching conditions. The inter-element matching conditions enforce various types of continuity of the solution between elements. The combination of a locally smooth function space and suitable matching conditions allows finite element spaces to be discrete subspaces of Sobolev spaces such as $L^2(\mathcal{D})$, $H^1(\mathcal{D})$, and $H(\mathrm{div};\mathcal{D})$. The following subsections define the element-local function spaces and matching conditions used to discretize the transport and moment equations in subsequent chapters.

4.5.1 Discontinuous Galerkin

The DG space is a discrete subspace of $L^2(\mathcal{D})$, the space of square-integrable functions. In other words, if u is an element of the DG space,

$$\int u^2 \,\mathrm{d}\mathbf{x} < \infty \,. \tag{4.106}$$

Since only square integrability is required, functions in $L^2(\mathcal{D})$, and thus DG spaces, do not need to be continuous. DG functions are represented using piecewise-discontinuous polynomials that are defined on the reference element and mapped to the physical element using the inverse mesh transformation $\mathbf{T}^{-1} : K \rightarrow \hat{K}$. In other words, on each element, the solution belongs to:

$$\mathbb{Q}_p(K) = \{u = \hat{u} \circ \mathbf{T}^{-1} : \hat{u} \in \mathcal{Q}_p(\hat{K})\} \,. \tag{4.107}$$

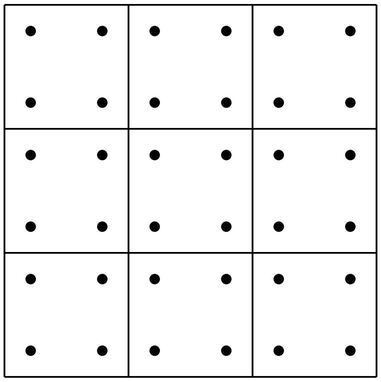


Figure 4.12: A depiction of the distribution of degrees of freedom in the linear DG space. The Legendre nodes are used to illustrate that degrees of freedom are not shared between elements.

The distinction between $\mathcal{Q}_p(\hat{K})$ and $\mathbb{Q}_p(K)$ is important for non-affine mesh transformations. In such case, the inverse mesh transformation is generally non-polynomial so that the composition $u = \hat{u} \circ \mathbf{T}^{-1}$ is also non-polynomial.

The degree-p DG space is

$$Y_p = \{u \in L^2(\mathcal{D}) : u|_K \in \mathbb{Q}_p(K) \,, \quad \forall K \in \mathcal{T}\} \,. \tag{4.108}$$

An example of the distribution of the degrees of freedom in a linear DG space on a 3×3 mesh is shown in Fig. 4.12. Note that degrees of freedom are not shared between elements. Since there are no continuity requirements in the DG space, the basis for the local polynomials can use either open or closed points. That is, a nodal basis can be formed with Lagrange interpolating polynomials through the dim-fold Cartesian product of either the closed Gauss-Lobatto points or the open Gauss-Legendre points.

We additionally define the vector-valued DG space

$$X_p = \{\boldsymbol{v} \in [L^2(\mathcal{D})]^{\dim} : \boldsymbol{v}_i \in Y_p \,, \quad 1 \le i \le \dim\} \,. \tag{4.109}$$

This space uses the scalar DG space for each component.

4.5.2 Continuous Finite Element

Let the degree-p, scalar continuous finite element space be

$$V_p = \{u \in C_0(\mathcal{D}) : u|_{K_e} \in \mathbb{Q}_p(K_e) \,, \quad \forall K_e \in \mathcal{T}\} \tag{4.110}$$

so that each function $u \in V_p$ is a piecewise-continuous polynomial mapped from the reference element. Since $u \in V_p$ is locally smooth and $V_p \subset C_0(\mathcal{D})$, it can be shown that given $u \in V_p$, $\nabla u = \nabla_h u \in [L^2(\mathcal{D})]^2$ [66, cf. Prop. 3.2.1]. Thus, $V_p \subset H^1(\mathcal{D})$. The distribution of degrees

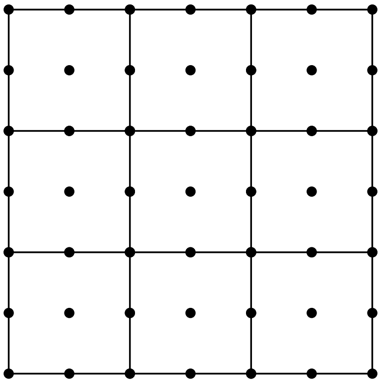


Figure 4.13: A depiction of the distribution of degrees of freedom for the quadratic continuous finite element space. Continuity of the members of the finite element space is enforced by sharing degrees of freedom across neighboring elements.

of freedom for the space V_2 is shown in Fig. 4.13. Here, continuity is enforced by sharing degrees of freedom between elements. Due to this, a nodal basis using closed points, such as the Gauss-Lobatto points, must be used.

The vector-valued analog

$$W_p = \{\boldsymbol{v} \in [H^1(\mathcal{D})]^{\dim} : v_i \in V_p\,, \quad 1 \leq i \leq \dim\} \tag{4.111}$$

uses the scalar continuous finite element space for each component. In this way, $\boldsymbol{v} \in W_p \subset [H^1(\mathcal{D})]^{\dim}$ and thus $\nabla \boldsymbol{v} = \nabla_h \boldsymbol{v} \in [L^2(\mathcal{D})]^{\dim \times \dim}$. Since each component is defined independently using the scalar space, vectors $\boldsymbol{v} \in W_p$ transform according to Eq. 4.74.

4.5.3 Raviart Thomas

The RT space [78, 79] is a discrete subspace of $H(\mathrm{div}; \mathcal{D})$, the space of vector-valued functions with square-integrable divergence. That is,

$$H(\mathrm{div}; \mathcal{D}) = \{\boldsymbol{v} \in [L^2(\mathcal{D})]^2 : \nabla \cdot \boldsymbol{v} \in L^2(\mathcal{D})\}\,. \tag{4.112}$$

The requirements of a discrete subspace are codified in the following proposition.

Proposition 4.2 (Cf. Quarteroni and Valli [66] Prop. 3.2.2)**.** *Let $\boldsymbol{v} : \mathcal{D} \to \mathbb{R}^2$ be such that*

1. $\boldsymbol{v}|_K \in [H^1(K)]^2$ *for each* $K \in \mathcal{T}$
2. $[\![\boldsymbol{v} \cdot \mathbf{n}]\!] = 0$ *for each* $\mathcal{F} \in \Gamma_0$

then $\boldsymbol{v} \in H(\mathrm{div}; \mathcal{D})$*. Conversely, if* $\boldsymbol{v} \in H(\mathrm{div}; \mathcal{D})$ *and (a) is satisfied, then (b) holds.*

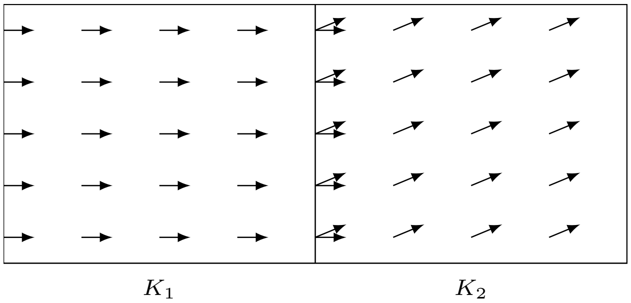


Figure 4.14: An example vector in $H(\text{div};\mathcal{D})$. Note that the normal component is continuous across the shared face between the two elements while the tangential component is discontinuous.

Proof. It must be shown that, given (a) and (b), $\nabla\cdot\boldsymbol{v}\in L^2(\mathcal{D})$. From (a), $\nabla_h\cdot\boldsymbol{v}\in L^2(\mathcal{D})$. Using Green's identity and (b):

$$\begin{aligned}\int u\nabla_h\cdot\boldsymbol{v}\,\mathrm{d}\mathbf{x} &= \sum_{K\in\mathcal{T}}\left[\int_{\partial K} u\boldsymbol{v}\cdot\mathbf{n}\,\mathrm{d}s-\int_K \nabla u\cdot\boldsymbol{v}\,\mathrm{d}\mathbf{x}\right]\\ &= \int_{\Gamma_0} u\,[\![\boldsymbol{v}\cdot\mathbf{n}]\!]\,\mathrm{d}s-\int\nabla u\cdot\boldsymbol{v}\,\mathrm{d}\mathbf{x}\\ &= \int u\,\nabla\cdot\boldsymbol{v}\,\mathrm{d}\mathbf{x}\,,\end{aligned}\tag{4.113}$$

holds for u sufficiently smooth and vanishing on the boundary (i.e. $u\in C_0^\infty(\mathcal{D})$). Thus, $\nabla\cdot\boldsymbol{v}=\nabla_h\cdot\boldsymbol{v}\in L^2(\mathcal{D})$.

On the other hand, if $\boldsymbol{v}\in H(\text{div};\mathcal{D})$ then $\nabla\cdot\boldsymbol{v}=\nabla_h\cdot\boldsymbol{v}$ and, given $\boldsymbol{v}|_K\in[H^1(K)]^2$, we obtain

$$\int_{\Gamma_0} u\,[\![\boldsymbol{v}\cdot\mathbf{n}]\!]\,\mathrm{d}s=0\,,\quad \forall u\in C_0^\infty(\mathcal{D})\,,\tag{4.114}$$

hence, (b) holds. □

In other words, a discrete subspace of $H(\text{div};\mathcal{D})$ must (a) have a smooth function space on each element and (b) have suitable matching conditions so that the normal component is continuous across interior mesh interfaces. Figure 4.14 depicts an example vector in the space $H(\text{div};\mathcal{D})$. The vector field is piecewise discontinuous on each element but since the normal component is the same in K_1 and K_2, this vector field is a member of $H(\text{div};\mathcal{D})$.

In two spatial dimensions, the RT space uses the local polynomial space $\mathcal{Q}_{p+1,p}(\hat{K})\times\mathcal{Q}_{p,p+1}(\hat{K})$. This choice can be motivated by the discrete de Rham complex [80] in that

$$\mathcal{Q}_{p+1}(\hat{K})\xrightarrow{\nabla\times}\mathcal{Q}_{p+1,p}(\hat{K})\times\mathcal{Q}_{p,p+1}(\hat{K})\xrightarrow{\nabla\cdot}\mathcal{Q}_p(\hat{K})\,.\tag{4.115}$$

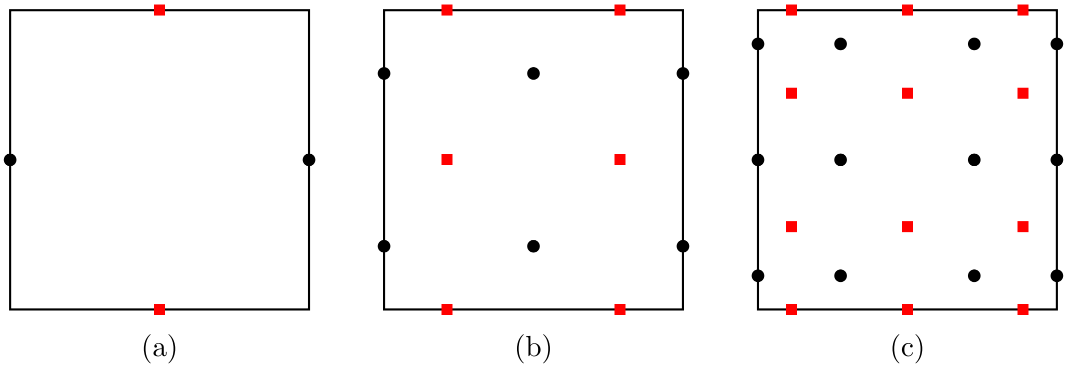


Figure 4.15: The interpolating points used for the nodal basis of the space $\mathcal{Q}_{p+1,p}(\hat{K}) \times \mathcal{Q}_{p,p+1}(\hat{K})$ for (a) $p = 0$, (b) $p = 1$, and (c) $p = 2$. Gauss-Legendre points are used in the tangential direction and Gauss-Lobatto in the normal direction for each component of the vector. Circles denote the degrees of freedom associated with the ξ component and squares the η component.

As an example, the lowest-order polynomial space is

$$\mathcal{Q}_{1,0}(\hat{K}) \times \mathcal{Q}_{0,1}(\hat{K}) = \text{span}\left\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} \xi \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ \eta \end{pmatrix} \right\}, \tag{4.116}$$

and thus we have that:

$$\hat{\nabla} \cdot \mathcal{Q}_{1,0}(\hat{K}) \times \mathcal{Q}_{0,1}(\hat{K}) = \text{span}\{1\} = \mathcal{Q}_0(\hat{K}) \,. \tag{4.117}$$

The nodal basis for $\mathcal{Q}_{p+1,p}(\hat{K}) \times \mathcal{Q}_{p,p+1}(\hat{K})$ uses the closed Gauss-Lobatto points in the normal direction and the open Gauss-Legendre points in the tangential direction. The interpolating points for the first three orders are shown in Fig. 4.15. The circles denote degrees of freedom corresponding to the ξ component while squares denote the η component. In three dimensions, the local polynomial space is spanned by $\mathcal{Q}_{p+1,p,p}(\hat{K}) \times \mathcal{Q}_{p,p+1,p}(\hat{K}) \times \mathcal{Q}_{p,p,p+1}(\hat{K})$.

The contravariant Piola transformation is used to allow sharing the degrees of freedom associated with the normal component with neighboring elements. Combining the local function space $\mathcal{Q}_{p+1,p}(\hat{K}) \times \mathcal{Q}_{p,p+1}(\hat{K})$ with the contravariant Piola transform yields:

$$\mathbb{D}_p(K) = \{\boldsymbol{v} = \frac{1}{J}\mathbf{F}\hat{\boldsymbol{v}} \circ \mathbf{T}^{-1} : \hat{\boldsymbol{v}} \in \mathcal{Q}_{p+1,p}(\hat{K}) \times \mathcal{Q}_{p,p+1}(\hat{K})\} \,. \tag{4.118}$$

In three dimensions, the element-local polynomial space is:

$$\mathbb{D}_p(K) = \{\boldsymbol{v} = \frac{1}{J}\mathbf{F}\hat{\boldsymbol{v}} \circ \mathbf{T}^{-1} : \hat{\boldsymbol{v}} \in \mathcal{Q}_{p+1,p,p}(\hat{K}) \times \mathcal{Q}_{p,p+1,p}(\hat{K}) \times \mathcal{Q}_{p,p,p+1}(\hat{K})\} \,. \tag{4.119}$$

Here, both the inverse mesh transformation and $1/J$ are generally non-polynomial when $\mathbf{T}$ is non-affine.

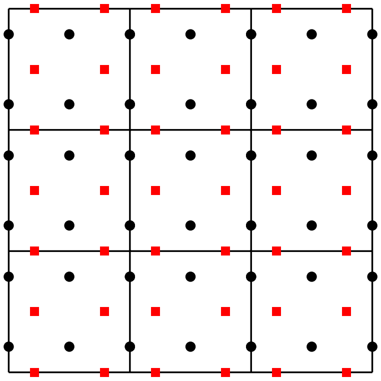


Figure 4.16: The distribution of degrees of freedom corresponding to the first degree Raviart Thomas space. Continuity of the normal component is enforced by sharing the degrees of freedom corresponding to the normal component along the interior face between neighboring elements. The circles and squares denote degrees of freedom in the x and y directions, respectively.

We now define the degree-p RT space as:

$$RT_p = \{\boldsymbol{v} \in [L^2(\mathcal{D})]^{\dim} : \boldsymbol{v}|_K \in \mathbb{D}_p(K) \ \forall K \in \mathcal{T} \text{ and } [\![\boldsymbol{v} \cdot \mathbf{n}]\!] = 0 \ \forall \mathcal{F} \in \Gamma_0\} \,. \tag{4.120}$$

In other words, RT_p uses the element-local polynomial space $\mathbb{D}_p(K)$ on each element and shares degrees of freedom so that the normal component is continuous across each interior face. Note that since the contravariant Piola transform is used, functions in RT_p transform according to Eqs. 4.77, 4.83, and 4.85. The location of the degrees of freedom for RT_1 are shown on a 3×3 mesh in Fig. 4.16. Continuity in the normal component is enforced by sharing the degrees of freedom corresponding to the normal component on interior faces. From Proposition 4.2, $\boldsymbol{v} \in RT_p$ satisfies $\nabla \cdot \boldsymbol{v} = \nabla_h \cdot \boldsymbol{v} \in L^2(\mathcal{D})$. However, the RT space does not have the continuity to allow a square-integrable gradient. In other words, $\nabla \boldsymbol{v} \notin [L^2(\mathcal{D})]^{2\times 2}$ and $\nabla \boldsymbol{v} \neq \nabla_h \boldsymbol{v}$.

4.5.4 Raviart Thomas Trace Space

The normal trace of the RT space is required for the hybridization procedure discussed in Section 7.4. This space is defined on the interior skeleton of the mesh Γ_0 and represents the normal component of the RT space along the interior mesh faces. The RT trace space is:

$$\Lambda_p = \{\mu \in L^2(\Gamma_0) : \mu|_{\mathcal{F}} \in \mathcal{Q}_p(\hat{K}^{\dim -1}) \circ \mathbf{T}_{\mathcal{F}}^{-1} \,, \quad \forall \mathcal{F} \in \Gamma_0\} \,. \tag{4.121}$$

The degrees of freedom in Λ_1 are depicted in Fig. 4.17. Note that these degrees of freedom are exactly the degrees of freedom corresponding to the normal component of RT_1 on the interior faces of the mesh.

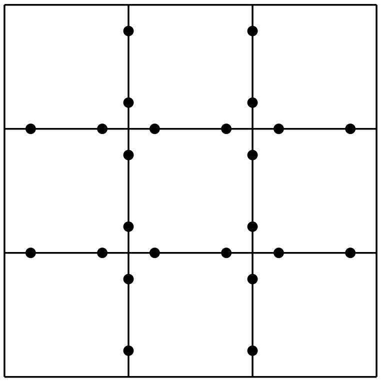


Figure 4.17: The distribution of degrees of freedom corresponding to Λ_1, the space defined as the normal trace of the first degree Raviart Thomas space, on a 3×3 mesh.

4.6 Computational Aspects

In the previous section, we defined piecewise polynomial spaces that, through a clever choice of degrees of freedom, are finite-dimensional subspaces of the Sobolev spaces introduced in Section 4.1.4. In this section, we discuss their computer implementation. A canonical basis with small support such that each basis function is non-zero on only a few elements in the mesh is built. We then discuss finite element assembly which leverages the small support of the basis functions to achieve an efficient algorithm for forming the matrices associated with a finite element space and bilinear or linear form.

4.6.1 The Canonical Basis

Let X_h denote one of the finite element spaces described in the previous section and $\mathcal{N}_h$ denote the set of nodes associated with X_h. We also let $\mathcal{Q}(K)$ denote the element-local polynomial space for X_h such that given $u \in X_h$, $u|_K \in \mathcal{Q}(K)$ for each $K \in \mathcal{T}$ and use $\{\ell_i\}$ to denote its nodal basis. For example, if $X_h = V_1$, the linear continuous finite element space, then $\mathcal{N}_h$ is the set of locations, $\mathbf{x}_i$, in the domain corresponding to a vertex in the mesh and $\mathcal{Q}(K) = \mathbb{Q}_p(K)$. A function $u_h \in X_h$ is determined by the values it takes at the nodes in $\mathcal{N}_h$. That is, $u_h \in X_h$ is determined by the set

$$\Sigma_h = \{u_h(\mathbf{x}_i)\,, \quad \forall \mathbf{x}_i \in \mathcal{N}_h\}\,, \tag{4.122}$$

called the set of degrees of freedom for the space X_h. Defining functions $b_i \in X_h$ that satisfy

$$b_i(\mathbf{x}_j) = \delta_{ij}\,, \quad 1 \le i, j \le \dim(X_h)\,, \tag{4.123}$$

it is seen that $\{b_i\}$ forms a basis for X_h. The b_i are referred to as the global basis functions. Since each $b_i \in X_h$, we have that $b_i|_K \in \mathcal{Q}(K)$. In particular, we choose the b_i so that their restrictions in each element exactly align with one of the element-local nodal basis functions ℓ_i. In this way, the global representation of members of X_h is built up from the local representation using $\mathcal{Q}(K)$ on each element. Through the sharing of nodes between neighboring elements (e.g. the choice of $\mathcal{N}_h$), the requirements of the finite element spaces in Section 4.5 are satisfied. Using this basis, a function $u \in X_h$ can be represented as

$$u(\mathbf{x}) = \sum_i b_i(\mathbf{x}) u_i \tag{4.124}$$

where $u_i \in \Sigma_h$ is the degree of freedom corresponding to b_i.

A selection of basis functions for V_1, V_2, and V_3 in one dimension are depicted in Fig. 4.18. Observe that each b_i is one at x_i and zero for each x_j where $i \neq j$ and that each basis function has a localized support. That is, the basis functions associated with the boundary of the domain are non-zero only on a single element. This is also true for the "bubble" basis functions corresponding to a node that is on the interior of an element (e.g. b_8 in the middle diagram). Basis functions associated with a node shared by multiple elements are non-zero only on those elements. This is demonstrated by b_4, b_7, and b_{10} in the upper, middle, and lower diagrams, respectively. Figure 4.19 shows a selection of basis functions for V_1 in two dimensions. Again, the basis functions are non-zero only in the elements that share a node. This property of local support was constructed by design to to keep the resulting algebraic system as computationally manageable and memory efficient as possible.

For DG spaces, degrees of freedom are not shared between elements. Thus, the global basis is spanned by the collection of element-local basis functions corresponding to each element in the mesh. In this case, the support of each basis function is limited to a single element.

4.6.2 Finite Element Assembly

Consider the abstract, finite-dimensional problem: find $u \in X_h$ such that

$$\mathcal{A}(v, u) = f(v)\,, \quad \forall v \in X_h\,. \tag{4.125}$$

Both the test function v and the solution u are represented as a linear combination of the global basis functions $\{b_i\}$ that span X_h. That is, we write

$$u = \sum_i b_i u_i\,, \quad v = \sum_i b_i v_i\,, \tag{4.126}$$

where $\{u_i\}$ and $\{v_i\}$ are the degrees of freedom that determine $u, v \in X_h$. Inserting this representation into the abstract problem yields

$$\mathcal{A}(\sum_i b_i v_i, \sum_j b_j u_j) = f(\sum_i b_i v_i) \iff \sum_i \sum_j v_i \mathcal{A}(b_i, b_j) u_j = \sum_i v_i f(b_i)\,, \tag{4.127}$$

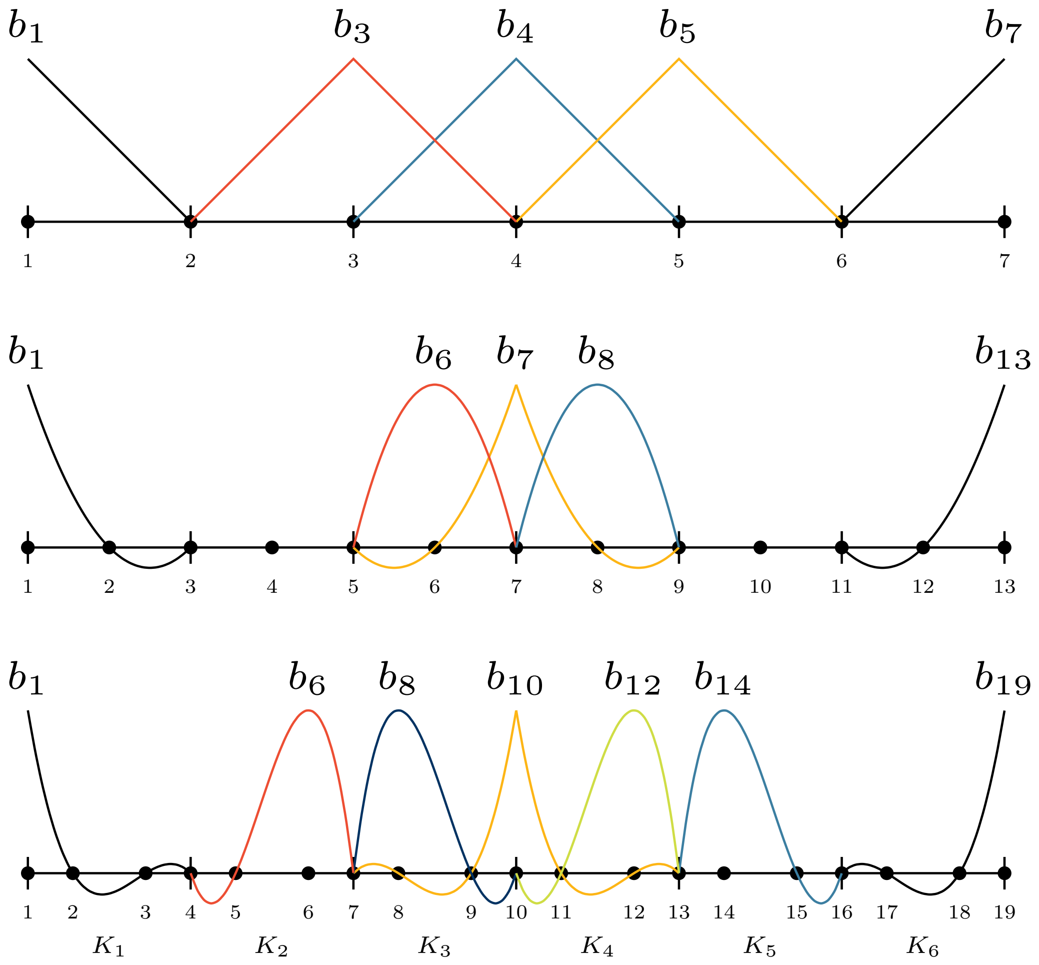


Figure 4.18: A selection of the global basis functions for a linear (upper), quadratic (middle), and cubic (lower) continuous finite element space in one dimension. The basis functions are zero outside of their local support.

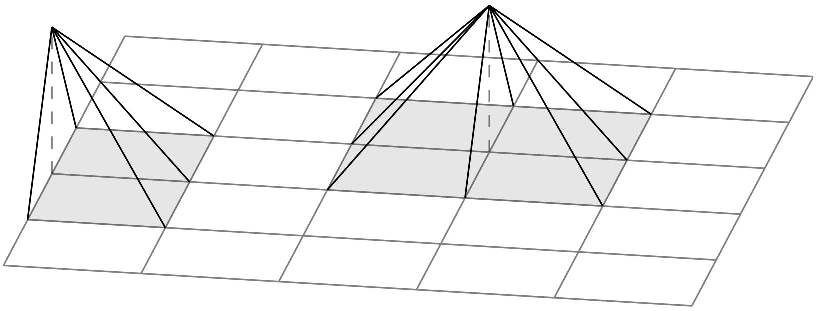


Figure 4.19: A selection of global basis functions for a linear continuous finite element space in two dimensions. The basis functions are non-zero only on the finite number of elements that share a node.

where we have used the bilinearity and linearity of $\mathcal{A}(\cdot,\cdot)$ and $f(\cdot)$, respectively. Letting $\mathbf{A}$ be the matrix with entries $\mathcal{A}(b_i, b_j)$ and $\underline{f}$ the vector with entries $f(b_i)$, the above is equivalent to

$$\underline{v}^T \mathbf{A} \underline{u} = \underline{v}^T \underline{f}\,, \tag{4.128}$$

where $\underline{w}$ represents the vector of degrees of freedom corresponding to some $w \in X_h$. We wish to have this hold $\forall v \in X_h \Rightarrow \forall v_i$. Therefore, solving the abstract problem is equivalent to solving:

$$\mathbf{A} \underline{u} = \underline{f}\,, \tag{4.129}$$

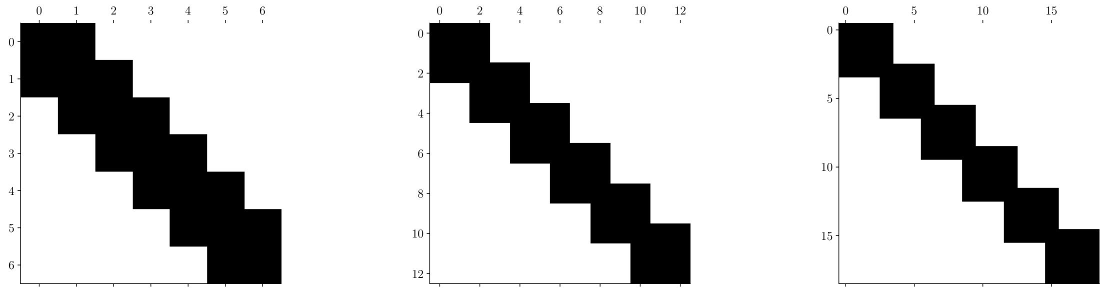
for the vector of degrees freedom $\underline{u}$.

We now show that the algebraic system $\mathbf{A}\underline{u} = \underline{f}$ is sparse in that most of its entries are zero. This fact arises from the local support of the basis functions b_i. Let $\operatorname{supp}(b_i) \subset \mathcal{D}$ denote the subset of the domain where b_i is non-zero. In other words, $\operatorname{supp}(b_i)$ is the union of adjacent elements that share the node located at $\mathbf{x}_i$. Then, $\mathcal{A}(b_i, b_j)$ is non-zero only where $\operatorname{supp}(b_i)$ and $\operatorname{supp}(b_j)$ overlap i.e. when $\operatorname{supp}(b_i) \cap \operatorname{supp}(b_j) \neq \emptyset$. Using the basis functions depicted in the upper diagram of Fig. 4.18 as an example, the term $\mathcal{A}(b_1, b_7)$ is zero since b_7 is zero where b_1 is non-zero and vice versa. On the other hand, $\mathcal{A}(b_4, b_5)$ is non-zero since b_4 and b_5 are both non-zero in K_4. Figure 4.20 shows the resulting sparsity pattern corresponding to the finite element spaces depicted in Fig. 4.18.

The piecewise polynomial nature of the finite element spaces also lends itself to an efficient algorithm for forming the matrices and vectors corresponding to bilinear and linear forms called finite element assembly. The goal is to split the integration over the entire domain into a sum over the individual elements and to group computations associated with each element together. Let $\mathcal{Q}$ be the element-local polynomial space associated with X_h and let $\{\ell_i\}$ be its nodal basis. Element-local matrices are computed according to:

$$(\mathbf{A}_K)_{ij} = \mathcal{A}_K(\ell_i, \ell_j)\,, \quad 1 \leq i,j \leq \dim(\mathcal{Q})\,, \tag{4.130}$$

where $\mathcal{A}_K(\cdot,\cdot)$ is the bilinear form $\mathcal{A}(\cdot,\cdot)$ restricted to element K. The local matrices $\mathbf{A}_K$ are then summed into the global matrix $\mathbf{A}$ using a local to global map, $\mathcal{G}_K$. This map converts



(a) (b) (c)

Figure 4.20: Sparsity patterns associated with the finite element spaces depicted in Fig. 4.18.

the local index i corresponding to the local basis function ℓ_i in element K to its associated global basis function $b_{\mathcal{G}_K(i)}$. In particular, $\mathcal{G}_K$ obeys the matching conditions associated with X_h such that entries corresponding to nodes shared between elements are mapped to the same global index. The finite element assembly algorithm is to perform:

$$\mathbf{A}_{\mathcal{G}_K(i),\mathcal{G}_K(j)} \mathrel{+}= (\mathbf{A}_K)_{ij}\,, \quad 1 \leq i,j \leq \dim(\mathcal{Q})\,, \tag{4.131}$$

for each element $K \in \mathcal{T}$. That is, each entry $(\mathbf{A}_K)_{ij}$ is summed into the global row and column specified by $\mathcal{G}_K(i)$ and $\mathcal{G}_K(j)$, respectively. Linear forms are assembled in an analogous process. The entries of the linear form are computed with

$$\underline{f}_{\mathcal{G}_K(i)} \mathrel{+}= (\underline{f}_K)_i\,, \quad 1 \leq i \leq \dim(\mathcal{Q})\,, \tag{4.132}$$

for each $K \in \mathcal{T}$ where

$$(\underline{f}_K)_i = f_K(\ell_i)\,, \quad 1 \leq i \leq \dim(\mathcal{Q})\,, \tag{4.133}$$

with $f_K = f|_K$.

In this document, we will frequently build the matrix, $\mathbf{A}$, corresponding to a bilinear form, $\mathcal{A}(\cdot,\cdot)$, on a space X_h using the following notation:

$$\underline{v}^T\mathbf{A}\underline{u} = \mathcal{A}(v,u)\,, \quad u,v \in X_h\,. \tag{4.134}$$

In the above, we implicitly write $u = \sum u_i b_i$ and $v = \sum v_i b_i$ to form the matrix $\mathbf{A}_{ij} = \mathcal{A}(b_i, b_j)$. Analogously, we build the vector $\underline{f}$ corresponding to the linear form $f(\cdot)$ using

$$\underline{v}^T\underline{f} = f(v)\,, \tag{4.135}$$

where $v = \sum v_i b_i$ is implicitly used to form $\underline{f}_i = f(b_i)$. In both cases, the efficient finite element assembly algorithm is used to form the matrix $\mathbf{A}$ and vector $\underline{f}$.

4.7 Iterative Solution Methods for Linear Systems

We now discuss iterative solution techniques for solving the linear systems of equations generated by finite element discretization. We consider the algebraic system

$$\mathbf{A}\underline{x} = \underline{b}\,, \tag{4.136}$$

where $\mathbf{A}$ is a non-singular matrix of dimension n. Iterative methods define a sequence of vectors $\underline{x}^k$ such that $\underline{x}^k \to \underline{x} = \mathbf{A}^{-1}\underline{b}$ as $k \to \infty$. In practice, iterative tolerances are used such that the sequence is terminated when the residual, defined as

$$\underline{r}^k = \underline{b} - \mathbf{A}\underline{x}^k\,, \tag{4.137}$$

is small enough. This allows finding an $\underline{x}^K$ that is "close enough" to the true solution in a number of iterations $K \ll n$.

On the other hand, direct methods, such as Gaussian elimination, produce the exact solution in a finite number of operations. In the case of Gaussian elimination, the matrix $\mathbf{A}$ is decomposed into lower and upper triangular matrices. The decomposed system can then be solved in $\mathcal{O}(n^2)$ operations through forward and backward substitution. The decomposition of $\mathbf{A}$ requires $\mathcal{O}(n^3)$ operations and thus direct methods scale as $\mathcal{O}(n^3)$.

In practice, iterative methods are preferred over direct methods for the sparse systems of equations generated by finite element discretization. This is due to the memory cost of storing the inverse of a matrix as $\mathbf{A}^{-1}$ is generally dense even when $\mathbf{A}$ is sparse. Thus, even sparse direct methods, such as Li and Demmel [81], which account for the sparsity in $\mathbf{A}$ in the decomposition process, have "fill-in" from storing the increased number of non-zeros associated with $\mathbf{A}^{-1}$. In addition, the matrix-vector products that make up the dominant cost of each iteration of an iterative method can be implemented in $\mathcal{O}(n)$ operations when sparsity is accounted for. If the iteration converges in K iterations, an iterative method can solve the problem in $\mathcal{O}(Kn)$ operations. An iterative method with $K \ll n$ can then be considerably cheaper in memory and floating point operations compared to a direct method.

4.7.1 Classical Iterative Schemes

Most iterative methods are based on a suitable splitting of $\mathbf{A}$ such that

$$\mathbf{A} = \mathbf{P} - \mathbf{N}\,, \tag{4.138}$$

where $\mathbf{P}$ is non-singular and ideally computationally efficient to invert. The iteration proceeds as

$$\mathbf{P}\underline{x}^{k+1} = \mathbf{N}\underline{x}^k + \underline{b}\,, \tag{4.139}$$

where the superscript denotes iteration index with $\underline{x}^0$ an initial guess. Such an iteration is called a Richardson iteration and converges if the spectral radius of the iteration matrix

$\mathbf{P}^{-1}\mathbf{N}$ is less than unity. By adding and subtracting $\mathbf{A}\underline{x}^k$ on the right hand side, this iteration is equivalent to

$$\mathbf{P}\underline{x}^{k+1} = (\mathbf{N}+\mathbf{A})\underline{x}^k + \underline{b} - \mathbf{A}\underline{x}^k = \mathbf{P}\underline{x}^k + \underline{r}^k\,, \tag{4.140}$$

where the residual vector is defined in Eq. 4.137. Operating by $\mathbf{P}^{-1}$ yields the iteration:

$$\underline{x}^{k+1} = \underline{x}^k + \mathbf{P}^{-1}\underline{r}^k\,. \tag{4.141}$$

Let $\mathbf{A} = \mathbf{D} + \mathbf{L} + \mathbf{U}$ where $\mathbf{D}$, $\mathbf{L}$, and $\mathbf{U}$ are the diagonal, strictly lower triangular, and strictly upper triangular parts of the matrix $\mathbf{A}$. That is,

$$\mathbf{D}_{ij} = \begin{cases} \mathbf{A}_{ij}\,, & i=j \\ 0\,, & \text{otherwise} \end{cases}\,, \quad \mathbf{L}_{ij} = \begin{cases} \mathbf{A}_{ij}\,, & i>j \\ 0\,, & \text{otherwise} \end{cases}\,, \quad \mathbf{U} = \begin{cases} \mathbf{A}_{ij}\,, & i<j \\ 0\,, & \text{otherwise} \end{cases}\,. \tag{4.142}$$

The Jacobi method sets $\mathbf{P} = \mathbf{D}$ and $\mathbf{N} = \mathbf{L} + \mathbf{U}$ so that

$$\underline{x}^{k+1} = \underline{x}_k + \mathbf{D}^{-1}\underline{r}^k\,. \tag{4.143}$$

Since $\mathbf{D}$ is a diagonal matrix, the action $\mathbf{D}^{-1}\underline{r}^k$ can be applied cheaply by dividing the entries of $\underline{r}$ by the diagonal entries of $\mathbf{A}$.

The Gauss-Seidel method uses the splitting $\mathbf{P} = \mathbf{D} + \mathbf{L}$ and $\mathbf{N} = \mathbf{U}$. The iteration is then

$$\underline{x}^{k+1} = \underline{x}^k + (\mathbf{D}+\mathbf{L})^{-1}\,\underline{r}^k\,. \tag{4.144}$$

Note that since the matrix $\mathbf{D} + \mathbf{L}$ is lower triangular, the action of its inverse can be applied using forward substitution. Forward substitution requires more floating point operations than inverting a diagonal matrix. However, this added work allows Gauss-Seidel to typically converge faster than Jacobi.

4.7.2 The Conjugate Gradient Method

For symmetric positive definite matrices, the solution $\mathbf{A}\underline{x} = \underline{b}$ is the unique minimizer of the potential

$$\Pi = \frac{1}{2}\underline{x}^T\mathbf{A}\underline{x} - \underline{x}^T\underline{b}\,. \tag{4.145}$$

This can be seen by setting the gradient of Π to zero:

$$0 = \nabla\Pi = \mathbf{A}\underline{x} - \underline{b} \iff \mathbf{A}\underline{x} = \underline{b}\,. \tag{4.146}$$

The conjugate gradient method [82, 83] is built around the idea of minimizing Π in such a way that the system is guaranteed to be solved in at most n steps. An excellent reference for understanding the conjugate gradient method is Shewchuk [84].

Note that the residual $\underline{r}^k = \underline{b} - \mathbf{A}\underline{x}^k = -\nabla\Pi$. Moving in the direction of the residual then moves opposite the gradient of the potential. The method of steepest descent is the method that updates the solution in the direction of the residual in such a way that the potential is minimized. That is, we seek to use the update $\underset{\sim}{x}^{k+1} = \underset{\sim}{x}^k + \lambda_k \underset{\sim}{r}^k$ where $\lambda_i \in \mathbb{R}$ is chosen so that $\underset{\sim}{x}^{k+1}$ minimizes the potential. Introducing this update into the potential:

$$\Pi = \frac{1}{2}\left(\underset{\sim}{x}^k + \lambda_k \underset{\sim}{r}^k\right)^T \mathbf{A}\left(\underset{\sim}{x}^k + \lambda_k \underset{\sim}{r}^k\right) - \left(\underset{\sim}{x}^k + \lambda_k \underset{\sim}{r}^k\right)^T \underset{\sim}{r}^k \,. \tag{4.147}$$

Setting $\frac{\partial \Pi}{\partial \lambda_k} = 0$ and solving for λ_k yields:

$$\lambda_k = \frac{\underset{\sim}{r}^k \cdot \underset{\sim}{r}^k}{\underset{\sim}{r}^k \cdot \mathbf{A}\underset{\sim}{r}^k} \,. \tag{4.148}$$

The conjugate gradient method is a type of steepest descent method where the search directions are chosen to be **A**-orthogonal. In this way, the algorithm only searches in new directions and avoids stalls associated with exploring parts of the solution space that have already been visited. In other words, both the parameter λ_k in the update for $\underset{\sim}{x}^{k+1}$ and the search direction are chosen optimally. The update used by conjugate gradient is

$$\underset{\sim}{x}^{k+1} = \underset{\sim}{x}^k + \lambda_k \underset{\sim}{z}^k \,, \tag{4.149}$$

where the search direction, $\underset{\sim}{z}^k$, is chosen such that

$$\underset{\sim}{z}^k = \underset{\sim}{r}^k + \theta_k \underset{\sim}{z}^{k-1} \,. \tag{4.150}$$

Here, θ_k is the parameter that forces $\underset{\sim}{z}^k$ and $\underset{\sim}{z}^{k-1}$ to be **A**-conjugate such that $\underset{\sim}{z}^k \cdot \mathbf{A}\underset{\sim}{z}^{k-1} = 0$:

$$0 = \underset{\sim}{z}^k \cdot \mathbf{A}\underset{\sim}{z}^{k-1} \Rightarrow \theta_k = -\frac{\underset{\sim}{r}^k \cdot \mathbf{A}\underset{\sim}{z}^{k-1}}{\underset{\sim}{z}^{k-1} \cdot \mathbf{A}\underset{\sim}{z}^{k-1}} \,. \tag{4.151}$$

With this update, the λ_k that minimizes the potential is

$$\lambda_k = \frac{\underset{\sim}{z}^k \cdot \underset{\sim}{r}^k}{\underset{\sim}{z}^k \cdot \mathbf{A}\underset{\sim}{z}^k} \,. \tag{4.152}$$

The solution procedure for a 2-dimensional system is shown in Fig. 4.21. The search directions are chosen such that they are **A**-orthogonal allowing the algorithm to find the solution in two steps.

Note that the solution being the minimizer of a potential is only true when **A** is symmetric positive definite. Furthermore, when **A** is not symmetric positive definite, $\underset{\sim}{x} \cdot \mathbf{A}\underset{\sim}{x}$ is not an inner product. This means **A**-orthogonality cannot be built using the three-term recursion process outlined above and that the previous search directions would need to be stored in order to enforce **A**-orthogonality.

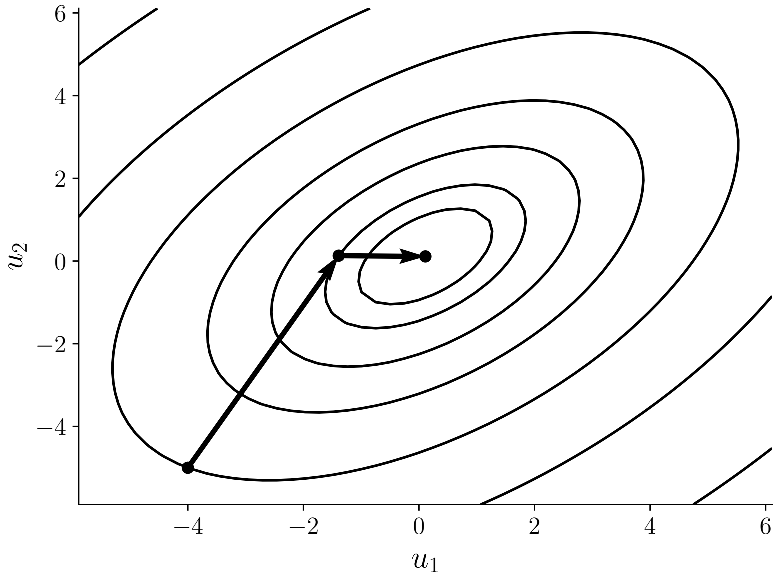


Figure 4.21: A depiction of the conjugate gradient solution process for solving a two-dimension symmetric positive definite system. The solution iterates are shown as dots and the search directions as arrows. The potential is minimized in two iterations.

4.7.3 The Stabilized Bi-Conjugate Gradient Method

In this document, the Stabilized Bi-Conjugate Gradient Method (BiCGStab) is used to solve non-symmetric systems where the conjugate gradient method cannot be used. The original Bi-Conjugate Gradient Method (BCG) [85, 86] implicitly solves both $\mathbf{A}\underset{\sim}{x} = \underset{\sim}{b}$ and the dual system $\mathbf{A}^T\underset{\sim}{x}^* = \underset{\sim}{b}^*$ [87]. The residuals for the original and dual problem form a bi-orthogonal sequence that allows the design of a conjugate gradient-like algorithm that is effective for non-symmetric problems. The BiCGStab algorithm [88] both rewrites the BCG algorithm to remove the requirement of the transpose matrix-vector product and introduces a stabilization process that leads to smoother and more robust convergence. The resulting algorithm is characterized by requiring two matrix-vector products per iteration. Thus, BiCGStab allows solving non-symmetric systems but at the cost of being more than twice as expensive per iteration than the conjugate gradient algorithm. The BiCGStab algorithm is described in *Templates for the Solution of Linear Systems* [89, p. 27] and is also implemented in the MFEM finite element library [90].

The Generalized Minimal Residual Method (GMRES) [91] is another algorithm that is effective for non-symmetric problems. GMRES performs one matrix-vector product per iteration but requires storing the residual vectors as the iteration proceeds. Since GPU storage is likely to be limited, we prefer BiCGStab over GMRES. This choice trades an additional matrix-vector product per iteration for reduced storage requirements.

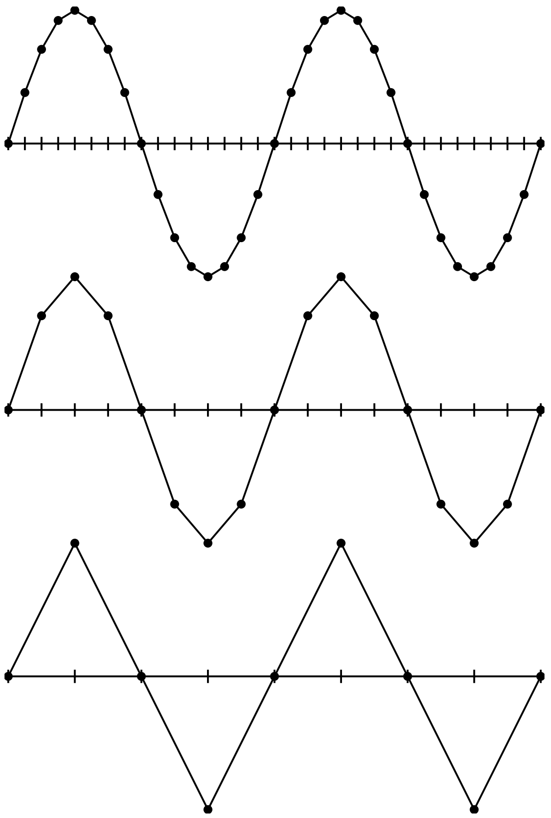


Figure 4.22: The effect of coarsening on a piecewise linear approximation of $\sin(x)$.

4.7.4 Multigrid

Many iterative schemes, such as the classical Jacobi and Gauss-Seidel iterations discussed above, have a smoothing property. That is, they are effective at eliminating high-frequency or oscillatory components of the error but leave the low-frequency or smooth components relatively unchanged. This intuition is supported by Fourier analysis [92]. Multigrid is an algorithm that effectively eliminates both the high and low frequency components of the error. The basic idea is to apply a smoothing scheme on recursively coarsened problems. Figure 4.22 shows the effect of coarsening in the mesh size h on an example smooth wave form. In each step from top to bottom, the number of elements is cut in half. The initially smooth waveform appears to be higher frequency on the coarsened meshes. Thus, smooth error modes can be eliminated by representing them on a coarse mesh and applying a smoothing iterative scheme.

The multigrid algorithm leverages this behavior. A two-level algorithm would: apply a smoothing iterative scheme to the fine-grid problem, restrict the residual onto a coarser grid, solve for a coarse-grid update, and then prolong the update to the fine-grid problem in order to iteratively update the fine-grid solution vector. A general multigrid algorithm recursively applies this algorithm by replacing each coarse-grid solve with another two-level smoothing and coarsening algorithm. The recursion is repeated until the coarse grid problem is small enough to be solved efficiently with a smoothing iteration. Recursively traversing from the finest grid to the coarsest grid and back is called a V-cycle.

Multigrid algorithms based around coarsening the mesh are termed Geometric Multigrid (GMG) because they rely on the geometry of the problem. Such algorithms typically require the user to supply the hierarchy of meshes used in the coarsening steps. The generation of such a hierarchy is a non-trivial task for sufficiently complicated geometries. In addition, GMG requires re-discretizing the problem on each of the meshes in the hierarchy.

Algebraic Multigrid (AMG) [93] is a multigrid scheme that defines the coarse-grid restrictions and prolongations using only information provided by the entries of the algebraic system of equations. In this way, knowledge of the mesh is not required and the expense of re-discretizing on each coarse problem is avoided. This ease-of-use comes at some cost. Typically, AMG performance is mildly degraded compared to GMG with a well-designed mesh hierarchy. In addition, there exists an expensive "setup phase" where the matrix entries are analyzed and the restriction and prolongation operators are formed.

In this document, the AMG solver from *hypre* [94] is used extensively as a preconditioner for the iterative solution of the discrete moment systems. AMG is preferred to GMG in this work due to its black box nature.

4.7.5 Preconditioning

The condition number of the matrix $\mathbf{A}$, denoted $\kappa(\mathbf{A})$, is defined as the ratio of the maximal and minimal singular values of $\mathbf{A}$. It is common for the performance of iterative schemes to degrade as the condition number of the algebraic system increases. For example, it is well known [66, cf. §2.4.2] that the conjugate gradient algorithm reduces the error at each iteration by at most a factor of

$$2\left(\frac{\sqrt{\kappa(\mathbf{A})}-1}{\sqrt{\kappa(\mathbf{A})}+1}\right)\,. \tag{4.153}$$

The condition numbers of matrices arising from finite element discretization generally increase as the mesh size decreases. Thus, Eq. 4.153 predicts more iterations will be needed to solve the linear system associated with increased mesh resolution. Note that as the mesh size decreases, n increases and thus the cost of all the linear algebra operations that comprise each iteration of the iterative solver also increases. Iterative solvers that have uniform convergence with respect to the mesh size are desired. This makes the cost of solving the system scale according to the increased cost associated with larger linear algebra operations only.

Such algorithms are achieved via preconditioning. For a non-singular matrix $\mathbf{P}$, the preconditioned system is given by:

$$\mathbf{P}^{-1}\mathbf{A}\underset{\sim}{x} = \mathbf{P}^{-1}\underset{\sim}{b}\,. \tag{4.154}$$

Note that the preconditioned system is equivalent to the original system $\mathbf{A}\underset{\sim}{x} = \underset{\sim}{b}$. The basic requirements for $\mathbf{P}$ to be a good preconditioner are that $\mathbf{P}$ is "easy" to invert and that $\kappa(\mathbf{P}^{-1}\mathbf{A})$ is smaller than $\kappa(\mathbf{A})$. In this way, an iterative scheme such as conjugate gradient will converge faster on the preconditioned system in Eq. 4.154 than on the original problem. $\mathbf{P}$ is said to be an optimal preconditioner if $\kappa(\mathbf{P}^{-1}\mathbf{A})$ is bounded with respect to n. Note that for conjugate gradient, the preconditioner must also be symmetric positive definite so that $\mathbf{P}^{-1}\mathbf{A}$ is also positive definite.

In practice, forming the matrix $\mathbf{P}^{-1}\mathbf{A}$ is avoided since it requires matrix-matrix multiplication along with storing the matrix $\mathbf{P}^{-1}$. Efficient algorithms are found by applying the preconditioned operator in the following two stages:

$$\underset{\sim}{y} = \mathbf{A}\underset{\sim}{r}\,, \tag{4.155a}$$

$$\underset{\sim}{z} = \mathbf{P}^{-1}\underset{\sim}{y}\,. \tag{4.155b}$$

In this way, iterative schemes can be recast to only require the action of $\mathbf{A}$ and a routine to solve systems of the form:

$$\mathbf{P}\underset{\sim}{z} = \underset{\sim}{y}\,. \tag{4.156}$$

We will often approximately solve Eq. 4.156 by setting $\mathbf{P} = \mathbf{A}$ and using one iteration of a classical iterative scheme or one AMG V-cycle. In such cases, the preconditioner is an approximate inversion of the matrix $\mathbf{A}$.

Chapter 5

Transport Discretization

This chapter presents the S_N and DG discretization for the Boltzmann transport equation. We consider the steady-state, mono-energetic transport problem with isotropic scattering given by:

$$\boldsymbol{\Omega} \cdot \nabla \psi + \sigma_t \psi = \frac{\sigma_s}{4\pi} \int \psi \, \mathrm{d}\Omega' + q \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{5.1a}$$

$$\psi(\mathbf{x}, \boldsymbol{\Omega}) = \bar{\psi}(\mathbf{x}, \boldsymbol{\Omega}) \,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega} \cdot \mathbf{n} < 0 \,, \tag{5.1b}$$

where $\psi(\mathbf{x}, \boldsymbol{\Omega})$ is the angular flux, $\mathcal{D}$ the domain of the problem with $\partial\mathcal{D}$ its boundary, $\sigma_t(\mathbf{x})$ and $\sigma_s(\mathbf{x})$ the total and scattering macroscopic cross sections, respectively, $q(\mathbf{x}, \boldsymbol{\Omega})$ the fixed-source, and $\bar{\psi}(\mathbf{x}, \boldsymbol{\Omega})$ the inflow boundary function. We discuss efficient solution strategies for solving the resulting algebraic system on both orthogonal and curved meshes and define so-called negative flux fixups which correct the discrete angular flux solution such that it is positive. The chapter concludes with the implementation details associated with using the transport discretization presented here in a moment method.

5.1 Discrete Ordinates

The S_N angular model collocates the transport equation at a set of discrete angles, $\boldsymbol{\Omega}_d$, and integration is numerically approximated using a suitable angular quadrature rule $\{\boldsymbol{\Omega}_d, w_d\}_{d=1}^{N_\Omega}$ on the unit sphere. The discrete-in-angle transport equation is:

$$\boldsymbol{\Omega}_d \cdot \nabla \psi_d + \sigma_t \psi_d = \frac{\sigma_s}{4\pi} \sum_{d'=1}^{N_\Omega} w_{d'} \psi_{d'} + q_d \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{5.2a}$$

$$\psi_d(\mathbf{x}) = \bar{\psi}(\mathbf{x}, \boldsymbol{\Omega}_d) \,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega}_d \cdot \mathbf{n} < 0 \,, \tag{5.2b}$$

where $d \in [1, N_\Omega]$, and $\psi_d(\mathbf{x}) = \psi(\mathbf{x}, \boldsymbol{\Omega}_d)$ and $q_d(\mathbf{x}) = q(\mathbf{x}, \boldsymbol{\Omega}_d)$ are the angular flux and fixed-source of particles traveling in the discrete direction $\boldsymbol{\Omega}_d$, respectively. We use the Level Symmetric quadrature rules described in Lewis and Miller Jr. [60]. Figure 5.1 shows the

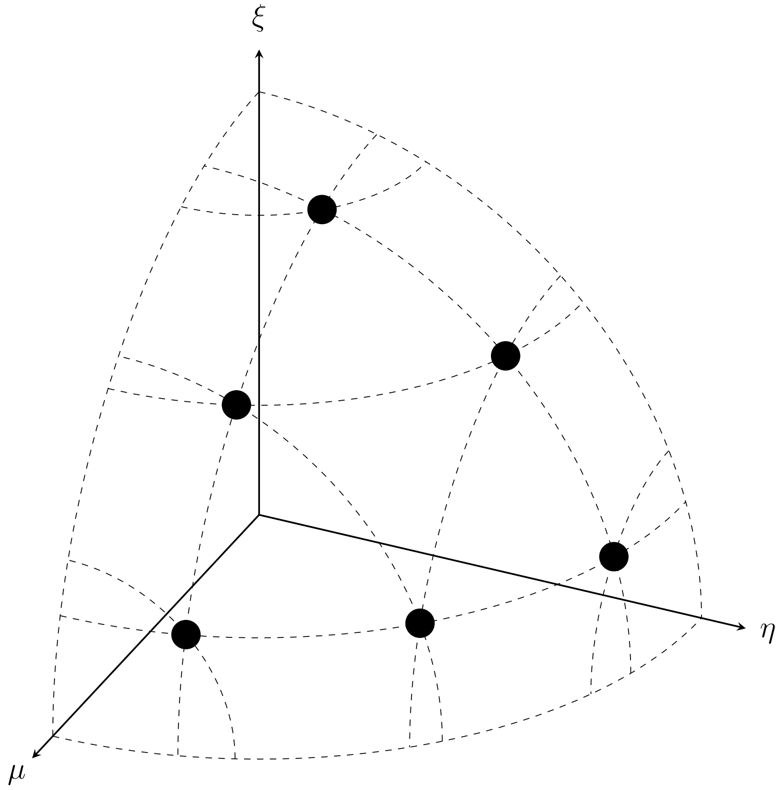


Figure 5.1: The positive octant for the level symmetric S_6 angular quadrature rule.

quadrature points in an octant of the Level Symmetric S_6 angular quadrature rule. A key property of S_N is that the angles are only coupled in the scattering term. That is, if the scattering source were known, each $\psi_d(\mathbf{x})$ could be solved for independently.

5.2 Discontinuous Galerkin

We now apply a DG discretization to the S_N transport equations. We derive a discretization for each angle independently by approximating each ψ_d in the degree-p DG space Y_p introduced in Section 4.5.1. The weak form is first derived on each element K. A global approximation is found by defining the upwind numerical flux that couples adjacent elements based on the direction of $\mathbf{\Omega}_d$. We delay discussion of the computation of the scattering source until Section 5.6 and assume for the moment that the scattering source is included in the fixed-source q.

The weak form on each element is: find $\psi_d \in Y_p$ such that for all $K \in \mathcal{T}$:

$$\int_{\partial K} \mathbf{\Omega}_d \cdot \mathbf{n}\, u\widehat{\psi}_d \,\mathrm{d}s - \int_K \mathbf{\Omega}_d \cdot \nabla u\, \psi_d \,\mathrm{d}\mathbf{x} + \int_K \sigma_t\, u\psi_d \,\mathrm{d}\mathbf{x} = \int_K u\, q_d \,\mathrm{d}\mathbf{x}\,, \tag{5.3}$$

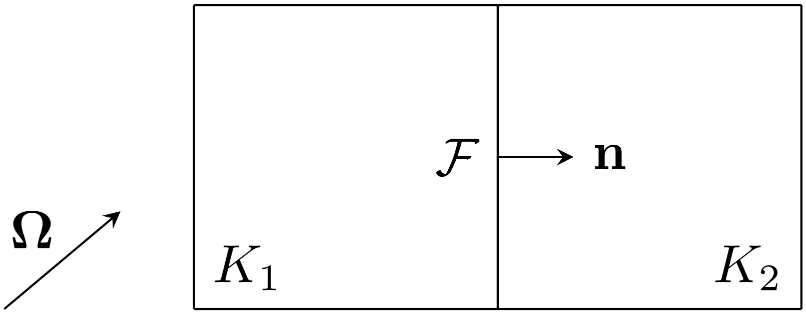


Figure 5.2: A depiction of a grouping of mesh elements where, due to the direction of $\boldsymbol{\Omega}$, the element K_1 is upwind of K_2. A transport solve in the direction $\boldsymbol{\Omega}$ would use the outflow from K_1 to compute the inflow for K_2.

where the numerical flux $\widehat{\psi}_d$ is either an approximation of ψ_d on interior mesh interfaces or given by the inflow boundary function $\bar{\psi}$ on an inflow boundary. We use the upwind numerical flux that defines the incoming angular flux as the outflow from the upwind element. On a face $\mathcal{F}$ between elements K_1 and K_2 with normal pointing from K_1 to K_2 (see Fig. 5.2), the upwind numerical flux is defined as

$$\widehat{\psi}_d = \begin{cases} \psi_{d,1}\,, & \boldsymbol{\Omega}_d \cdot \mathbf{n} > 0 \\ \psi_{d,2}\,, & \boldsymbol{\Omega}_d \cdot \mathbf{n} < 0 \end{cases}\,, \quad \text{on } \mathcal{F} \in \Gamma_0\,, \tag{5.4}$$

where $\psi_{d,i} = \psi_d|_{K_i}$. For boundary faces, we set

$$\widehat{\psi}_d(\mathbf{x}) = \begin{cases} \bar{\psi}(\mathbf{x}, \boldsymbol{\Omega}_d)\,, & \boldsymbol{\Omega} \cdot \mathbf{n} < 0 \\ \psi_d(\mathbf{x})\,, & \boldsymbol{\Omega} \cdot \mathbf{n} > 0 \end{cases}\,, \quad \text{on } \mathcal{F} \in \Gamma_b\,. \tag{5.5}$$

Note that for $\mathcal{F} \in \Gamma_b$, we use the convention that $\mathbf{n}$ is the outward unit normal. Thus, Eq. 5.5 applies the inflow boundary condition when $\boldsymbol{\Omega}_d \cdot \mathbf{n} < 0$. Observe that the conditions in Eqs. 5.4 and 5.5 can equivalently be written using the switch functions:

$$\boldsymbol{\Omega} \cdot \mathbf{n}\,\widehat{\psi}_d = \begin{cases} \frac{1}{2}\,(\boldsymbol{\Omega} \cdot \mathbf{n} + |\boldsymbol{\Omega} \cdot \mathbf{n}|)\psi_{d,1} + \frac{1}{2}(\boldsymbol{\Omega} \cdot \mathbf{n} - |\boldsymbol{\Omega} \cdot \mathbf{n}|)\psi_{d,2}\,, & \text{on } \mathcal{F} \in \Gamma_0 \\ \frac{1}{2}(\boldsymbol{\Omega} \cdot \mathbf{n} + |\boldsymbol{\Omega} \cdot \mathbf{n}|)\psi_d + \frac{1}{2}(\boldsymbol{\Omega} \cdot \mathbf{n} - |\boldsymbol{\Omega} \cdot \mathbf{n}|)\bar{\psi}(\mathbf{x}, \boldsymbol{\Omega}_d)\,, & \text{on } \mathcal{F} \in \Gamma_b \end{cases}\,, \tag{5.6}$$

since when $\boldsymbol{\Omega} \cdot \mathbf{n} > 0$, $\frac{1}{2}(\boldsymbol{\Omega} \cdot \mathbf{n} + |\boldsymbol{\Omega} \cdot \mathbf{n}|) = \boldsymbol{\Omega} \cdot \mathbf{n}$ and $\frac{1}{2}(\boldsymbol{\Omega} \cdot \mathbf{n} - |\boldsymbol{\Omega} \cdot \mathbf{n}|) = 0$ with the opposite holding when $\boldsymbol{\Omega} \cdot \mathbf{n} < 0$. Using the definitions of the jump and average in Eq. 4.30, the switch functions can be rewritten as

$$\boldsymbol{\Omega} \cdot \mathbf{n}\,\widehat{\psi}_d = \boldsymbol{\Omega} \cdot \mathbf{n}\,\{\!\{\psi_d\}\!\} + \frac{1}{2}|\boldsymbol{\Omega} \cdot \mathbf{n}|\,\llbracket \psi_d \rrbracket\,, \quad \text{on } \mathcal{F} \in \Gamma_0 \tag{5.7}$$

with the boundary case left unchanged. Note that $\widehat{\psi}_d$ is single-valued on all faces in the mesh. Thus, the jumps and averages identity (Eq. 4.32) simplifies to

$$\left\llbracket u\widehat{\psi}_d \right\rrbracket = \llbracket u \rrbracket\,\widehat{\psi}_d\,. \tag{5.8}$$

Using the upwind numerical flux and summing over all elements yields the global weak form: find $\psi_d \in Y_p$ such that

$$\frac{1}{2}\int_{\Gamma_b} u(\mathbf{\Omega}_d\cdot\mathbf{n}+|\mathbf{\Omega}_d\cdot\mathbf{n}|)\,\psi_d\,\mathrm{d}s+\int_{\Gamma_0}[\![u]\!]\left(\mathbf{\Omega}_d\cdot\mathbf{n}\,\{\!\{\psi_d\}\!\}+\frac{1}{2}|\mathbf{\Omega}_d\cdot\mathbf{n}|\,[\![\psi_d]\!]\right)\mathrm{d}s-\int\mathbf{\Omega}_d\cdot\nabla_h u\,\psi_d\,\mathrm{d}\mathbf{x}$$
$$+\int\sigma_t\,u\psi_d\,\mathrm{d}\mathbf{x}=\int u\,q_d\,\mathrm{d}\mathbf{x}-\frac{1}{2}\int_{\Gamma_b}u(\mathbf{\Omega}_d\cdot\mathbf{n}-|\mathbf{\Omega}_d\cdot\mathbf{n}|)\,\bar{\psi}(\mathbf{x},\mathbf{\Omega}_d)\,\mathrm{d}s\,,\quad\forall u\in Y_p\,,\quad(5.9)$$

where ∇_h denotes the broken gradient defined in Eq. 4.33. Defining the bilinear forms

$$\underline{u}^T\mathbf{M}_t\underline{\psi}_d=\int\sigma_t\,u\psi_d\,\mathrm{d}\mathbf{x}\,,\tag{5.10a}$$

$$\underline{u}^T\mathbf{G}_d\underline{\psi}_d=-\int\mathbf{\Omega}_d\cdot\nabla u\,\psi_d\,\mathrm{d}\mathbf{x}\,,\tag{5.10b}$$

$$\underline{u}^T\mathbf{F}_d\underline{\psi}_d=\frac{1}{2}\int_{\Gamma_b}u(\mathbf{\Omega}_d\cdot\mathbf{n}+|\mathbf{\Omega}_d\cdot\mathbf{n}|)\,\psi_d\,\mathrm{d}s+\int_{\Gamma_0}[\![u]\!]\left(\mathbf{\Omega}_d\cdot\mathbf{n}\,\{\!\{\psi_d\}\!\}+\frac{1}{2}|\mathbf{\Omega}_d\cdot\mathbf{n}|\,[\![\psi_d]\!]\right)\mathrm{d}s\,,\tag{5.10c}$$

and the linear form

$$\underline{u}^T\underline{b}_d=\int u\,q_d\,\mathrm{d}\mathbf{x}-\frac{1}{2}\int_{\Gamma_b}u(\mathbf{\Omega}_d\cdot\mathbf{n}-|\mathbf{\Omega}_d\cdot\mathbf{n}|)\,\bar{\psi}(\mathbf{x},\mathbf{\Omega}_d)\,\mathrm{d}s\,,\tag{5.11}$$

the discrete transport system is:

$$(\mathbf{F}_d+\mathbf{G}_d+\mathbf{M}_t)\,\underline{\psi}_d=\underline{b}_d\,,\quad 1\le d\le N_\Omega\,.\tag{5.12}$$

Since the space Y_p does not share degrees of freedom across interior element interfaces, the matrices $\mathbf{G}_d$ and $\mathbf{M}_t$ are block diagonal by element. However, the numerical flux couples neighboring elements meaning $\mathbf{F}_d$ has coupling between neighboring elements. On meshes without mesh cycles or reentrant faces, the elements in the mesh can be reordered so that the transport system $\mathbf{F}_d+\mathbf{G}_d+\mathbf{M}_t$ is block lower triangular by element. This means each direction of the angular flux can be solved with an element-by-element forward solve. This procedure is described in the next section and extended to high-order meshes in Section 5.4.

5.3 The Transport Sweep

Here, we present the efficient procedure for solving the discrete transport equation known as the transport sweep. The use of the transport sweep is motivated by the memory and computational cost associated with the extreme number of degrees of freedom in transport calculations. Let $N_{\mathbf{x}}=\dim(Y_p)$ be the number of spatial degrees of freedom corresponding to the space Y_p and $N=N_\Omega\times N_{\mathbf{x}}$ be the total number of degrees of freedom in the discrete phase space. When N is also multiplied by the number of discrete frequency groups in a time-dependent calculation, storing the angular flux solution vector, a vector of size N, is challenging on even the largest computers. Thus, forming and storing an $N \times N$ system of equations is impractical even if the sparsity of the finite element system was accounted for. In this section, we assume the mesh is linear and delay discussion of the transport sweep on meshes with curved surfaces to Section 5.4.

We first discuss properties of the discrete transport system corresponding to the entire phase space. We define

$$\underline{\psi} = \begin{bmatrix} \underline{\psi}_1^T & \dots & \underline{\psi}_{N_\Omega}^T \end{bmatrix}^T \in \mathbb{R}^N \tag{5.13}$$

such that the solution vector groups the all the spatial unknowns corresponding to each angle together. In other words, $\underline{\psi}$ strides in space first then angle. Let $\mathbf{L}_d = \mathbf{F}_d + \mathbf{G}_d + \mathbf{M}_t$ be the streaming and collision operator in each direction $\mathbf{\Omega}_d$ and $\mathbf{L} = \operatorname{diag}(\mathbf{L}_d) \in \mathbb{R}^{N\times N}$ be the streaming and collision operator for all directions such that

$$\mathbf{L} = \begin{bmatrix} \mathbf{L}_1 & & \\ & \ddots & \\ & & \mathbf{L}_{N_\Omega} \end{bmatrix} . \tag{5.14}$$

For the scattering source, let $\mathbf{D} \in \mathbb{R}^{N_\mathbf{x}\times N}$ be the operator that represents computing the zeroth angular moment of each spatial degree of freedom. That is,

$$\left[\mathbf{D}\underline{\psi}\right]_i = \sum_d w_d \psi_{d,i} \,, \quad 1 \le i \le N_\mathbf{x} \,, \tag{5.15}$$

where $\psi_{d,i}$ is the i^{th} spatial degree of freedom in direction $\mathbf{\Omega}_d$. The scattering mass matrix is defined as

$$\underline{u}^T \mathbf{S} \underline{\phi} = \int \sigma_s \, u\phi \, \mathrm{d}\mathbf{x} \,, \tag{5.16}$$

where $u, \phi \in Y_p$. In addition, we define $\mathbf{M} = \operatorname{diag}(\mathbf{I}_{N_\mathbf{x}}) \in \mathbb{R}^{N\times N_\mathbf{x}}$ as the operator that copies the isotropic scattering source into each discrete angle. The scattering source can then be written as $\mathbf{MSD}\underline{\psi}$. Finally, the fixed-source for the entire phase space is $\underline{b} = \begin{bmatrix} \underline{b}_1^T & \dots & \underline{b}_{N_\Omega}^T \end{bmatrix}^T$.

With these definitions the transport equation can be written

$$(\mathbf{L} - \mathbf{MSD})\underline{\psi} = \underline{b} \,. \tag{5.17}$$

Figure 5.4a depicts the sparsity pattern for a one-dimensional transport problem using S_4 angular quadrature and five spatial elements. This matrix corresponds to the mesh and discrete directions depicted in Fig. 5.3. The sparsity pattern of the streaming and collision operator, $\mathbf{L}$, is shown in Fig. 5.4b. The vertical and horizontal lines split the system by angle so that each block corresponds to the spatial degrees of freedom associated with a single direction. In one dimension, the trivial ordering of elements from left to right leads to $\mathbf{L}$ having an upper block triangular by element structure for negative angles and a lower block triangular by element structure for positive angles. These structures for negative and positive

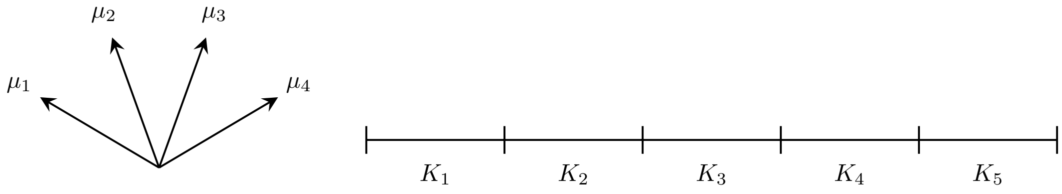
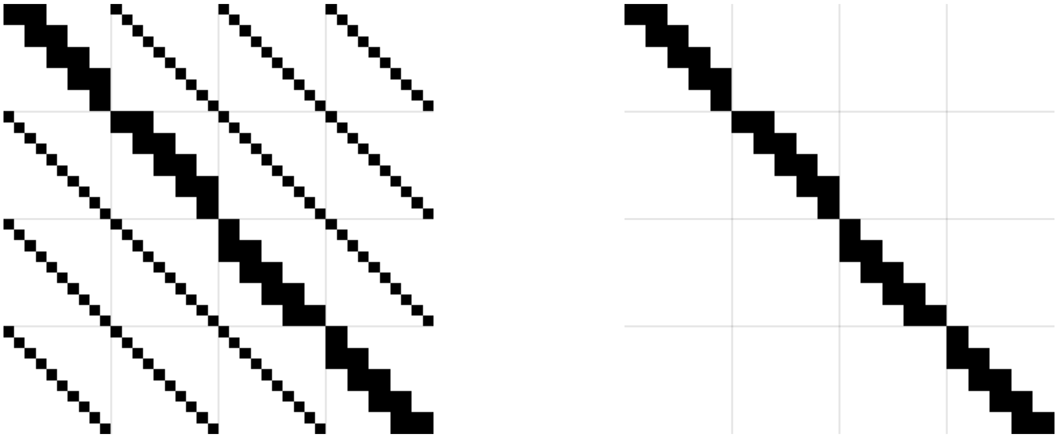


Figure 5.3: The discrete phase space for an example transport solve in one spatial dimension. The trivial ordering of elements from left to right results in a lower/upper block triangular transport operator for positive/negative angles. The angles are ordered from negative to positive.



(a)

(b)

Figure 5.4: Sparsity plots for a 1D DG S_N transport problem. The unknowns stride in space and then angle. In (a), scattering is included showing that all angles are coupled. By lagging the scattering term, (b) shows that each angle can now be solved independently. Furthermore, for each angle the system is either lower or upper block triangular by element. This means each angle can be solved element-by-element in a “sweep”. In two and three dimensions, the block triangular structure can be revealed by a suitable reordering of the elements.

angles are seen in the upper two and lower two diagonal blocks of Fig. 5.4b. Including the scattering contribution, $\mathbf{MSD}\underline{\psi}$, couples all the angles corresponding to each spatial degree of freedom. This is seen in Fig. 5.4a which has off-diagonal blocks corresponding to the quadrature sum and prolongation performed by the operator $\mathbf{MSD}$. These off-diagonal blocks couple the entire phase space and thus forming the $N \times N$ system corresponding to Fig. 5.4a is impractical to store and invert.

Practical transport algorithms use iterative methods that only require the inversion of

$\mathbf{L}$. Observe that $\mathbf{L}$ does not couple the degrees of freedom in angle. That is, each $\mathbf{L}_d$ is decoupled and can be inverted independently. Furthermore, since each $\mathbf{L}_d$ is either upper or lower block triangular by element, an element-by-element forward or backward substitution can be applied. This is implemented as a sweep over the mesh along the direction μ_d. For negative angles, the sweep begins on the right edge of the domain where the inflow boundary condition is defined and solves each element in sequence. The outflow from the previous element is used to provide the inflow condition for the next. Positive angles begin on the left side of the domain and sweep from left to right. This allows solving the full operator $\mathbf{L}$ using only computations associated with the spatial degrees of freedom corresponding to a single element of a single angle at a time, drastically reducing the storage and computational costs of inverting the full $N \times N$ matrix $\mathbf{L}$.

In multiple dimensions, it is typically possible to reorder the elements such that $\mathbf{L}_d$ is lower block triangular by element. Such a sweep ordering can be found by finding an element with no upwind dependencies (e.g. a boundary element where all inflows are computed from the inflow boundary condition) and traversing the directed acyclic graph associated with the connectivity of the mesh. A solution procedure for approximately inverting $\mathbf{L}_d$ when curved faces in the mesh prevent reordering to a block triangular by element system is discussed in Section 5.4.

The most classical algorithm for solving the transport problem using only inversions of the streaming and collision operator is Source Iteration (SI). SI can be viewed as a form of Richardson iteration where a matrix splitting is used to form an iterative solution scheme. The splitting is such that the scattering term is lagged. In equations,

$$\mathbf{L}\underline{\psi}^{\ell+1} = \mathbf{MSD}\underline{\psi}^{\ell} + \underline{b} \iff \underline{\psi}^{\ell+1} = \mathbf{L}^{-1}\big(\mathbf{MSD}\underline{\psi}^{\ell} + \underline{b}\big) \ , \tag{5.18}$$

where superscripts denote iteration index. Here, $\mathbf{L}^{-1}$ represents application of the transport sweep to solve the streaming and collision operator. Unfortunately, this iteration can be arbitrarily slow to converge when the scattering ratio, σ_s/σ_t, is large. In such case, the spectral radius of the iteration matrix $\mathbf{L}^{-1}\mathbf{MSD}$ is very close to unity. This motivates the use of preconditioning schemes such as DSA or an acceleration scheme such as the moment methods discussed in this dissertation.

5.4 Solving on a High-Order Mesh

The normal vector on a high-order surface is in general not constant. It is then possible for the quantity $\mathbf{\Omega}_d \cdot \mathbf{n}$ to change sign along a face between elements due to the variation of the normal vector. This induces a mesh "cycle" where two elements are upwind of each other causing their inflows and outflows to be coupled. When mesh cycles exist, the matrix $\mathbf{F}_d$ cannot be re-ordered to be block lower triangular by element, precluding the use of the classical transport sweep. In addition, the change of sign causes a discontinuity in the integrands of the bilinear forms that comprise $\mathbf{F}_d$. This means $\mathbf{F}_d$ will be difficult to compute accurately using numerical quadrature. In this section, we summarize recent advances in the literature that address the issues of sweeping on a high-order mesh and approximating $\mathbf{F}_d$ with numerical quadrature.

5.4.1 Sweeping on a High-Order Mesh

Consider the example mesh of four cubic elements shown in Fig. 5.5a. Observe that in the direction $\boldsymbol{\Omega}$ depicted in the diagram, the faces $K_1 \cap K_3$ and $K_2 \cap K_4$ are reentrant. That is, particles traveling in the direction $\boldsymbol{\Omega}$ can originate in K_1 or K_2, exit into their neighbor, and then return back. This is possible since $\boldsymbol{\Omega} \cdot \mathbf{n}$ changes sign along the faces between these elements. This is shown for K_3 in Fig. 5.5b where the normal vectors are plotted along the bottom face of K_3. The vectors are colored to denote the subsets of $\mathcal{F} = K_1 \cap K_3$ that correspond to the inflow and outflow conditions, respectively. Thus, one cannot compute the solution in K_1 without already knowing the solution in K_3 and vice versa; the inflow for K_1 depends on the solution in K_3 but the inflow for K_3 also depends on the outflow of K_1. The same is true for the pair K_2 and K_4. Thus, it is not possible to sweep this mesh one element at a time. A direct solve on this mesh would require grouping the degrees of freedom corresponding to (K_1, K_3) and (K_2, K_4) and solving them simultaneously. Such a process would require extra communication to solve across a parallel boundary and would present a non-uniform computation pattern that may be difficult to perform efficiently on a GPU.

In Haut *et al.* [15], an approximate sweep based on a pseudo-optimal reordering of the elements was developed. A graph algorithm is used to find an ordering of the elements that minimizes the strictly upper triangular components of the matrix $\mathbf{F}_d$. The strictly upper triangular contributions are iteratively lagged so that the classical sweep can be applied. This is implemented as a splitting of the matrix $\mathbf{F}_d$ such that:

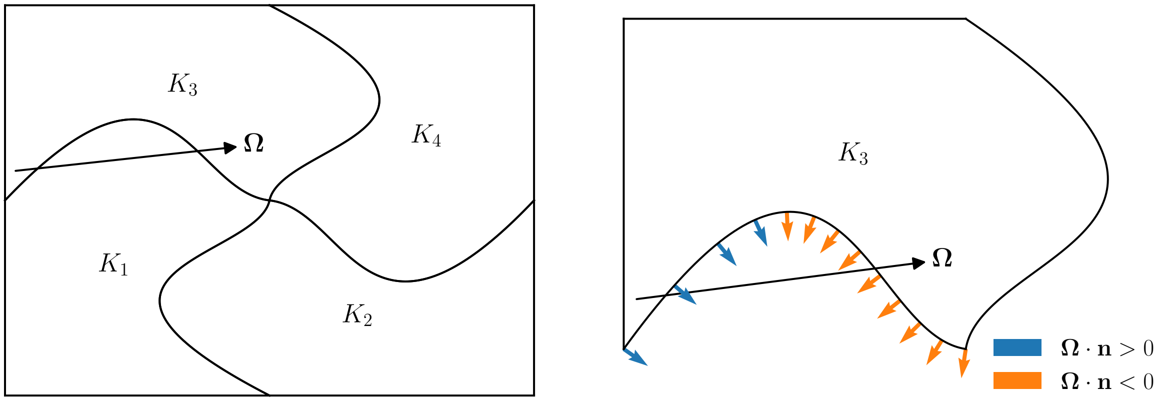
$$\mathbf{F}_d = \mathbf{F}_{d,\downarrow} + \mathbf{F}_{d,\uparrow} \,, \tag{5.19}$$

where $\mathbf{F}_{d,\downarrow}$ and $\mathbf{F}_{d,\uparrow}$ represent the lower and strictly upper triangular parts of $\mathbf{F}_d$, respectively. The graph algorithm finds the element ordering such that $\|\mathbf{F}_{d,\uparrow}\|$ is as small as possible. The inversion of the streaming and collision operator in direction $\boldsymbol{\Omega}_d$ can then be iteratively solved using

$$(\mathbf{F}_{d,\downarrow} + \mathbf{G}_d + \mathbf{M}_t)\underline{\psi}_d^{k+1} = \underline{b}_d - \mathbf{F}_{d,\uparrow}\underline{\psi}_d^{k} \,, \tag{5.20}$$

where superscripts denote the iteration index. Since $\mathbf{F}_{d,\downarrow}$ is block lower triangular by element and $\mathbf{G}_d$ and $\mathbf{M}_t$ are block diagonal by element, the left hand side of Eq. 5.20 can be inverted element-by-element using a transport sweep.

When used in a source iteration solver (e.g. Eq. 5.18), the action of $\mathbf{L}^{-1}$ is approximated using the splitting of $\mathbf{F}_d$ for each direction. Thus, solving the transport problem on a mesh with reentrant faces requires a nested iteration scheme: for each source iteration, the streaming and collision operator is iteratively inverted. However, in practice, it has been seen that a single iteration of the iterative scheme in Eq. 5.20 per outer source iteration is


K_3
$\boldsymbol{\Omega}$
K_4
K_1
K_2
K_3
$\boldsymbol{\Omega}$
🟦 $\boldsymbol{\Omega}\cdot\mathbf{n} > 0$
🟧 $\boldsymbol{\Omega}\cdot\mathbf{n} < 0$

(a) (b)

Figure 5.5: (a) a mesh of four cubic elements. The face $\mathcal{F} = K_1 \cap K_3$ is curved such that K_3 has a concave face. For the direction $\boldsymbol{\Omega}$, this means particles can originate in K_3, exit into K_1, and then reenter K_3. Note that $\mathcal{F} = K_2 \cap K_4$ is also reentrant in the direction $\boldsymbol{\Omega}$. (b) depicts the element K_3 and plots the normal vectors along the face $\mathcal{F} = K_1 \cap K_3$. The normal vector varying along the face causes $\boldsymbol{\Omega}\cdot\mathbf{n}$ to switch sign meaning this face acts as both an outflow and an inflow face for K_3.

enough for robust convergence when the previous outer iteration is used as the initial guess for the inner iteration. This algorithm solves:

$$\mathbf{L}_{\downarrow}\underline{\psi}^{\ell+1} = \mathbf{MSD}\underline{\psi}^{\ell} + \underline{b} - \mathbf{F}_{\uparrow}\underline{\psi}^{\ell}\,, \tag{5.21}$$

where $\mathbf{L}_{\downarrow} = \operatorname{diag}(\mathbf{F}_{d,\downarrow} + \mathbf{G}_d + \mathbf{M}_t)$ and $\mathbf{F}_{\uparrow} = \operatorname{diag}(\mathbf{F}_{d,\uparrow})$. Note that since more information is lagged, solving on a mesh with reentrant faces requires more iterations than a corresponding orthogonal mesh problem.

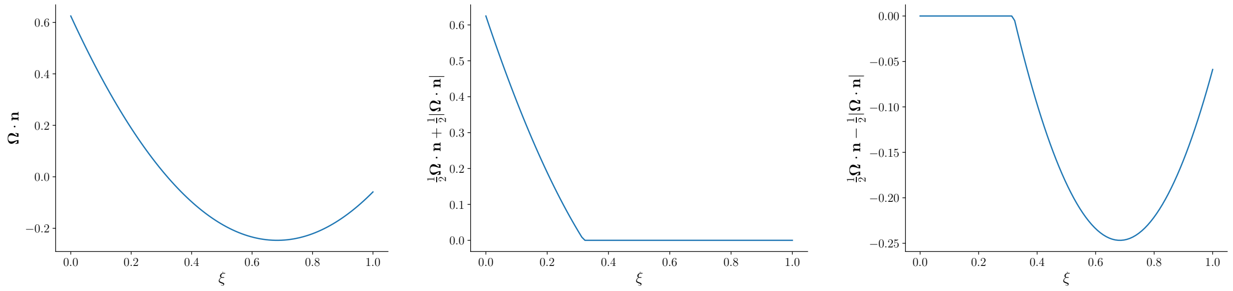
5.4.2 Numerical Integration on Curved Surfaces

Consider the bilinear form

$$\int_{\Gamma_0} [\![u]\!] \left(\boldsymbol{\Omega}_d\cdot\mathbf{n}\,\{\!\{\psi_d\}\!\} + \frac{1}{2}|\boldsymbol{\Omega}_d\cdot\mathbf{n}|\,[\![\psi_d]\!]\right)\mathrm{d}s \tag{5.22}$$

which corresponds to the matrix $\mathbf{F}_d$ defined in Eq. 5.10c without the boundary term. Recall that Eq. 5.22 is equivalent to

$$\int_{\Gamma_0} [\![u]\!]\,\boldsymbol{\Omega}\cdot\mathbf{n}\,\widehat{\psi}_d\,\mathrm{d}s \tag{5.23}$$



(a) (b) (c)

Figure 5.6: Plots of $\mathbf{\Omega}\cdot\mathbf{n}$ and its upwind and downwind components, respectively, along the face $\mathcal{F} = K_1 \cap K_3$ and direction $\mathbf{\Omega}$ from Fig. 5.5a. The plots are provided as a function of the reference coordinate, $\xi \in [0, 1]$, corresponding to $\mathcal{F}$. Observe that the upwind and downwind components shown in (b) and (c) have discontinuous derivatives.

where $\widehat{\psi}_d$ is the upwind numerical flux defined in Eq. 5.4. Multiplying $\widehat{\psi}_d$ by $\mathbf{\Omega}\cdot\mathbf{n}$ yields

$$\mathbf{\Omega}\cdot\mathbf{n}\,\widehat{\psi}_d = \begin{cases} \mathbf{\Omega}\cdot\mathbf{n}\,\psi_{d,1}\,, & \mathbf{\Omega}\cdot\mathbf{n} < 0 \\ \mathbf{\Omega}\cdot\mathbf{n}\,\psi_{d,2}\,, & \mathbf{\Omega}\cdot\mathbf{n} > 0 \end{cases} . \tag{5.24}$$

Thus, as $\mathbf{\Omega}\cdot\mathbf{n}$ passes through zero both cases evaluate to zero meaning $\mathbf{\Omega}\cdot\mathbf{n}\,\widehat{\psi}_d$ is continuous. However, since $\psi_d \in Y_p$ is generally discontinuous across interior mesh interfaces, $\mathbf{\Omega}\cdot\mathbf{n}\,\widehat{\psi}_d$ will generally have a discontinuous derivative. This is shown in Fig. 5.6 where $\mathbf{\Omega}\cdot\mathbf{n}$ is plotted as a function of the reference coordinate $\xi \in [0, 1]$ along the face $K_1 \cap K_3$ from the mesh in Fig. 5.5a. Here, $\mathbf{n}$ corresponds to the normal vectors depicted in Fig. 5.5b. The upwind and downwind parts of $\mathbf{\Omega}\cdot\mathbf{n}$ are also plotted. Observe that the variation of the normal vector with space causes $\mathbf{\Omega}\cdot\mathbf{n}$ to change sign and that the upwind and downwind parts of $\mathbf{\Omega}\cdot\mathbf{n}$ are continuous with a discontinuous derivative.

Numerical quadrature is expected to converge with first-order accuracy when the integrand is continuous but has discontinuous derivatives. Thus, accurate computation of the bilinear forms in $\mathbf{F}_d$ would require a large number of quadrature points. However, Pazner and Haut [95] proved that high-order accuracy is maintained even when the bilinear forms in $\mathbf{F}_d$ are not computed exactly. This suggests that specialized quadrature rules designed to accurately integrate $\mathbf{\Omega}\cdot\mathbf{n}\,\widehat{\psi}_d$ do not need to be used.

5.5 Positivity-Preserving Flux Fixups

One significant challenge of using high-order methods is that they are prone to negativities in under-resolved regions of the solution, especially near material boundaries and other discontinuities. For any physical problem, the continuous S_N transport equations yield nonnegative solutions. Thus, an ideal discretization of the transport equation should be positivity-preserving. However, there is a well-known tradeoff between accuracy and positivity [96]. In addition, Godunov's theorem states that a linear method that guarantees a positive solution can be at most first-order accurate [97]. This precludes the possibility of a positive numerical method for the S_N transport equations that is more than first-order accurate.

Negative solutions are particularly problematic for multiphysics simulations as any of the numerical physics packages may fail due to unphysical inputs from other physics packages. For example, negative transport solutions produce a negative absorption term in the material energy balance equation which significantly increases the likelihood of a negative temperature. Without a positive temperature, many equations of states and opacity models are not well-defined and may cause the simulation to fail. Of particular importance in this work is that the VEF data are not well-defined when the transport solution is not positive. In such case, the angular flux is no longer a valid weight function for computing the average of $\boldsymbol{\Omega}\otimes\boldsymbol{\Omega}$ and $|\boldsymbol{\Omega}\cdot\mathbf{n}|$ in the definitions of the Eddington tensor and boundary factor, respectively. Use of a non-positive transport solution to compute the VEF data can cause the VEF data to diverge by effectively dividing by zero. Thus, a negativity correction or "fixup" is needed.

It is important for the fixup to locally conserve the number of particles in each spatial element. A statement of local balance is found by integrating the transport equation over each element (i.e. taking the zeroth spatial moment). Let the transport equation in direction $\boldsymbol{\Omega}_d$ be written:

$$\int_\Gamma \boldsymbol{\Omega}\cdot\mathbf{n}\, [\![u]\!]\, \widehat{\psi}_d - \int \boldsymbol{\Omega}\cdot\nabla_h u\, \psi_d \,\mathrm{d}\mathbf{x} + \int \sigma_t\, u\psi \,\mathrm{d}\mathbf{x} = \int u\, q \,\mathrm{d}\mathbf{x} \tag{5.25}$$

where q includes the fixed and scattering sources and $\widehat{\psi}$ is the upwind numerical flux defined for both interior and boundary faces in Eqs. 5.4 and 5.5. By setting the test function $u = \mathbb{1}_K$ where

$$\mathbb{1}_K(\mathbf{x}) = \begin{cases} 1\,, & \mathbf{x}\in K \\ 0\,, & \text{otherwise} \end{cases} \tag{5.26}$$

is the indicator function for element K, a statement of local balance over element K is found:

$$\int_{\partial K} \boldsymbol{\Omega}\cdot\mathbf{n}\, \widehat{\psi}_d \,\mathrm{d}s + \int_K \sigma_t\, \psi_d \,\mathrm{d}\mathbf{x} = \int_K q \,\mathrm{d}\mathbf{x}\,. \tag{5.27}$$

Note that $\nabla_h \mathbb{1}_K$ is zero since $\mathbb{1}_K$ is constant on each element $K \in \mathcal{T}$. Defining $\partial K^\pm$ as the outflow and inflow parts of the boundary of the element K, the local balance statement is equivalently written

$$\int_{\partial K^+} \boldsymbol{\Omega}\cdot\mathbf{n}\, \psi_d \,\mathrm{d}s + \int_K \sigma_t\, \psi_d \,\mathrm{d}\mathbf{x} = \int_K q \,\mathrm{d}\mathbf{x} - \int_{\partial K^-} \boldsymbol{\Omega}\cdot\mathbf{n}\, \psi_{d,\mathrm{in}} \,\mathrm{d}s\,, \tag{5.28}$$

where $\psi_{d,\mathrm{in}}$ is the boundary inflow function when $\partial K^+ \cap \Gamma_b \neq \emptyset$ or incoming angular flux from an upwind element otherwise. We now write the balance statement as $\underline{c}_{d,K}^T \underline{\psi}_{d,K} = s_{d,K}$

where

$$\underset{\sim}{c}_{d,K}^{T}\underline{\psi}_{d,K} = \int_{\partial K^+} \mathbf{\Omega}\cdot\mathbf{n}\,\psi_d\,\mathrm{d}s + \int_K \sigma_t\,\psi_d\,\mathrm{d}\mathbf{x}\,, \tag{5.29}$$

$s_{d,K}$ is the right hand side of Eq. 5.28, and $\underline{\psi}_{d,K}$ is the vector of $\underline{\psi}_d$ degrees of freedom corresponding to element K.

Here, we present two methods for correcting the transport solution to be positive. Both methods are "sweep-compatible" in that they can be performed in each element as the sweep progresses. This ensures that the inflow for subsequent elements is positive. In addition, both methods preserve local balance such that the corrected angular flux, denoted ψ^*, satisfies $\underset{\sim}{c}_{d,K}^{T}\underline{\psi}_{d,K}^{*} = s_{d,K}$. In other words, the corrections preserve the original solution's number of particles.

5.5.1 Zero and Rescale

The zero and rescale fixup [98] is a simple scheme for producing a positive solution that maintains particle balance. The scheme is characterized by setting any negative degrees of freedom to zero and then rescaling the solution in each element so that particle balance is maintained before and after zeroing the degrees of freedom. Let

$$[\underline{\psi}_{d,K}]_i^Z = \max([\underline{\psi}_{d,K}]_i, 0)\,, \quad 1 \le i \le \dim(\mathbb{Q}_p(K))\,, \tag{5.30}$$

denote the angular flux degrees of freedom in direction $\mathbf{\Omega}_d$ in element K where negative values have been replaced with zero. These degrees of freedom are then rescaled so that particle balance is preserved. The corrected angular flux is then:

$$\underline{\psi}_{d,K}^{*} = \frac{s_{d,K}}{\underset{\sim}{c}_{d,K}^{T}\underline{\psi}_{d,K}^{Z}}\underline{\psi}_{d,K}^{Z}\,. \tag{5.31}$$

5.5.2 Quadratic Programming Negative Flux Fixup

Here we summarize the quadratic programming-based fixup method from Yee *et al.* [43]. This method seeks to find a positive solution that is as close as possible to the original solution as measured in the ℓ_2 norm (e.g. the usual Euclidean norm for vectors). The objective function is:

$$f(\underline{\psi}_{d,K}^{*}) = \|\underline{\psi}_{d,K}^{*} - \underline{\psi}_{d,K}\|_2^2 = (\underline{\psi}_{d,K}^{*} - \underline{\psi}_{d,K})\cdot(\underline{\psi}_{d,K}^{*} - \underline{\psi}_{d,K})\,. \tag{5.32}$$

The constrained minimization problem is: find $\underline{\psi}_{d,K}^{*}$ such that

$$\underline{\psi}_{d,K}^{*} = \min_{\underline{y}} f(\underline{y})\,, \tag{5.33}$$

under the constraints of particle balance

$$\underset{\sim}{c}_{d,K}^{T}\underline{y} = s_{d,k}\,, \tag{5.34}$$

and that the degrees of freedom are positive

$$[\underline{y}]_i \geq 0\,, \quad 1 \leq i \leq \dim(\mathbb{Q}_p(K))\,. \tag{5.35}$$

Note that this method acts on the degrees of freedom directly and not the finite element interpolation function. That is, while the degrees of freedom of ψ^* may be positive, the interpolation may not be. It is then crucial that a positive interpolating scheme is used. Such schemes guarantee the interpolation function is positive when the degrees of freedom are positive. In this document, we use the positive Bernstein polynomials [99] when the quadratic programming fixup is used.

5.6 Connection to Moment Algorithm

Moment methods simultaneously solve the transport equation coupled to the moment system. That is, we solve

$$\boldsymbol{\Omega}_d \cdot \nabla \psi_d + \sigma_t \psi_d = \frac{\sigma_s}{4\pi}\varphi + q_d\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{5.36a}$$

$$\psi_d(\mathbf{x}) = \bar{\psi}(\mathbf{x}, \boldsymbol{\Omega}_d)\,, \quad \mathbf{x} \in \partial\mathcal{D} \text{ and } \boldsymbol{\Omega}_d \cdot \mathbf{n} < 0\,, \tag{5.36b}$$

for each $1 \leq d \leq N_\Omega$ using the DG discretization discussed in this chapter coupled to a discretization of either the VEF moment system:

$$\nabla \cdot \boldsymbol{J} + \sigma_a \varphi = Q_0\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{5.37a}$$

$$\nabla \cdot (\mathbf{E}\varphi) + \sigma_t \boldsymbol{J} = \boldsymbol{Q}_1\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{5.37b}$$

$$\boldsymbol{J} \cdot \mathbf{n} = E_b \varphi + 2J_{\text{in}}\,, \quad \mathbf{x} \in \partial\mathcal{D}\,, \tag{5.37c}$$

or the SMM moment system:

$$\nabla \cdot \boldsymbol{J} + \sigma_a \varphi = Q_0\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{5.38a}$$

$$\frac{1}{3}\nabla\varphi + \sigma_t \boldsymbol{J} = \boldsymbol{Q}_1 - \nabla \cdot \mathbf{T}\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{5.38b}$$

$$\boldsymbol{J} \cdot \mathbf{n} = \frac{1}{2}\varphi + 2J_{\text{in}} + \beta\,, \quad \mathbf{x} \in \partial\mathcal{D}\,. \tag{5.38c}$$

The discrete transport and moment equations overlap in the computation of the transport equation's scattering source and in computing the moment system's closures from the discrete angular flux. This section discusses the implementation details associated with these connections. We present the computation of the VEF and SMM closures from the discrete angular flux, the evaluation of the transport scattering source from the VEF scalar flux, and the integration of the angular moments for the moment system.

5.6.1 Discrete Closures

The closures are computed using the discrete representation of the angular flux in space and angle along with the S$_N$ angular quadrature rule. The Eddington tensor and boundary factor are then:

$$\mathbf{E}(\mathbf{x}) = \frac{\sum_{d=1}^{N_\Omega} w_d\, \boldsymbol{\Omega}_d \otimes \boldsymbol{\Omega}_d\, \psi_d(\mathbf{x})}{\sum_{d=1}^{N_\Omega} w_d \psi_d(\mathbf{x})} \,, \tag{5.39a}$$

$$E_b(\mathbf{x}) = \frac{\sum_{d=1}^{N_\Omega} w_d\, |\boldsymbol{\Omega}_d \cdot \mathbf{n}|\, \psi_d(\mathbf{x})}{\sum_{d=1}^{N_\Omega} w_d \psi_d(\mathbf{x})} \,. \tag{5.39b}$$

The standard finite element interpolation procedure is used to evaluate ψ_d at any location in the mesh. Note that it is important to interpolate the numerator and denominator of the VEF data *separately.* That is, each component of the Eddington tensor and the boundary factor are represented as q/p where $q, p \in \mathbb{Q}_p(K)$ for each K in the mesh and are thus piecewise discontinuous, improper rational polynomials mapped from the reference element. The SMM correction tensor and factor are analogously computed with:

$$\mathbf{T}(\mathbf{x}) = \sum_{d=1}^{N_\Omega} w_d\, \boldsymbol{\Omega}_d \otimes \boldsymbol{\Omega}_d\, \psi_d(\mathbf{x}) - \frac{1}{3}\mathbf{I} \sum_{d=1}^{N_\Omega} w_d\, \psi_d(\mathbf{x}) \,, \tag{5.40a}$$

$$\beta(\mathbf{x}) = \sum_{d=1}^{N_\Omega} w_d\, |\boldsymbol{\Omega}_d \cdot \mathbf{n}|\, \psi_d(\mathbf{x}) - \frac{1}{2} \sum_{d=1}^{N_\Omega} w_d\, \psi_d(\mathbf{x}) \,. \tag{5.40b}$$

However, since the SMM closures are not normalized, each component of the the correction tensor and the correction factor belong in the finite element space used for the angular flux in each direction. $\boldsymbol{\Omega} = \sum_i^{\dim} \Omega_i \mathbf{e}_i$ is defined on the canonical basis $\mathbf{e}_i$, so each component of the Eddington tensor and correction tensor transform independently as a scalar. In other words, the Piola transform is not required to map the Eddington tensor or correction tensor between reference and physical space.

Note that since $\psi_d \in Y_p$ is in general discontinuous across interior mesh interfaces, the closures are also generally discontinuous. Thus, their global derivatives are not well defined and, in particular, the Eddington and correction tensors are not single-valued on interior mesh interfaces. However, they are *locally* differentiable since $\psi_d|_K \in \mathbb{Q}_p(K)$ is differentiable. Let

$$\phi(\mathbf{x}) = \sum_{d=1}^{N_\Omega} w_d\, \psi_d(\mathbf{x}) \,, \tag{5.41a}$$

$$\mathbf{P}(\mathbf{x}) = \sum_{d=1}^{N_\Omega} w_d\, \boldsymbol{\Omega}_d \otimes \boldsymbol{\Omega}_d\, \psi_d(\mathbf{x}) \,, \tag{5.41b}$$

denote the discrete zeroth and second moments of the angular flux. Using the quotient rule, the local divergence of the Eddington tensor is:

$$\nabla_h \cdot \mathbf{E} = \frac{(\nabla_h \cdot \mathbf{P})\phi - \mathbf{P} \cdot \nabla_h \phi}{\phi^2} \,. \tag{5.42}$$

For SMM, the local divergence of the correction tensor is computed with

$$\nabla_h \cdot \mathbf{T} = \nabla_h \cdot (\mathbf{P} - \frac{1}{3}\mathbf{I}\phi) = \nabla_h \cdot \mathbf{P} - \frac{1}{3}\nabla_h \phi \, . \tag{5.43}$$

Here, the divergence of a second-order tensor is the vector formed by taking the divergence of each of the columns of the tensor. Note that the boundary and correction factors are only needed on the boundary of the domain (e.g. $\mathbf{x} \in \Gamma_b$) and are thus always single-valued.

We restrict our attention to problems where $\psi \geq \delta$ for some $\delta > 0$. This assumption is reasonable for our applications but may be violated in shielding or deep penetration problems. Application of a positivity-preserving negative flux fixup ensures that the numerical approximation of ϕ is bounded away from zero so that $\mathbf{E}$, E_b, and $\nabla_h \cdot \mathbf{E}$ are all well defined. Note that this restriction is not required for the SMM correction tensor and factor to be well defined.

5.6.2 Computation of the Scattering Source

To support generality and algorithmic flexibility, we assume that $\psi_d \in Y_p$ but that the moment system's scalar flux $\varphi \in X$ where X may be different than the finite element space used for the angular flux in each direction. For example, we will develop moment methods where φ is discretized using continuous finite elements and when φ and ψ_d are approximated using different finite element polynomial degrees. In certain cases, it is also advantageous to use a different nodal basis for the angular flux and moment scalar flux. For example, the quadratic programming negative flux fixup in Section 5.5.2 requires the use of the Bernstein polynomials for the transport solve but this choice may not always be advantageous for the discretization of the moment system.

The scattering source is then computed as a mixed-space mass matrix that has test functions in the space for the angular flux and trial functions in the space for the moment system's scalar flux. In other words, for $\varphi \in X$, the scattering operator is

$$\frac{1}{4\pi} \int \sigma_s \, u\varphi \, \mathrm{d}\mathbf{x} \, , \quad \forall u \in Y_p \, . \tag{5.44}$$

The scattering mass matrix $\mathbf{M}_s \in \mathbb{R}^{\dim(Y_p)\times\dim(X)}$ is then

$$\underline{u}^T \mathbf{M}_s \underline{\varphi} = \frac{1}{4\pi} \int \sigma_s \, u\varphi \, \mathrm{d}\mathbf{x} \, , \quad u \in Y_p \, , \varphi \in X \, . \tag{5.45}$$

In a moment method, the scattering source is computed from the moment system's scalar flux. Thus, the scattering source $\mathbf{MSD}\underline{\psi}$ is replaced with $\mathbf{MM}_s\underline{\varphi}$. The resulting discrete transport problem is:

$$\mathbf{L}\underline{\psi} = \mathbf{MM}_s\underline{\varphi} + \underline{b} \, . \tag{5.46}$$

5.6.3 Moments of the Fixed-Source

In forming the right hand sides for the moment system, angular moments of the fixed-source, q, are required. These moments can be computed analytically if q is a simple enough function of angle. However, analytic integration of the source requires an additional input from the user not required by many other transport algorithms. Thus, these source moments are approximated using S_N angular quadrature. Note that the angular-dependence of the source is in general independent from the angular-dependence of the solution. For example, a problem with an isotropic source (a very simple dependence on direction) could have a solution that is anisotropic in angle due to the presence of discontinuous materials. In such case, the solution would require high angular resolution but the moments of the source could be computed with a much coarser angular quadrature rule. Thus, we evaluate the moments of the fixed-source using an angular quadrature rule that is separate from the angular quadrature used to approximate the angular flux. That is, we use the quadrature rule $\{w_d, \mathbf{\Omega}_d\}_{d=1}^{N_\Omega^q}$ for the source terms where N_Ω^q and N_Ω are allowed to differ. The source moments are then computed as

$$Q_0(\mathbf{x}) = \sum_{d=1}^{N_\Omega^q} w_d \, q(\mathbf{x}, \mathbf{\Omega}_d) \, , \tag{5.47}$$

and

$$\boldsymbol{Q}_1(\mathbf{x}) = \sum_{d=1}^{N_\Omega^q} w_d \, \mathbf{\Omega}_d \, q(\mathbf{x}, \mathbf{\Omega}_d) \, . \tag{5.48}$$

In addition to angular integration, the discretization of the moment system also requires spatial integration of the source. Thus, nested quadrature sums arise in forming the right hand side for the discrete moment system. Judiciously choosing N_Ω^q has been seen to drastically reduce the setup cost associated with forming the moment system's source terms, especially when a high angular resolution for the solution is desired.

Chapter 6

Discontinuous Galerkin VEF Discretizations

In this chapter, we design a family of independent VEF discretizations for the linear, steady-state transport problem that can be efficiently and scalably solved with high-order accuracy, in multiple dimensions, and on curved meshes. Our approach is to begin with discretization techniques known to have effective preconditioners on the simpler case of radiation diffusion (i.e. the model Poisson problem) and adapt them to the VEF equations. By using the Eddington tensor and boundary factor interpolation procedure established in [22], these methods achieve both rapid convergence in outer fixed-point iterations and in inner linear solver iterations when paired with a high-order DG discretization of S_N transport.

In particular, we extend the unified analysis of DG methods developed for elliptic problems presented by Arnold *et al.* [58] to the VEF equations to derive analogs of the Interior Penalty (IP), Second Method of Bassi and Rebay (BR2), Minimal Dissipation Local Discontinuous Galerkin (MDLDG), and continuous finite element (CG) techniques. We show that the IP and BR2 VEF methods are effectively preconditioned by the subspace correction method from Pazner and Kolev [59] and that AMG is effective for the CG and MDLDG discretizations. Anistratov and Warsa [52] also applied DG techniques to the VEF equations but they discretize the first-order form of the VEF equations and only consider lowest-order elements in one dimension. We note that our CG operator is equivalent to extending the continuous finite element VEF discretization in [22] to multiple dimensions, arbitrary-order, and curved meshes.

The chapter proceeds as follows. We first introduce the high-level concepts used to derive DG discretizations of the second-order form of elliptic partial differential equations. We then extend Arnold's unified framework to the VEF moment system. The IP, BR2, and MDLDG methods are derived from this framework through the specification of the numerical fluxes. A discussion of the implementation of the so-called lifting operators that are used to stabilize the BR2 and MDLDG discretizations is provided. A continuous finite element discretization is then extracted from the DG framework. Section 6.6 discusses the USC preconditioner and its implementation. We conclude with computational results. We show that all the

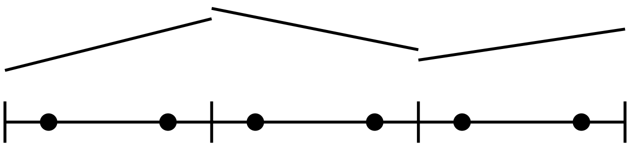


Figure 6.1: An example of a one-dimensional piecewise discontinuous function that is a member of the linear DG space, Y_1.

methods presented achieve high-order accuracy on curved meshes through the method of manufactured solutions, preserve the thick diffusion limit both on orthogonal and a severely distorted curved mesh, and are effective on the linearized, steady-state crooked pipe problem in both outer fixed-point iterations and inner preconditioned linear solver iterations. The parallel performance of the IP and CG methods is demonstrated with a weak scaling study.

6.1 Arnold's Unified Framework

Arnold *et al.* [58] derive a family of DG methods for the model elliptic problem:

$$\boldsymbol{q} = \nabla u \,, \tag{6.1a}$$

$$-\nabla \cdot \boldsymbol{q} = f \,, \tag{6.1b}$$

with Dirichlet boundary conditions. Recall that the DG finite element space Y_p is defined as the space of piecewise polynomials with no continuity enforced between elements. An example of a one-dimensional member of the space Y_1 is shown in Fig. 6.1. Global derivatives of functions in the DG space are not well defined due to their discontinuity along interior mesh interfaces. We must then use integration by parts formulae applied on each element to "offload" derivatives away from such functions. For each element, this generates a volumetric term and a surface term. For the surface term, we must provide an additional piece of information known as the numerical flux. Note that, for some function $u \in Y_p$, $u|_K$ is a smooth polynomial function meaning that u is *locally* differentiable on each element so that $\nabla_h u$ is well defined.

The procedure laid forth in Arnold *et al.* [58] is to discretize and manipulate the weak form of the vector-valued equation so that the vector variable can be eliminated. This discrete elimination is then inserted into the scalar equation. To particularize, we discretize Eq. 6.1a in such a way that the variable $\boldsymbol{q}$ can be eliminated on each element and insert this discrete elimination into a discretization for Eq. 6.1b. The discretization is complete by defining suitable numerical fluxes that ensure the resulting algebraic system will be non-singular.

These steps are demonstrated using sparsity plots in Fig. 6.2. The left diagram shows a naive discretization of the first-order form of Poisson's equation where the vector variable is approximated in such a way that it is coupled to its neighbors. This leads to the coupling shown in the $(1,1)$ block. This coupling causes the $(1,1)$ block to have a dense inverse

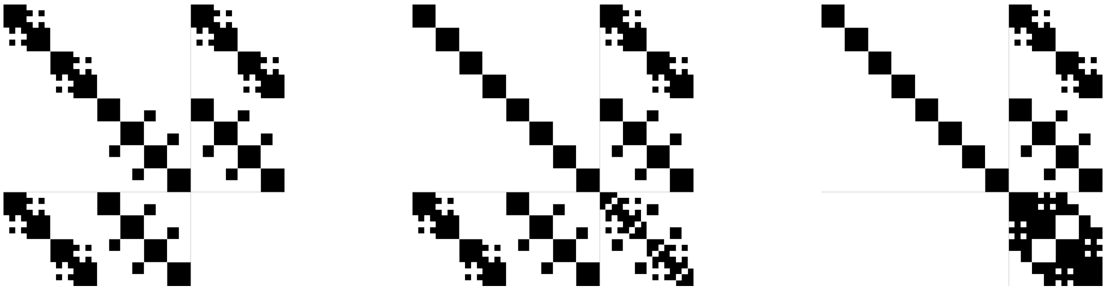


Figure 6.2: Sparsity plots depicting DG discretizations of the Poisson equation in first-order form. The degrees of freedom are ordered so that the first block row corresponds to the vector variable and the second to the scalar variable. (a) shows a “naive” approach where the vector variable is coupled to its neighbor. This causes the $(1,1)$ block to be globally connected, preventing practical elimination of the vector variable. (b) shows the unified approach where the vector variable is discretized so that it can be eliminated locally on each element. (c) shows the resulting block system after performing this elimination.

which makes elimination of the vector variable impractical. The center diagram shows the approach of Arnold *et al.* [58]: the vector variable is approximated such that it is not coupled to its neighbors. This allows the $(1,1)$ block to be block diagonal by element. We can then eliminate the vector variable by inverting the block of degrees of freedom corresponding to each element individually. The right diagram shows the resulting system after performing this element-by-element block Gaussian elimination. The $(2,2)$ block is now dependent only on the scalar variable's degrees of freedom. By solving the global system for the scalar variable in the $(2,2)$ block, the vector variable can be computed using back substitution on each element.

The present goal is to adapt this framework to the VEF equations:

$$\nabla \cdot (\mathbf{E}\varphi) + \sigma_t \boldsymbol{J} = \boldsymbol{Q}_1 \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{6.2a}$$

$$\nabla \cdot \boldsymbol{J} + \sigma_a \varphi = Q_0 \,, \quad \mathbf{x} \in \mathcal{D} \,, \tag{6.2b}$$

$$\boldsymbol{J} \cdot \mathbf{n} = E_b \varphi + 2J_{\text{in}} \,, \quad \mathbf{x} \in \partial\mathcal{D} \,. \tag{6.2c}$$

By extending this framework to the VEF equations, we can easily derive analogs of any of the DG methods presented in [58]. We will see significant differences in the final bilinear form since the Eddington tensor is inside the divergence. Additionally, the presence of a right-hand side in the first moment equation as well as non-unit coefficients and Robin-style boundary conditions introduce further complications not discussed in [58].

6.2 Adaption of the Unified Framework to VEF

We seek the VEF scalar flux in the degree-p DG finite element space Y_p and the current in the degree-p vector-valued DG finite element space X_p. The weak form is then: find $(\varphi, \boldsymbol{J}) \in Y_p \times X_p$ such that for all $K \in \mathcal{T}$:

$$\int_{\partial K} \boldsymbol{v} \cdot \widehat{\mathbf{E}\varphi} \mathbf{n}_K \, \mathrm{d}s - \int_K \nabla \boldsymbol{v} : \mathbf{E}\varphi \, \mathrm{d}\mathbf{x} + \int_K \sigma_t \, \boldsymbol{v} \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} = \int_K \boldsymbol{v} \cdot \boldsymbol{Q}_1 \, \mathrm{d}\mathbf{x} \,, \quad \forall \boldsymbol{v} \in [\mathbb{Q}_p(K)]^{\mathrm{dim}} \,, \tag{6.3a}$$

$$\int_{\partial K} u \, \widehat{\boldsymbol{J}} \cdot \mathbf{n}_K \, \mathrm{d}s - \int_K \nabla u \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} + \int_K \sigma_a \, u\varphi \, \mathrm{d}\mathbf{x} = \int_K u \, Q_0 \, \mathrm{d}\mathbf{x} \,, \quad \forall u \in \mathbb{Q}_p(K) \,, \tag{6.3b}$$

where the *numerical fluxes* $\widehat{\mathbf{E}\varphi}$ and $\widehat{\boldsymbol{J}}$ are approximations of $\mathbf{E}\varphi$ and $\boldsymbol{J}$ on the boundaries of the elements in the mesh. We group the product $\mathbf{E}\varphi$ as the numerical flux to mimic the integration by parts of a tensor times a vector. Here, the gradient of a vector is

$$(\nabla \boldsymbol{v})_{ij} = \left(\frac{\partial \boldsymbol{v}_i}{\partial \mathbf{x}_j} \right) \in \mathbb{R}^{\mathrm{dim} \times \mathrm{dim}} \tag{6.4}$$

and

$$\mathbf{A} : \mathbf{B} = \sum_{i=1}^{\mathrm{dim}} \sum_{j=1}^{\mathrm{dim}} \mathbf{A}_{ij} \mathbf{B}_{ij} \,, \quad \mathbf{A}, \mathbf{B} \in \mathbb{R}^{\mathrm{dim} \times \mathrm{dim}} \tag{6.5}$$

is the scalar contraction of two tensors. Note that if $\mathbf{E} = \frac{1}{3}\mathbf{I}$ then

$$\nabla \boldsymbol{v} : \mathbf{E} = \frac{1}{3} \nabla \cdot \boldsymbol{v} \tag{6.6}$$

and the symmetric weak form for radiation diffusion is recovered.

Summing the zeroth moment over all elements:

$$\int_\Gamma [\![u]\!] \left\{\!\!\left\{ \widehat{\boldsymbol{J}} \cdot \mathbf{n} \right\}\!\!\right\} \mathrm{d}s + \int_{\Gamma_0} \{\!\{u\}\!\} \left[\!\left[\widehat{\boldsymbol{J}} \cdot \mathbf{n} \right]\!\right] \mathrm{d}s - \int \nabla_h u \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} + \int \sigma_a \, u\varphi \, \mathrm{d}\mathbf{x} = \int u \, Q_0 \, \mathrm{d}\mathbf{x} \,, \tag{6.7}$$

where the jumps and averages identity (Eq. 4.32) was used. We will now use the discrete first moment to determine a functional form for $\boldsymbol{J}$. Integrating by parts locally over element K, we have that

$$\int_K \nabla \boldsymbol{v} : \mathbf{E}\varphi \, \mathrm{d}\mathbf{x} = \int_{\partial K} \boldsymbol{v} \cdot \mathbf{E}\varphi \mathbf{n}_K \, \mathrm{d}s - \int_K \boldsymbol{v} \cdot \nabla \cdot (\mathbf{E}\varphi) \, \mathrm{d}\mathbf{x} \,. \tag{6.8}$$

The first moment's weak form on each element becomes:

$$\int_{\partial K} \boldsymbol{v} \cdot \left(\widehat{\mathbf{E}\varphi} \mathbf{n}_K - \mathbf{E}\varphi \mathbf{n}_K \right) \mathrm{d}s + \int_K \boldsymbol{v} \cdot \nabla \cdot (\mathbf{E}\varphi) \, \mathrm{d}\mathbf{x} + \int_K \sigma_t \, \boldsymbol{v} \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} = \int_K \boldsymbol{v} \cdot \boldsymbol{Q}_1 \, \mathrm{d}\mathbf{x} \,, \quad \forall \boldsymbol{v} \in [\mathbb{Q}_p(K)]^{\mathrm{dim}} \,. \tag{6.9}$$

Summing over all elements and using the jumps and averages identity, the weak form for the first moment is:

$$\int_\Gamma \{\!\{\boldsymbol{v}\}\!\} \cdot \left[\!\!\left[\widehat{\mathbf{E}\varphi}\mathbf{n} - \mathbf{E}\varphi\mathbf{n}\right]\!\!\right] \mathrm{d}s + \int_{\Gamma_0} [\![\boldsymbol{v}]\!] \cdot \left\{\!\!\left\{\widehat{\mathbf{E}\varphi}\mathbf{n} - \mathbf{E}\varphi\mathbf{n}\right\}\!\!\right\} \mathrm{d}s \\ + \int \boldsymbol{v} \cdot \nabla_h \cdot (\mathbf{E}\varphi)\,\mathrm{d}\mathbf{x} + \int \sigma_t\,\boldsymbol{v} \cdot \boldsymbol{J}\,\mathrm{d}\mathbf{x} = \int \boldsymbol{v} \cdot \boldsymbol{Q}_1\,\mathrm{d}\mathbf{x}\,, \quad \forall \boldsymbol{v} \in X_p\,, \quad (6.10)$$

where $\nabla_h \cdot (\mathbf{E}\varphi)$ is evaluated as $\nabla_h \cdot (\mathbf{E}\varphi) = \mathbf{E}\nabla_h\varphi + (\nabla_h \cdot \mathbf{E})\varphi$, and the term $\nabla_h \cdot \mathbf{E}$ is computed using Eq. 5.42.

We now wish to write all terms as volumetric integrals so that a functional form for the current can be found. To that end, define *lifting operators* $\boldsymbol{r}(\boldsymbol{\tau}) \in X_p$ and $\boldsymbol{\ell}(\boldsymbol{\chi}) \in X_p$ such that

$$\int \sigma_t\,\boldsymbol{v} \cdot \boldsymbol{r}(\boldsymbol{\tau})\,\mathrm{d}\mathbf{x} = -\int_\Gamma \{\!\{\boldsymbol{v}\}\!\} \cdot \boldsymbol{\tau}\,\mathrm{d}s\,, \quad \forall \boldsymbol{v} \in X_p\,, \quad (6.11a)$$

$$\int \sigma_t\,\boldsymbol{v} \cdot \boldsymbol{\ell}(\boldsymbol{\chi})\,\mathrm{d}\mathbf{x} = -\int_{\Gamma_0} [\![\boldsymbol{v}]\!] \cdot \boldsymbol{\chi}\,\mathrm{d}s\,, \quad \forall \boldsymbol{v} \in X_p\,, \quad (6.11b)$$

where $\boldsymbol{\tau}$ and $\boldsymbol{\chi}$ are vector functions that are singled-valued on Γ_0. Note that the lifting operators are finite element grid functions just as the current is and that the left hand sides are simply the X_p total interaction mass matrix. Since X_p is piecewise discontinuous, the X_p mass matrix is block-diagonal by element and thus the systems of equations corresponding to Eqs. 6.11a and 6.11b are amenable to efficient direct factorization. The computational aspects of lifting operators are discussed in Section 6.4.

Setting $\boldsymbol{\tau} = \left[\!\!\left[\widehat{\mathbf{E}\varphi}\mathbf{n} - \mathbf{E}\varphi\mathbf{n}\right]\!\!\right]$ and $\boldsymbol{\chi} = \left\{\!\!\left\{\widehat{\mathbf{E}\varphi}\mathbf{n} - \mathbf{E}\varphi\mathbf{n}\right\}\!\!\right\}$ and using the definitions of the lifting operators, Eq. 6.10 can be written entirely in terms of volumetric integrals as:

$$\int \sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} = \int \sigma_t\,\boldsymbol{v}\cdot\left[\frac{1}{\sigma_t}\left(\boldsymbol{Q}_1 - \nabla_h \cdot (\mathbf{E}\varphi)\right) + \boldsymbol{r}\Big(\left[\!\!\left[\widehat{\mathbf{E}\varphi}\mathbf{n} - \mathbf{E}\varphi\mathbf{n}\right]\!\!\right]\Big) + \boldsymbol{\ell}\Big(\left\{\!\!\left\{\widehat{\mathbf{E}\varphi}\mathbf{n} - \mathbf{E}\varphi\mathbf{n}\right\}\!\!\right\}\Big)\right]\mathrm{d}\mathbf{x} \quad (6.12)$$

for all $\boldsymbol{v} \in X_p$. Subtracting the right hand side and setting the integrand to zero implies that

$$\boldsymbol{J} = \frac{1}{\sigma_t}\left(\boldsymbol{Q}_1 - \nabla_h \cdot (\mathbf{E}\varphi)\right) + \boldsymbol{r}\Big(\left[\!\!\left[\widehat{\mathbf{E}\varphi}\mathbf{n} - \mathbf{E}\varphi\mathbf{n}\right]\!\!\right]\Big) + \boldsymbol{\ell}\Big(\left\{\!\!\left\{\widehat{\mathbf{E}\varphi}\mathbf{n} - \mathbf{E}\varphi\mathbf{n}\right\}\!\!\right\}\Big)\,. \quad (6.13)$$

Observe that the above represents the element-local strong form of the current, $\frac{1}{\sigma_t}\left(\boldsymbol{Q}_1 - \nabla_h \cdot (\mathbf{E}\varphi)\right)$ found by analytically eliminating the current, with additional terms that capture the effect of the numerical fluxes. In other words, we have derived the *discrete* elimination of the current.

Using this discrete form for the current and the definitions of the lifting operators to convert from volumetric integrals back to surface integrals, the zeroth moment becomes:

$$\begin{aligned}\int_{\Gamma} [\![u]\!] \left\{\!\!\left\{\widehat{\boldsymbol{J}}\cdot\mathbf{n}\right\}\!\!\right\} \mathrm{d}s + \int_{\Gamma_0} \{\!\!\{u\}\!\!\} \left[\!\!\left[\widehat{\boldsymbol{J}}\cdot\mathbf{n}\right]\!\!\right] \mathrm{d}s + \int_{\Gamma} \left\{\!\!\left\{\frac{\nabla_h u}{\sigma_t}\right\}\!\!\right\} \cdot \left[\!\!\left[\widehat{\mathbf{E}\varphi\mathbf{n}} - \mathbf{E}\varphi\mathbf{n}\right]\!\!\right] \mathrm{d}s \\ + \int_{\Gamma_0} \left[\!\!\left[\frac{\nabla_h u}{\sigma_t}\right]\!\!\right] \cdot \left\{\!\!\left\{\widehat{\mathbf{E}\varphi\mathbf{n}} - \mathbf{E}\varphi\mathbf{n}\right\}\!\!\right\} \mathrm{d}s + \int \nabla_h u \cdot \frac{1}{\sigma_t}\nabla_h \cdot (\mathbf{E}\varphi)\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} \\ = \int u\, Q_0\,\mathrm{d}\mathbf{x} + \int \nabla_h u \cdot \frac{\boldsymbol{Q}_1}{\sigma_t}\,\mathrm{d}\mathbf{x}\,, \quad \forall u \in Y_p\,. \end{aligned} \tag{6.14}$$

On boundary faces, we apply the Miften-Larsen boundary conditions by setting

$$\widehat{\boldsymbol{J}}\cdot\mathbf{n} = E_b\varphi + 2J_{\text{in}}\,, \quad \widehat{\mathbf{E}\varphi\mathbf{n}} = \mathbf{E}\varphi\mathbf{n}\,, \quad \text{on } \mathcal{F}\in\Gamma_b\,. \tag{6.15}$$

All the methods we consider use so-called conservative numerical fluxes such that

$$\left[\!\!\left[\widehat{\boldsymbol{J}}\cdot\mathbf{n}\right]\!\!\right] = 0\,, \quad \left\{\!\!\left\{\widehat{\boldsymbol{J}}\cdot\mathbf{n}\right\}\!\!\right\} = \widehat{\boldsymbol{J}}\cdot\mathbf{n}\,, \quad \text{on } \mathcal{F}\in\Gamma_0\,, \tag{6.16a}$$

$$\left[\!\!\left[\widehat{\mathbf{E}\varphi\mathbf{n}}\right]\!\!\right] = 0\,, \quad \left\{\!\!\left\{\widehat{\mathbf{E}\varphi\mathbf{n}}\right\}\!\!\right\} = \widehat{\mathbf{E}\varphi\mathbf{n}}\,, \quad \text{on } \mathcal{F}\in\Gamma_0\,. \tag{6.16b}$$

Using the boundary conditions and the assumption of conservative numerical fluxes, Eq. 6.14 becomes:

$$\begin{aligned}\int_{\Gamma_b} E_b\, u\varphi\,\mathrm{d}s + \int_{\Gamma_0} [\![u]\!]\, \widehat{\boldsymbol{J}}\cdot\mathbf{n}\,\mathrm{d}s - \int_{\Gamma_0} \left\{\!\!\left\{\frac{\nabla_h u}{\sigma_t}\right\}\!\!\right\} \cdot [\![\mathbf{E}\varphi\mathbf{n}]\!]\,\mathrm{d}s \\ + \int_{\Gamma_0} \left[\!\!\left[\frac{\nabla_h u}{\sigma_t}\right]\!\!\right] \cdot \left\{\!\!\left\{\widehat{\mathbf{E}\varphi\mathbf{n}} - \mathbf{E}\varphi\mathbf{n}\right\}\!\!\right\} \mathrm{d}s + \int \nabla_h u \cdot \frac{1}{\sigma_t}\nabla_h \cdot (\mathbf{E}\varphi)\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} \\ = \int u\, Q_0\,\mathrm{d}\mathbf{x} + \int \nabla_h u \cdot \frac{\boldsymbol{Q}_1}{\sigma_t}\,\mathrm{d}\mathbf{x} - 2\int_{\Gamma_b} u\, J_{\text{in}}\,\mathrm{d}s\,, \quad \forall u \in Y_p\,. \end{aligned} \tag{6.17}$$

Equation 6.17 defines a *family* of DG methods. That is, through the specification of the numerical fluxes on interior faces, analogs of all the methods listed in [58] can be derived.

6.3 Specification of Numerical Fluxes

All the methods we consider use numerical fluxes of the form

$$\widehat{\boldsymbol{J}}\cdot\mathbf{n} = \left\{\!\!\left\{\frac{1}{\sigma_t}\left(\boldsymbol{Q}_1 - \nabla_h\cdot(\mathbf{E}\varphi)\right)\cdot\mathbf{n}\right\}\!\!\right\} + \alpha(\varphi)\,, \quad \text{on } \Gamma_0\,, \tag{6.18a}$$

$$\widehat{\mathbf{E}\varphi\mathbf{n}} = \{\!\!\{\mathbf{E}\varphi\mathbf{n}\}\!\!\} + \boldsymbol{\theta}(\varphi)\,, \quad \text{on } \Gamma_0\,, \tag{6.18b}$$

where $\alpha(\varphi)$ and $\boldsymbol{\theta}(\varphi)$ are single-valued functions whose purpose is to ensure a stable discretization. The IP, BR2, and Local Discontinuous Galerkin (LDG) methods differ only in the choice of $\alpha(\varphi)$ and $\boldsymbol{\theta}(\varphi)$. With these numerical fluxes, Eq. 6.17 becomes:

$$\begin{aligned}\int_{\Gamma_b} E_b\, u\varphi \,\mathrm{d}s + \int_{\Gamma_0} [\![u]\!]\, \alpha(\varphi)\,\mathrm{d}s - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{1}{\sigma_t}\nabla_h\cdot(\mathbf{E}\varphi)\cdot\mathbf{n} \right\}\!\!\right\} \mathrm{d}s - \int_{\Gamma_0} \left\{\!\!\left\{ \frac{\nabla_h u}{\sigma_t} \right\}\!\!\right\}\cdot [\![\mathbf{E}\varphi\mathbf{n}]\!]\,\mathrm{d}s \\ + \int_{\Gamma_0} \left[\!\!\left[\frac{\nabla_h u}{\sigma_t} \right]\!\!\right]\cdot \boldsymbol{\theta}(\varphi)\,\mathrm{d}s + \int \nabla_h u \cdot \frac{1}{\sigma_t}\nabla_h\cdot(\mathbf{E}\varphi)\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} \\ = \int u\, Q_0\,\mathrm{d}\mathbf{x} + \int \nabla_h u\cdot \frac{\boldsymbol{Q}_1}{\sigma_t}\,\mathrm{d}\mathbf{x} - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{\boldsymbol{Q}_1\cdot\mathbf{n}}{\sigma_t} \right\}\!\!\right\} \mathrm{d}s - 2\int_{\Gamma_b} u\, J_{\mathrm{in}}\,\mathrm{d}s\,, \quad \forall u \in Y_p\,. \quad (6.19)\end{aligned}$$

Recall that this form has already applied boundary conditions according to Eq. 6.15. In other words, the above corresponds to a DG scheme with the following numerical fluxes:

$$\widehat{\boldsymbol{J}}\cdot\mathbf{n} = \begin{cases} \left\{\!\!\left\{ \frac{1}{\sigma_t}\left(\boldsymbol{Q}_1 - \nabla_h\cdot(\mathbf{E}\varphi)\right)\cdot\mathbf{n} \right\}\!\!\right\} + \alpha(\varphi)\,, & \text{on } \Gamma_0 \\ E_b\varphi + 2J_{\mathrm{in}}\,, & \text{on } \Gamma_b \end{cases}, \quad (6.20\text{a})$$

$$\widehat{\mathbf{E}\varphi\mathbf{n}} = \begin{cases} \{\!\{\mathbf{E}\varphi\mathbf{n}\}\!\} + \boldsymbol{\theta}(\varphi)\,, & \text{on } \Gamma_0 \\ \mathbf{E}\varphi\mathbf{n}\,, & \text{on } \Gamma_b \end{cases}. \quad (6.20\text{b})$$

6.3.1 Interior Penalty

An interior penalty-like method uses

$$\alpha(\varphi) = \kappa\, [\![\varphi]\!]\,, \quad \boldsymbol{\theta}(\varphi) = 0\,, \quad (6.21)$$

where κ is the penalty parameter. IP methods require that $\kappa \propto \sigma_t^{-1}p^2/h$ in order to guarantee stability. The full IP weak form is then: find $\varphi \in Y_p$ such that

$$\begin{aligned}\int_{\Gamma_b} E_b\, u\varphi \,\mathrm{d}s + \int_{\Gamma_0} \kappa\, [\![u]\!]\, [\![\varphi]\!]\,\mathrm{d}s - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{1}{\sigma_t}\nabla_h\cdot(\mathbf{E}\varphi)\cdot\mathbf{n} \right\}\!\!\right\} \mathrm{d}s - \int_{\Gamma_0} \left\{\!\!\left\{ \frac{\nabla_h u}{\sigma_t} \right\}\!\!\right\}\cdot [\![\mathbf{E}\varphi\mathbf{n}]\!]\,\mathrm{d}s \\ + \int \nabla_h u \cdot \frac{1}{\sigma_t}\nabla_h\cdot(\mathbf{E}\varphi)\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} \\ = \int u\, Q_0\,\mathrm{d}\mathbf{x} + \int \nabla_h u\cdot \frac{\boldsymbol{Q}_1}{\sigma_t}\,\mathrm{d}\mathbf{x} - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{\boldsymbol{Q}_1\cdot\mathbf{n}}{\sigma_t} \right\}\!\!\right\} \mathrm{d}s - 2\int_{\Gamma_b} u\, J_{\mathrm{in}}\,\mathrm{d}s\,, \quad \forall u \in Y_p\,. \quad (6.22)\end{aligned}$$

6.3.2 The Second Method of Bassi and Rebay (BR2)

The BR2 method uses an alternative penalty term. Let $\boldsymbol{\rho}_f(\omega) \in X_p$ be a face-local lifting operator defined by

$$\int \boldsymbol{v}\cdot\boldsymbol{\rho}_f(\omega)\,\mathrm{d}\mathbf{x} = -\int_f \{\!\{\boldsymbol{v}\cdot\mathbf{n}\}\!\}\,\omega\,\mathrm{d}s\,, \quad \forall \boldsymbol{v} \in X_p\,, \quad \text{on } f \in \Gamma_0\,. \quad (6.23)$$

Here, ω is a scalar function that is single-valued on the interior face f. Note that the integration on the left hand side is over the entire domain while the right hand side is localized to a single interior face. This means the right hand side, and thus $\boldsymbol{\rho}_f(\omega)$, will be non-zero only for the degrees of freedom in elements that share the face f.

A BR2-like discretization sets

$$\alpha(\varphi) = -\eta \left\{\!\!\left\{ \boldsymbol{\rho}_f([\![\varphi]\!]) \cdot \mathbf{n} \right\}\!\!\right\} \,, \quad \text{on } f \in \Gamma_0 \,, \quad \boldsymbol{\theta}(\varphi) = 0 \,, \tag{6.24}$$

so that the relevant term is

$$\begin{aligned} \int_{\Gamma_0} [\![u]\!] \, \alpha(\varphi) \, \mathrm{d}s &= -\sum_{f \in \Gamma_0} \int_f \eta \; [\![u]\!] \left\{\!\!\left\{ \boldsymbol{\rho}_f([\![u]\!]) \cdot \mathbf{n} \right\}\!\!\right\} \mathrm{d}s \\ &= \sum_{f \in \Gamma_0} \int \eta \, \boldsymbol{\rho}_f([\![u]\!]) \cdot \boldsymbol{\rho}_f([\![\varphi]\!]) \, \mathrm{d}\mathbf{x} \,. \end{aligned} \tag{6.25}$$

This BR2 numerical flux avoids the need to tune the penalty parameter while still allowing element-by-element assembly (see Section 6.4).

The BR2 DG VEF discretization is then: find $\varphi \in Y_p$ such that

$$\begin{aligned} \int_{\Gamma_b} E_b \, u\varphi \, \mathrm{d}s - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{1}{\sigma_t} \nabla_h \cdot (\mathbf{E}\varphi) \cdot \mathbf{n} \right\}\!\!\right\} \mathrm{d}s - \int_{\Gamma_0} \left\{\!\!\left\{ \frac{\nabla_h u}{\sigma_t} \right\}\!\!\right\} \cdot [\![\mathbf{E}\varphi\mathbf{n}]\!] \, \mathrm{d}s \\ + \sum_{f \in \Gamma_0} \int \eta \, \boldsymbol{\rho}_f([\![u]\!]) \cdot \boldsymbol{\rho}_f([\![\varphi]\!]) \, \mathrm{d}\mathbf{x} + \int \nabla_h u \cdot \frac{1}{\sigma_t} \nabla_h \cdot (\mathbf{E}\varphi) \, \mathrm{d}\mathbf{x} + \int \sigma_a \, u\varphi \, \mathrm{d}\mathbf{x} \\ = \int u \, Q_0 \, \mathrm{d}\mathbf{x} + \int \nabla_h u \cdot \frac{\boldsymbol{Q}_1}{\sigma_t} \, \mathrm{d}\mathbf{x} - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{\boldsymbol{Q}_1 \cdot \mathbf{n}}{\sigma_t} \right\}\!\!\right\} \mathrm{d}s - 2 \int_{\Gamma_b} u \, J_{\mathrm{in}} \, \mathrm{d}s \,, \quad \forall u \in Y_p \,. \end{aligned} \tag{6.26}$$

6.3.3 Minimal Dissipation Local Discontinuous Galerkin

Finally, we consider the LDG method. In general, LDG uses the following numerical fluxes:

$$\widehat{\boldsymbol{J}} \cdot \mathbf{n} = \{\!\{ \boldsymbol{J} \cdot \mathbf{n} \}\!\} + \beta \, [\![\boldsymbol{J} \cdot \mathbf{n}]\!] + \kappa \, [\![\varphi]\!] \,, \tag{6.27a}$$

$$\widehat{\mathbf{E}\varphi\mathbf{n}} = \{\!\{ \mathbf{E}\varphi\mathbf{n} \}\!\} - \beta \, [\![\mathbf{E}\varphi\mathbf{n}]\!] \,, \tag{6.27b}$$

where $\boldsymbol{J}$ is defined as the discrete elimination of the current derived in Eq. 6.13. The scalar parameter β can be defined as

$$\beta = \begin{cases} 1/2 \,, & \boldsymbol{w} \cdot \mathbf{n} > 0 \\ -1/2 \,, & \boldsymbol{w} \cdot \mathbf{n} < 0 \end{cases} \,, \quad \mathcal{F} \in \Gamma_0 \,, \tag{6.28}$$

where $\boldsymbol{w}$ is any constant, non-zero vector. This choice imposes an arbitrary upwinding on the current that is balanced by an opposing choice for the scalar flux. With this choice of β, the LDG method is stable for any $\kappa \geq 0$; if $\kappa \equiv 0$, the method is referred to as the minimal dissipation LDG method [100]. Using the numerical flux for the scalar flux, the discrete current simplifies to

$$\boldsymbol{J} = \frac{1}{\sigma_t}\left(\boldsymbol{Q}_1 - \nabla_h \cdot (\mathbf{E}\varphi)\right) - \boldsymbol{r}_0(\llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket) - \boldsymbol{\ell}(\beta \llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket) \,, \tag{6.29}$$

where $\boldsymbol{r}_0(\boldsymbol{\tau}) \in X_p$ is another lifting operator defined by

$$\int \sigma_t \, \boldsymbol{v} \cdot \boldsymbol{r}_0(\boldsymbol{\tau}) \, \mathrm{d}\mathbf{x} = -\int_{\Gamma_0} \{\!\{\boldsymbol{v}\}\!\} \cdot \boldsymbol{\tau} \, \mathrm{d}s \,, \quad \forall \boldsymbol{v} \in X_p \,, \tag{6.30}$$

that differs from $\boldsymbol{r}(\boldsymbol{\tau})$ only in the region of integration on the right hand side. The LDG method is then equivalent to setting

$$\begin{aligned}\alpha(\varphi) = &-\{\!\{\boldsymbol{r}_0(\llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket)\cdot\mathbf{n} + \boldsymbol{\ell}(\beta \llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket)\cdot\mathbf{n}\}\!\} \\ &+ \beta \left\llbracket \frac{1}{\sigma_t}\left(\boldsymbol{Q}_1 - \nabla_h \cdot (\mathbf{E}\varphi)\right)\cdot\mathbf{n} - \boldsymbol{r}_0(\llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket)\cdot\mathbf{n} - \boldsymbol{\ell}(\beta \llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket)\cdot\mathbf{n}\right\rrbracket + \kappa \llbracket \varphi \rrbracket \,, \end{aligned} \tag{6.31a}$$

$$\boldsymbol{\theta}(\varphi) = -\beta \llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket \,. \tag{6.31b}$$

We then have that

$$\begin{aligned}\int_{\Gamma_0} \llbracket u \rrbracket \, \alpha(\varphi) \, \mathrm{d}s = &\int_{\Gamma_0} \beta \llbracket u \rrbracket \left\llbracket \frac{\boldsymbol{Q}_1 \cdot \mathbf{n}}{\sigma_t} \right\rrbracket \mathrm{d}s - \int_{\Gamma_0} \beta \llbracket u \rrbracket \left\llbracket \frac{1}{\sigma_t}\nabla_h \cdot (\mathbf{E}\varphi)\cdot\mathbf{n} \right\rrbracket \mathrm{d}s \\ &+ \int \left(\boldsymbol{\rho}_0(\llbracket u \rrbracket) + \boldsymbol{\lambda}(\beta \llbracket u \rrbracket)\right)\cdot\left(\boldsymbol{r}_0(\llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket) + \boldsymbol{\ell}(\beta \llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket)\right) \mathrm{d}\mathbf{x} + \int_{\Gamma_0} \kappa \llbracket u \rrbracket \llbracket \varphi \rrbracket \, \mathrm{d}s \end{aligned} \tag{6.32}$$

where $\boldsymbol{\rho}_0(\omega), \boldsymbol{\lambda}(\upsilon) \in X_p$ such that

$$\int \boldsymbol{v} \cdot \boldsymbol{\rho}_0(\omega) \, \mathrm{d}\mathbf{x} = -\int_{\Gamma_0} \{\!\{\boldsymbol{v}\cdot\mathbf{n}\}\!\} \, \omega \, \mathrm{d}s \,, \quad \forall \boldsymbol{v} \in X_p \,, \tag{6.33}$$

$$\int \boldsymbol{v} \cdot \boldsymbol{\lambda}(\upsilon) \, \mathrm{d}\mathbf{x} = -\int_{\Gamma_0} \llbracket \boldsymbol{v}\cdot\mathbf{n} \rrbracket \, \upsilon \, \mathrm{d}s \,, \quad \forall \boldsymbol{v} \in X_p \,, \tag{6.34}$$

are analogs of $\boldsymbol{r}_0(\boldsymbol{\tau})$ and $\boldsymbol{\ell}(\boldsymbol{\chi})$, respectively, that do not include the total interaction cross section in the left hand side mass matrices and have scalar arguments. The LDG VEF discretization is then: find $\varphi \in Y_p$ such that

$$\begin{aligned}\int_{\Gamma_b} E_b \, u\varphi \, \mathrm{d}s + \int_{\Gamma_0} \kappa \, \llbracket u \rrbracket \llbracket \varphi \rrbracket \, \mathrm{d}s - \int_{\Gamma_0} \llbracket u \rrbracket \left\{\!\!\left\{ \frac{1}{\sigma_t}\nabla_h \cdot (\mathbf{E}\varphi)\cdot\mathbf{n} \right\}\!\!\right\} \mathrm{d}s - \int_{\Gamma_0} \left\{\!\!\left\{ \frac{\nabla_h u}{\sigma_t} \right\}\!\!\right\} \cdot \llbracket \mathbf{E}\varphi\mathbf{n} \rrbracket \, \mathrm{d}s \\ + \int \left(\boldsymbol{\rho}_0(\llbracket u \rrbracket) + \boldsymbol{\lambda}(\beta \llbracket u \rrbracket)\right)\cdot\left(\boldsymbol{r}_0(\llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket) + \boldsymbol{\ell}(\beta \llbracket \mathbf{E}\varphi\mathbf{n}\rrbracket)\right) \mathrm{d}\mathbf{x} \\ + \int \nabla_h u \cdot \frac{1}{\sigma_t}\nabla_h \cdot (\mathbf{E}\varphi) \, \mathrm{d}\mathbf{x} + \int \sigma_a \, u\varphi \, \mathrm{d}\mathbf{x} \\ = \int u \, Q_0 \, \mathrm{d}\mathbf{x} + \int \nabla_h u \cdot \frac{\boldsymbol{Q}_1}{\sigma_t} \, \mathrm{d}\mathbf{x} - \int_{\Gamma_0} \llbracket u \rrbracket \left(\left\{\!\!\left\{ \frac{\boldsymbol{Q}_1 \cdot \mathbf{n}}{\sigma_t} \right\}\!\!\right\} + \beta \left\llbracket \frac{\boldsymbol{Q}_1 \cdot \mathbf{n}}{\sigma_t} \right\rrbracket \right) \mathrm{d}s - 2\int_{\Gamma_b} u \, J_{\mathrm{in}} \, \mathrm{d}s \,, \quad \forall u \in Y_p \,. \end{aligned} \tag{6.35}$$

The advantage of LDG is that the penalty parameter does not need to scale with the mesh size. However, the LDG stabilization term has a non-compact stencil that connects neighbors of neighbors, leading to less sparsity than the IP or BR2 methods.

6.4 Implementation of Lifting Operators

Consider the face-local lifting operator $\boldsymbol{\rho}_f(\omega)$ used in the BR2 stabilization term defined in Eq. 6.23 with $\omega = [\![u]\!]$ which satisfies

$$\int \boldsymbol{v} \cdot \boldsymbol{\rho}_f([\![u]\!])\,\mathrm{d}\mathbf{x} = -\int_f \{\!\{\boldsymbol{v} \cdot \mathbf{n}\}\!\}\,[\![u]\!]\,\mathrm{d}s\,, \quad \forall \boldsymbol{v} \in X_p\,, \quad \text{on } f \in \Gamma_0\,. \tag{6.36}$$

Let $\underline{y}$ represent the vector of degrees of freedom corresponding to a Y_p or X_p grid function y. Let $\boldsymbol{v}, \boldsymbol{w} \in X_p$ and define

$$\underline{v}^T \mathbf{M} \underline{w} = \int \boldsymbol{v} \cdot \boldsymbol{w}\,\mathrm{d}\mathbf{x} \tag{6.37}$$

as the X_p mass matrix. Further, define

$$\underline{v}^T \mathbf{A}_f \underline{u} = -\int_f \{\!\{\boldsymbol{v} \cdot \mathbf{n}\}\!\}\,[\![u]\!]\,\mathrm{d}s\,, \quad \text{on } f \in \Gamma_0\,, \tag{6.38}$$

for $u \in Y_p$. Equation 6.36 is then equivalent to

$$\mathbf{M}\underline{\rho}_f([\![u]\!]) = \mathbf{A}_f\underline{u} \iff \underline{\rho}_f([\![u]\!]) = \mathbf{M}^{-1}\mathbf{A}_f\underline{u}\,. \tag{6.39}$$

Since the X_p mass matrix is block diagonal by element, its inverse can be computed and stored without fill-in by simply inverting each block individually. The BR2 stabilization term can then be written as

$$\begin{aligned}\sum_{f\in\Gamma_0} \int \boldsymbol{\rho}_f([\![u]\!]) \cdot \boldsymbol{\rho}_f([\![\varphi]\!])\,\mathrm{d}\mathbf{x} &= \sum_{f\in\Gamma_0} \underline{\rho}_f([\![u]\!])^T \mathbf{M} \underline{\rho}_f([\![\varphi]\!]) \\ &= \sum_{f\in\Gamma_0} \underline{u}^T \mathbf{A}_f^T \mathbf{M}^{-T}\mathbf{M}\mathbf{M}^{-1}\mathbf{A}_f\underline{\varphi} \\ &= \sum_{f\in\Gamma_0} \underline{u}^T \mathbf{A}_f^T \mathbf{M}^{-1}\mathbf{A}_f\underline{\varphi}\end{aligned} \tag{6.40}$$

since $\mathbf{M}$ is symmetric. Again, since $\mathbf{M}^{-1}$ is block diagonal by element and the products $\mathbf{A}_f\underline{\varphi}$ and $\underline{u}^T\mathbf{A}_f^T$ are non-zero only on degrees of freedom that share the face f, each argument of the sum only contributes to the degrees of freedom that share the face f. Due to this, the matrix $\sum_{f\in\Gamma_0} \mathbf{A}_f^T\mathbf{M}^{-1}\mathbf{A}_f$ can be assembled face by face.

Next, consider one part of the LDG stabilization term:

$$\int \boldsymbol{\rho}_0([\![u]\!]) \cdot \boldsymbol{r}_0([\![\mathbf{E}\varphi\mathbf{n}]\!])\,\mathrm{d}\mathbf{x}\,. \tag{6.41}$$

Let,

$$\underline{v}^T \mathbf{B} \underline{\varphi} = - \int_{\Gamma_0} \{\!\{ \boldsymbol{v} \}\!\} \cdot [\![\mathbf{E}\varphi\mathbf{n}]\!] \, \mathrm{d}s \,, \tag{6.42}$$

and further define the total interaction X_p mass matrix as

$$\underline{v}^T \mathbf{M}_t \underline{w} = \int \sigma_t \, \boldsymbol{v} \cdot \boldsymbol{w} \, \mathrm{d}\mathbf{x} \,, \tag{6.43}$$

so that $\underline{r}_0([\![\mathbf{E}\varphi\mathbf{n}]\!]) = \mathbf{M}_t^{-1}\mathbf{B}\underline{\varphi}$. In addition, define

$$\underline{v}^T \mathbf{A} \underline{u} = - \int_{\Gamma_0} \{\!\{ \boldsymbol{v} \cdot \mathbf{n} \}\!\} \, [\![u]\!] \, \mathrm{d}s \,, \tag{6.44}$$

such that $\mathbf{A} = \sum_{f \in \Gamma_0} \mathbf{A}_f$. The LDG stabilization term under consideration is then

$$\begin{aligned} \int \boldsymbol{\rho}_0([\![u]\!]) \cdot \boldsymbol{r}_0([\![\mathbf{E}\varphi\mathbf{n}]\!]) \, \mathrm{d}\mathbf{x} &= \underline{\rho}_0([\![u]\!])^T \mathbf{M} \underline{r}_0([\![\mathbf{E}\varphi\mathbf{n}]\!]) \\ &= \underline{u}^T \mathbf{A}^T \mathbf{M}^{-T} \mathbf{M} \mathbf{M}_t^{-1} \mathbf{B} \underline{\varphi} \\ &= \underline{u}^T \mathbf{A}^T \mathbf{M}_t^{-1} \mathbf{B} \underline{\varphi} \,. \end{aligned} \tag{6.45}$$

Note that since the matrices $\mathbf{A}$ and $\mathbf{B}$ are not face-local, this term cannot be assembled locally. The LDG stabilization term is instead formed through matrix multiplication as $\mathbf{A}^T \mathbf{M}_t^{-1} \mathbf{B}$.

6.5 Extracting a Continuous Discretization from the DG Framework

We now show how a CG discretization of the VEF drift-diffusion equation can be extracted from the DG framework presented above. An approximate inversion of this operator is one stage of the subspace correction preconditioner described in Section 6.6 that is used to efficiently solve the IP and BR2 VEF discretizations. This CG operator is also a VEF method itself and represents an extension to multiple dimensions, arbitrary-order, and curved meshes of the algorithm in [22]. A CG VEF method has fewer unknowns than an analogous DG method and requires simpler methods to solve the resulting linear system. We will show that this CG discretization has similar accuracy to DG and does not degrade convergence of the fixed-point iteration even in the asymptotic thick diffusion limit. However, it is unclear if using a continuous finite element space would negatively impact robustness and stability in the larger radiation-hydrodynamics multiphysics setting.

Let $u, \varphi \in V_p$, the degree-p continuous finite element space, then

$$[\![u]\!] = 0 \,, \quad [\![\varphi]\!] = 0 \,, \quad \text{on } \mathcal{F} \in \Gamma_0 \,. \tag{6.46}$$

However, since the Eddington tensor is still discontinuous, we have that

$$[\![\mathbf{E}\varphi\mathbf{n}]\!] = [\![\mathbf{E}\mathbf{n}]\!]\,\varphi\,. \tag{6.47}$$

Note that, for $u \in V_p$, $\nabla u \in X_p$. In other words, while $u \in V_p$ is continuous ∇u is not. Thus, by starting from the DG VEF discretization and assembling onto V_p, we arrive at a CG VEF discretization of the form: find $\varphi \in V_p$ such that

$$\begin{aligned}\int_{\Gamma_b} E_b\, u\varphi \,\mathrm{d}s - \int_{\Gamma_0} \left\{\!\!\left\{ \frac{\nabla u}{\sigma_t} \right\}\!\!\right\} \cdot [\![\mathbf{E}\mathbf{n}]\!]\varphi \,\mathrm{d}s + \int \nabla u \cdot \frac{1}{\sigma_t}\nabla_h \cdot (\mathbf{E}\varphi)\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi \,\mathrm{d}\mathbf{x} \\ = \int u\, Q_0 \,\mathrm{d}\mathbf{x} + \int \nabla u \cdot \frac{\boldsymbol{Q}_1}{\sigma_t}\,\mathrm{d}\mathbf{x} - 2\int_{\Gamma_b} u\, J_{\mathrm{in}}\,\mathrm{d}s\,, \quad \forall u \in V_p\,. \end{aligned} \tag{6.48}$$

Observe that in the thick diffusion limit, where $\mathbf{E} = \frac{1}{3}\mathbf{I}$ and $E_b = 1/2$, a CG discretization of radiation diffusion with Marshak boundary conditions arises since $[\![\mathbf{E}\mathbf{n}]\!] = 0$ and $\frac{1}{\sigma_t}\nabla_h \cdot (\mathbf{E}\varphi) = \frac{1}{3\sigma_t}\nabla\varphi$.

6.6 Uniform Subspace Correction Preconditioner

Pazner and Kolev [59] presents a preconditioning strategy for DG discretizations of elliptic problems based on a decomposition of the DG space. The decomposition is chosen so that the resulting preconditioner can leverage the efficiency and scalability of AMG applied to a continuous finite element discretization of an elliptic problem. The resulting scheme is closely related to preconditioning a DG operator with a CG operator. However, this method includes an additional step where a classical smoother is applied to a DG operator in order to provide a preconditioner that scales optimally with respect to the mesh size, polynomial order, and penalty parameter. Here, we present the implementation details in applying this preconditioner to the DG discretizations developed in this chapter.

6.6.1 Decomposition of the DG Space

The DG space is split into a continuous finite element part and a discontinuous finite element part. In order to allow definition of a continuous finite element space, we restrict to the case where the DG space uses the closed Gauss-Lobatto basis on each element. Let V_C denote the continuous subspace of Y_p. In other words, $V_C = Y_p \cap H^1(\mathcal{D})$. Due to the use of Gauss-Lobatto nodes, $V_C = V_p$, the degree-p continuous finite element space.

We now seek to construct a space, V_B, such that $Y_p = V_C + V_B$. Here, the "B" subscript stands for "boundary." Let $\mathcal{B}(K)$ denote the set of nodes located on the boundary of the element, ∂K, and $\mathcal{I}(K)$ the set of nodes located on the interior of the element. The space V_B consists of functions v_b such that $v_b(\mathbf{x}_{K,i}) = 0$ for each $\mathbf{x}_{K,i} \in \mathcal{I}(K)$ for all elements $K \in \mathcal{T}$. In other words, $v_b \in V_B$ attains a value of zero at each interior node of every element in the

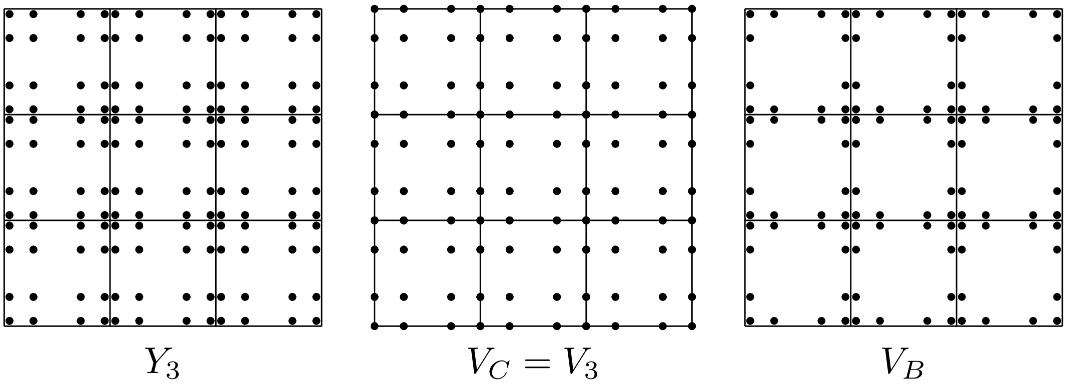


Figure 6.3: A depiction of the degrees of freedom associated with the decomposition of the DG space. The left diagram shows the discontinuous space Y_3 where the nodal basis for each element interpolates through the Gauss-Lobatto points. The nodes on the boundary of each element are offset inwards in order to show the placement of nodes on both sides of the interface between interior elements. The middle diagram shows the node placement for the corresponding continuous finite element space. Here, the nodes are shared between neighboring elements. The right diagram depicts the boundary space consisting of only the nodes on the boundary of each element.

mesh. Alternatively, a function $u \in Y_p$ could be represented as $u = u_c + u_b$ where $u_c \in V_C$ and $u_b \in Y_p$. However, V_B is preferred due to its smaller dimension.

This decomposition is depicted in Fig. 6.3 for the space Y_3. The nodes $\boldsymbol{\xi}_i \in \mathcal{B}(K)$ of Y_3 and V_B are offset toward the interior of the element in order to show the presence of multiple nodes on each interior face in the mesh. Here, the continuous space V_C is equivalent to V_3 the cubic continuous finite element space. The right diagram depicts the boundary space V_B which has only the nodes of Y_3 corresponding to $\mathcal{B}(K)$ for each $K \in \mathcal{T}$. Note that V_B has nodes on both sides of each interior interface.

We now present operators $\mathbf{Q} : Y_p \to V_C$ and $\mathbf{B} : Y_p \to V_B$ that take functions in Y_p and convert them to V_C and V_B, respectively. Let $u \in Y_p$ be given. The operator $\mathbf{Q}u$ copies the degrees of freedom corresponding to nodes interior to each element or on the boundary of the domain so that $\mathbf{Q}u(\mathbf{x}_i) = u(\mathbf{x}_i)$ at each $\mathbf{x}_i \in \mathcal{I}(K)$ for each K and $\mathbf{x}_i \in \partial\mathcal{D}$. For nodes located on interior faces of the mesh, $\mathbf{Q}u$ averages the values of u in the elements that share the face. That is for a face $\Gamma_0 \ni \mathcal{F} = K_1 \cap K_2$, we set

$$\mathbf{Q}u(\mathbf{x}_i) = \frac{1}{2}\left(u|_{K_1} + u|_{K_2}\right) , \tag{6.49}$$

for each node $\mathbf{x}_i \in \mathcal{F}$. The action of the operator $\mathbf{B}$ on $u \in Y_p$ restricts u to the space V_B. That is, degrees of freedom associated with $\mathcal{I}(K)$ for each K are removed leaving only the nodes corresponding to $\mathcal{B}(K)$ for each K. The handling of the complications arising in hp refinement (e.g. non-conforming hanging nodes) is presented in [59].

6.6.2 Additive Schwarz Preconditioner

We now define a preconditioner built on the decomposition $Y_p = V_C + V_B$. Let $\mathcal{A} : Y_p \times Y_p \to \mathbb{R}$ denote one of the DG VEF bilinear forms derived in this chapter. We also define the linear operator $\mathbf{A} : Y_p \to Y_p$ such that

$$\underline{v}^T \mathbf{A} \underline{u} = \mathcal{A}(v, u)\,, \quad u, v \in Y_p\,. \tag{6.50}$$

In other words, $\mathcal{A}$ is a DG VEF bilinear form (e.g. Eq. 6.22) while $\mathbf{A}$ is the corresponding matrix. Further, we define the restrictions of $\mathbf{A}$ to V_C and V_B as

$$\underline{v}_c^T \mathbf{A}_c \underline{u}_c = \mathcal{A}(v_c, u_c)\,, \quad u_c, v_c \in V_c\,, \tag{6.51a}$$

$$\underline{v}_b^T \mathbf{A}_b \underline{u}_b = \mathcal{A}(v_b, u_b)\,, \quad u_b, v_b \in V_B\,. \tag{6.51b}$$

Note that the restriction of $\mathcal{A}$ to V_C represented by $\mathbf{A}_c$ is exactly the matrix corresponding to the CG VEF operator in Eq. 6.48. We define the so-called *elliptic projections* onto the spaces V_C and V_B as $\mathbf{P}_c : Y_p \to V_C$ and $\mathbf{P}_b : Y_p \to V_B$, respectively, which satisfy

$$\mathbf{A}_c \mathbf{P}_c = \mathbf{Q}\mathbf{A}\,, \tag{6.52a}$$

$$\mathbf{A}_b \mathbf{P}_b = \mathbf{B}\mathbf{A}\,. \tag{6.52b}$$

Applying the inverse, the projections are given by

$$\mathbf{P}_c = \mathbf{A}_c^{-1} \mathbf{Q}\mathbf{A}\,, \tag{6.53}$$

$$\mathbf{P}_b = \mathbf{A}_b^{-1} \mathbf{B}\mathbf{A}\,. \tag{6.54}$$

The spaces V_C and V_B are sufficiently large that $\mathbf{A}_c$ and $\mathbf{A}_b$ are impractical to invert directly. We thus approximate the inverses using $\mathbf{R}_c \approx \mathbf{A}_c^{-1}$ and $\mathbf{R}_b \approx \mathbf{A}_b^{-1}$. We use one AMG V-cycle for $\mathbf{R}_c$ and one iteration of point Jacobi for $\mathbf{R}_b$. The projections are prolonged to the space Y_p using the transpose of their associated restriction operators. The preconditioned operator is given by the sum of the prolonged, approximate elliptic projections:

$$\mathbf{P}^{-1}\mathbf{A} = (\mathbf{Q}^T \mathbf{P}_c + \mathbf{B}^T \mathbf{P}_b)\mathbf{A} = (\mathbf{Q}^T \mathbf{R}_c \mathbf{Q} + \mathbf{B}^T \mathbf{R}_b \mathbf{B})\mathbf{A}\,. \tag{6.55}$$

The preconditioner $\mathbf{P}^{-1}$ is then

$$\mathbf{P} = \mathbf{Q}^T \mathbf{R}_c \mathbf{Q} + \mathbf{B}^T \mathbf{R}_b \mathbf{B}\,. \tag{6.56}$$

It is applied in two stages corresponding to V_C and V_B each of which include restriction to the subspace, approximately inverting the operator derived by restricting $\mathbf{A}$ to the subspace, and prolonging back to the original space Y_p. In Pazner and Kolev [59], it is shown that the preconditioned system $\mathbf{P}^{-1}\mathbf{A}$ has condition number independent of the mesh size, polynomial order, and penalty parameter (if applicable).

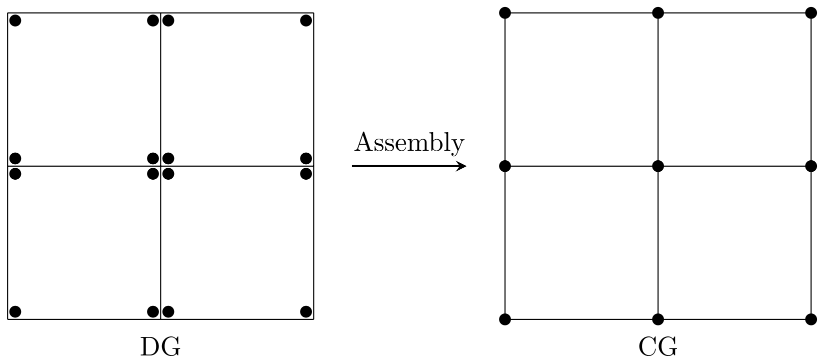


Figure 6.4: A depiction of the assembly operator applied to the space Y_1 to yield an operator defined on the continuous finite element space V_1. By assembling a DG discretization of the VEF equations onto a suitable CG space, the CG operator can be computed without rediscretizing.

In the implementation, the operator $\mathbf{A}_c$ is computed by *assembling* $\mathbf{A}$ onto the space V_C. This is an algebraic process that allows the CG operator to be formed without rediscretizing and re-evaluating the bilinear forms that comprise $\mathbf{A}$. This process is depicted in Fig. 6.4. The operators $\mathbf{Q}$ and $\mathbf{B}$ are implemented in a matrix-free fashion. That is, we define only their action and transpose action onto the residual vector. Finally, the operator $\mathbf{R}_b$ is computed by assembling the diagonal of $\mathbf{A}$ and applying $\mathbf{B}$ so that only the degrees of freedom corresponding to V_B remain. The action of $\mathbf{R}_b$ onto a vector $\underset{\sim}{r}$ is then equivalent to dividing the entries of $\underset{\sim}{r}$ corresponding to $\mathcal{B}(K)$ for each K by the associated diagonal entry of $\mathbf{A}$.

6.7 Results

The VEF algorithms described in this chapter were implemented using the MFEM [101, 102] finite element framework. The BiCGStab and Jacobi solvers from MFEM were used to solve the VEF discretizations along with BoomerAMG, the AMG solver from the sparse linear algebra library *hypre* [94]. KINSOL, from the Sundials package [103], provided the Anderson-accelerated fixed-point solver. When iterative solver results are not presented, the parallel implementation of the sparse direct solver SuperLU [81] was used. We use the high-order DG S$_N$ transport solver from [15].

Unless otherwise specified, we set the penalty parameter to

$$\kappa_e = \left\{\!\!\left\{ \frac{(p+1)^2}{\sigma_t h_e} \right\}\!\!\right\} \tag{6.57}$$

and the BR2 stabilization parameter to $\eta = 4$. We use the MDLDG method, the variant of the LDG method where $\kappa \equiv 0$ and set the upwinding vector $\boldsymbol{w}$ to be a unit vector at a 45°

angle from the x-axis. The VEF discretizations all use the element-local basis defined using the Gauss-Lobatto points to enable the use of the subspace correction preconditioner where required. The transport discretization is always solved with the same finite element order as the VEF scalar flux. However, we use the positive Bernstein basis [99] for the transport discretization.

6.7.1 Method of Manufactured Solutions

The accuracy of the methods are determined with the MMS. The solution is set to

$$\psi = \frac{1}{4\pi}[\alpha(\mathbf{x}) + \boldsymbol{\Omega}\cdot\boldsymbol{\beta}(\mathbf{x}) + \boldsymbol{\Omega}\otimes\boldsymbol{\Omega} : \boldsymbol{\Theta}(\mathbf{x})] \,, \tag{6.58}$$

where

$$\alpha(\mathbf{x}) = \sin(\pi x)\sin(\pi y) + \delta \,, \tag{6.59a}$$

$$\boldsymbol{\beta}(\mathbf{x}) = \begin{bmatrix} \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \\ \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \end{bmatrix} \,, \tag{6.59b}$$

$$\boldsymbol{\Theta}(\mathbf{x}) = \begin{bmatrix} \frac{1}{2}\sin\left(\frac{3\pi(x+\zeta)}{1+2\zeta}\right)\sin\left(\frac{3\pi(y+\zeta)}{1+2\zeta}\right) & \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \\ \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) & \frac{1}{4}\sin\left(\frac{3\pi(x+\zeta)}{1+2\zeta}\right)\sin\left(\frac{3\pi(y+\zeta)}{1+2\zeta}\right) \end{bmatrix} \,. \tag{6.59c}$$

The parameter $\delta = 1.25$ is used to ensure $\psi > 0$ and $\zeta = 0.1$ and $\omega = 0.05$ are used to test spatially-dependent, non-isotropic inflow boundary conditions. The domain is $\mathcal{D} = [0,1]^2$. With this definition:

$$\phi(\mathbf{x}) = \alpha(\mathbf{x}) + \frac{1}{3}\operatorname{trace}\boldsymbol{\Theta}(\mathbf{x}) \,, \tag{6.60a}$$

$$\boldsymbol{J}(\mathbf{x}) = \frac{1}{3}\boldsymbol{\beta}(\mathbf{x}) \,, \tag{6.60b}$$

$$\mathbf{P}(\mathbf{x}) = \frac{\alpha(\mathbf{x})}{3}\mathbf{I} + \frac{1}{15}\begin{bmatrix} 3\Theta_{11}(\mathbf{x}) + \Theta_{22}(\mathbf{x}) & \Theta_{12}(\mathbf{x}) \\ \Theta_{21}(\mathbf{x}) & \Theta_{11}(\mathbf{x}) + 3\Theta_{22}(\mathbf{x}) \end{bmatrix} \,. \tag{6.60c}$$

This leads to an exact Eddington tensor $\mathbf{E} = \mathbf{P}/\phi$ that is dense and spatially varying. The MMS ψ and ϕ are substituted into the transport equation to solve for the MMS source q that forces the solution to Eq. 6.58.

The accuracy of the VEF discretizations can be investigated in isolation by computing the VEF data from the MMS angular flux and setting the sources Q_0 and $\boldsymbol{Q}_1$ to the moments of the transport MMS source. This is accomplished by computing the VEF data from the MMS angular flux projected onto a finite element space of equal order to the VEF finite element space. An open, Gauss-Legendre basis is used for the angular flux so that the Eddington tensor will have discontinuities of magnitude $\mathcal{O}(h^{p+1})$ on interior mesh faces. The VEF data and source moments are computed using level symmetric S_4 angular quadrature. The VEF equations are then solved as if $\mathbf{E}$, E_b, Q_0, $\boldsymbol{Q}_1$, and J_{in} are provided data.

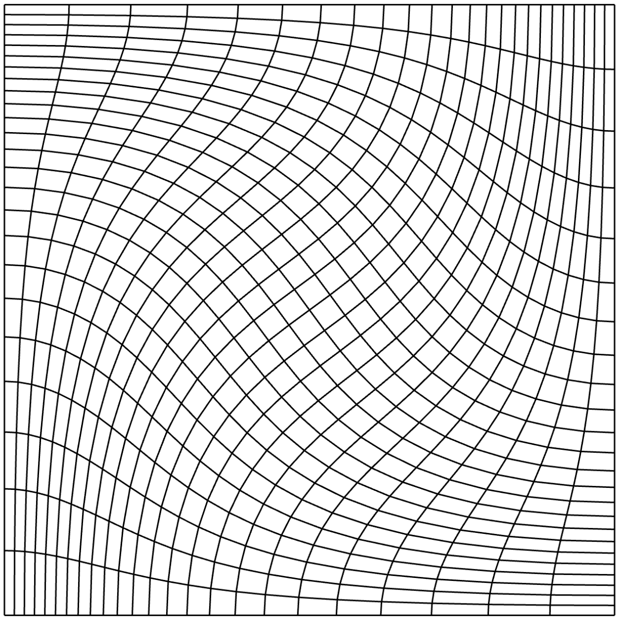


Figure 6.5: An example of a third-order mesh distorted by advecting the interior nodes according to the velocity field of the Taylor-Green hydrodynamics problem.

We use refinements of a third-order curved mesh created by distorting an orthogonal mesh according to the velocity field of the Taylor Green vortex. This mesh distortion is generated by advecting the mesh control points with

$$\mathbf{x} = \int_0^T \mathbf{v}\,\mathrm{d}t\,, \tag{6.61}$$

where the final time $T = 0.3\pi$ and

$$\mathbf{v} = \begin{bmatrix} \sin(x_1)\cos(x_2) \\ -\cos(x_1)\sin(x_2) \end{bmatrix} \tag{6.62}$$

is the analytic solution of the Taylor Green vortex. The time integration is calculated with 300 forward Euler time steps. An example mesh is shown in Fig. 6.5.

Figure 6.6 shows the $L^2(\mathcal{D})$ error between the VEF solution and the exact MMS scalar flux solution as the mesh is refined for the IP, BR2, MDLDG, and CG VEF discretizations when quadratic basis functions are used. Here, h is the *maximum* value of the characteristic element length in the mesh. All methods have nearly identical error behavior and converge with third-order accuracy as expected. This experiment is repeated with $p = 1$ and $p = 3$ in Tables 6.1 and 6.2, respectively. Logarithmic regression is used to compute the exponent and constant of the error function $E = Ch^{\bar{p}}$ with C the constant and $\bar{p}$ the method's experimentally-observed order of accuracy. The standard deviation of the four error values

Table 6.1: MMS error for each method as a function of the *maximum* characteristic mesh size, h. The standard deviation of the four error values in each row is also provided. First-order polynomial basis functions were used. The order of accuracy and error constant were computed with logarithmic regression.

h	IP	BR2	MDLDG	CG	Deviation
3.994×10^{-2}	4.018×10^{-3}	3.999×10^{-3}	3.557×10^{-3}	4.022×10^{-3}	1.978×10^{-4}
1.997×10^{-2}	1.006×10^{-3}	1.004×10^{-3}	8.819×10^{-4}	1.006×10^{-3}	5.350×10^{-5}
1.331×10^{-2}	4.472×10^{-4}	4.468×10^{-4}	3.890×10^{-4}	4.472×10^{-4}	2.515×10^{-5}
9.985×10^{-3}	2.515×10^{-4}	2.514×10^{-4}	2.178×10^{-4}	2.515×10^{-4}	1.459×10^{-5}
Order	1.999	1.996	2.015	2.000	
Constant	2.510	2.473	2.340	2.518	

Table 6.2: MMS error for each method as a function of the *maximum* characteristic mesh size, h. The standard deviation of the four error values in each row is also provided. Third-order polynomial basis functions were used. The order of accuracy and error constant were computed with logarithmic regression.

h	IP	BR2	MDLDG	CG	Deviation
7.681×10^{-2}	1.451×10^{-4}	1.437×10^{-4}	1.396×10^{-4}	1.456×10^{-4}	2.364×10^{-6}
3.994×10^{-2}	1.040×10^{-5}	1.037×10^{-5}	9.616×10^{-6}	1.041×10^{-5}	3.360×10^{-7}
2.628×10^{-2}	1.953×10^{-6}	1.950×10^{-6}	1.779×10^{-6}	1.953×10^{-6}	7.501×10^{-8}
1.997×10^{-2}	6.522×10^{-7}	6.516×10^{-7}	5.904×10^{-7}	6.523×10^{-7}	2.669×10^{-8}
Order	4.012	4.006	4.058	4.014	
Constant	4.280	4.172	4.615	4.320	

for each value of h is also provided to quantify the variance in the error behavior. Accuracy of $\mathcal{O}(h^{p+1})$ is observed and the four variants are shown to have variance below the discretization error.

6.7.2 Thick Diffusion Limit

Next, we investigate the iterative convergence properties of the VEF methods in the regime known as the asymptotic thick diffusion limit [25]. The material data are set to

$$\sigma_t = \frac{1}{\epsilon}\,, \quad \sigma_a = \epsilon\,, \quad \sigma_s = \frac{1}{\epsilon} - \epsilon\,, \quad q = \epsilon \tag{6.63}$$

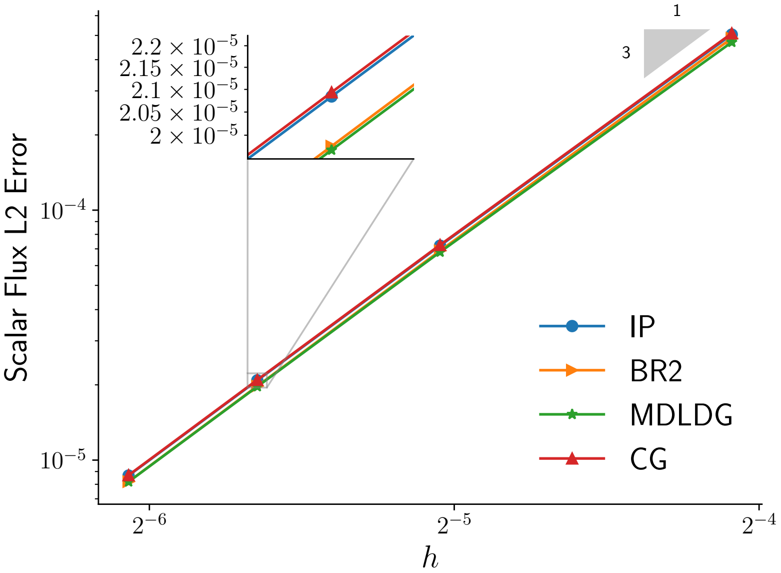


Figure 6.6: The error in the scalar flux as the mesh is refined. Quadratic basis functions were used. Comparison to the reference third-order slope indicates all methods converge with optimal third-order accuracy.

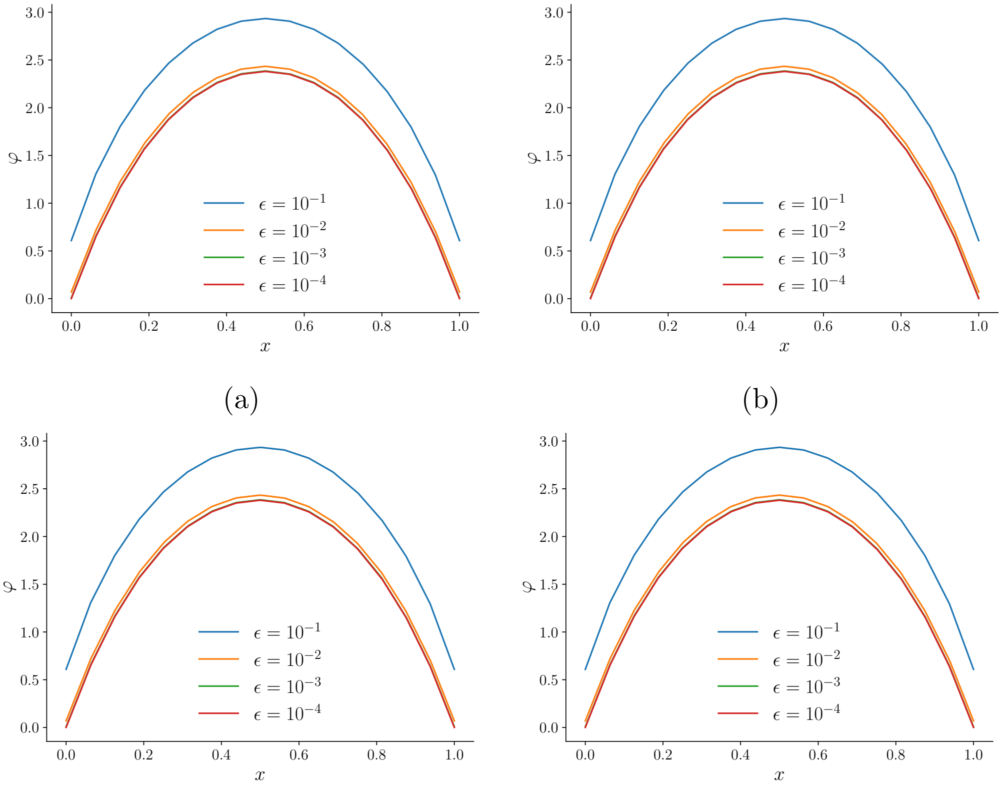
with $\epsilon \in (0, 1]$ and the thick diffusion limit corresponding to $\epsilon \to 0$. A coarse mesh that does not adequately resolve the mean free path is used to stress the convergence of the VEF algorithm. This is a numerically challenging, but common in practice, regime where robust performance is crucial.

We first demonstrate robust convergence on an 8×8 linear mesh with $\mathcal{D} = [0, 1]^2$. Convergence was identical for linear, quadratic, and cubic basis functions so we present results for $p = 2$ only. Level symmetric S_4 angular quadrature is used. Fixed-point iteration without Anderson acceleration is used to solve the coupled transport-VEF system. Table 6.3 shows the number of iterations required to converge to a fixed-point tolerance of 10^{-6} as $\epsilon \to 0$. All four VEF variants converge robustly and in an identical number of iterations for each value of ϵ. All methods converged to the non-trivial diffusion limit solution. Lineouts of the 2D solutions are shown in Fig. 6.7 to demonstrate that the non-trivial, diffusion solution is obtained by each of the four methods. Note that even the continuous finite element discretization paired with the discontinuous finite element transport discretization was(robust in the thick diffusion limit.

This experiment is repeated on the triple point mesh shown in Fig. 6.8. This mesh was generated by running a purely Lagrangian hydrodynamics simulation on a third-order mesh. The mesh contains concave/reentrant interior faces meaning the matrix corresponding to the transport discretization *cannot* be re-ordered to be strictly lower block triangular. The pseudo-optimally reordered sweep from [15], which lags the incoming angular flux on

Table 6.3: Number of iterations to convergence in the thick diffusion limit on a coarse, orthogonal mesh.

ϵ	IP	BR2	MDLDG	CG
10^{-1}	8	8	8	8
10^{-2}	6	6	6	6
10^{-3}	4	4	4	4
10^{-4}	3	3	3	3



(a) (b)

(c) (d)

Figure 6.7: Lineouts of the 2D thick diffusion limit solutions taken at $y = \frac{1}{2}$ for the (a) IP, (b) BR2, (c) MDLDG, and (d) CG methods.

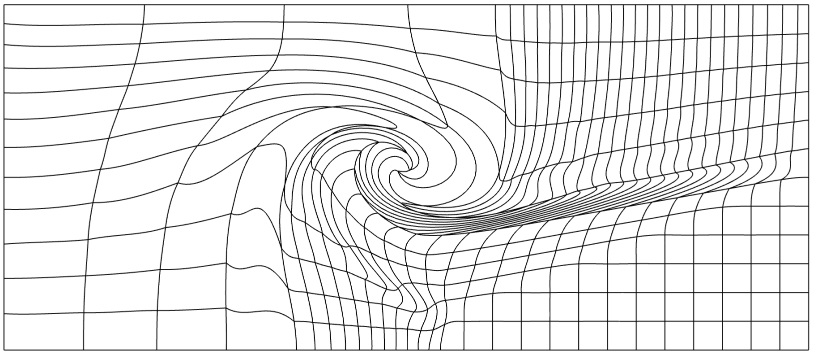


Figure 6.8: A depiction of the triple point mesh used to stress test the VEF algorithms on a severely distorted, third-order mesh. The mesh was generated with a purely Lagrangian hydrodynamics simulation.

reentrant faces, is used to enable an element-by-element transport solve. Since the incoming fluxes on reentrant faces are lagged, the angular flux on these faces is not linearly eliminated. In other words, the presence of reentrant faces means that the transport equation is not fully inverted at every fixed-point iteration. In addition, the mesh elements in the "swirl" at the center are severely distorted and thus have poor approximation ability. In practice, the mesh would be remapped before this level of distortion were present. Due to this severe distortion, stability of the IP VEF discretization required scaling the penalty parameter according to

$$\kappa_e = C\frac{(p+1)^2}{\sigma_t h_e} \,, \tag{6.64}$$

where $C = \max_{K_e\in\mathcal{T}} C_e$ with C_e the condition number of the Jacobian matrix for element K_e. For the triple point mesh, $C = 169$.

Table 6.4 shows the number of fixed-point iterations to converge the thick diffusion limit problems on the triple point mesh. The IP, BR2, and CG methods converged equivalently with MDLDG generally converging slower. This is likely due to MDLDG being less numerically diffusive compared to the IP, BR2, and CG methods which either have a penalization term that regularizes towards a continuous solution or is a continuous method.

6.7.3 Crooked Pipe

We now demonstrate the efficacy of the methods on a more realistic, multi-material problem. A common benchmark is the crooked pipe problem. The geometry and materials are shown in Fig. 6.9. The problem consists of two materials, the wall and the pipe, which have an 1000x difference in total interaction cross section. We mock the time-dependent benchmark as a steady-state problem by adding artificial absorption and fixed-source terms corresponding to backward Euler time integration. We use a large time step such that $c\Delta t = 10^3$ with an

Table 6.4: Number of fixed-point iterations required for convergence on the triple point mesh as $\epsilon \to 0$. On the triple point mesh, reentrant faces mean the transport equation is not fully inverted at each iteration.

ϵ	IP	BR2	MDLDG	CG
10^{-1}	19	19	23	19
10^{-2}	11	11	19	11
10^{-3}	8	8	9	8
10^{-4}	6	6	6	6

initial condition $\psi_0 = 10^{-4}$ for all $(\mathbf{x}, \mathbf{\Omega}) \in \mathcal{D} \times \mathbb{S}^2$. The absorption and source terms are then

$$\sigma_a = \frac{1}{c\Delta t} = 10^{-3}\,\frac{1}{\mathrm{cm}}\,, \tag{6.65a}$$

$$q = \frac{1}{c\Delta t}\psi_0 = 10^{-1}\,\frac{1}{\mathrm{cm}^3 \cdot \mathrm{s} \cdot \mathrm{str}}\,. \tag{6.65b}$$

The boundary conditions are

$$f = \begin{cases} \frac{1}{2\pi}\,, & x = 0 \text{ and } y \in [-1/2, 1/2] \\ 0\,, & \text{otherwise} \end{cases}\,, \tag{6.66}$$

so that radiation enters the pipe at the left side of the domain. We use a Level Symmetric S_{12} angular quadrature set. The zero and scale [98] negative flux fixup, a sweep compatible method that zeros out negativity and rescales so that particle balance is preserved, is used inside the transport inversion to ensure positivity.

A VEF solution to the crooked pipe using the IP method with $p = 2$ is shown in Fig. 6.10 where a non-uniform mesh is used to adequately resolve the interface between the optically thin pipe and the optically thick wall. Here, we see that VEF does capture the "shadow" induced by the radiation turning the corner around the inner wall. A radiation diffusion solution would non-physically show illumination on the back side of the wall.

The outer fixed-point and inner linear iterative efficiency is demonstrated by refining in h and p. Note that to simplify the refinement process we use a uniform mesh. The outer solver is Anderson-accelerated fixed-point iteration with two Anderson vectors. Anderson acceleration is not required for convergence on this problem but does provide more uniform convergence in h. Since the mesh is orthogonal, the transport equation is fully inverted at each outer iteration. This allows use of the low memory variant so that the storage cost of Anderson acceleration is two scalar flux-sized vectors. The outer tolerance is 10^{-6}. The inner BiCGStab tolerance is 10^{-8}. The USC preconditioner with one Jacobi iteration and one AMG V-cycle per application is used for the IP and BR2 discretizations. The CG and MDLDG discretizations use one V-cycle of AMG as a preconditioner. The previous

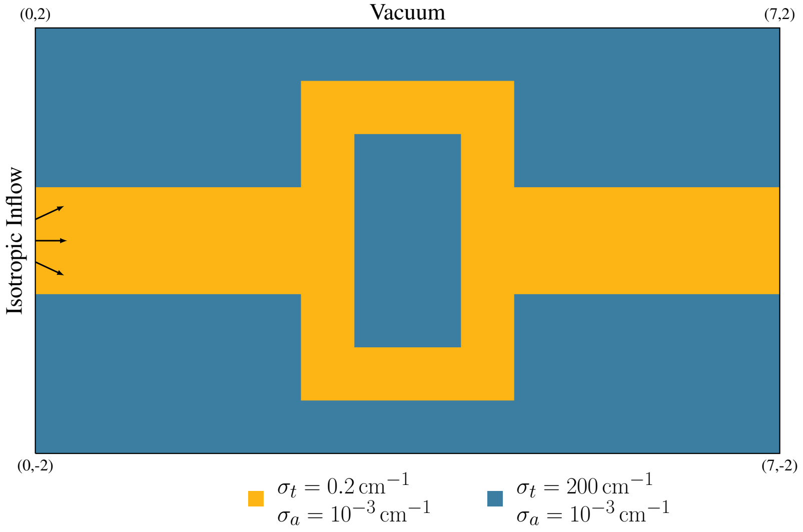

(0,2)
Vacuum
(7,2)
Isotropic Inflow
(0,-2)
(7,-2)
$\sigma_t = 0.2\,\text{cm}^{-1}$
$\sigma_a = 10^{-3}\,\text{cm}^{-1}$
$\sigma_t = 200\,\text{cm}^{-1}$
$\sigma_a = 10^{-3}\,\text{cm}^{-1}$

Figure 6.9: Geometry, material data, and boundary conditions for the linearized crooked pipe problem.

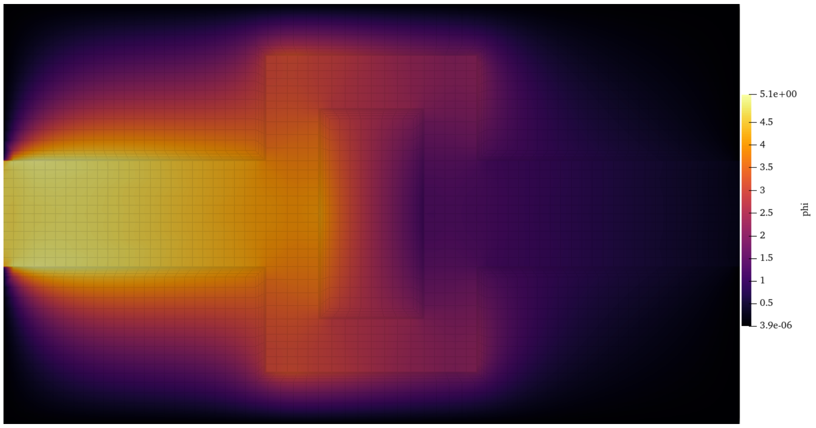


Figure 6.10: VEF scalar flux solution to the linearized crooked pipe problem to show that VEF does capture the transport solution. The mesh is refined at the interface between thick and thin to adequately resolve the material interface. The IP VEF method with $p = 2$ was used.

Table 6.5: The number of Anderson-accelerated fixed-point iterations until convergence to a tolerance of 10^{-6} for the IP, BR2, MDLDG, and CG discretizations of VEF on the linearized crooked pipe problem refined in h and p. An Anderson space of size two is used.

		Outer				Max Inner				Min Inner				Avg. Inner			
	N_e	IP	BR2	CG	MDLDG	IP	BR2	CG	MDLDG	IP	BR2	CG	MDLDG	IP	BR2	CG	MDLDG
$p = 1$	112	10	10	10	14	16	16	7	10	6	6	3	3	11.50	11.20	4.90	6.43
	448	11	11	12	16	17	16	7	11	7	7	2	3	12.00	11.18	4.75	6.94
	1792	13	13	13	16	18	18	7	11	4	4	2	4	11.23	11.00	4.85	7.38
	7168	14	14	14	18	18	17	8	12	6	6	2	3	11.50	11.21	5.00	6.94
$p = 2$	112	13	13	13	16	16	16	9	11	5	5	3	4	10.69	10.85	6.23	7.62
	448	15	15	15	18	17	17	10	12	5	5	3	4	11.20	10.80	6.07	7.94
	1792	16	16	16	18	17	16	10	14	4	4	3	4	11.12	11.12	6.44	9.11
	7168	17	17	17	19	17	19	11	14	5	5	3	4	11.18	11.35	6.41	9.32
$p = 3$	112	15	15	15	17	19	18	11	22	5	6	3	5	12.47	12.20	7.53	12.29
	448	16	16	16	18	22	18	12	17	7	6	5	6	14.00	13.19	8.56	11.61
	1792	17	17	17	19	22	22	13	18	6	6	4	6	14.06	14.35	8.71	12.58
	7168	18	18	18	19	22	20	13	20	7	7	4	7	14.39	14.17	9.11	12.95

outer iteration is used as an initial guess to BiCGStab so that the initial guess becomes progressively better as the outer iteration converges.

Table 6.5 shows the number of outer Anderson-accelerated fixed-point iterations and the maximum, minimum, and average number of inner BiCGStab iterations performed at each outer iteration as the mesh is refined and the polynomial order is increased. At each refinement and polynomial order, the IP, BR2, and CG methods outer iteration converged equivalently with MDLDG converging 1-4 iterations slower. All methods were scalable in h and p with CG requiring the fewest iterations followed by MDLDG and then IP and BR2.

6.7.4 Weak Scaling

Finally, we show that the IP and CG VEF system with $p = 2$ can be solved efficiently in parallel on larger problems. The parallel partitioning is such that there are ~ 9000 VEF scalar flux unknowns per processor. The results were generated on 32 nodes of the `rztopaz` machine at LLNL which has two 18-core Intel Xeon E5-2695 CPUs per node.

First, we investigate weak scaling on a mock problem where the VEF data are provided as inputs to the problem (as opposed to being solved for through fixed-point iteration). This allows the VEF system to be solved in isolation from the transport equation. We use the materials, geometry, and boundary conditions from the crooked pipe problem shown in Fig. 6.9 but set the Eddington tensor and boundary factor to

$$\mathbf{E} = \begin{cases} \begin{bmatrix} 9/11 & 0 \\ 0 & 1/11 \end{bmatrix} , & \mathbf{x} \in \text{pipe} \\ \begin{bmatrix} 1/3 & 0 \\ 0 & 1/3 \end{bmatrix} , & \mathbf{x} \in \text{wall} \end{cases} , \tag{6.67a}$$

Table 6.6: Weak scaling the IP VEF method with $p = 2$ on a non-physically difficult problem with mock VEF data. The preconditioner is parameterized by the method used for the approximate inverse of the continuous operator. The standard USC preconditioner with AMG on the continuous operator did not converge due to the large discontinuity in the VEF data. Convergence is recovered by applying AMG to a symmetrized version of the CG operator.

Processors	N_e	AMG	Direct	AMG-S	AMG-S3
36	36 288	–	27	33	26
72	70 000	–	24	33	28
144	145 152	–	28	31	27
288	285 628	–	26	31	29
576	580 608	–	26	32	26
864	867 328	–	27	33	29
1152	1 153 852	–	27	34	30

$$E_b = \begin{cases} 9/10\,, & \mathbf{x} \in \partial(\text{pipe}) \\ 1/2\,, & \mathbf{x} \in \partial(\text{wall}) \end{cases} . \tag{6.67b}$$

This corresponds to a linearly-anisotropic/diffusive angular flux in the wall and an extremely forward peaked solution

$$\psi = \mathbf{\Omega}_x^8 \tag{6.68}$$

in the pipe. The motivation for this choice is that the solvers are predicted to struggle when the Eddington tensor is discontinuous. We stress that this setup does not correspond to a physically realistic problem.

Table 6.6 shows the number of BiCGStab iterations to convergence for the USC-preconditioned IP VEF system on this mock problem. The columns of the table parameterize the solver used for the continuous stage of the USC preconditioner. The standard USC preconditioner used in the previous results did not converge on this mock problem. However, when a sparse direct solver is used instead of AMG, uniform convergence is recovered. This suggests that AMG is failing to adequately solve the continuous operator.

Note that AMG is effective on the standard continuous finite element discretization of diffusion. It is then expected that AMG will be effective in approximating the inverse of a symmetrized CG operator. By lagging the terms

$$-\int_{\Gamma_0} \left\{\!\!\left\{ \frac{\nabla u}{\sigma_t} \right\}\!\!\right\} \cdot [\![\mathbf{E}\varphi\mathbf{n}]\!] \,\mathrm{d}s + \int \nabla u \cdot \frac{1}{\sigma_t} (\nabla_h \cdot \mathbf{E})\varphi \,\mathrm{d}\mathbf{x} \tag{6.69}$$

in the CG VEF discretization (Eq. 6.48), a symmetric operator more amenable to accurate inversion via AMG is found. The symmetrized operator is then:

$$\int_{\Gamma_b} E_b \, u\varphi \, \mathrm{d}s + \int \nabla_h u \cdot \frac{1}{\sigma_t}\mathbf{E}\nabla_h \varphi \, \mathrm{d}\mathbf{x} + \int \sigma_a \, u\varphi \, \mathrm{d}\mathbf{x} \,, \tag{6.70}$$

with $u, v \in V_p$. This is a CG discretization of

$$-\nabla \cdot \frac{1}{\sigma_t}\mathbf{E}\nabla\varphi + \sigma_a \varphi \tag{6.71}$$

which corresponds to the VEF drift-diffusion equation where the advective term $(\nabla \cdot \mathbf{E})\varphi$ is lagged and moved to the right hand side.

The "AMG-S" column of Table 6.6 shows the convergence on the mock problem when one AMG V-cycle is applied to the symmetrized CG operator instead of the non-symmetric CG operator. The method converges and is roughly uniform in iteration counts as the mesh is refined. The "AMG-S3" column corresponds to the use of three iterations of an inner Richardson iteration to approximate the inverse of the non-symmetric CG operator. The Richardson iteration is preconditioned using one V-cycle of AMG on the symmetrized CG operator. In other words, this option approximates the inverse of the non-symmetric CG operator with three approximate inversions of the symmetrized CG operator. For this option, iterative efficiency generally fell between that of the sparse direct solver and using only AMG on the symmetrized CG operator. Inner iterations do reduce the number of total iterations to convergence but, since three V-cycles are performed per preconditioner application, not to the degree that fewer V-cycles are performed.

Next, we show weak scaling of the IP and CG VEF linear solves on the first outer iteration of the linearized crooked pipe problem from Section 6.7.3 with $p = 2$. One parallel block Jacobi transport sweep is performed to provide angular fluxes to compute the VEF data. The VEF system is then solved using BiCGStab. Table 6.7 shows the number of BiCGStab iterations to convergence to a tolerance of 10^{-8}. For the IP method, the USC preconditioner is used where the continuous operator is left in non-symmetric form and when it is symmetrized. The CG method uses AMG as a preconditioner. In addition, the number of iterations to solve the corresponding IP and CG diffusion problems (by setting $\mathbf{E} = \frac{1}{3}\mathbf{I}$ and $E_b = 1/2$) are shown. These results indicate that on a physically realistic problem the standard USC preconditioner and USC preconditioner with the symmetrized CG operator are both effective. IP VEF only required 5-7 more iterations than IP diffusion and CG VEF only required 2-7 more iterations than CG diffusion. Since no problems where the USC preconditioner with symmetrized CG operator failed to converge were found, the preconditioner with symmetrized CG operator may be more robust. Note that this discrepancy was not observed on physically realistic problems.

Table 6.7: Weak scaling the IP and CG VEF methods with $p = 2$ on the first iteration of the linearized crooked pipe problem. A parallel block Jacobi sweep was used to generate the VEF data needed to form the VEF system. On this physically realistic problem, both the standard USC and USC with symmetrized CG operator converged uniformly. The iterative efficiency is compared to solving the corresponding IP and CG radiation diffusion problems.

		IP			CG	
Processors	N_e	USC	USC-S	Diffusion	AMG	Diffusion
36	36 288	20	22	19	14	12
72	70 000	22	20	17	13	10
144	145 152	23	20	17	15	11
288	285 628	23	20	17	17	10
576	580 608	24	25	18	15	10
864	867 328	26	22	22	17	11
1152	1 153 852	26	25	19	16	10

Chapter 7

Mixed Finite Element VEF Discretizations

Mixed finite element methods are a class of discretization techniques for solving the mixed variational form of a partial differential equation. This variational form is characterized by the inclusion of multiple (typically two) physically disparate quantities resulting in a saddle point problem [67]. By contrast, primal formulations operate on a single quantity and produce minimization problems. Mixed methods were invented to 1) allow incorporation of a constraint (e.g. divergence free velocity in fluid flow), 2) provide direct access to an intermediate variable (e.g. the stress in elasticity), and 3) allow a weaker formulation than the corresponding primal formulation [80]. In the context of neutron diffusion, mixed methods are applied to the first-order, or P_1, form of radiation diffusion and 1) explicitly include the constraint of particle balance, 2) solve for the current in addition to the scalar flux, and 3) allow scalar flux solutions with no continuity requirements at interior element interfaces. Furthermore, through a process called hybridization [104, 105] the resulting system can be efficiently solved with AMG.

In this chapter, we investigate the use of mixed finite elements to solve the VEF equations in two spatial dimensions. We are interested in designing a discretization of the VEF equations that matches as closely as possible to that of Maginot and Brunner [106], the mixed finite element method used for radiation diffusion at LLNL in the BLAST hydrodynamics code [12]. Such a method would 1) have element-local particle balance, 2) solve for the current directly potentially leading to high accuracy coupling to the hydrodynamics' momentum equation, and 3) allow the scalar flux to be approximated in the same finite element space as the hydrodynamics' thermodynamic variables. In addition, a mixed finite element VEF discretization could serve as a drop-in replacement for radiation diffusion at LLNL providing a transport algorithm that allows reuse of the linear and nonlinear solvers already in place for diffusion. Note that mixed finite element discretizations of radiation diffusion have also been used in reactor analysis [107–109].

A lowest-order mixed finite element discretization of the VEF equations in one spatial dimension was developed in [40] for the linear transport problem. Lou *et al.* [41] and Lou and

Morel [42] used this algorithm to form efficient, VEF-based thermal radiative transfer and radiation-hydrodynamics algorithms, respectively. The 1D linear transport algorithm was extended to multiple dimensions and high-order in [50] but scalable preconditioned iterative solvers were not developed. The DG methods presented in Chapter 6 do not have element-local particle conservation and do not directly solve for the current. Furthermore, a mixed finite element VEF discretization has immediate compatibility with the mixed methods used in the hydrodynamics framework of [12].

The chapter begins by deriving the weak form of the VEF equations in first-order form. We show that, due to the presence of the Eddington tensor in the VEF first moment equation, the standard RT mixed finite element methods are not appropriate for the VEF equations. We present two alternatives: a method where each component of the current is approximated with continuous finite elements and a non-conforming approach where the RT space is used along with DG-like numerical fluxes. We provide background on the discrete inf-sup condition, a key mathematical aspect in the design of mixed finite element methods, and show how the inf-sup condition indicates both methods will be non-singular but the first method will suffer from the presence of non-physical spurious modes that plague solution quality and degrade the performance of iterative solvers. Block preconditioners for the mixed finite element system are presented. We then derive a hybridized version of the RT method that uses Lagrange multipliers to reduce the number of globally coupled unknowns. The chapter concludes with numerical results. We investigate the accuracy of the methods, their fixed-point convergence rates in the thick diffusion limit and on the linearized crooked pipe, and the performance of the preconditioned iterative solvers in both serial and parallel.

7.1 Weak Form

We seek approximations to the scalar flux and current in the finite-dimensional spaces $\mathcal{E}$ and $\mathcal{V}$, respectively, and test the zeroth and first moments with functions in the spaces $\mathcal{E}'$ and $\mathcal{V}'$, respectively. We consider Galerkin discretizations so that the test and trial spaces for the scalar flux and current are the same. In other words, we restrict ourselves to the case that $\mathcal{E}' = \mathcal{E}$ and $\mathcal{V}' = \mathcal{V}$. We proceed by first informally deriving the weak form assuming the spaces $\mathcal{E}$ and $\mathcal{V}$ have the requisite regularity to allow the resulting weak form to be well defined. We will see that there is no ambiguity in the choice $\mathcal{E} = Y_p \subset L^2(\mathcal{D})$. However, due to the presence of the Eddington tensor, the standard Raviart Thomas methods are inappropriate and so two choices for $\mathcal{V}$ are presented: a method with $\mathcal{V} = W_{p+1} \subset [H^1(\mathcal{D})]^2$ and a non-conforming method where $\mathcal{V} = RT_p \subset H(\text{div};\mathcal{D})$.

Multiplying the zeroth and first moments with sufficiently smooth functions u and $\boldsymbol{v}$, respectively, and integrating over the domain yields:

$$\int u\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int \sigma_a\,u\varphi\,\mathrm{d}\mathbf{x} = \int u\,Q_0\,\mathrm{d}\mathbf{x}\,, \tag{7.1a}$$

$$\int \boldsymbol{v}\cdot\nabla\cdot(\mathbf{E}\varphi)\,\mathrm{d}\mathbf{x} + \int \sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} = \int \boldsymbol{v}\cdot\boldsymbol{Q}_1\,\mathrm{d}\mathbf{x}\,. \tag{7.1b}$$

Note that the Eddington tensor is not globally differentiable due to the spatial interpolation used to approximate the angular flux. Thus, we integrate by parts to arrive at the weak form of the VEF equations:

$$\int u \, \nabla \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} + \int \sigma_a \, u\varphi \, \mathrm{d}\mathbf{x} = \int u \, Q_0 \, \mathrm{d}\mathbf{x} \,, \tag{7.2a}$$

$$\int_{\partial\mathcal{D}} \boldsymbol{v} \cdot \mathbf{E}\mathbf{n} \, \bar{\varphi} \, \mathrm{d}s - \int \nabla \boldsymbol{v} : \mathbf{E}\varphi \, \mathrm{d}\mathbf{x} + \int \sigma_t \, \boldsymbol{v} \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} = \int \boldsymbol{v} \cdot \boldsymbol{Q}_1 \, \mathrm{d}\mathbf{x} \,, \tag{7.2b}$$

where $\varphi = \bar{\varphi}$ on the boundary of the domain. We have used Green's identity for a tensor multiplied by a vector:

$$\int \nabla \cdot (\boldsymbol{v} \cdot \mathbf{P}) \, \mathrm{d}\mathbf{x} = \int \boldsymbol{v} \cdot \nabla \cdot \mathbf{P} \, \mathrm{d}\mathbf{x} + \int \nabla \boldsymbol{v} : \mathbf{P} \, \mathrm{d}\mathbf{x} = \oint \boldsymbol{v} \cdot \mathbf{P}\mathbf{n} \, \mathrm{d}s \,, \tag{7.3}$$

where

$$\mathbf{A} : \mathbf{B} = \sum_{i=1}^{2} \sum_{j=1}^{2} \mathbf{A}_{ij} \mathbf{B}_{ij} \,, \quad \mathbf{A}, \mathbf{B} \in \mathbb{R}^{2\times 2} \,. \tag{7.4}$$

Integrating by parts moves derivatives from the Eddington tensor and VEF scalar flux to the test function $\boldsymbol{v}$ allowing weaker requirements for $\mathbf{E}$ and φ. In addition, we assume $\boldsymbol{J} \in \mathcal{V}$ has enough regularity to allow $\nabla \cdot \boldsymbol{J} \in L^2(\mathcal{D})$ (i.e. $\mathcal{V} \subset H(\mathrm{div};\mathcal{D})$) so that $\int u \, \nabla \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x}$ is well defined. Thus, we can unambiguously take $u, \varphi \in \mathcal{E} \subset L^2(\mathcal{D})$. However, the test function $\boldsymbol{v}$ now has increased regularity requirements. Namely, we must have $\nabla \boldsymbol{v} : \mathbf{E} \in L^2(\mathcal{D})$ instead of the typical requirement that $\nabla \cdot \boldsymbol{v} = \nabla \boldsymbol{v} : \mathbf{I} \in L^2(\mathcal{D})$. In the thick diffusion limit, $\mathbf{E} = \frac{1}{3}\mathbf{I}$ and this requirement reduces to $\nabla \boldsymbol{v} : \mathbf{E} = \frac{1}{3} \nabla \cdot \boldsymbol{v} \in L^2(\mathcal{D})$. In this case, RT methods apply directly for both $\boldsymbol{v}$ and $\boldsymbol{J}$. However, for a general Eddington tensor, the RT space does not have the continuity requirements to allow the term $\int \nabla \boldsymbol{v} : \mathbf{E}\varphi \, \mathrm{d}\mathbf{x} < \infty$.

Proposition 7.1. *For a symmetric tensor* $\mathbf{S}$, *let* $\boldsymbol{v} : \mathcal{D} \to \mathbb{R}^2$ *be such that*

1. $\boldsymbol{v}|_K \in [H^1(K)]^2$ *for each* $K \in \mathcal{T}$
2. $[\![\boldsymbol{v} \cdot \mathbf{S}\mathbf{n}]\!] = 0$

then $\int \nabla \boldsymbol{v} : \mathbf{S} \, \mathrm{d}\mathbf{x} < \infty$. *Conversely, if* $\int \nabla \boldsymbol{v} : \mathbf{S} \, \mathrm{d}\mathbf{x} < \infty$ *and (a) is satisfied, then (b) holds.*

Proof. From (a), $\nabla_h \boldsymbol{v} \in [L^2(K)]^2$. Given a sufficiently smooth symmetric tensor $\mathbf{S}$ that vanishes on the boundary, the following holds:

$$\begin{aligned} \int \nabla_h \boldsymbol{v} : \mathbf{S} \, \mathrm{d}\mathbf{x} &= \sum_{K\in\mathcal{T}} \left[\int_{\partial K} \boldsymbol{v} \cdot \mathbf{S}\mathbf{n} \, \mathrm{d}s - \int_K \boldsymbol{v} \cdot \nabla \cdot \mathbf{S} \, \mathrm{d}\mathbf{x} \right] \\ &= \int_{\Gamma_0} [\![\boldsymbol{v} \cdot \mathbf{S}\mathbf{n}]\!] \, \mathrm{d}s - \int \boldsymbol{v} \cdot \nabla \cdot \mathbf{S} \, \mathrm{d}\mathbf{x} \\ &= \int \nabla \boldsymbol{v} : \mathbf{S} \, \mathrm{d}\mathbf{x} \,. \end{aligned} \tag{7.5}$$

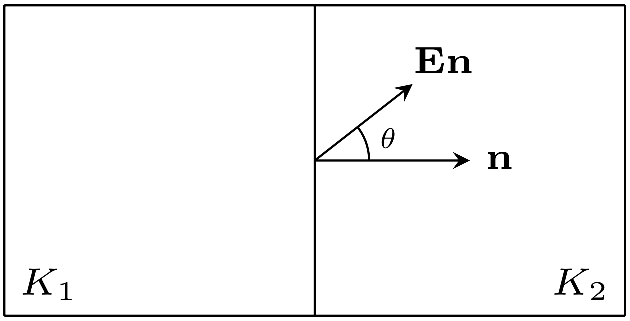


Figure 7.1: A depiction of the rotation and scaling of the normal vector induced by the Eddington tensor. Since the Eddington tensor is symmetric positive definite, the angle θ cannot be larger than ±90°. Due to the presence of the Eddington tensor in the VEF first moment equation, continuity of the $\mathbf{En}$ component is required.

Since the left hand side is bounded, $\int \nabla \boldsymbol{v} : \mathbf{S} \,\mathrm{d}\mathbf{x} < \infty$.

On the other hand, if $\int \nabla \boldsymbol{v} : \mathbf{S} \,\mathrm{d}\mathbf{x} < \infty$, then $\nabla \boldsymbol{v} : \mathbf{S} = \nabla_h \boldsymbol{v} : \mathbf{S}$ and, given $\boldsymbol{v}|_K \in [H^1(K)]^2$, we obtain

$$\int_{\Gamma_0} [\![\boldsymbol{v} \cdot \mathbf{Sn}]\!] \,\mathrm{d}s = 0 \,, \quad \forall \mathbf{S} \in [C_0^\infty(\mathcal{D})]^{2\times 2} \,, \tag{7.6}$$

hence, (b) holds. □

Proposition 7.1 generalizes Proposition 4.2 in that Proposition 7.1 reduces to Proposition 4.2 when $\mathbf{S} = \mathbf{I}$. Applying this result to the VEF equations, we have that

$$\int \nabla \boldsymbol{v} : \mathbf{E}\varphi \,\mathrm{d}\mathbf{x} < \infty \iff [\![\boldsymbol{v} \cdot \mathbf{En}]\!] = 0 \,, \quad \forall \mathcal{F} \in \Gamma_0 \,. \tag{7.7}$$

Figure 7.1 depicts an example of the Eddington tensor rotating and scaling the normal vector, altering the continuity requirement of the space. Note that since the Eddington tensor is symmetric positive definite, $\mathbf{n} \cdot \mathbf{En} > 0$ and thus $\theta \in (-\pi/2, \pi/2)$. In other words, the Eddington tensor cannot rotate the normal past a direction tangential to the face.

In light of Eq. 7.7, the weak form in Eq. 7.2 will hold only when the space $\mathcal{V}$ is chosen so that both $[\![\boldsymbol{J} \cdot \mathbf{n}]\!] = 0$ and $[\![\boldsymbol{v} \cdot \mathbf{En}]\!] = 0$ on all interior faces. These conditions can only be met by using $\boldsymbol{v}, \boldsymbol{J} \in \mathcal{V} \subset [H^1(\mathcal{D})]^2$ so that all components of $\boldsymbol{v}$ and $\boldsymbol{J}$ are continuous. A Petrov-Galerkin discretization where the test space satisfies $[\![\boldsymbol{v} \cdot \mathbf{En}]\!] = 0$ and the trial space satisfies $[\![\boldsymbol{J} \cdot \mathbf{n}]\!] = 0$ may be possible. In this case, the test space would need to use a more general Piola transform that preserves the $\mathbf{En}$ component of a vector, making the test space dependent on the angular flux. The Petrov-Galerkin discretization is not considered here due to this complication. Alternatively, non-conforming DG-like techniques can be used to allow use of the RT space for both the test and trial spaces. That is, both $\boldsymbol{v}, \boldsymbol{J} \in \mathcal{V} = RT_p \subset H(\mathrm{div}; \mathcal{D})$ and the discontinuity in $\boldsymbol{v} \cdot \mathbf{En}$ is handled with numerical fluxes.

7.1.1 $[H^1(\mathcal{D})]^2$

Setting $\boldsymbol{v}, \boldsymbol{J} \in \mathcal{V} \subset [H^1(\mathcal{D})]^2$ and $u, \varphi \in \mathcal{E} \subset L^2(\mathcal{D})$ allows the weak form in Eq. 7.2 to hold. The inf-sup [80, 110] condition states that the discretization arising from the pairing of equal degree interpolation for the scalar flux and current will be singular. That is, the $Y_p \times W_p$ discretization does not have a unique solution. The smallest non-singular pairing of spaces is then $Y_p \times W_{p+1}$. In other words, if the scalar flux is piecewise-constant, continuous linear finite elements for each component of the current must be used.

The discretization is complete by supplying boundary conditions. Solving the Miften-Larsen boundary conditions (Eq. 3.13c) for φ yields

$$\bar{\varphi} = \frac{1}{E_b}(\boldsymbol{J} \cdot \mathbf{n} - 2J_{\mathrm{in}}) \,. \tag{7.8}$$

The $[H^1(\mathcal{D})]^2 \times L^2(\mathcal{D})$ mixed finite element VEF discretization is then: find $(\varphi, \boldsymbol{J}) \in Y_p \times W_{p+1}$ such that

$$\int u \, \nabla \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} + \int \sigma_a \, u\varphi \, \mathrm{d}\mathbf{x} = \int u \, Q_0 \, \mathrm{d}\mathbf{x} \,, \quad \forall u \in Y_p \,, \tag{7.9a}$$

$$\begin{aligned} - \int \nabla \boldsymbol{v} : \mathbf{E}\varphi \, \mathrm{d}\mathbf{x} + \int \sigma_t \, \boldsymbol{v} \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} + \int_{\Gamma_b} \frac{1}{E_b}(\boldsymbol{v} \cdot \mathbf{E}\mathbf{n})(\boldsymbol{J} \cdot \mathbf{n}) \, \mathrm{d}s \\ = \int \boldsymbol{v} \cdot \boldsymbol{Q}_1 \, \mathrm{d}\mathbf{x} + 2 \int_{\Gamma_b} \frac{1}{E_b} \boldsymbol{v} \cdot \mathbf{E}\mathbf{n} \, J_{\mathrm{in}} \, \mathrm{d}s \,, \quad \forall \boldsymbol{v} \in W_{p+1} \,. \end{aligned} \tag{7.9b}$$

Equation 4.76 is used to compute the gradient and divergence of a W_p vector in reference space.

Using $\mathcal{V} \subset [H^1(\mathcal{D})]^2$ is simple to implement in that it relies only on the scalar continuous finite element space and does not require interior face bilinear forms. However, this choice has been seen to degrade both solution quality and solver performance due to allowing non-physical, spurious modes. These so-called checkerboard modes are a well-known issue with $[H^1(\mathcal{D})]^2 \times L^2(\mathcal{D})$ discretizations in the context of fluid flow [111] and are a consequence of the mismatch between the spaces $\nabla \cdot W_{p+1}$ and Y_p. The space $\mathcal{V} \subset [H^1(\mathcal{D})]^2$ is either too small with respect to Y_p, leading to a singular system in the case $\mathcal{V} = W_p$ or too large, allowing spurious modes for $\mathcal{V} = W_{p+1}$. The presence of these modes are analyzed in Section 7.3 and their effect on solution quality and solver performance is investigated in Section 7.5.5 in the context of radiation diffusion.

7.1.2 Raviart Thomas

If $\boldsymbol{v}, \boldsymbol{J} \in \mathcal{V} \subset H(\mathrm{div}; \mathcal{D})$, a non-conforming approach must be used for the first moment equation due to the presence of the Eddington tensor. On each element K, consider the weak first moment equation:

$$\int_{\partial K} \boldsymbol{v} \cdot \widehat{\mathbf{E}\varphi}\mathbf{n} \,\mathrm{d}s - \int_K \nabla \boldsymbol{v} : \mathbf{E}\varphi \,\mathrm{d}\mathbf{x} + \int_K \sigma_t \,\boldsymbol{v} \cdot \boldsymbol{J} \,\mathrm{d}\mathbf{x} = \int_K \boldsymbol{v} \cdot \boldsymbol{Q}_1 \,\mathrm{d}\mathbf{x}\,, \quad \forall \boldsymbol{v} \in \mathbb{D}_p(K)\,, \tag{7.10}$$

where $\widehat{\mathbf{E}\varphi}$ is an approximation to $\mathbf{E}\varphi$ provided on the boundary of the element known as the numerical flux. Summing over all elements $K \in \mathcal{T}$:

$$\int_\Gamma [\![\boldsymbol{v}]\!] \cdot \widehat{\mathbf{E}\varphi}\mathbf{n} \,\mathrm{d}s - \int \nabla_h \boldsymbol{v} : \mathbf{E}\varphi \,\mathrm{d}\mathbf{x} + \int \sigma_t \,\boldsymbol{v} \cdot \boldsymbol{J} \,\mathrm{d}\mathbf{x} = \int \boldsymbol{v} \cdot \boldsymbol{Q}_1 \,\mathrm{d}\mathbf{x}\,, \quad \forall \boldsymbol{v} \in RT_p\,. \tag{7.11}$$

We have used the fact that on a face $\mathcal{F} = K_1 \cap K_2$, $\mathbf{n} = \mathbf{n}_1 = -\mathbf{n}_2$ and the definitions of the jump and broken gradient in Eqs. 4.30 and 4.33, respectively. To facilitate the connection to radiation diffusion in the thick diffusion limit, we set

$$\widehat{\mathbf{E}\varphi}\mathbf{n} = \{\!\{\mathbf{E}\mathbf{n}\}\!\}\{\!\{\varphi\}\!\}\ , \quad \text{on } \mathcal{F} \in \Gamma_0 \tag{7.12}$$

where the average is defined in Eq. 4.30. The Miften-Larsen boundary conditions are applied with

$$\widehat{\mathbf{E}\varphi}\mathbf{n} = \frac{\mathbf{E}\mathbf{n}}{E_b}(\boldsymbol{J} \cdot \mathbf{n} - 2J_{\mathrm{in}})\ , \quad \text{on } \mathcal{F} \in \Gamma_b\,. \tag{7.13}$$

This is derived by solving Eq. 3.13c for the scalar flux and multiplying by $\mathbf{E}\mathbf{n}$. The $Y_p \times RT_p$ discretization is then: find $(\varphi, \boldsymbol{J}) \in Y_p \times RT_p$ such that

$$\int u \,\nabla \cdot \boldsymbol{J} \,\mathrm{d}\mathbf{x} + \int \sigma_a \,u\varphi \,\mathrm{d}\mathbf{x} = \int u \,Q_0 \,\mathrm{d}\mathbf{x}\,, \quad \forall u \in Y_p\,, \tag{7.14a}$$

$$\begin{aligned}\int_{\Gamma_0} [\![\boldsymbol{v} \cdot \{\!\{\mathbf{E}\mathbf{n}\}\!\}]\!] \,\{\!\{\varphi\}\!\} \,\mathrm{d}s - \int \nabla_h \boldsymbol{v} : \mathbf{E}\varphi \,\mathrm{d}\mathbf{x} + \int \sigma_t \,\boldsymbol{v} \cdot \boldsymbol{J} \,\mathrm{d}\mathbf{x} + \int_{\Gamma_b} \frac{1}{E_b}(\boldsymbol{v} \cdot \mathbf{E}\mathbf{n})(\boldsymbol{J} \cdot \mathbf{n}) \,\mathrm{d}s \\ = \int \boldsymbol{v} \cdot \boldsymbol{Q}_1 \,\mathrm{d}\mathbf{x} + 2\int_{\Gamma_b} \frac{1}{E_b} \boldsymbol{v} \cdot \mathbf{E}\mathbf{n} \,J_{\mathrm{in}} \,\mathrm{d}s\,, \quad \forall \boldsymbol{v} \in RT_p\,. \end{aligned} \tag{7.14b}$$

Since RT vectors use the contravariant Piola transform, we substitute $\boldsymbol{v} = \frac{1}{J}\mathbf{F}\hat{\boldsymbol{v}}$ in all terms involving $\boldsymbol{v}$ and use Eqs. 4.83 and 4.85 to evaluate $\nabla_h \boldsymbol{v}$ and $\nabla \cdot \boldsymbol{J}$, respectively.

In the thick diffusion limit, $\mathbf{E} = \frac{1}{3}\mathbf{I}$ and

$$[\![\boldsymbol{v} \cdot \{\!\{\mathbf{E}\mathbf{n}\}\!\}]\!] = \frac{1}{3}\,[\![\boldsymbol{v} \cdot \mathbf{n}]\!] = 0\,, \tag{7.15}$$

since $\boldsymbol{v} \in RT_p$ has a continuous normal component. This can be seen with the aid of Fig. 7.2. The vector $\{\!\{\mathbf{E}\mathbf{n}\}\!\}$ has a projection $\alpha = \{\!\{\mathbf{E}\mathbf{n}\}\!\} \cdot \mathbf{n}$ onto the normal component and a projection $\beta = \{\!\{\mathbf{E}\mathbf{n}\}\!\} \cdot \mathbf{t}$ onto the tangential component such that $\{\!\{\mathbf{E}\mathbf{n}\}\!\} = \alpha\mathbf{n} + \beta\mathbf{t}$. In the thick diffusion limit, $\mathbf{E} = 1/3\mathbf{I}$ meaning $\beta = 0$. This leaves only $[\![\alpha\boldsymbol{v} \cdot \mathbf{n}]\!]$ which is zero due to $\boldsymbol{v} \in RT_p$. Furthermore, $\nabla_h : \mathbf{E} = \frac{1}{3}\nabla \cdot \boldsymbol{v}$ and thus this discretization with this choice

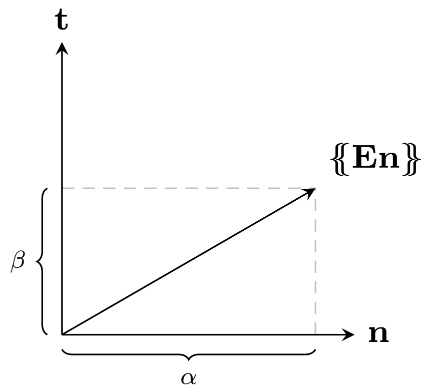


Figure 7.2: A depiction of the angle between the normal vector and the average of the Eddington tensor applied to the normal vector. In the thick diffusion limit, the Eddington tensor does not rotate the normal making $\beta = 0$.

of numerical flux is equivalent to the standard RT discretization of diffusion in the thick diffusion limit.

The RT space satisfies $\nabla \cdot RT_p = Y_p$ avoiding the spurious modes seen for the $[H^1(\mathcal{D})]^2 \times L^2(\mathcal{D})$ discretization. This allows superior solution quality and excellent solver performance. However, the RT method is more complex due to the need for interior face bilinear forms, the contravariant Piola transform, and the comparatively less simple RT space.

7.2 Block Solvers

The above discretizations admit the following block system

$$\begin{bmatrix} \mathbf{A} & \mathbf{G} \\ \mathbf{D} & \mathbf{M}_a \end{bmatrix} \begin{bmatrix} \underline{J} \\ \underline{\varphi} \end{bmatrix} = \begin{bmatrix} \underline{g} \\ \underline{f} \end{bmatrix} , \tag{7.16}$$

where for $u, \varphi \in \mathcal{E}$ and $\boldsymbol{v}, \boldsymbol{J} \in \mathcal{V}$:

$$\underline{v}^T \mathbf{A} \underline{J} = \int \sigma_t \, \boldsymbol{v} \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} + \int_{\Gamma_b} \frac{1}{E_b} (\boldsymbol{v} \cdot \mathbf{En})(\boldsymbol{J} \cdot \mathbf{n}) \, \mathrm{d}s \, , \tag{7.17a}$$

$$\underline{u}^T \mathbf{M}_a \underline{\varphi} = \int \sigma_a, u\varphi \, \mathrm{d}\mathbf{x} \, , \tag{7.17b}$$

$$\underline{u}^T \mathbf{D} \underline{J} = \int u \, \nabla \cdot \boldsymbol{J} \, \mathrm{d}\mathbf{x} \, , \tag{7.17c}$$

$$\underline{v}^T \mathbf{G} \underline{\varphi} = \begin{cases} -\int \nabla \boldsymbol{v} : \mathbf{E}\varphi \, \mathrm{d}\mathbf{x} \, , & \mathcal{V} = W_{p+1} \\ \int_{\Gamma_0} [\![\boldsymbol{v} \cdot \{\!\{ \mathbf{En} \}\!\}]\!] \{\!\{ \varphi \}\!\} \, \mathrm{d}s - \int \nabla_h \boldsymbol{v} : \mathbf{E}\varphi \, \mathrm{d}\mathbf{x} \, , & \mathcal{V} = RT_p \end{cases} , \tag{7.17d}$$

$$\underset{\sim}{v}^T \underset{\sim}{g} = \int \boldsymbol{v} \cdot \boldsymbol{Q}_1 \,\mathrm{d}\mathbf{x} + 2 \int_{\Gamma_b} \frac{1}{E_b} \boldsymbol{v} \cdot \mathbf{En}\, J_{\mathrm{in}} \,\mathrm{d}s \,, \tag{7.17e}$$

$$\underset{\sim}{u}^T \underset{\sim}{f} = \int u\, Q_0 \,\mathrm{d}\mathbf{x} \,. \tag{7.17f}$$

Note that the integration transformations described in Section 4.4 are implicitly used and in particular the contravariant Piola transform is implicitly used when $\mathcal{V} = RT_p$.

We use a lower block triangular preconditioner of the form

$$\mathbf{M} = \begin{bmatrix} \mathbf{A} & \\ \mathbf{D} & \tilde{\mathbf{S}} \end{bmatrix} , \tag{7.18}$$

where $\tilde{\mathbf{S}}$ is an approximation to the Schur complement $\mathbf{S} = \mathbf{M}_a - \mathbf{D}\mathbf{A}^{-1}\mathbf{G}$. Block preconditioners seek to modify the system such that it has a minimal polynomial with small degree [112]. Iterative solvers with an optimality condition, such as GMRES, can then converge in a small number of iterations. However, computing the generally dense Schur complement and exactly inverting it are impractical. Instead, we use an approximate Schur complement formed from a sparse approximation to $\mathbf{A}^{-1}$ and sparse matrix multiplication. That is, we use

$$\tilde{\mathbf{S}} = \mathbf{M}_a - \mathbf{D}\tilde{\mathbf{A}}^{-1}\mathbf{G} \tag{7.19}$$

where $\tilde{\mathbf{A}}$ is the lumped mass matrix and boundary term. On elements with no boundary faces (i.e. $\partial K \cap \Gamma_b = \emptyset$), the lumping procedure is to sum the rows of the matrix into the diagonal. This is computed on the element-local matrix as:

$$\tilde{\mathbf{A}}^e_{ij} = \begin{cases} \sum_k \mathbf{A}^e_{ik} \,, & i = j \\ 0 \,, & i \neq j \end{cases} , \tag{7.20}$$

where $\mathbf{A}^e$ and $\tilde{\mathbf{A}}^e$ are the matrices associated with the degrees of freedom corresponding to element K_e. On elements with a boundary face, the boundary integral over Γ_b contributes. Due to the Eddington tensor, $\boldsymbol{v} \cdot \mathbf{En}$ couples degrees of freedom corresponding to the normal and tangential components of $\boldsymbol{v}$. We leverage the block structure of the local matrices to lump the boundary elements. Let

$$\mathbf{A}^e = \begin{bmatrix} \mathbf{A}^e_{11} & \mathbf{A}^e_{12} \\ \mathbf{A}^e_{21} & \mathbf{A}^e_{22} \end{bmatrix} \tag{7.21}$$

where $\mathbf{A}^e_{ij}$ is the sub-block corresponding to the degrees of freedom of the i^{th} and j^{th} components of the test and trial functions, respectively. We then lump each of these sub-blocks separately so that:

$$\tilde{\mathbf{A}}^e = \begin{bmatrix} \tilde{\mathbf{A}}^e_{11} & \tilde{\mathbf{A}}^e_{12} \\ \tilde{\mathbf{A}}^e_{21} & \tilde{\mathbf{A}}^e_{22} \end{bmatrix} . \tag{7.22}$$

The lumped local matrix $\tilde{\mathbf{A}}^e$ is diagonal by vector component. That is, each row has at most two entries corresponding to the two components of a vector in $\mathbb{R}^2$.

For both interior and boundary elements, the local matrices $\tilde{\mathbf{A}}^e$ are assembled into the global matrix $\tilde{\mathbf{A}}$. For rows corresponding to interior degrees of freedom, the lumped matrix is diagonal and thus the inverse is simply $1/\tilde{A}_{ii}$. For rows corresponding to boundary degrees of freedom, $\tilde{\mathbf{A}}$ is a diagonal matrix for each vector component. The inverse is computed by gathering the entries corresponding to each vector component into a 2×2 matrix, inverting it, and scattering the inverse back to a sparse matrix representing $\tilde{\mathbf{A}}^{-1}$. Finally, the lumped Schur complement is formed with sparse matrix multiplication. Note that computing the Schur complement is numerically analogous to eliminating the current in the analytic equations to form a second-order, elliptic partial differential equation. Thus, AMG is expected to be a spectrally equivalent approximation to $\tilde{\mathbf{S}}^{-1}$.

The approximate inverse of the block preconditioner in Eq. 7.18 is applied with forward substitution. In other words, we solve

$$\begin{bmatrix} \mathbf{A} & \\ \mathbf{D} & \tilde{\mathbf{S}} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \end{bmatrix} \tag{7.23}$$

by approximately solving the block problems:

$$\mathbf{A} x_1 = r_1 \,, \tag{7.24a}$$

$$\tilde{\mathbf{S}} x_2 = r_2 - \mathbf{D} x_1 \,. \tag{7.24b}$$

The approximate inverse of $\mathbf{A}$ and $\tilde{\mathbf{S}}$ are applied with one iteration of Jacobi smoothing and AMG, respectively.

7.3 Discrete Inf-Sup Condition

Here, we discuss the the inf-sup condition that governs the solvability of the 2×2 block systems arising in mixed finite element discretizations. Two excellent references for this topic are Brezzi [110] and Benzi *et al.* [112]. We present an analysis for Poisson's equation. Methods not effective for this simpler problem have no hope of being effective for the VEF equations.

7.3.1 Conditions for Solvability

Consider the linear system:

$$\begin{bmatrix} \mathbf{M} & -\mathbf{D}^T \\ \mathbf{D} & \end{bmatrix} \begin{bmatrix} \underline{q} \\ \underline{u} \end{bmatrix} = \begin{bmatrix} 0 \\ f \end{bmatrix} , \tag{7.25}$$

which corresponds to the mixed finite element discretization of

$$\boldsymbol{q} + \nabla u = 0 \,, \tag{7.26a}$$

$$\nabla \cdot \boldsymbol{q} = f \,. \tag{7.26b}$$

Note that the above is Poisson's equation, $-\nabla^2 u = f$, in mixed form. The matrices are of the form:

$$\underset{\sim}{v}^T \mathbf{M} \underset{\sim}{q} = \int \boldsymbol{v} \cdot \boldsymbol{q} \, \mathrm{d}\mathbf{x} \, , \quad \underset{\sim}{w}^T \mathbf{D} \underset{\sim}{q} = \int w \, \nabla \cdot \boldsymbol{q} \, \mathrm{d}\mathbf{x} \, . \tag{7.27}$$

We wish to demonstrate the conditions for when the block system in Eq. 7.25 is non-singular. To show the solution is unique, it must be verified that $f = 0$ implies that $u = 0$ and $\boldsymbol{q} = 0$. When $f = 0$, we have that

$$\mathbf{D} \underset{\sim}{q} = 0 \iff \underset{\sim}{q} \in N(\mathbf{D}) \, , \tag{7.28}$$

where $N(\mathbf{D})$ denotes the nullspace of $\mathbf{D}$ such that

$$N(\mathbf{D}) = \{ \underset{\sim}{v} : \mathbf{D} \underset{\sim}{v} = 0 \} \, . \tag{7.29}$$

For some $\underset{\sim}{v} \in N(\mathbf{D})$, the first equation reads

$$\underset{\sim}{v}^T \mathbf{M} \underset{\sim}{q} - \underset{\sim}{v}^T \mathbf{D}^T \underset{\sim}{u} = 0 \, . \tag{7.30}$$

Since $\underset{\sim}{v} \in N(\mathbf{D})$, $\underset{\sim}{v}^T \mathbf{D}^T = 0$. Thus, we have that

$$\underset{\sim}{v}^T \mathbf{M} \underset{\sim}{q} = 0 \, . \tag{7.31}$$

Since $\mathbf{M}$ is a mass matrix, it is symmetric positive definite and thus, $\underset{\sim}{v}^T \mathbf{M} \underset{\sim}{q} = 0 \iff \underset{\sim}{q} = 0$. In other words, we have shown that $f = 0 \Rightarrow \underset{\sim}{q} = 0$. Setting $\underset{\sim}{q} = 0$ in the first row of Eq. 7.25 yields:

$$\mathbf{D}^T \underset{\sim}{u} = 0 \, . \tag{7.32}$$

For the block system to be non-singular, we must have that

$$\mathbf{D}^T \underset{\sim}{u} = 0 \iff \underset{\sim}{u} = 0 \tag{7.33}$$

Equivalently, we require that the nullspace of $\mathbf{D}^T$ has only the trivial nullspace of zero (i.e. $N(\mathbf{D}^T) = \{0\}$). The discrete inf-sup condition is precisely this condition on the matrix $\mathbf{D}$ that $N(\mathbf{D}^T) = \{0\}$.

7.3.2 Presence of Spurious Modes

We now particularize the matrix $\mathbf{D}$ for a single element, $K = [0,1]^2$, of the lowest-order $[H^1(\mathcal{D})]^2 \times L^2(\mathcal{D})$ and $H(\mathrm{div}; \mathcal{D}) \times L^2(\mathcal{D})$ discretizations of the Poisson problem. Let

$$RT_0 = \mathrm{span}\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} , \begin{pmatrix} x \\ 0 \end{pmatrix} , \begin{pmatrix} 0 \\ 1 \end{pmatrix} , \begin{pmatrix} 0 \\ y \end{pmatrix} \} \, , \tag{7.34a}$$

$$W_1 = \mathrm{span}\{ \begin{pmatrix} 1 \\ 0 \end{pmatrix} , \begin{pmatrix} x \\ 0 \end{pmatrix} , \begin{pmatrix} y \\ 0 \end{pmatrix} , \begin{pmatrix} xy \\ 0 \end{pmatrix} , \begin{pmatrix} 0 \\ 1 \end{pmatrix} , \begin{pmatrix} 0 \\ x \end{pmatrix} , \begin{pmatrix} 0 \\ y \end{pmatrix} , \begin{pmatrix} 0 \\ xy \end{pmatrix} \} \, , \tag{7.34b}$$

be the lowest-order polynomial spaces for a single element of the Raviart Thomas and $[H^1(\mathcal{D})]^2$ finite element spaces, respectively. Observe that the divergence of RT_0 is exactly the constant polynomial space, $\mathcal{Q}_0(K)$:

$$\nabla \cdot RT_0 = \text{span}\{0, 1, 0, 1\} = \text{span}\{1\} = \mathcal{Q}_0(K)\,, \tag{7.35}$$

while the divergence of the $[H^1(\mathcal{D})]^2$ space is:

$$\nabla \cdot W_1 = \text{span}\{0, 1, 0, y, 0, 0, 1, x\} = \text{span}\{1, x, y\}\,, \tag{7.36}$$

which is a space larger than $\mathcal{Q}_0(K)$. The nullspaces of the divergence of the RT and $[H^1(\mathcal{D})]^2$ local polynomial spaces are spanned by

$$N(\nabla \cdot RT_0) = \text{span}\{\begin{pmatrix} -x \\ y \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}\}\,, \tag{7.37a}$$

$$N(\nabla \cdot W_1) = \text{span}\{\begin{pmatrix} -x \\ y \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ x \end{pmatrix}, \begin{pmatrix} y \\ 0 \end{pmatrix}\}\,. \tag{7.37b}$$

Here, we can already see an issue forming: the nullspace for W_1 is larger than the nullspace for RT_0.

We are interested in the bilinear form

$$\int_K u \nabla \cdot \boldsymbol{v} \, \mathrm{d}\mathbf{x}\,, \quad u \in \mathcal{Q}_0(K)\,,\ \boldsymbol{v} \in X\,, \tag{7.38}$$

where X is either W_1 or RT_0. The above bilinear form has the same properties as the $L^2(K)$ inner product of $u \in \mathcal{Q}_0(K)$ and $w \in \nabla \cdot X$:

$$\int_K uw \, \mathrm{d}\mathbf{x}\,, \quad u \in \mathcal{Q}_0(K)\,,\ w \in \nabla \cdot X\,. \tag{7.39}$$

For $w \in \nabla \cdot RT_0 = \mathcal{Q}_0(K)$, $\int_K uw \, \mathrm{d}\mathbf{x} = 0$ only when $u = 0$ or $w = 0$. This means that for RT_0, the matrix $\mathbf{D}$ has the same nullspace as $N(\nabla \cdot RT_0)$ and $\mathbf{D}^T$ has only the trivial nullspace corresponding to $u = 0$.

On the other hand, for $X = W_1$, there exists $w \in \nabla \cdot W_1 = \text{span}\{1, x, y\}$ such that $w \neq 0$ but $\int_K uw \, \mathrm{d}\mathbf{x} = 0$. This nullspace consists of all non-zero $w \in \text{span}\{x - 1/2, y - 1/2\}$. Thus, when W_1 is used $\mathbf{D}$ will have a nullspace that is larger than the nullspace for $\nabla \cdot W_1$. Note that the nullspace for $\mathbf{D}^T$ will still be the trivial case corresponding to $u = 0$ only. This allows the $[H^1(\mathcal{D})]^2 \times L^2(\mathcal{D})$ discretization to be non-singular but since $N(\mathbf{D}) \supset N(\nabla \cdot W_1)$, $\mathbf{D}$ will allow non-physical spurious modes. The takeaway is that for the pairing W_1 and $\mathcal{Q}_0(K)$, W_1 is rich enough to ensure non-singularity but is *too* rich for $Q_0(K)$ such that spurious modes are allowed. The RT space avoids these spurious modes through its careful design such that $\nabla \cdot RT_p = Y_p$.

7.4 Hybridization

A hybridized version of the RT mixed method is obtained by relaxing the continuity requirements of the space RT_p and reimposing them weakly. Removing the continuity requirement from RT_p yields the broken space

$$\hat{RT}_p = \{\boldsymbol{v} \in [L^2(\mathcal{D})]^2 : \boldsymbol{v}|_{K_e} \in \mathbb{D}_k(K_e)\,, \quad \forall K_e \in \mathcal{T}\}\,. \tag{7.40}$$

This space is equivalent to RT_p on each element but $\hat{RT}_p$ does not have the matching conditions that strongly enforce continuity in the normal component. Note that $RT_p \subset \hat{RT}_p$ and that $\boldsymbol{v} \in \hat{RT}_p$ belongs to RT_p if and only if $[\![\boldsymbol{v} \cdot \mathbf{n}]\!] = 0$ on all interior mesh interfaces. In other words, the mixed problem can be reformulated to use the space $\hat{RT}_p$ instead of RT_p by adding the constraint that $[\![\boldsymbol{J} \cdot \mathbf{n}]\!] = 0$ for each $\mathcal{F} \in \Gamma_0$. The methods presented in this section enforce this constraint with a Lagrange multiplier.

Hybridized methods are attractive for three reasons. First, since $\boldsymbol{J} \in \hat{RT}_p$ and $\varphi \in Y_p$ are both discontinuous, their degrees of freedom are coupled only locally on each element. It is then possible to locally eliminate the scalar flux and current arriving at a system of equations for just the Lagrange multiplier. This reduced system is much smaller than the original 2×2 system. Second, the reduced system for the Lagrange multiplier will be positive definite and AMG can be applied directly, avoiding the need for block preconditioners. Finally, the Lagrange multiplier provides an additional approximation for the scalar flux not provided by the original mixed problem.

Since the VEF equations are not symmetric, the variational principles typically used to derive hybridized mixed finite element methods are not appropriate. We first show the derivation of a hybridized method for the symmetric case of radiation diffusion using variational principles. This method is extended to the VEF equations by emulating the properties of the symmetric case. Finally, we discuss the details of an efficient implementation.

7.4.1 Derivation for Radiation Diffusion

The radiation diffusion equation with zero boundary conditions is

$$\nabla \cdot \boldsymbol{J} + \sigma_a \varphi = Q_0\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{7.41a}$$

$$\nabla \varphi + 3\sigma_t \boldsymbol{J} = 0\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{7.41b}$$

$$\varphi = 0\,, \quad \mathbf{x} \in \partial\mathcal{D}\,, \tag{7.41c}$$

where the source has been assumed to be isotropic. The $Y_p \times RT_p$ mixed finite element discretization is then: find $(\varphi, \boldsymbol{J}) \in Y_p \times RT_p$ such that

$$\int 3\sigma_t\, \boldsymbol{v} \cdot \boldsymbol{J} \,\mathrm{d}\mathbf{x} - \int \nabla \cdot \boldsymbol{v}\, \varphi \,\mathrm{d}\mathbf{x} = 0\,, \quad \forall \boldsymbol{v} \in RT_p\,, \tag{7.42a}$$

$$\int u\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} = \int u\,Q_0\,\mathrm{d}\mathbf{x}\,, \quad \forall u \in Y_p\,. \tag{7.42b}$$

This discretization arises from a mixed variational principle. Consider the saddle point problem:

$$\inf_{\boldsymbol{J}\in RT_p}\sup_{\varphi\in Y_p}\mathcal{L}(\boldsymbol{J},\varphi)\,, \tag{7.43}$$

where

$$\mathcal{L}(\boldsymbol{J},\varphi) = \int 3\sigma_t\,\boldsymbol{J}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} - \left(\int \varphi\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \frac{1}{2}\int\sigma_a\,\varphi^2\,\mathrm{d}\mathbf{x} - \int\varphi\,Q_0\;\mathrm{d}\mathbf{x}\right)\,. \tag{7.44}$$

This saddle point problem minimizes the so-called minimum complementary energy principle, defined as $\int 3\sigma_t\,\boldsymbol{J}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x}$, under the constraint of particle balance. The solution is found by setting $\nabla\mathcal{L} = 0$:

$$\frac{\partial\mathcal{L}}{\partial\boldsymbol{J}} = \int 3\sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} - \int\nabla\cdot\boldsymbol{v}\,\varphi\,\mathrm{d}\mathbf{x} = 0\,, \quad \forall\boldsymbol{v}\in RT_p\,, \tag{7.45a}$$

$$\frac{\partial\mathcal{L}}{\partial\varphi} = \int u\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int\sigma_a\,u\varphi\,\mathrm{d}\mathbf{x} - \int u\,Q_0\,\mathrm{d}\mathbf{x} = 0\,, \quad \forall u\in Y_p\,. \tag{7.45b}$$

In other words, the solution of the mixed discretization in Eq. 7.42 is also the saddle point of $\mathcal{L}$.

The hybridized form is found by replacing RT_p with $\hat{RT}_p$ and adding the constraint that the normal component of the current is continuous. The resulting constrained saddle point problem is:

$$\inf_{\boldsymbol{J}\in\hat{RT}_p}\sup_{\varphi\in Y_p}\hat{\mathcal{L}}(\boldsymbol{J},\varphi) \quad \text{such that} \quad [\![\boldsymbol{J}\cdot\mathbf{n}]\!] = 0 \quad \forall\mathcal{F}\in\Gamma_0\,, \tag{7.46}$$

where $\hat{\mathcal{L}}$ is the broken form of $\mathcal{L}$ that applies $\mathcal{L}$ on each element independently:

$$\hat{\mathcal{L}}(\boldsymbol{J},\varphi) = \int 3\sigma_t\,\boldsymbol{J}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} - \left(\int\varphi\,\nabla_h\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \frac{1}{2}\int\sigma_a\,\varphi^2\,\mathrm{d}\mathbf{x} - \int\varphi\,Q_0\;\mathrm{d}\mathbf{x}\right)\,. \tag{7.47}$$

Since the $\boldsymbol{J}\in\hat{RT}_p$ and $\varphi\in Y_p$ are discontinuous, the above holds element-wise due to the use of the broken divergence. Introducing the Lagrange multiplier λ, the constrained saddle point problem is equivalent to

$$\inf_{\boldsymbol{J}\in\hat{RT}_p}\sup_{\varphi\in Y_p}\sup_{\lambda\in\Lambda_p}\mathcal{H}(\boldsymbol{J},\varphi,\lambda)\,, \tag{7.48}$$

where

$$\mathcal{H}(\boldsymbol{J},\varphi,\lambda) = \hat{\mathcal{L}}(\boldsymbol{J},\varphi) + \int_{\Gamma_0}\lambda\,[\![\boldsymbol{J}\cdot\mathbf{n}]\!]\,\mathrm{d}s\,. \tag{7.49}$$

As before, the solution is found by setting $\nabla\mathcal{H} = 0$:

$$\frac{\partial\mathcal{H}}{\partial\boldsymbol{J}} = \int 3\sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} - \int\nabla_h\cdot\boldsymbol{v}\,\varphi\,\mathrm{d}\mathbf{x} + \int_{\Gamma_0}[\![\boldsymbol{v}\cdot\mathbf{n}]\!]\,\lambda\,\mathrm{d}s = 0\,, \quad \forall\boldsymbol{v}\in\hat{RT}_p\,, \tag{7.50a}$$

$$\frac{\partial \mathcal{H}}{\partial \varphi} = \int u\,\nabla_h \cdot \boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} - \int u\,Q_0\,\mathrm{d}\mathbf{x} = 0\,, \quad \forall u \in Y_p\,, \tag{7.50b}$$

$$\frac{\partial \mathcal{H}}{\partial \lambda} = \int_{\Gamma_0} \mu\,[\![\boldsymbol{J}\cdot\mathbf{n}]\!]\,\mathrm{d}s = 0\,, \quad \forall \mu \in \Lambda_p\,. \tag{7.50c}$$

Since $\hat{RT}_p$ and Y_p are discontinuous spaces, the hybridized mixed method is equivalent to:

$$\int_K 3\sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} - \int_K \nabla\cdot\boldsymbol{v}\,\varphi\,\mathrm{d}\mathbf{x} + \int_{\partial K\cap\Gamma_0} \boldsymbol{v}\cdot\mathbf{n}_K\lambda\,\mathrm{d}s = 0\,, \quad \forall \boldsymbol{v}\in\mathbb{D}_p(K)\,,\ K\in\mathcal{T}\,, \tag{7.51a}$$

$$\int_K u\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int_K \sigma_a\,u\varphi\,\mathrm{d}\mathbf{x} = \int_K u\,Q_0\,\mathrm{d}\mathbf{x}\,, \quad \forall u\in\mathbb{Q}_p(K)\,,\ K\in\mathcal{T}\,, \tag{7.51b}$$

$$\int_{\Gamma_0} \mu\,[\![\boldsymbol{J}\cdot\mathbf{n}]\!]\,\mathrm{d}s = 0\,, \quad \forall \mu\in\Lambda_p\,. \tag{7.51c}$$

Here it can be seen that the degrees of freedom for the scalar flux and current are no longer globally coupled. In fact, if λ were known, the scalar flux and current could be recovered by solving element-local radiation diffusion problems where λ plays the role of a weak boundary condition applied on each element. Note that the non-zero boundary condition $\varphi = \bar{\varphi}$ for $\mathbf{x}\in\Gamma_b$ can be applied by subtracting $\int_{\Gamma_b} \boldsymbol{v}\cdot\mathbf{n}\,\bar{\varphi}\,\mathrm{d}s$ from the right hand side of Eq. 7.50a.

In hybridization, continuity of the normal component is enforced weakly (e.g. see Eq. 7.51c). However, it is well known that the resulting discrete solution will actually satisfy continuity of the normal component in a strong sense. In fact, hybridization has also been viewed as an algebraic technique similar to static condensation in [105].

7.4.2 Extension to VEF

The above variational process cannot be applied directly to the VEF equations due to their lack of symmetry. Without symmetry, it is unclear which potential the weak VEF equations correspond to. However, we can define a hybrid method for the VEF equations by mimicking the properties seen above for the symmetric case. In particular, we use the broken RT space, $\hat{RT}_p$, and a Lagrange multiplier that 1) weakly enforces continuity of the normal component of the current and 2) provides inter-element boundary conditions for element-local VEF problems. As in the symmetric case, this will allow elimination of the scalar flux and current, leading to a smaller system for just the Lagrange multiplier where AMG can be applied directly. However, since the resulting method cannot be derived from a variational principle it is unclear whether the resulting hybrid formulation will be equivalent to the original mixed formulation.

The hybridized diffusion method can be extended to the VEF equations with Miften-Larsen boundary conditions by replacing the diffusion first moment with the VEF first moment equation and using the boundary condition $\bar{\varphi} = \frac{1}{E_b}(\boldsymbol{J}\cdot\mathbf{n} - 2J_{\mathrm{in}})$. This can be accomplished by using the element-local weak form of the first moment equation in Eq. 7.10 and setting

$$\widehat{\mathbf{E}\varphi\mathbf{n}} = \{\!\{\mathbf{E}\mathbf{n}\}\!\}\,\lambda\,, \quad \text{on } \mathcal{F}\in\Gamma_0\,. \tag{7.52}$$

The numerical flux on the boundary is the same as in Eq. 7.13. For each K, the element-local VEF problem is then

$$\begin{aligned}\int_{\partial K\cap\Gamma_0} \boldsymbol{v}\cdot\{\!\{\mathbf{E}\mathbf{n}_K\}\!\}\,\lambda\,\mathrm{d}s - \int_K \nabla\boldsymbol{v} : \mathbf{E}\varphi\,\mathrm{d}\mathbf{x} + \int_K \sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int_{\partial K\cap\Gamma_b} \frac{1}{E_b}(\boldsymbol{v}\cdot\mathbf{E}\mathbf{n}_K)(\boldsymbol{J}\cdot\mathbf{n}_K)\,\mathrm{d}s \\ = \int_K \boldsymbol{v}\cdot\boldsymbol{Q}_1\,\mathrm{d}\mathbf{x} + 2\int_{\partial K\cap\Gamma_b}\frac{1}{E_b}\boldsymbol{v}\cdot\mathbf{E}\mathbf{n}_K\,J_{\mathrm{in}}\,\mathrm{d}s\,, \quad \forall\boldsymbol{v}\in\mathbb{D}_p(K)\,, \quad (7.53\mathrm{a})\end{aligned}$$

$$\int_K u\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int_K \sigma_a\,u\varphi\,\mathrm{d}\mathbf{x} = \int_K u\,Q_0\,\mathrm{d}\mathbf{x}\,, \quad \forall u\in\mathbb{Q}_p(K)\,. \quad (7.53\mathrm{b})$$

The resulting hybrid VEF method is: find $(\boldsymbol{J},\varphi,\lambda)\in\hat{RT}_p\times Y_p\times\Lambda_p$ such that

$$\begin{aligned}\int_{\Gamma_0} \llbracket\boldsymbol{v}\cdot\{\!\{\mathbf{E}\mathbf{n}\}\!\}\rrbracket\,\lambda\,\mathrm{d}s - \int \nabla_h\boldsymbol{v} : \mathbf{E}\varphi\,\mathrm{d}\mathbf{x} + \int \sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int_{\Gamma_b}\frac{1}{E_b}(\boldsymbol{v}\cdot\mathbf{E}\mathbf{n})(\boldsymbol{J}\cdot\mathbf{n})\,\mathrm{d}s \\ = \int \boldsymbol{v}\cdot\boldsymbol{Q}_1\,\mathrm{d}\mathbf{x} + 2\int_{\Gamma_b}\frac{1}{E_b}\boldsymbol{v}\cdot\mathbf{E}\mathbf{n}\,J_{\mathrm{in}}\,\mathrm{d}s\,, \quad \forall\boldsymbol{v}\in\hat{RT}_p\,, \quad (7.54\mathrm{a})\end{aligned}$$

$$\int_K u\,\nabla_h\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int_K \sigma_a\,u\varphi\,\mathrm{d}\mathbf{x} = \int_K u\,Q_0\,\mathrm{d}\mathbf{x}\,, \quad \forall u\in Y_p\,, \quad (7.54\mathrm{b})$$

$$\int_{\Gamma_0}\mu\,\llbracket\boldsymbol{J}\cdot\mathbf{n}\rrbracket\,\mathrm{d}s = 0\,, \quad \forall\mu\in\Lambda_p\,. \quad (7.54\mathrm{c})$$

Observe that this represents element-local VEF problems where the boundary conditions are provided either by the Miften-Larsen boundary conditions on the boundary of the domain or by the Lagrange multiplier λ for interior elements. Thus, if λ were known, the scalar flux and current could be solved for independently on each element.

7.4.3 Implementation Details

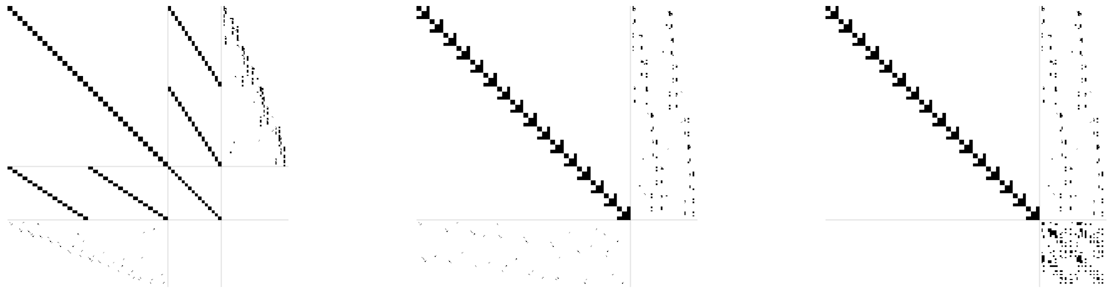
In matrix form, the hybridized system is

$$\begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{G}} & \mathbf{C}_2 \\ \hat{\mathbf{D}} & \mathbf{M}_a & \\ \mathbf{C}_1 & & \end{bmatrix}\begin{bmatrix}\underline{J} \\ \underline{\varphi} \\ \underline{\lambda}\end{bmatrix} = \begin{bmatrix}\underline{\hat{g}} \\ \underline{f} \\ 0\end{bmatrix}\,, \quad (7.55)$$

where $\hat{\mathbf{A}}$, $\hat{\mathbf{D}}$, and $\underline{\hat{g}}$ are defined in Eqs. 7.17a, 7.17c, and 7.17e, respectively, but use $\mathcal{V}=\hat{RT}_p$ and

$$\hat{\mathbf{G}} = -\int \nabla_h\boldsymbol{v} : \mathbf{E}\varphi\,\mathrm{d}\mathbf{x} \quad (7.56)$$

is the analog of $\mathbf{G}$ in Eq. 7.17d that uses $\mathcal{V}=\hat{RT}_p$ and does not include the interior face bilinear form. The DG absorption mass matrix, $\mathbf{M}_a$, and right hand side, $\underline{f}$, are unchanged



(a) (b) (c)

Figure 7.3: Sparsity plots for the block system corresponding to the hybridized Raviart Thomas discretization for the VEF equations. In (a), the degrees of freedom are organized as J_1, J_2, φ, λ. In (b), the rows and columns of the matrix in (a) are permuted to group the currents and scalar fluxes associated with each element together. With this ordering, it is clear that the scalar flux and current can be eliminated on each element without fill-in, leaving a system for λ only. Note that in practice, the elimination of the element-local problems is performed locally with dense operations and global sparse matrices are used to form the reduced system for the Lagrange multiplier. (c) shows the resulting system after performing block Gaussian elimination. This produces a globally coupled system for λ that is sparse. Once λ is known, the scalar flux and current can be solved with element-local back substitution.

from the original mixed form defined in Eqs. 7.17b and 7.17f, respectively. The constraint matrices are defined as:

$$\underset{\sim}{\mu}^T \mathbf{C}_1 \underset{\sim}{J} = \int_{\Gamma_0} \mu \, \llbracket \boldsymbol{J} \cdot \mathbf{n} \rrbracket \, \mathrm{d}s \,, \tag{7.57a}$$

$$\underset{\sim}{v}^T \mathbf{C}_2 \underset{\sim}{\lambda} = \int_{\Gamma_0} \llbracket \boldsymbol{v} \cdot \{\!\{ \mathbf{En} \}\!\} \rrbracket \, \lambda \, \mathrm{d}s \,, \tag{7.57b}$$

where $\mu \in \Lambda_p$ and $\boldsymbol{v} \in \hat{RT}_p$.

Only the constraint matrices $\mathbf{C}_1$ and $\mathbf{C}_2$ are globally coupled. The matrices $\hat{\mathbf{A}}$, $\hat{\mathbf{G}}$, $\hat{\mathbf{D}}$, and $\mathbf{M}_a$ are all block diagonal by element and can thus be eliminated on each element without fill-in. Figure 7.3a shows the sparsity pattern of the block system in Eq. 7.55. Note that this matrix can be permuted to be block diagonal by element by grouping the current and scalar flux degrees of freedom associated with each element together. This matrix is shown in Fig. 7.3b where it is clear that the block system has a structure amenable to efficient solution via block Gaussian elimination. The block system after Gaussian elimination is shown in Fig. 7.3c. After solving for the Lagrange multiplier, element-local back substitution can be applied to solve for the scalar flux and current.

Performing block Gaussian elimination on each element, the reduced system for the Lagrange multiplier reads

$$\mathbf{H}\underline{\lambda} = \begin{bmatrix}\mathbf{C}_1 & \mathbf{0}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{G}} \\ \hat{\mathbf{D}} & \mathbf{M}_a\end{bmatrix}^{-1}\begin{bmatrix}\mathbf{C}_2 \\ \mathbf{0}\end{bmatrix}\underline{\lambda} = \begin{bmatrix}\mathbf{C}_1 & \mathbf{0}\end{bmatrix}\begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{G}} \\ \hat{\mathbf{D}} & \mathbf{M}_a\end{bmatrix}^{-1}\begin{bmatrix}\hat{\underline{g}} \\ \underline{f}\end{bmatrix} . \tag{7.58}$$

The inverse of the local VEF problems is derived by finding the blocks $\mathbf{W}$, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ that satisfy

$$\begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{G}} \\ \hat{\mathbf{D}} & \mathbf{M}_a\end{bmatrix}\begin{bmatrix}\mathbf{W} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z}\end{bmatrix} = \begin{bmatrix}\mathbf{I} & \\ & \mathbf{I}\end{bmatrix} . \tag{7.59}$$

We assume that $\hat{\mathbf{A}}$ and the Schur complement $\hat{\mathbf{S}} = \mathbf{M}_a - \hat{\mathbf{D}}\hat{\mathbf{A}}^{-1}\hat{\mathbf{G}}$ are non-singular. This is justified in non-void regions where $\sigma_t > 0$. However, we do not assume $\mathbf{M}_a$ is non-singular since $\sigma_a \geq 0$ can be zero. Solving Eq. 7.59 for the blocks $\mathbf{W}$, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ under these constraints yields:

$$\mathbf{W} = \hat{\mathbf{A}}^{-1}(\mathbf{I} + \hat{\mathbf{G}}\hat{\mathbf{S}}^{-1}\hat{\mathbf{D}}\hat{\mathbf{A}}^{-1}) , \tag{7.60a}$$

$$\mathbf{X} = -\hat{\mathbf{A}}^{-1}\hat{\mathbf{G}}\hat{\mathbf{S}}^{-1} , \tag{7.60b}$$

$$\mathbf{Y} = -\hat{\mathbf{S}}^{-1}\hat{\mathbf{D}}\hat{\mathbf{A}}^{-1} , \tag{7.60c}$$

$$\mathbf{Z} = \hat{\mathbf{S}}^{-1} . \tag{7.60d}$$

The reduced system for the Lagrange multiplier is then

$$\mathbf{H}\underline{\lambda} = \mathbf{C}_1\mathbf{W}\mathbf{C}_2 = \mathbf{C}_1\hat{\mathbf{A}}^{-1}\left(\mathbf{I} + \hat{\mathbf{G}}\hat{\mathbf{S}}^{-1}\hat{\mathbf{D}}\hat{\mathbf{A}}^{-1}\right)\mathbf{C}_2\underline{\lambda} = \mathbf{C}_1\big(\mathbf{W}\hat{\underline{g}} + \mathbf{X}\underline{f}\big) . \tag{7.61}$$

We can now rewrite the 3 × 3 block system as

$$\begin{bmatrix}\hat{\mathbf{A}} & \hat{\mathbf{G}} & \mathbf{C}_2 \\ \hat{\mathbf{D}} & \mathbf{M}_a & \\ & & \mathbf{H}\end{bmatrix}\begin{bmatrix}\underline{J} \\ \underline{\varphi} \\ \underline{\lambda}\end{bmatrix} = \begin{bmatrix}\hat{\underline{g}} \\ \underline{f} \\ \mathbf{C}_1\big(\mathbf{W}\hat{\underline{g}} + \mathbf{X}\underline{f}\big)\end{bmatrix} . \tag{7.62}$$

This system can be solved with block back substitution. First, solve the globally coupled system

$$\mathbf{H}\underline{\lambda} = \mathbf{C}_1\big(\mathbf{W}\hat{\underline{g}} + \mathbf{X}\underline{f}\big) \tag{7.63}$$

for $\underline{\lambda}$. The element-local inverse can then be used to solve for the scalar flux and current with

$$\begin{bmatrix}\underline{J} \\ \underline{\varphi}\end{bmatrix} = \begin{bmatrix}\mathbf{W} & \mathbf{X} \\ \mathbf{Y} & \mathbf{Z}\end{bmatrix}\begin{bmatrix}\hat{\underline{g}} - \mathbf{C}_2\underline{\lambda} \\ \underline{f}\end{bmatrix} = \begin{bmatrix}\mathbf{W}\big(\hat{\underline{g}} - \mathbf{C}_2\underline{\lambda}\big) + \mathbf{X}\underline{f} \\ \mathbf{Y}\big(\hat{\underline{g}} - \mathbf{C}_2\underline{\lambda}\big) + \mathbf{Z}\underline{f}\end{bmatrix} . \tag{7.64}$$

In this way, only $\dim(\Lambda_p)$ globally coupled unknowns must be solved for as opposed to the $\dim(RT_p) + \dim(Y_p)$ required by the original mixed formulation.

In practice, the blocks of the inverse $\mathbf{W}$, $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{Z}$ are formed using dense matrix operations applied on the element-local matrices corresponding to the degrees of freedom of a single element. The local matrices are then broadcast to a global sparse matrix in order to perform the sparse matrix multiplication required to form the reduced system for the Lagrange multiplier. The Lagrange multiplier can be scalably solved for by preconditioning **H** with AMG. In addition, recovering the scalar flux and current is a post-processing step that is independent on each element and thus scales optimally.

7.5 Results

The VEF algorithms presented here were implemented using the MFEM [90, 102] finite element framework. The BiCGStab solver from MFEM was used to solve the discretized VEF equations. Lower block triangular preconditioners were built using MFEM's Jacobi smoother and BoomerAMG from the sparse linear algebra package *hypre* [94]. KINSOL, from the Sundials package [103], provided the fixed-point and Anderson-accelerated fixed-point solvers. When iterative solver results are not presented, the parallel implementation of the sparse direct solver SuperLU [81] was used. The high-order DG S_N transport solver from [15] was used. Unless otherwise noted, the angular flux and VEF scalar flux are approximated using the same degree finite element spaces. However, the positive Bernstein polynomials [99] are used for the transport discretization's local polynomial basis whereas the Lagrange basis through the Gauss-Legendre points is used for the VEF scalar flux. The use of a positive transport basis facilitates the use of the quadratic programming negative flux fixup from [43] that is used on the crooked pipe problem.

Since all methods produce a VEF scalar flux in Y_p, the methods are parameterized by their choice of space for the current. Thus, we refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and $Y_p \times \hat{R}T_p \times \Lambda_p$ methods as H1, RT, and HRT, respectively.

7.5.1 Method of Manufactured Solutions

The accuracy of the methods are determined with MMS. The solution is set to

$$\psi = \frac{1}{4\pi}\left[\alpha(\mathbf{x}) + \mathbf{\Omega}\cdot\boldsymbol{\beta}(\mathbf{x}) + \mathbf{\Omega}\otimes\mathbf{\Omega} : \mathbf{\Theta}(\mathbf{x})\right] , \tag{7.65}$$

where

$$\alpha(\mathbf{x}) = \sin(\pi x)\sin(\pi y) + \delta\,, \tag{7.66a}$$

$$\boldsymbol{\beta}(\mathbf{x}) = \begin{bmatrix} \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \\ \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \end{bmatrix} , \tag{7.66b}$$

$$\mathbf{\Theta}(\mathbf{x}) = \begin{bmatrix} \frac{1}{2}\sin\left(\frac{3\pi(x+\zeta)}{1+2\zeta}\right)\sin\left(\frac{3\pi(y+\zeta)}{1+2\zeta}\right) & \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \\ \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) & \frac{1}{4}\sin\left(\frac{3\pi(x+\zeta)}{1+2\zeta}\right)\sin\left(\frac{3\pi(y+\zeta)}{1+2\zeta}\right) \end{bmatrix} , \tag{7.66c}$$

where $\delta = 1.25$ is used to ensure $\psi > 0$ and $\zeta = 0.1$ and $\omega = 0.05$ are used to test spatially-dependent, non-isotropic inflow boundary conditions. The domain is $\mathcal{D} = [0,1]^2$. With this definition:

$$\phi(\mathbf{x}) = \alpha(\mathbf{x}) + \frac{1}{3}\operatorname{trace}\mathbf{\Theta}(\mathbf{x}) \,, \tag{7.67a}$$

$$\boldsymbol{J}(\mathbf{x}) = \frac{1}{3}\boldsymbol{\beta}(\mathbf{x}) \,, \tag{7.67b}$$

$$\mathbf{P}(\mathbf{x}) = \frac{\alpha(\mathbf{x})}{3}\mathbf{I} + \frac{1}{15}\begin{bmatrix} 3\Theta_{11}(\mathbf{x}) + \Theta_{22}(\mathbf{x}) & \Theta_{12}(\mathbf{x}) \\ \Theta_{21}(\mathbf{x}) & \Theta_{11}(\mathbf{x}) + 3\Theta_{22}(\mathbf{x}) \end{bmatrix} . \tag{7.67c}$$

This leads to an exact Eddington tensor $\mathbf{E} = \mathbf{P}/\phi$ that is dense and spatially varying. The MMS ψ and ϕ are substituted into the transport equation to solve for the MMS source q that forces the solution to Eq. 7.65.

The accuracy of the VEF discretizations are investigated in isolation by computing the VEF data from the MMS angular flux and setting the sources Q_0 and $\boldsymbol{Q}_1$ to the moments of the MMS source. This is accomplished by projecting the MMS angular flux onto a degree-p DG finite element space and using Level Symmetric S_4 angular quadrature to compute the VEF data, the moments of the MMS source, and the inflow boundary function. The VEF equations are then solved as if $\mathbf{E}$, E_b, Q_0, $\boldsymbol{Q}_1$, and J_{in} are given data. Errors are calculated with the $L^2(\mathcal{D})$ norm for scalars and the $[L^2(\mathcal{D})]^2$ norm for vectors given by

$$\|u\| = \sqrt{\int u^2 \,\mathrm{d}\mathbf{x}} \,, \tag{7.68}$$

and

$$\|\boldsymbol{v}\| = \sqrt{\int \boldsymbol{v}\cdot\boldsymbol{v} \,\mathrm{d}\mathbf{x}} \,, \tag{7.69}$$

respectively. We also use the $L^2(\mathcal{D})$ projection operator $\Pi_p : L^2(\mathcal{D}) \to Y_p$ such that

$$\int u(v - \Pi_p v) \,\mathrm{d}\mathbf{x} = 0 \,, \quad \forall u \in Y_p \,, \tag{7.70}$$

for some $v \in L^2(\mathcal{D})$. In particular, Π_p is used to project the exact MMS scalar flux onto a Y_p finite element grid function in order to investigate a superconvergence property of mixed finite elements.

We use refinements of a third-order mesh created by distorting an orthogonal mesh according to the velocity field of the Taylor Green vortex. This mesh distortion is generated by advecting the mesh control points with

$$\mathbf{x} = \int_0^T \mathbf{v} \,\mathrm{d}t \,, \tag{7.71}$$

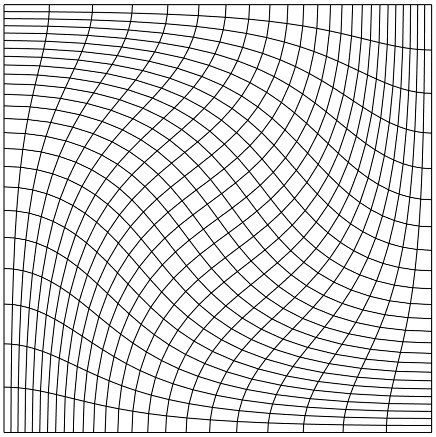


Figure 7.4: A depiction of a third-order mesh generated by distorting an orthogonal mesh according to the Taylor Green vortex. Refinements of this mesh are used in calculating the error with the method of manufactured solutions.

where the final time $T = 0.3\pi$ and

$$\mathbf{v} = \begin{bmatrix} \sin(x)\cos(y) \\ -\cos(x)\sin(y) \end{bmatrix} \tag{7.72}$$

is the analytic solution of the Taylor Green vortex. 300 forward Euler steps were used to advect the mesh. An example mesh is shown in Fig. 7.4. Logarithmic regression is used to fit the constant and order of accuracy according to

$$E = Ch^{\tilde{p}} \tag{7.73}$$

where E is the error, C the constant, and $\tilde{p}$ the order of accuracy. Four values of h were used for each MMS problem considered in this section. The raw error values are provided in Appendix A.

We first show the accuracy of the three methods on a simple radiation diffusion problem. The above process is used with $\mathbf{\Theta} = 0$ so that the angular flux is linearly anisotropic. This forces the Eddington tensor and boundary factor to $\mathbf{E} = \frac{1}{3}\mathbf{I}$ and $E_b = \frac{1}{2}$, mimicking a radiation diffusion problem. Table 7.1 shows the estimated order of accuracy and constant for $p \in [0, 3]$. The error in the scalar flux is computed with two methods: 1) by comparing to the analytic MMS scalar flux solution directly and 2) by projecting the analytic MMS solution onto the corresponding Y_p space. For all orders, the first error measure for the scalar flux converges $\mathcal{O}(h^{p+1})$ while the second converges $\mathcal{O}(h^{p+2})$. This is a mixed finite element superconvergence result that indicates that the nodal values of the scalar flux solution converge one order higher than the Y_p interpolation allows. The current converges as $\mathcal{O}(h^{p+1})$ for all three methods and all orders except for $Y_0 \times W_1$ which converges as $\mathcal{O}(h^{3/2})$ instead of $\mathcal{O}(h)$. On this diffusive problem, the scalar flux and current solutions from the unhybridized and hybridized RT methods are equivalent to machine precision.

Table 7.1: Estimates of the order of accuracy and constant from an isotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the VEF data are constant in space and thus are represented exactly.

		$\|\varphi - \varphi_{\text{ex}}\|$			$\|\varphi - \Pi\varphi_{\text{ex}}\|$			$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$		
p	Value	H1	RT	HRT	H1	RT	HRT	H1	RT	HRT
1	Order	2.000	2.000	2.000	3.017	3.053	3.053	2.001	2.000	2.000
	Constant	0.261	0.261	0.261	0.163	0.197	0.197	0.353	0.785	0.785
2	Order	3.001	3.003	3.003	4.144	4.096	4.096	3.150	2.989	2.989
	Constant	0.070	0.070	0.070	0.090	0.142	0.142	0.202	0.780	0.780
3	Order	3.995	4.016	4.016	5.098	5.125	5.125	4.018	4.016	4.016
	Constant	0.027	0.030	0.030	0.048	0.132	0.132	0.118	0.928	0.928
4	Order	4.971	4.971	4.971	6.013	5.964	5.963	5.096	4.675	4.675
	Constant	0.034	0.034	0.034	0.048	0.045	0.045	0.157	0.217	0.217

Table 7.2: Estimates of the order of accuracy and constant from a quadratically anisotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the angular flux used to calculate the VEF data is represented with Y_p. Due to this, the maximum accuracy expected is order $p + 1$.

		$\|\varphi - \varphi_{\text{ex}}\|$			$\|\varphi - \Pi\varphi_{\text{ex}}\|$			$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$		
p	Value	H1	RT	HRT	H1	RT	HRT	H1	RT	HRT
1	Order	2.004	2.004	2.004	2.310	2.332	2.318	1.939	0.974	0.980
	Constant	1.200	1.200	1.198	0.430	0.488	0.451	3.831	0.394	0.353
2	Order	2.958	2.957	2.963	2.995	2.979	3.054	2.486	2.564	2.522
	Constant	1.233	1.225	1.263	0.352	0.329	0.485	1.601	1.605	1.477
3	Order	4.046	4.045	4.044	4.348	4.313	4.263	4.003	2.857	2.905
	Constant	2.612	2.599	2.592	0.942	0.837	0.710	12.054	0.555	0.584
4	Order	4.787	4.785	4.783	5.033	4.921	4.845	4.221	4.351	4.454
	Constant	0.931	0.923	0.923	0.421	0.283	0.258	1.011	1.458	2.050

Table 7.3: Estimates of the order of accuracy and constant from a quadratically anisotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the angular flux used to calculate the VEF data is represented with Y_{p+1}. Due to this, the maximum accuracy expected is order $p+2$.

		$\|\varphi - \varphi_{\text{ex}}\|$			$\|\varphi - \Pi\varphi_{\text{ex}}\|$			$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$		
p	Value	H1	RT	HRT	H1	RT	HRT	H1	RT	HRT
0	Order	0.999	0.999	0.999	2.019	2.002	2.001	1.477	1.001	1.001
	Constant	0.781	0.780	0.780	1.439	1.338	1.304	2.561	0.517	0.516
1	Order	2.001	2.001	2.001	3.012	2.954	2.969	1.941	0.987	1.887
	Constant	1.180	1.179	1.178	1.683	1.488	1.390	2.377	0.083	0.583
2	Order	2.961	2.960	2.960	3.990	4.028	4.006	3.065	2.967	2.903
	Constant	1.208	1.204	1.204	2.383	2.447	2.347	3.312	1.273	0.783
3	Order	4.042	4.041	4.041	4.965	4.732	4.759	3.931	2.726	3.667
	Constant	2.554	2.545	2.545	1.883	0.896	0.864	6.673	0.102	0.575

This test is repeated for the quadratically-anisotropic MMS problem (i.e. using $\boldsymbol{\Theta}(\mathbf{x})$ as defined in Eq. 7.66c) in Table 7.2. Since the MMS solution is projected onto Y_p, it is expected that this problem can converge at a maximum of order $p+1$. This can be seen in the loss of the superconvergence property. Here, both error measures for the scalar flux converge with $\mathcal{O}(h^{p+1})$. On this transport MMS problem, the current convergence is also reduced. Compared to the diffusion case, the H1 current error is maintained for p odd but is reduced by 1/2 for p even. The RT and HRT methods lose one order for p odd but only half an order for p even. In addition, the RT and HRT discretizations are no longer equivalent to machine precision. This loss of equivalence may be due to inexact numerical quadrature in terms involving the VEF data. The VEF data are improper rational polynomials in space and thus cannot be exactly integrated with Gaussian quadrature.

Finally, we repeat the transport MMS problem in the case where the angular flux solution is projected onto Y_{p+1} instead of Y_p. This allows a maximum accuracy in the problem of $\mathcal{O}(h^{p+2})$. The estimated orders of convergence and constants are provided in Table 7.3. Convergence rates similar to the diffusion problem are observed: the scalar flux solutions converge optimally for all methods and superconvergence of the scalar flux returns. The H1 and HRT methods produce currents that converge at similar rates as in the diffusion case. However, the unhybridized RT method converges suboptimally by one order for p even. The difference in convergence rates between the RT and HRT methods indicates the HRT method

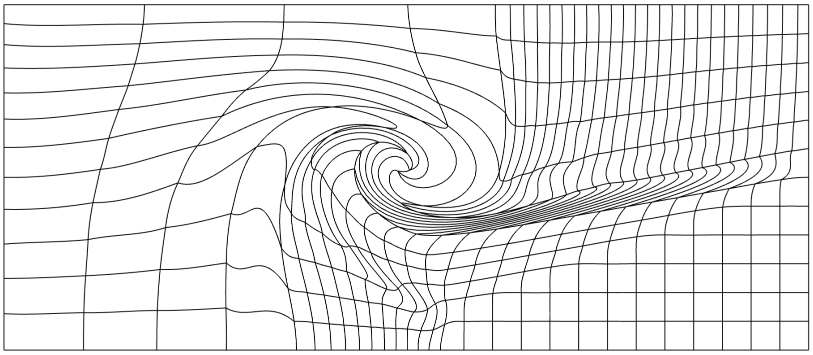


Figure 7.5: A depiction of the triple point mesh used to stress the VEF algorithms on a severely distorted, third-order mesh. This mesh was generated with a Lagrangian hydrodynamics simulation.

is in fact a new discretization for the VEF equations and not simply an algebraic method to reduce the number of globally coupled unknowns.

7.5.2 Thick Diffusion Limit

The convergence of the VEF methods are investigated in the thick diffusion limit. The material data are set to

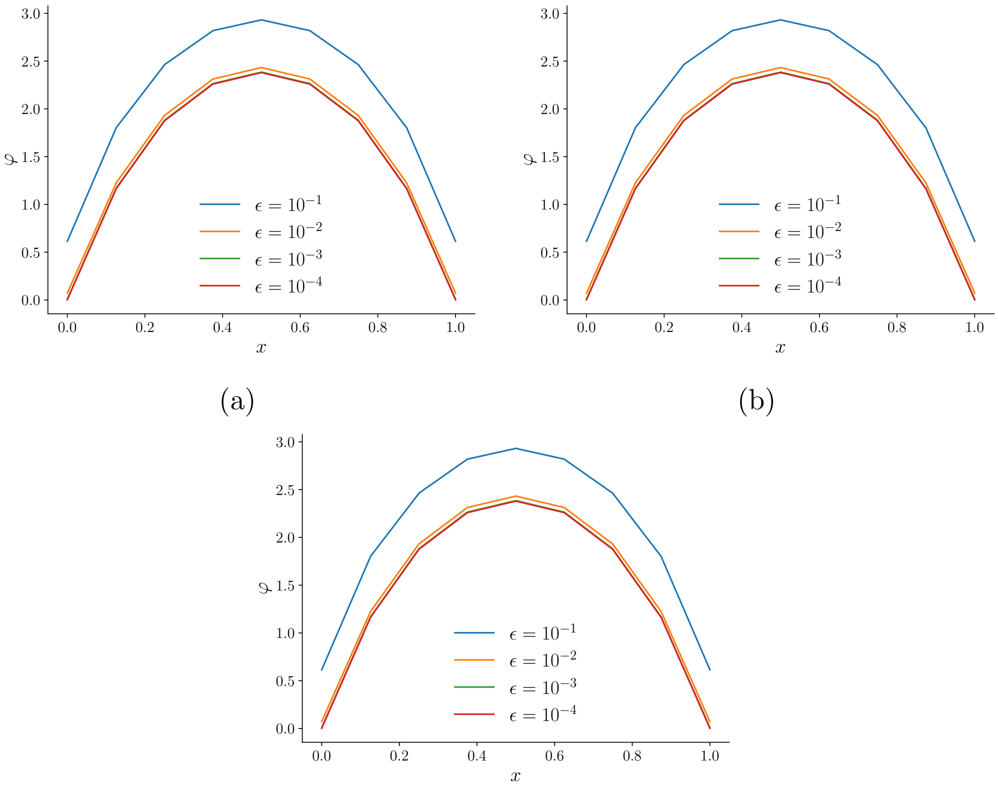
$$\sigma_t = 1/\epsilon\,, \quad \sigma_a = \epsilon\,, \quad \sigma_s = 1/\epsilon - \epsilon\,, \quad q = \epsilon\,, \tag{7.74}$$

where $\epsilon \in (0,1]$ and the thick diffusion limit corresponds to $\epsilon \to 0$. We use two coarse meshes that do not resolve the mean free path to stress the convergence of the VEF method. The first is an orthogonal 8×8 mesh with $\mathcal{D} = [0,1]^2$. The second is the triple point mesh shown in Fig. 7.5, a third-order mesh generated with a Lagrangian hydrodynamics code where $\mathcal{D} = [0,7] \times [0,3]$. On the triple point mesh, the angular flux is only approximately inverted due to the lagging of reentrant faces and thus it is expected that convergence will degrade. In addition, highly distorted elements have poor approximation properties. We use Level Symmetric S_4 angular quadrature. The three methods are compared when $p = 2$. The coupled transport-VEF system is solved with fixed-point iteration.

Table 7.4 shows the number of fixed-point iterations until convergence to a tolerance of 10^{-6} for each method on the orthogonal and triple point meshes. Rapid convergence is seen for all methods on both problems. The three methods converge equivalently on the orthogonal mesh. On the triple point mesh, the RT and HRT methods converged equivalently. Lineouts of the 2D VEF scalar flux solutions for each method as $\epsilon \to 0$ are provided in Fig. 7.6 for the orthogonal mesh. In all cases, a non-trivial solution is found.

Table 7.4: The number of fixed-point iterations required for convergence as the thick diffusion limit parameter $\epsilon \to 0$. The H1, RT, and HRT columns refer to the $Y_2 \times W_3$, $Y_2 \times RT_2$, and hybridized $Y_2 \times RT_2$ discretizations, respectively. Convergence is tested on an orthogonal 8×8 mesh and on the triple point mesh, a mesh with re-entrant faces. Due to the re-entrant faces, a partial transport sweep is used making convergence slower on the triple point mesh.

	Orthogonal			Triple Point		
ϵ	H1	RT	HRT	H1	RT	HRT
10^{-1}	8	8	8	20	21	21
10^{-2}	6	6	6	13	19	19
10^{-3}	4	4	4	9	13	13
10^{-4}	3	3	3	6	8	8

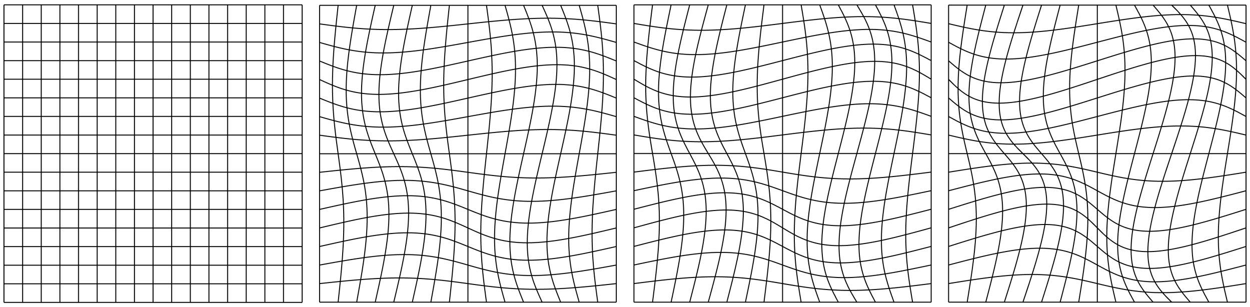


(a)

(b)

(c)

Figure 7.6: Lineouts of the 2D solution as $\epsilon \to 0$. The methods all converge to the asymptotic solution indicating they preserve the thick diffusion limit.



(a) $\alpha = 0.000$ (b) $\alpha = 0.050$ (c) $\alpha = 0.060$ (d) $\alpha = 0.080$

Figure 7.7: A selection of meshes generated by distorting a third-order, orthogonal 16×16 mesh according to the sine distortion. The parameter α controls the amount of distortion. These meshes are used to assess linear solver robustness against mesh distortion.

7.5.3 Solver Performance on Curved Meshes

Here, we investigate the robustness of the preconditioned iterative solvers on increasingly distorted meshes. The meshes were created by moving the interior control points of an initially orthogonal, third-order mesh according to the sine distortion:

$$\mathbf{x} = \mathbf{x} + \alpha \begin{bmatrix} \sin(2\pi x)\sin(2\pi y) \\ \sin(2\pi x)\sin(2\pi y) \end{bmatrix} , \tag{7.75}$$

where α controls the amount of distortion. When $\alpha = 0$, the mesh is unchanged. The initial mesh was 16×16 with $\mathcal{D} = [0, 1]^2$. A range of meshes are shown in Fig. 7.7. Solver performance is evaluated on the first iteration of the thick diffusion limit problem introduced in the previous section. We use $\epsilon = 10^{-1}$. The number of BiCGStab iterations until convergence to a tolerance of 10^{-6} are shown for a range of mesh distortions in Table 7.5 for the H1, RT, and HRT VEF methods. The solvers for the RT method did not converge in 250 iterations once the mesh became too distorted. The H1 discretization converged on all the meshes tested but the iteration counts varied between 46 and 69 whereas HRT was solved more uniformly, varying only between 7 and 11 iterations. This indicates the solvers for the RT method are sensitive to mesh distortion.

7.5.4 Crooked Pipe

We now show convergence in outer fixed-point iterations and inner preconditioned linear solver iterations on a more realistic, multi-material problem. The geometry and materials are shown in Fig. 7.8. The problem consists of two materials, the wall and the pipe, which have an 1000x difference in total interaction cross section. Time dependence is mocked by including artificial absorption and sources that correspond to backward Euler time integration. The time step is set so that $c\Delta t = 10^3$ and the initial condition is $\psi_0 = 10^{-4}$. The absorption

Table 7.5: Number of BiCGStab iterations until convergence on the first iteration of a thick diffusion limit problem with $\epsilon = 10^{-1}$ as the mesh distortion parameter increases. A – indicates BiCGStab did not converge in 250 iterations. Here, H1, RT, and HRT rows refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively.

		Distortion Amount					
p	Method	0.000	0.025	0.050	0.060	0.070	0.080
1	H1	46	48	48	48	48	50
	RT	20	22	26	31	72	–
	HRT	7	10	8	8	8	8
2	H1	59	61	52	55	54	57
	RT	28	27	31	–	–	–
	HRT	11	10	9	9	10	9
3	H1	54	54	56	69	55	57
	RT	29	28	41	–	–	–
	HRT	9	9	8	8	9	9
4	H1	51	55	55	66	57	61
	RT	41	44	46	78	–	–
	HRT	7	9	10	10	10	11

and source are then $\sigma_a = 1/c\Delta t = 10^{-3}\frac{1}{\text{cm}}$ and $q = \psi_0/c\Delta t = 10^{-1}\frac{1}{\text{cm}^3\cdot\text{s}\cdot\text{str}}$. The boundary conditions are set so that isotropic inflow of magnitude $1/2\pi$ enters on the left entrance of the pipe. A Level Symmetric S_{12} angular quadrature set is used. The quadratic programming negative flux fixup from [43] is used inside the transport sweep to ensure positivity so that the VEF data are well defined.

The outer fixed-point and inner linear iterative efficiency are shown by refining in h and p on an orthogonal mesh. Anderson acceleration with two Anderson vectors is used. The previous outer iteration's solution is used as an initial guess for the inner solver so that the initial guess becomes progressively more accurate as the outer iteration converges. The outer tolerance is 10^{-6} and the inner BiCGStab tolerance is 10^{-8}.

Table 7.6 shows the number of Anderson-accelerated fixed-point iterations to convergence and the maximum, minimum, and average number of inner iterations performed across all outer iterations for the H1, RT, and HRT methods. The RT and HRT methods had equivalent convergence in outer iterations with H1 requiring up to 15 more iterations. The slowdown of H1 is likely caused by its increased reliance on the negative flux fixup compared to the RT and HRT methods. The RT and HRT inner solvers were scalable in h and p while the H1 solvers were not. On the problem with the smallest value of h, the H1 inner solver did not converge within 100 iterations on at least one of the solves.

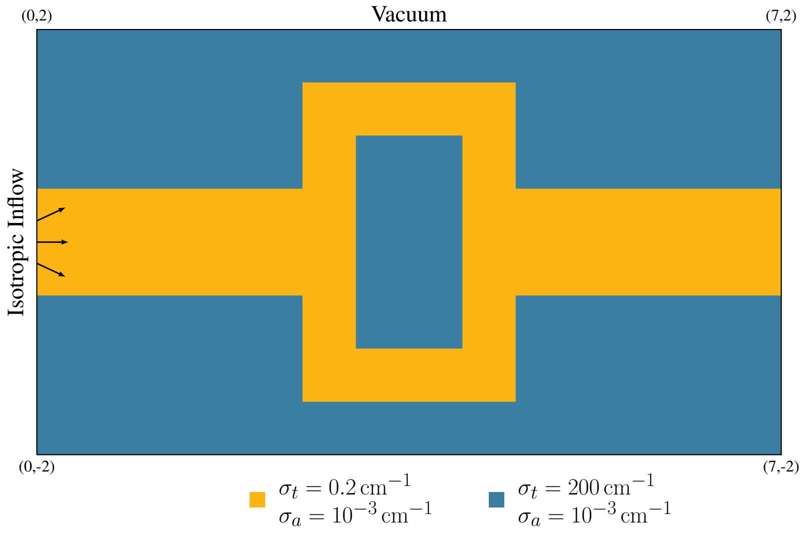

(0,2)
Vacuum
(7,2)
Isotropic Inflow
(0,-2)
(7,-2)
$\sigma_t = 0.2\,\mathrm{cm}^{-1}$
$\sigma_a = 10^{-3}\,\mathrm{cm}^{-1}$
$\sigma_t = 200\,\mathrm{cm}^{-1}$
$\sigma_a = 10^{-3}\,\mathrm{cm}^{-1}$

Figure 7.8: The geometry, material data, and boundary conditions for the linearized crooked pipe problem.

7.5.5 Eigenvalue Problem

It was observed that the $Y_p \times W_{p+1}$ discretization exhibited poor solution quality in under resolved problems and could not be scalably solved using block preconditioners. In particular, AMG struggled to adequately precondition the lumped Schur complement. To investigate this issue, we consider the following eigenvalue problem:

$$-\nabla^2 u = \lambda u\,, \quad \mathbf{x} \in \mathcal{D}\,, \tag{7.76}$$

$$u = 0\,, \quad \mathbf{x} \in \partial\mathcal{D}\,, \tag{7.77}$$

with $\mathcal{D} = [0, 1]^2$. The exact solutions are

$$u = \sin(k_x \pi x)\sin(k_y \pi y)\,, \quad \lambda = \pi^2(k_x^2 + k_y^2)\,. \tag{7.78}$$

The $Y_1 \times W_2$ discretization's lumped Schur complement is used to discretize this problem as: find $u \in Y_1$ such that

$$\tilde{\mathbf{S}}\underline{u} = \lambda \mathbf{M}\underline{u}\,, \tag{7.79}$$

where $\mathbf{M}$ is the Y_1 mass matrix. The Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) solver from *hypre* was used to solve for the first 5 eigenvalues and eigenvectors of this system. The solver correctly found the first four eigenvalues and eigenvectors but produced the high-frequency, "checkerboard" mode shown in Fig. 7.9 for the fifth. This checkerboard mode corresponded to a non-physically degenerate eigenvalue of $8\pi^2$. The presence of this mode indicates the $Y_p \times W_{p+1}$ discretization allows non-physical,

Table 7.6: The number of outer Anderson-accelerated fixed-point iterations until convergence along with the maximum, minimum, and average numbers of inner BiCGStab iterations until convergence on the linearized crooked pipe problem. Two Anderson vectors were used. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The H1 and RT methods were preconditioned with a block lower triangular preconditioner with AMG applied to the lumped Schur complement. HRT was preconditioned with AMG. The previous outer iteration's solution was used as the initial guess for the inner iteration.

		Outer			Max Inner			Min Inner			Avg. Inner		
	N_e	H1	RT	HRT	H1	RT	HRT	H1	RT	HRT	H1	RT	HRT
$p = 1$	112	18	13	13	37	11	7	5	3	2	21.39	6.54	4.54
	448	20	13	13	69	12	7	7	5	3	37.95	8.31	5.31
	1792	27*	16	16	100	13	8	7	5	2	51.30	8.12	5.19
	7168	30*	16	16	100	15	8	8	5	3	57.57	8.50	5.50
$p = 2$	112	23	15	15	46	22	10	9	7	4	26.57	13.53	6.53
	448	26	16	16	76	21	10	11	7	4	37.38	14.19	7.00
	1792	29*	18	17	100	23	12	11	6	3	53.76	14.22	7.53
	7168	25*	17	17	100	23	13	12	9	4	66.28	15.59	8.18
$p = 3$	112	23	16	16	50	20	9	7	6	4	26.17	11.69	6.12
	448	26	17	17	74	19	9	12	6	3	39.35	12.23	6.24
	1792	30*	17	17	100	21	9	10	8	4	50.07	13.41	6.18
	7168	30*	19	19	100	21	11	9	6	3	58.30	13.21	6.26

spurious modes that are slowly decaying and high frequency. Such modes are slow to remove with relaxation and also cannot be accurately represented on a coarser grid, meaning AMG will not be an effective preconditioner.

7.5.6 Weak Scaling

Finally, we show that the RT and HRT methods scale in parallel. The parallel partitioning is such that there are $\approx 9\,000$ VEF scalar flux unknowns per processor. The results were generated on 32 nodes of the `rztopaz` machine at LLNL which has two 18-core Intel Xeon E5-2695 CPUs per node. The materials and geometry from the crooked pipe in Section 7.5.4 are used. In Table 7.7, the number of BiCGStab iterations until convergence to a tolerance of 10^{-8} is shown when 1) a transport solve is used to compute the VEF data and 2) the VEF data are set to their thick diffusion limit values of $\mathbf{E} = \frac{1}{3}\mathbf{I}$ and $E_b = \frac{1}{2}$. The RT and HRT discretizations were used with $p = 2$. Both the RT and HRT methods are scalable in parallel out to over 10 million scalar flux unknowns. Compared to the corresponding diffusion problems, HRT required at most 5 more iterations to solve the VEF equations

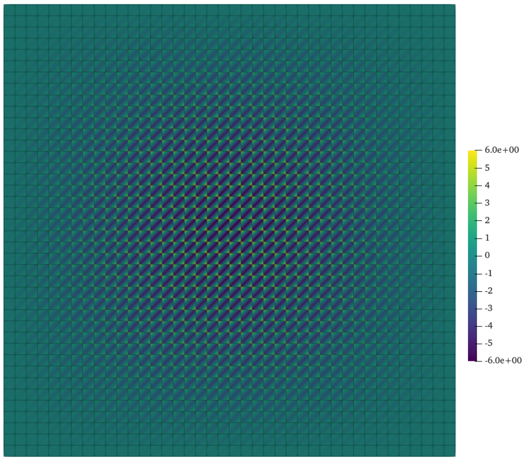


Figure 7.9: A depiction of an eigenmode corresponding to an eigenvalue of $8\pi^2$ of the Poisson eigenvalue problem discretized with the $Y_1 \times W_2$ discretization's lumped Schur complement. For this eigenvalue, the exact solution is $\sin(2\pi x)\sin(2\pi y)$ meaning this mode is spurious. The presence of high-frequency spurious modes in the $Y_p \times W_{p+1}$ discretization's lumped Schur complement degrades the effectiveness of AMG and thus the performance of the block preconditioners used to solve the full $Y_p \times W_{p+1}$ discretization.

while RT required at most 9 more iterations.

Table 7.7: A weak scaling study on the first iteration of the linearized crooked pipe problem. The RT and HRT columns refer to the $Y_2 \times RT_p$ and hybridized $Y_2 \times RT_P$ discretizations, respectively. The RT method uses lower block triangular preconditioning with AMG on the lumped Schur complement. HRT is preconditioned with AMG. BiCGStab iteration counts are compared when a parallel block Jacobi sweep is used to compute the VEF data (VEF) and when the VEF data are set to mock a radiation diffusion problem (Diffusion). The DOF column corresponds to the number of VEF scalar flux unknowns.

		RT		HRT	
Processors	N_e	VEF	Diffusion	VEF	Diffusion
36	36 288	29	27	14	12
72	70 000	34	26	15	12
144	145 152	31	24	15	12
288	285 628	33	28	16	12
576	580 608	32	26	15	12
864	867 328	32	26	18	13
1152	1 153 852	33	27	16	12

Chapter 8

Second Moment Methods

Lewis and Miller's SMM [53] couples the transport equation to the moment equations with additive closures that depend linearly on the transport solution. This scheme is characterized by a two-stage process where 1) the transport equation is inverted with a scattering source defined from the previous iteration's SMM scalar flux and 2) a *diffusion approximation* is solved with transport-dependent correction sources. By contrast, VEF uses nonlinear closures and the generally non-symmetric VEF moment system must be solved at each iteration. Since SMM solves the symmetric diffusion equation instead of the non-symmetric VEF equations, less expensive solvers, such as conjugate gradient, can be used instead of the general purpose methods such as BiCGStab or GMRES which generally require more computation and storage. In addition, where the VEF closures are well defined only when $\psi > 0$, the additive closures used in SMM are well-defined for any ψ.

Here, we pursue independent discretizations of the SMM moment system. An independent method can be designed such that its diffusion system exactly matches that of an existing radiation diffusion package. The existing diffusion package could then be extended to a transport algorithm by simply supplying a modified, transport-dependent source term corresponding to the SMM correction sources. Such a method would ease code coupling between radiation and hydrodynamics and allow reuse of the existing linear and nonlinear solvers implemented for radiation diffusion. Finally, the independent approach allows the moment system to be agnostic to the transport solver. For example, the correction sources could be computed using a stochastic solver such as IMC.

We note that, since the left hand side of the SMM moment system is simply radiation diffusion, consistent SMMs can be designed by using any of the diffusion discretizations developed for consistent DSA methods such as Haut *et al.* [17], Warsa *et al.* [113], Adams and Martin [114], and Wang and Ragusa [115]. Furthermore, many consistent DSA schemes can be scalably solved with preconditioned iterative solvers (e.g. [17, 115]). The design of efficient, consistent SMM algorithms then only requires developing consistent discretizations for the SMM correction sources. Thus, in the case of SMM, the consistent approach is likely to be effective. However, such a method would not enjoy the flexibilities discussed above.

In this chapter, we use the connection between VEF and SMM established in Section 3.6

to systematically convert the VEF methods developed in Chapters 6 and 7 to derive discrete SMMs. In particular, we derive SMM analogs of the interior penalty, continuous finite element, and mixed finite element VEF methods. The chapter concludes with numerical results demonstrating the accuracy and performance of the methods.

8.1 Discrete Second Moment Methods

The equivalence of SMM and VEF linearized about a linearly anisotropic solution provides a systematic path toward deriving discrete SMMs. Any VEF method can be converted to an SMM through the linearization process described in Section 3.6. We make use of the structure of the coupled transport-VEF system to simplify the derivations. Consider

$$\mathbf{F}(\psi, \mathbf{X}) = \begin{bmatrix} \boldsymbol{\Theta}(\psi, \varphi) \\ \mathbf{V}(\psi, \mathbf{X}) \end{bmatrix} = 0 \,, \tag{8.1}$$

where $\boldsymbol{\Theta}(\psi, \varphi)$ represents a generic transport discretization and $\mathbf{V}(\psi, \varphi)$ a generic discretization of the VEF moment equations. The moment system's unknowns are represented by $\mathbf{X}$ which includes the scalar flux and current for discretizations of the first-order form of the VEF equations and just the scalar flux for discretizations in second-order form. Here, $\boldsymbol{\Theta}$ and $\mathbf{V}$ include the inflow and Miften-Larsen boundary conditions, respectively. Given that the operators that make up the discretization of the transport equation are linear in both the angular and scalar flux and that the operators in the VEF discretization are nonlinear in the angular flux and linear in the scalar flux (and current where applicable), the linearization process will produce a system of the form:

$$\begin{bmatrix} \boldsymbol{\Theta}(\psi, \varphi) \\ \mathbf{V}(\psi_0, \mathbf{X}) + \left.\dfrac{\partial \mathbf{V}}{\partial \psi}\right|_{\psi_0} \end{bmatrix} = 0 \,, \tag{8.2}$$

where ψ_0 is the diffusion approximation of the transport problem. In other words, we have the original transport equation coupled to a discretization of diffusion with a ψ-dependent correction term arising from the derivative of the VEF operator, $\frac{\partial \mathbf{V}}{\partial \psi}$, evaluated at the diffusion approximation to the transport problem.

In the following subsections, the methods derived in Chapters 6 and 7 are linearized to form discrete SMMs by determining 1) the diffusion problem, $\mathbf{V}(\psi_0, \mathbf{X})$, found by evaluating the VEF system using a linearly anisotropic angular flux and 2) the transport-dependent correction terms corresponding to $\left.\frac{\partial \mathbf{V}}{\partial \psi}\right|_{\psi_0}$. We will see that linearizing the *discrete* VEF system provides a straightforward path toward discretizing the correction terms.

8.1.1 Interior Penalty

From Section 6.3.1, the IP VEF discretization is: find $\varphi \in Y_p$ such that

$$\int_{\Gamma_b} E_b\, u\varphi \,\mathrm{d}s + \int_{\Gamma_0} \kappa \, [\![u]\!] \, [\![\varphi]\!] \,\mathrm{d}s - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{1}{\sigma_t} \nabla_h \cdot (\mathbf{E}\varphi) \cdot \mathbf{n} \right\}\!\!\right\} \mathrm{d}s - \int_{\Gamma_0} \left\{\!\!\left\{ \frac{\nabla_h u}{\sigma_t} \right\}\!\!\right\} \cdot [\![\mathbf{E}\varphi\mathbf{n}]\!] \,\mathrm{d}s$$
$$+ \int \nabla_h u \cdot \frac{1}{\sigma_t} \nabla_h \cdot (\mathbf{E}\varphi) \,\mathrm{d}\mathbf{x} + \int \sigma_a \, u\varphi \,\mathrm{d}\mathbf{x}$$
$$= \int u \, Q_0 \,\mathrm{d}\mathbf{x} + \int \nabla_h u \cdot \frac{\boldsymbol{Q}_1}{\sigma_t} \,\mathrm{d}\mathbf{x} - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{\boldsymbol{Q}_1 \cdot \mathbf{n}}{\sigma_t} \right\}\!\!\right\} \mathrm{d}s - 2 \int_{\Gamma_b} u \, J_{\mathrm{in}} \,\mathrm{d}s \,, \quad \forall u \in Y_p \,, \tag{8.3}$$

where κ is the penalty parameter. Evaluating the VEF data when the angular flux is linearly anisotropic gives

$$\mathbf{E} = \frac{1}{3}\mathbf{I} \,, \quad E_b = \frac{1}{2} \,. \tag{8.4}$$

The diffusion operator is then

$$\frac{1}{2} \int_{\Gamma_b} u\varphi \,\mathrm{d}s + \int_{\Gamma_0} \kappa \, [\![u]\!] \, [\![\varphi]\!] \,\mathrm{d}s - \int_{\Gamma_0} [\![u]\!] \, \{\!\!\{ D \nabla_h \varphi \cdot \mathbf{n} \}\!\!\} \,\mathrm{d}s - \int_{\Gamma_0} \{\!\!\{ D \nabla_h u \cdot \mathbf{n} \}\!\!\} \, [\![\varphi]\!] \,\mathrm{d}s$$
$$+ \int \nabla_h u \cdot D \nabla_h \varphi \,\mathrm{d}\mathbf{x} + \int \sigma_a \, u\varphi \,\mathrm{d}\mathbf{x}$$
$$= \int u \, Q_0 \,\mathrm{d}\mathbf{x} + \int \nabla_h u \cdot \frac{\boldsymbol{Q}_1}{\sigma_t} \,\mathrm{d}\mathbf{x} - \int_{\Gamma_0} [\![u]\!] \left\{\!\!\left\{ \frac{\boldsymbol{Q}_1 \cdot \mathbf{n}}{\sigma_t} \right\}\!\!\right\} \mathrm{d}s - 2 \int_{\Gamma_b} u \, J_{\mathrm{in}} \,\mathrm{d}s \,, \tag{8.5}$$

where $D = \frac{1}{3\sigma_t}$ is the diffusion coefficient.

Next, we must determine the correction terms by computing the partial derivative with respect to the angular flux of each of the terms in the IP VEF discretization (Eq. 8.3). We set the diffusion solution $\mathbf{y}_0 = \begin{bmatrix} \psi_0 & \phi_0 \end{bmatrix}^T$. For terms without VEF data, the partial derivative is zero. Consider

$$\frac{\partial}{\partial \psi} \int_{\Gamma_b} E_b \, u\varphi \,\mathrm{d}s \bigg|_{\mathbf{y}_0} = \int_{\Gamma_b} \frac{\partial E_b}{\partial \psi}\bigg|_{\psi_0} u\varphi_0 \,\mathrm{d}s = \int_{\Gamma_b} u \, \beta(\psi) \,\mathrm{d}s \,, \tag{8.6}$$

where $\beta(\psi)$ is the correction factor defined in Eq. 3.17. We have used the directional derivative of the Eddington boundary factor computed in Eq. 3.60 to simplify the term $\frac{\partial \mathbf{E}}{\partial \psi}$. Here, $\frac{\partial \varphi}{\partial \psi} = 0$ since we consider ψ and φ as independent variables. Analogously, for terms with the Eddington tensor

$$\frac{\partial (\mathbf{E}\varphi)}{\partial \psi}\bigg|_{\mathbf{y}_0} = \frac{\partial \mathbf{E}}{\partial \psi}\bigg|_{\psi_0} \varphi_0 = \mathbf{T}(\psi) \,, \tag{8.7}$$

where the directional derivative of the Eddington tensor is given by Eq. 3.59 and $\mathbf{T}(\psi)$ is the correction tensor from Eq. 3.16. Thus, the correction terms can be derived by setting terms without angular flux dependence to zero and replacing

$$E_b \varphi \to \beta(\psi) \,, \quad \mathbf{E}\varphi \to \mathbf{T}(\psi) \,. \tag{8.8}$$

The IP SMM discretization is then: find $\varphi \in Y_p$ such that

$$\frac{1}{2}\int_{\Gamma_b} u\varphi \,\mathrm{d}s + \int_{\Gamma_0} \kappa \,[\![u]\!]\,[\![\varphi]\!]\,\mathrm{d}s - \int_{\Gamma_0} [\![u]\!]\,\{\!\{D\nabla_h\varphi\cdot\mathbf{n}\}\!\}\,\mathrm{d}s - \int_{\Gamma_0} \{\!\{D\nabla_h u\cdot\mathbf{n}\}\!\}\,[\![\varphi]\!]\,\mathrm{d}s \\ + \int \nabla_h u\cdot D\nabla_h\varphi\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} \\ = \int u\,Q_0\,\mathrm{d}\mathbf{x} + \int \nabla_h u\cdot\frac{\boldsymbol{Q}_1}{\sigma_t}\,\mathrm{d}\mathbf{x} - \int_{\Gamma_0}[\![u]\!]\left\{\!\!\left\{\frac{\boldsymbol{Q}_1\cdot\mathbf{n}}{\sigma_t}\right\}\!\!\right\}\mathrm{d}s - \int_{\Gamma_b} u\,(2J_{\mathrm{in}}+\beta)\,\mathrm{d}s \\ + \int_{\Gamma_0}[\![u]\!]\left\{\!\!\left\{\frac{1}{\sigma_t}\nabla_h\cdot\mathbf{T}\cdot\mathbf{n}\right\}\!\!\right\}\mathrm{d}s + \int_{\Gamma_0}\left\{\!\!\left\{\frac{\nabla_h u}{\sigma_t}\right\}\!\!\right\}\cdot[\![\mathbf{T}\mathbf{n}]\!]\,\mathrm{d}s - \int \nabla_h u\cdot\frac{1}{\sigma_t}\nabla_h\cdot\mathbf{T}\,\mathrm{d}\mathbf{x}\,, \quad \forall u\in Y_p\,, \tag{8.9}$$

where the local divergence of the correction tensor is computed with Eq. 5.43. This represents the standard IP discretization of diffusion with Marshak boundary conditions that is corrected by transport-dependent volumetric and boundary source terms.

8.1.2 Continuous Finite Element

As in Section 6.5, a continuous finite element discretization can be extracted from a DG method by setting $u, \varphi \in V_p$, where V_p is the degree-p continuous finite element space. For a continuous function $u \in Y_p$:

$$[\![u]\!] = 0\,, \quad \{\!\{u\}\!\} = u\,, \quad \forall\mathcal{F}\in\Gamma_0\,. \tag{8.10}$$

As before with the Eddington tensor, the correction tensor is generally discontinuous across interior mesh interfaces since we assume DG is used for the transport discretization. The CG discretization is then: find $\varphi \in V_p$ such that

$$\frac{1}{2}\int_{\Gamma_b} u\varphi\,\mathrm{d}s + \int \nabla u\cdot D\nabla\varphi\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} \\ = \int u\,Q_0\,\mathrm{d}\mathbf{x} + \int \nabla u\cdot\frac{\boldsymbol{Q}_1}{\sigma_t}\,\mathrm{d}\mathbf{x} - \int_{\Gamma_b} u\,(2J_{\mathrm{in}}+\beta)\,\mathrm{d}s \\ + \int_{\Gamma_0}\left\{\!\!\left\{\frac{\nabla u}{\sigma_t}\right\}\!\!\right\}\cdot[\![\mathbf{T}\mathbf{n}]\!]\,\mathrm{d}s - \int \nabla u\cdot\frac{1}{\sigma_t}\nabla_h\cdot\mathbf{T}\,\mathrm{d}\mathbf{x}\,, \quad \forall u\in V_p\,. \tag{8.11}$$

Alternatively, a CG SMM discretization can be derived by linearizing the CG VEF discretization in Eq. 6.48.

8.1.3 Raviart Thomas

From Eq. 7.14, a Raviart Thomas discretization of VEF is: find $(\varphi, \boldsymbol{J}) \in Y_p \times RT_p$ such that

$$\int u\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} = \int u\,Q_0\,\mathrm{d}\mathbf{x}\,, \quad \forall u\in Y_p\,, \tag{8.12a}$$

$$\int_{\Gamma_0} [\![\boldsymbol{v}\cdot\{\!\{\mathbf{En}\}\!\}]\!]\,\{\!\{\varphi\}\!\}\,\mathrm{d}s - \int \nabla_h \boldsymbol{v} : \mathbf{E}\varphi\,\mathrm{d}\mathbf{x} + \int \sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int_{\Gamma_b} \frac{1}{E_b}(\boldsymbol{v}\cdot\mathbf{En})(\boldsymbol{J}\cdot\mathbf{n})\,\mathrm{d}s$$
$$= \int \boldsymbol{v}\cdot\boldsymbol{Q}_1\,\mathrm{d}\mathbf{x} + 2\int_{\Gamma_b}\frac{1}{E_b}\boldsymbol{v}\cdot\mathbf{En}\,J_{\mathrm{in}}\,\mathrm{d}s\,, \quad \forall \boldsymbol{v}\in RT_p\,. \quad (8.12\mathrm{b})$$

The corresponding diffusion discretization is found by setting $\mathbf{E}\to\frac{1}{3}\mathbf{I}$ and $E_b \to E_{b0} = E_b(\psi_0)$:

$$\int u\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int \sigma_a\,u\varphi\,\mathrm{d}\mathbf{x} = \int u\,Q_0\,\mathrm{d}\mathbf{x}\,, \quad \forall u \in Y_p\,, \quad (8.13\mathrm{a})$$

$$-\frac{1}{3}\int \nabla\cdot\boldsymbol{v}\,\varphi\,\mathrm{d}\mathbf{x} + \int \sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int_{\Gamma_b}\frac{1}{3E_{b0}}(\boldsymbol{v}\cdot\mathbf{n})(\boldsymbol{J}\cdot\mathbf{n})\,\mathrm{d}s$$
$$= \int \boldsymbol{v}\cdot\boldsymbol{Q}_1\,\mathrm{d}\mathbf{x} + 2\int_{\Gamma_b}\frac{1}{3E_{b0}}\boldsymbol{v}\cdot\mathbf{n}\,J_{\mathrm{in}}\,\mathrm{d}s\,, \quad \forall \boldsymbol{v}\in RT_p\,, \quad (8.13\mathrm{b})$$

where the jump term $[\![\boldsymbol{v}\cdot\{\!\{\mathbf{En}\}\!\}]\!] = 0$ since $\boldsymbol{v}\in RT_p$ is continuous in the normal component. Here, we evaluate

$$E_{b0} = \frac{\int |\boldsymbol{\Omega}\cdot\mathbf{n}|\,\psi_0\,\mathrm{d}\Omega}{\int \psi_0\,\mathrm{d}\Omega} = \frac{\int |\boldsymbol{\Omega}\cdot\mathbf{n}|\,\mathrm{d}\Omega}{4\pi} \quad (8.14)$$

with angular quadrature since it was found to be important that J_{in} and $\int |\boldsymbol{\Omega}\cdot\mathbf{n}|\,\mathrm{d}\Omega$ are integrated with the same numerical quadrature rule. This is because both $\int |\boldsymbol{\Omega}\cdot\mathbf{n}|\,\mathrm{d}\Omega$ and

$$J_{\mathrm{in}} = \int_{\boldsymbol{\Omega}\cdot\mathbf{n}<0} \boldsymbol{\Omega}\cdot\mathbf{n}\,\psi\,\mathrm{d}\Omega = \int \mathbb{1}_{\boldsymbol{\Omega}\cdot\mathbf{n}<0}(\boldsymbol{\Omega})\,\boldsymbol{\Omega}\cdot\mathbf{n}\,\psi\,\mathrm{d}\Omega\,, \quad (8.15)$$

where $\mathbb{1}_A(x)$ is the indicator function that is one when $x\in A$ and zero otherwise, have non-smooth integrands and therefore cannot be integrated exactly with an angular quadrature rule.

The correction terms are found by computing the partial derivative of the VEF discretization with respect to the angular flux evaluated at the linearly anisotropic function $\mathbf{y}_0 = \begin{bmatrix}\psi_0 & \varphi_0 & \boldsymbol{J}_0\end{bmatrix}^T$ where $\boldsymbol{J}_0 = \int \boldsymbol{\Omega}\,\psi_0\,\mathrm{d}\Omega$. As before, terms without VEF data are zero and we can replace $E_b\varphi \to \beta(\psi)$ and $\mathbf{E}\varphi \to \mathbf{T}(\psi)$. However, this replacement is valid only for terms with *only* E_b or $\mathbf{E}$ and not both.

An additional linearization is required for the boundary terms due to the presence of both the Eddington tensor and boundary factor. This linearization is:

$$\frac{\partial}{\partial\psi}\int_{\Gamma_b}\frac{1}{E_b}(\boldsymbol{v}\cdot\mathbf{En})(\boldsymbol{J}\cdot\mathbf{n} - 2J_{\mathrm{in}})\,\mathrm{d}s\bigg|_{\mathbf{y}_0} = \int_{\Gamma_b}\boldsymbol{v}\cdot\frac{\partial(\mathbf{E}/E_b)}{\partial\psi}\bigg|_{\psi_0}\mathbf{n}\,(\boldsymbol{J}_0\cdot\mathbf{n} - 2J_{\mathrm{in}})\,\mathrm{d}s$$
$$= \int_{\Gamma_b}\boldsymbol{v}\cdot\left(\frac{1}{E_{b0}}\frac{\partial\mathbf{E}}{\partial\psi}\bigg|_{\psi_0} - \frac{\mathbf{E}_0}{E_{b0}^2}\frac{\partial E_b}{\partial\psi}\bigg|_{\psi_0}\right)\mathbf{n}(\boldsymbol{J}_0\cdot\mathbf{n} - 2J_{\mathrm{in}})\,\mathrm{d}s\,. \quad (8.16)$$

Since we have assumed that φ_0 and $\boldsymbol{J}_0$ satisfy the boundary conditions, we can substitute

$$\varphi_0 = \frac{1}{E_{b0}}\left(\boldsymbol{J}_0 \cdot \mathbf{n} - 2J_{\text{in}}\right) \tag{8.17}$$

to arrive at

$$\begin{aligned}
\frac{\partial}{\partial \psi}\int_{\Gamma_b} \frac{1}{E_b}(\boldsymbol{v}\cdot\mathbf{En})(\boldsymbol{J}\cdot\mathbf{n} - 2J_{\text{in}})\,\mathrm{d}s\bigg|_{\mathbf{y}_0} &= \int_{\Gamma_b} \boldsymbol{v}\cdot\left(\frac{\partial \mathbf{E}}{\partial\psi}\bigg|_{\psi_0} - \frac{\mathbf{E}_0}{E_{b0}}\frac{\partial E_b}{\partial \psi}\bigg|_{\psi_0}\right)\mathbf{n}\frac{1}{E_{b0}}(\boldsymbol{J}_0\cdot\mathbf{n} - 2J_{\text{in}})\,\mathrm{d}s \\
&= \int_{\Gamma_b} \boldsymbol{v}\cdot\left(\frac{\partial \mathbf{E}}{\partial\psi}\bigg|_{\psi_0} - \frac{\mathbf{E}_0}{E_{b0}}\frac{\partial E_b}{\partial \psi}\bigg|_{\psi_0}\right)\mathbf{n}\,\varphi_0\,\mathrm{d}s \\
&= \int_{\Gamma_b} \boldsymbol{v}\cdot\left(\mathbf{Tn} - \frac{\mathbf{E}_0\mathbf{n}}{E_{b0}}\beta\right)\mathrm{d}s \\
&= \int_{\Gamma_b} \boldsymbol{v}\cdot\left(\mathbf{Tn} - \frac{\beta\mathbf{n}}{3E_{b0}}\right)\mathrm{d}s\,,
\end{aligned} \tag{8.18}$$

where we have used the definition of the Miften-Larsen boundary condition (Eq. 3.13c), $\frac{\partial \mathbf{E}}{\partial \psi}|_{\psi_0}\varphi_0 = \mathbf{T}$, $\frac{\partial E_b}{\partial \psi}|_{\psi_0}\varphi_0 = \beta$, and $\mathbf{E}_0 = \frac{1}{3}\mathbf{I}$.

The Raviart Thomas SMM discretization is then: find $(\varphi, \boldsymbol{J}) \in Y_p \times RT_p$ such that

$$\int u\,\nabla\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi\,\mathrm{d}\mathbf{x} = \int u\,Q_0\,\mathrm{d}\mathbf{x}\,, \quad \forall u \in Y_p\,, \tag{8.19a}$$

$$\begin{aligned}
-\frac{1}{3}\int \nabla\cdot\boldsymbol{v}\,\varphi\,\mathrm{d}\mathbf{x} + \int \sigma_t\,\boldsymbol{v}\cdot\boldsymbol{J}\,\mathrm{d}\mathbf{x} + \int_{\Gamma_b}\frac{1}{3E_{b0}}(\boldsymbol{v}\cdot\mathbf{n})(\boldsymbol{J}\cdot\mathbf{n})\,\mathrm{d}s = \int \boldsymbol{v}\cdot\boldsymbol{Q}_1\,\mathrm{d}\mathbf{x} - \int_{\Gamma_b}\boldsymbol{v}\cdot\mathbf{Tn}\,\mathrm{d}s \\
+ \int_{\Gamma_b}\frac{1}{3E_{b0}}\boldsymbol{v}\cdot\mathbf{n}\,(2J_{\text{in}} + \beta)\,\mathrm{d}s - \int_{\Gamma_0} [\![\boldsymbol{v}]\!]\cdot\{\!\{\mathbf{Tn}\}\!\}\,\mathrm{d}s + \int \nabla_h\boldsymbol{v} : \mathbf{T}\,\mathrm{d}\mathbf{x}\,, \quad \forall \boldsymbol{v}\in RT_p\,.
\end{aligned} \tag{8.19b}$$

Note that the Minimal Residual Method (MINRES) solver requires the system to be symmetric indefinite. The RT SMM discretization can be written in symmetric indefinite form by by multiplying the zeroth moment by negative one and the first moment by positive three.

8.1.4 Hybridized Raviart Thomas

The left hand side of the RT SMM discretization is equivalent to an RT discretization of radiation diffusion. Thus, the standard RT hybridization process outlined in Section 7.4.1 can be applied directly to the RT SMM method. That is, the current is approximated in the discontinuous space $\hat{RT}_p$ and continuity of the current in the normal component is enforced with a Lagrange multiplier defined on the interior skeleton of the mesh, Γ_0. The use of a discontinuous approximation for the scalar flux and current allows eliminating these variables in favor of the Lagrange multiplier. This reduced system is both significantly smaller than the block system and can be effectively preconditioned with AMG.

The hybridized system is: find $(\boldsymbol{J}, \varphi, \lambda) \in \hat{RT}_p \times Y_p \times \Lambda_p$ such that

$$\int u\, \nabla_h \cdot \boldsymbol{J} \,\mathrm{d}\mathbf{x} + \int \sigma_a\, u\varphi \,\mathrm{d}\mathbf{x} = \int u\, Q_0 \,\mathrm{d}\mathbf{x}\,, \quad \forall u \in Y_p\,, \tag{8.20a}$$

$$-\frac{1}{3}\int \nabla_h \cdot \boldsymbol{v}\, \varphi \,\mathrm{d}\mathbf{x} + \int \sigma_t\, \boldsymbol{v} \cdot \boldsymbol{J} \,\mathrm{d}\mathbf{x} + \int_{\Gamma_b} \frac{1}{3E_{b0}} (\boldsymbol{v} \cdot \mathbf{n})(\boldsymbol{J} \cdot \mathbf{n}) \,\mathrm{d}s + \int_{\Gamma_0} [\![\boldsymbol{v} \cdot \mathbf{n}]\!]\, \lambda \,\mathrm{d}s = \mathcal{S}\,, \quad \forall \boldsymbol{v} \in \hat{RT}_p \tag{8.20b}$$

$$\int \mu\, [\![\boldsymbol{J} \cdot \mathbf{n}]\!] \,\mathrm{d}s = 0\,, \quad \forall \mu \in \Lambda_p\,, \tag{8.20c}$$

where

$$\mathcal{S} = \int \boldsymbol{v} \cdot \boldsymbol{Q}_1 \,\mathrm{d}\mathbf{x} - \int_{\Gamma_b} \boldsymbol{v} \cdot \mathbf{T}\mathbf{n} \,\mathrm{d}s + \int_{\Gamma_b} \frac{1}{3E_{b0}} \boldsymbol{v} \cdot \mathbf{n}\, (2J_{\mathrm{in}} + \beta) \,\mathrm{d}s - \int_{\Gamma_0} [\![\boldsymbol{v}]\!] \cdot \{\!\{ \mathbf{T}\mathbf{n} \}\!\} \,\mathrm{d}s + \int \nabla_h \boldsymbol{v} : \mathbf{T} \,\mathrm{d}\mathbf{x} \tag{8.21}$$

is the source term for the RT SMM discretization. This system can be efficiently solved by eliminating the scalar flux and current to form a reduced system for the Lagrange multiplier only. Once λ is known, the scalar flux and current can be solved in a post-processing step through element-local back substitution. Details of an efficient implementation are provided in Section 7.4.3 for the VEF system.

8.2 Results

The methods presented in this chapter were implemented in the MFEM finite element framework [90]. We use the conjugate gradient and BiCGStab solvers from MFEM along with *hypre*'s [94] BoomerAMG solver. KINSOL, from the Sundials [103] package, is used to solve the fixed-point problems with fixed-point iteration and Anderson-accelerated fixed-point iteration. The high-order DG S$_N$ solver from [15] was used. Unless otherwise noted, the angular flux and scalar flux are approximated with equal degree finite element spaces. We refer to the four methods derived in this section as the IP, CG, RT, and Hybridized Raviart Thomas (HRT) methods.

8.2.1 Method of Manufactured Solutions

The accuracy of the methods are determined with MMS. The solution is set to

$$\psi = \frac{1}{4\pi}\left[\alpha(\mathbf{x}) + \boldsymbol{\Omega} \cdot \boldsymbol{\beta}(\mathbf{x}) + \boldsymbol{\Omega} \otimes \boldsymbol{\Omega} : \boldsymbol{\Theta}(\mathbf{x})\right]\,, \tag{8.22}$$

where

$$\alpha(\mathbf{x}) = \sin(\pi x)\sin(\pi y) + \delta\,, \tag{8.23a}$$

$$\boldsymbol{\beta}(\mathbf{x}) = \begin{bmatrix} \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right) \sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \\ \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right) \sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \end{bmatrix}\,, \tag{8.23b}$$

$$\mathbf{\Theta}(\mathbf{x}) = \begin{bmatrix} \frac{1}{2}\sin\left(\frac{3\pi(x+\zeta)}{1+2\zeta}\right)\sin\left(\frac{3\pi(y+\zeta)}{1+2\zeta}\right) & \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) \\ \sin\left(\frac{2\pi(x+\omega)}{1+2\omega}\right)\sin\left(\frac{2\pi(y+\omega)}{1+2\omega}\right) & \frac{1}{4}\sin\left(\frac{3\pi(x+\zeta)}{1+2\zeta}\right)\sin\left(\frac{3\pi(y+\zeta)}{1+2\zeta}\right) \end{bmatrix} , \tag{8.23c}$$

where $\delta = 1.25$ is used to ensure $\psi > 0$ and $\zeta = 0.1$ and $\omega = 0.05$ are used to test spatially-dependent, non-isotropic inflow boundary conditions. The domain is $\mathcal{D} = [0,1]^2$. With this definition:

$$\phi(\mathbf{x}) = \alpha(\mathbf{x}) + \frac{1}{3}\operatorname{trace}\mathbf{\Theta}(\mathbf{x})\,, \tag{8.24a}$$

$$\boldsymbol{J}(\mathbf{x}) = \frac{1}{3}\boldsymbol{\beta}(\mathbf{x})\,, \tag{8.24b}$$

$$\mathbf{P}(\mathbf{x}) = \frac{\alpha(\mathbf{x})}{3}\mathbf{I} + \frac{1}{15}\begin{bmatrix} 3\Theta_{11}(\mathbf{x}) + \Theta_{22}(\mathbf{x}) & \Theta_{12}(\mathbf{x}) \\ \Theta_{21}(\mathbf{x}) & \Theta_{11}(\mathbf{x}) + 3\Theta_{22}(\mathbf{x}) \end{bmatrix} . \tag{8.24c}$$

This leads to an exact Eddington tensor $\mathbf{E} = \mathbf{P}/\phi$ that is dense and spatially varying. The MMS ψ and ϕ are substituted into the transport equation to solve for the MMS source q that forces the solution to Eq. 8.22.

The accuracy of the VEF discretizations are investigated in isolation by computing the VEF data from the MMS angular flux and setting the sources Q_0 and $\boldsymbol{Q}_1$ to the moments of the MMS source. This is accomplished by projecting the MMS angular flux onto a degree-p DG finite element space and using Level Symmetric S_4 angular quadrature to compute the VEF data, the moments of the MMS source, and the inflow boundary function. The VEF equations are then solved as if $\mathbf{E}$, E_b, Q_0, $\boldsymbol{Q}_1$, and J_{in} are given data. Errors are calculated with the $L^2(\mathcal{D})$ norm for scalars and the $[L^2(\mathcal{D})]^2$ norm for vectors given by

$$\|u\| = \sqrt{\int u^2 \,\mathrm{d}\mathbf{x}}\,, \tag{8.25}$$

and

$$\|\boldsymbol{v}\| = \sqrt{\int \boldsymbol{v}\cdot\boldsymbol{v}\,\mathrm{d}\mathbf{x}}\,, \tag{8.26}$$

respectively. We also use the $L^2(\mathcal{D})$ projection operator $\Pi_p : L^2(\mathcal{D}) \to Y_p$ such that

$$\int u(v - \Pi_p v)\,\mathrm{d}\mathbf{x} = 0\,, \quad \forall u \in Y_p\,, \tag{8.27}$$

for some $v \in L^2(\mathcal{D})$. In particular, Π_p is used to project the exact MMS scalar flux onto a Y_p finite element grid function in order to investigate a superconvergence property of mixed finite elements.

We use refinements of a third-order mesh created by distorting an orthogonal mesh according to the velocity field of the Taylor Green vortex. This mesh distortion is generated by advecting the mesh control points with

$$\mathbf{x} = \int_0^T \mathbf{v}\,\mathrm{d}t\,, \tag{8.28}$$

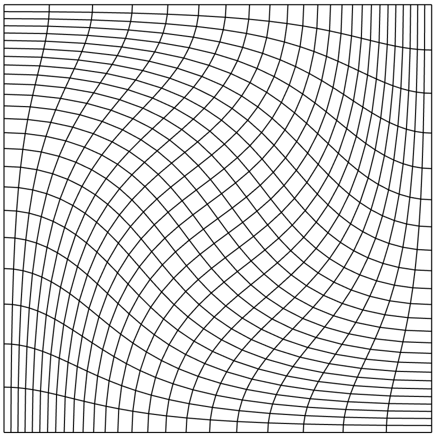


Figure 8.1: A depiction of a third-order mesh generated by distorting an orthogonal mesh according to the Taylor Green vortex. Refinements of this mesh are used in calculating the error with the method of manufactured solutions.

where the final time $T = 0.3\pi$ and

$$\mathbf{v} = \begin{bmatrix} \sin(x)\cos(y) \\ -\cos(x)\sin(y) \end{bmatrix} \tag{8.29}$$

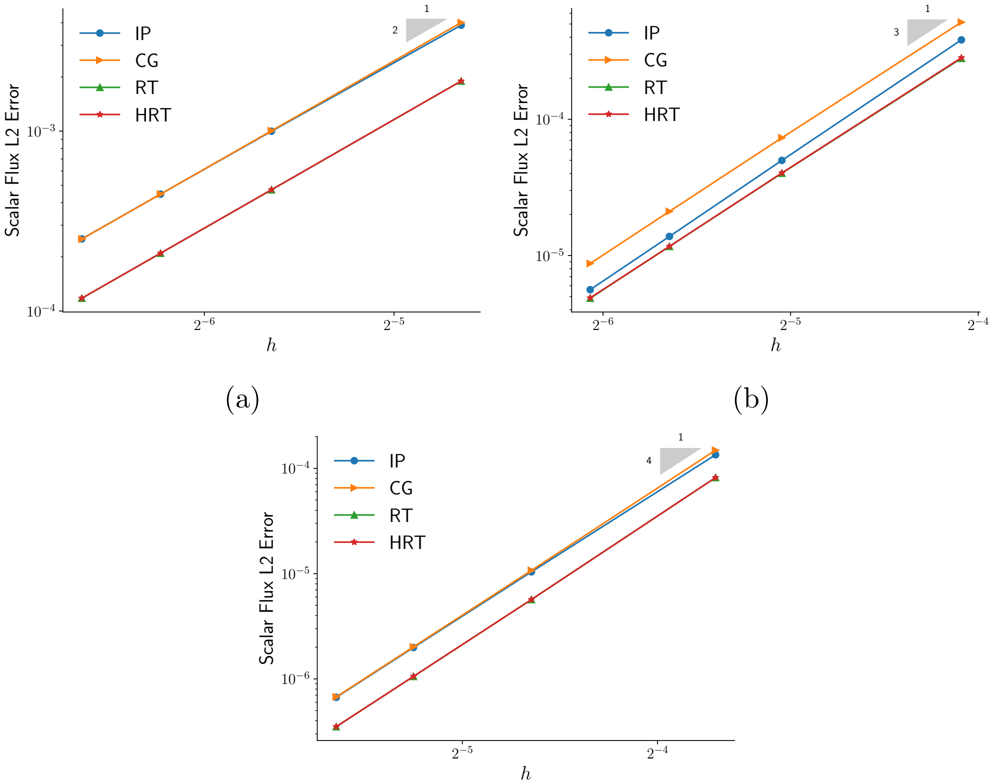
is the analytic solution of the Taylor Green vortex. 300 forward Euler steps were used to advect the mesh. An example mesh is shown in Fig. 8.1. Logarithmic regression is used to fit the constant and order of accuracy according to

$$E = Ch^{\tilde{p}} \tag{8.30}$$

where E is the error, C the constant, and $\tilde{p}$ the order of accuracy. Four values of h were used for each MMS problem considered in this section. The raw error values are provided in Appendix A.

Figure 8.2 shows the accuracy of the scalar flux on this MMS problem for a range of finite element polynomial orders. We see that all four methods have optimal $\mathcal{O}(h^{p+1})$ convergence for the scalar flux. As in the case for VEF, the IP and CG methods and RT and HRT methods have similar error behavior.

Next, we repeat the MMS problems from Section 7.5.1 used to probe the convergence of the current for the RT methods. Table 8.1 shows the error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto a Y_p grid function, and the error in the current on a diffusive MMS problem found by setting $\mathbf{\Theta} = 0$ in Eq. 8.22. It is observed that the two error values for the scalar flux converge with $\mathcal{O}(h^{p+1})$ and $\mathcal{O}(h^{p+2})$, respectively. In addition, the current converges with order $p + 1$. On a diffusive problem such as this one, the SMM and VEF discretizations are equivalent. This is verified by the equivalent convergence rates and constants compared to the RT VEF methods in Table 7.1.



(a)

(b)

(c)

Figure 8.2: Plots of the error in the scalar flux as the mesh is refined for (a) linear, (b) quadratic, and (c) cubic finite elements. A quadratically anisotropic MMS problem is used to ascertain the error.

Table 8.2 repeats this test on the quadratically anisotropic MMS solution given in Eq. 8.22 (i.e. with $\boldsymbol{\Theta}$ as defined in Eq. 8.23c). Since the MMS solution is projected onto Y_p, it is expected that this problem can converge with a maximum of order $p+1$. This can be seen in the loss of superconvergence property. Here, both measures of the scalar flux error converge with $\mathcal{O}(h^{p+1})$. On this transport problem, the current convergence is also degraded. Both RT and HRT converge the current with $\mathcal{O}(h^p)$ for p odd and $\mathcal{O}(h^{p+1/2})$ for p even. Compared to Table 7.2, the SMM methods are no longer equivalent to their VEF counterparts.

Finally, the MMS problem is repeated with the MMS angular flux projected onto the space Y_{p+1} allowing a maximum accuracy of $\mathcal{O}(h^{p+2})$. The error measures are provided in Table 8.3. Mixed finite element superconvergence is recovered as seen by $\|\varphi - \Pi\varphi_{\text{ex}}\|$ converging with order $p+2$ as in the diffusion case. In addition, the same behavior where the HRT method converges the current with order $p+1$ is observed with the RT method achieving $p+1$ only for p even.

Table 8.1: Estimates of the order of accuracy and constant from an isotropic MMS test problem. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the VEF data are constant in space and thus are represented exactly.

		$\|\varphi - \varphi_{\text{ex}}\|$		$\|\varphi - \Pi\varphi_{\text{ex}}\|$		$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$	
p	Value	RT	HRT	RT	HRT	RT	HRT
1	Order	2.000	2.000	3.053	3.053	2.000	2.000
	Constant	0.261	0.261	0.197	0.197	0.785	0.785
2	Order	3.003	3.003	4.096	4.096	2.989	2.989
	Constant	0.070	0.070	0.142	0.142	0.780	0.780
3	Order	4.016	4.016	5.125	5.125	4.016	4.016
	Constant	0.030	0.030	0.132	0.132	0.928	0.928
4	Order	4.971	4.971	5.964	5.964	4.675	4.675
	Constant	0.034	0.034	0.045	0.045	0.217	0.217

Table 8.2: Estimates of the order of accuracy and constant from a quadratically anisotropic MMS test problem. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the angular flux used to calculate the VEF data is represented with Y_p. Due to this, the maximum accuracy expected is order $p + 1$.

		$\|\varphi - \varphi_{\text{ex}}\|$		$\|\varphi - \Pi\varphi_{\text{ex}}\|$		$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$	
p	Value	RT	HRT	RT	HRT	RT	HRT
1	Order	2.005	2.005	2.280	2.283	0.979	0.980
	Constant	1.208	1.207	0.481	0.484	0.403	0.405
2	Order	2.954	2.958	2.941	2.991	2.521	2.527
	Constant	1.220	1.245	0.319	0.407	1.474	1.652
3	Order	4.047	4.045	4.270	4.230	2.895	2.906
	Constant	2.627	2.617	0.923	0.813	0.648	0.673
4	Order	4.779	4.776	4.814	4.756	4.240	4.244
	Constant	0.910	0.908	0.225	0.213	1.065	1.126

Table 8.3: Estimates of the order of accuracy and constant from a quadratically anisotropic MMS test problem. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the angular flux used to calculate the VEF data is represented with Y_{p+1}. Due to this, the maximum accuracy expected is order $p+2$.

		$\|\varphi - \varphi_{\text{ex}}\|$		$\|\varphi - \Pi\varphi_{\text{ex}}\|$		$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$	
p	Value	RT	HRT	RT	HRT	RT	HRT
0	Order	0.999	0.999	2.002	2.001	1.001	1.001
	Constant	0.780	0.780	1.338	1.304	0.517	0.516
1	Order	2.001	2.001	2.954	2.969	0.987	1.887
	Constant	1.179	1.178	1.488	1.390	0.083	0.583
2	Order	2.960	2.960	4.028	4.006	2.967	2.903
	Constant	1.204	1.204	2.447	2.347	1.273	0.783
3	Order	4.041	4.041	4.732	4.759	2.726	3.667
	Constant	2.545	2.545	0.896	0.864	0.102	0.575

8.2.2 Thick Diffusion Limit

The performance of the SMM algorithm is investigated in the thick diffusion limit. The material data are set to

$$\sigma_t = 1/\epsilon\,, \quad \sigma_a = \epsilon\,, \quad \sigma_s = 1/\epsilon - \epsilon\,, \quad q = \epsilon\,, \tag{8.31}$$

where $\epsilon \in (0,1]$ and the thick diffusion limit corresponds to $\epsilon \to 0$. We use two coarse meshes that do not resolve the mean free path to stress the convergence of the VEF method. The first is an orthogonal 8×8 mesh with $\mathcal{D} = [0,1]^2$. The second is the triple point mesh shown in Fig. 8.3, a third-order mesh generated with a Lagrangian hydrodynamics code where $\mathcal{D} = [0,7] \times [0,3]$. On the triple point mesh, the angular flux is only approximately inverted due to the lagging of reentrant faces and thus it is expected that convergence will degrade. In addition, highly distorted elements have poor approximation properties. We use Level Symmetric S_4 angular quadrature. The three methods are compared when $p = 2$. The coupled transport-VEF system is solved with fixed-point iteration.

Table 8.4 shows the number of fixed-point iterations until convergence to a tolerance of 10^{-6} for each method on the orthogonal and triple point meshes. Rapid convergence is seen for all methods on both problems. The IP and CG and RT and HRT methods converged equivalently on the orthogonal mesh and nearly equivalently on the triple point mesh (RT converged in one fewer iterations for $\epsilon = 10^{-2}$). Compared to VEF, SMM required 1-3 more iterations on the orthogonal mesh. Lineouts of the 2D VEF scalar flux solutions for each

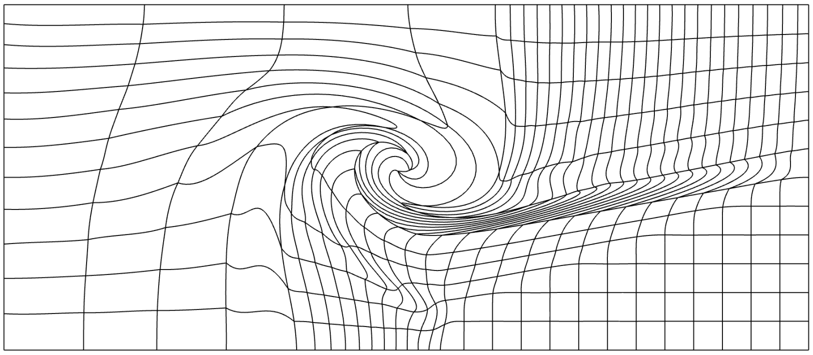


Figure 8.3: A depiction of the triple point mesh used to stress the VEF algorithms on a severely distorted, third-order mesh. This mesh was generated with a Lagrangian hydrodynamics simulation.

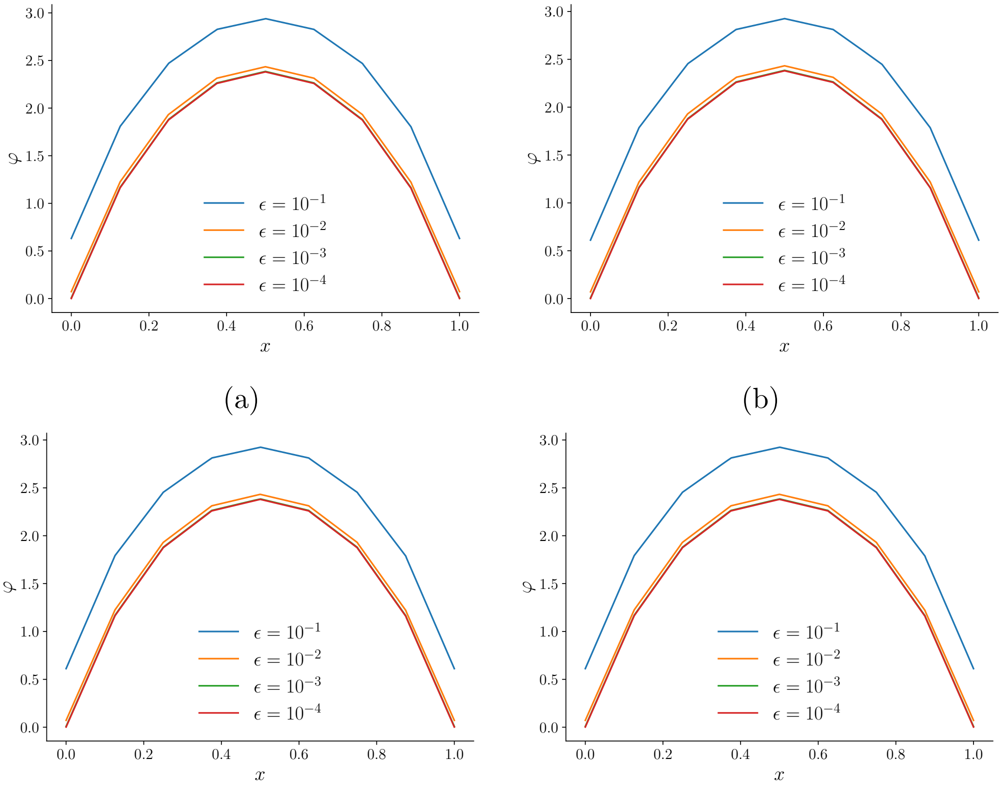
Table 8.4: The number of fixed-point iterations required for convergence as the thick diffusion limit parameter $\epsilon \to 0$. Convergence is tested on an orthogonal 8×8 mesh and on the triple point mesh, a mesh with re-entrant faces. Due to the re-entrant faces, a partial transport sweep is used making convergence slower on the triple point mesh.

	Orthogonal				Triple Point			
ϵ	IP	CG	RT	HRT	IP	CG	RT	HRT
10^{-1}	11	11	10	10	21	21	20	20
10^{-2}	7	7	7	7	11	11	15	16
10^{-3}	5	5	5	5	8	8	12	12
10^{-4}	5	4	4	4	6	6	8	8

method as $\epsilon \to 0$ are provided in Fig. 8.4 for the orthogonal mesh. In all cases, a non-trivial solution is found.

8.2.3 Crooked Pipe

We now show convergence in outer fixed-point iterations and inner preconditioned linear solver iterations on a more realistic, multi-material problem. The geometry and materials are shown in Fig. 8.5. The problem consists of two materials, the wall and the pipe, which have an 1000x difference in total interaction cross section. Time dependence is mocked by including artificial absorption and sources that correspond to backward Euler time integration. The time step is set so that $c\Delta t = 10^3$ and the initial condition is $\psi_0 = 10^{-4}$. The absorption and source are then $\sigma_a = 1/c\Delta t = 10^{-3}\frac{1}{\text{cm}}$ and $q = \psi_0/c\Delta t = 10^{-1}\frac{1}{\text{cm}^3\cdot\text{s}\cdot\text{str}}$. The boundary conditions are set so that isotropic inflow of magnitude $1/2\pi$ enters on the left entrance of the



(a)

(b)

(c)

(d)

Figure 8.4: Lineouts of the 2D solution as $\epsilon \to 0$. The methods all converge to the asymptotic solution indicating they preserve the thick diffusion limit.

pipe. A Level Symmetric S_{12} angular quadrature set is used. The quadratic programming negative flux fixup from [43] is used inside the transport sweep to ensure positivity.

The outer fixed-point and inner linear iterative efficiency are shown by refining in h and p on an orthogonal mesh. Anderson acceleration with two Anderson vectors is used. The previous outer iteration's solution is used as an initial guess for the inner solver so that the initial guess becomes progressively more accurate as the outer iteration converges. The outer tolerance is 10^{-6} and the inner tolerance is 10^{-8}. The IP, CG, and HRT methods used the conjugate gradient solver. CG and HRT are preconditioned with one AMG V-cycle. The IP method is preconditioned by the USC preconditioner which uses one AMG V-cycle and one iteration of Jacobi smoothing. Finally, the RT method uses BiCGStab with a lower block triangular preconditioner that performs one iteration of Gauss-Seidel smoothing and one AMG V-cycle per iteration.

Table 8.5 shows the number of Anderson-accelerated fixed-point iterations to convergence and the maximum, minimum, and average number of inner iterations performed across all

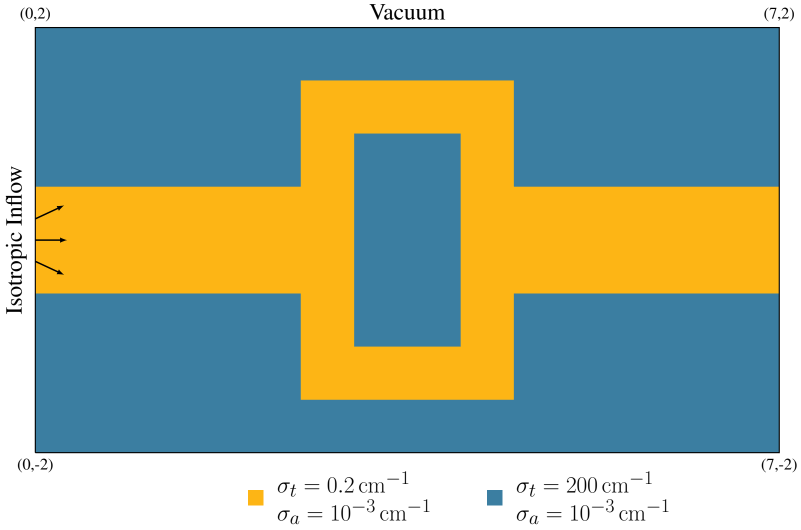

(0,2)
Vacuum
(7,2)
Isotropic Inflow
(0,-2)
(7,-2)
$\sigma_t = 0.2\,\mathrm{cm}^{-1}$
$\sigma_a = 10^{-3}\,\mathrm{cm}^{-1}$
$\sigma_t = 200\,\mathrm{cm}^{-1}$
$\sigma_a = 10^{-3}\,\mathrm{cm}^{-1}$

Figure 8.5: The geometry, material data, and boundary conditions for the linearized crooked pipe problem.

the outer iterations for each of the methods. The IP and CG methods and RT and HRT methods converged within one iteration of each other. The inner solvers were all scalable in both the mesh size and the polynomial order.

8.2.4 Weak Scaling

Finally, we show that the methods scale in parallel. The parallel partitioning is such that there are $\approx 9\,000$ VEF scalar flux unknowns per processor. The results were generated on 32 nodes of the `rztopaz` machine at LLNL which has two 18-core Intel Xeon E5-2695 CPUs per node. The materials and geometry from the crooked pipe in Section 8.2.3 are used. In Table 8.6, the number of linear iterations until convergence to a tolerance of 10^{-8} is shown when 1) a transport solve is used to compute the SMM correction sources and 2) the SMM correction sources are set to zero. The solvers are the same as those in Section 8.2.3. We have used $p = 2$. All the methods are scalable out to over a million elements and 1152 processors. Interestingly, the number of iterations for solving the SMM equations is increased compared to solving the corresponding diffusion problem. This suggests that the SMM correction sources make the inner solve more difficult.

Table 8.5: The number of outer Anderson-accelerated fixed-point iterations until convergence along with the maximum, minimum, and average numbers of inner linear iterations until convergence on the linearized crooked pipe problem. Two Anderson vectors were used. The previous outer iteration's solution was used as the initial guess for the inner iteration.

		Outer				Max Inner				Min Inner				Avg. Inner			
	N_e	IP	CG	RT	HRT	IP	CG	RT	HRT	IP	CG	RT	HRT	IP	CG	RT	HRT
$p = 1$	112	11	11	13	13	26	10	11	12	6	3	3	4	16.73	7.18	7.69	8.08
	448	12	12	14	14	26	12	13	13	8	4	5	4	17.25	8.25	9.07	9.00
	1792	14	14	15	15	27	14	14	13	7	4	5	5	17.43	9.00	9.07	9.07
	7168	15	14	17	17	27	14	14	14	7	5	5	4	17.40	9.29	9.00	9.06
$p = 2$	112	14	15	15	15	26	15	21	17	9	4	6	5	17.50	9.73	13.20	11.27
	448	15	15	16	16	27	16	22	19	7	4	7	5	17.33	10.40	14.56	11.69
	1792	16	16	17	17	27	16	25	21	7	4	7	6	17.06	9.81	15.59	13.47
	7168	16	16	18	18	29	17	26	23	8	5	8	7	18.38	10.81	15.56	14.17
$p = 3$	112	16	15	15	15	34	19	20	17	8	6	6	5	22.12	12.73	12.67	11.20
	448	15	15	16	17	36	20	20	17	8	5	8	5	23.67	13.40	13.12	10.88
	1792	18	18	18	18	38	21	21	17	11	6	6	5	23.67	13.33	13.28	10.89
	7168	20	20	18	18	38	22	22	20	11	6	6	5	23.20	13.45	13.39	12.06

Table 8.6: A weak scaling study on the first iteration of the linearized crooked pipe problem. Inner linear iteration counts are compared when a parallel block Jacobi sweep is used to compute the SMM correction sources (SMM) and when the SMM sources are set to mock a radiation diffusion problem (Diffusion).

		IP		CG		RT		HRT	
Processors	N_e	SMM	Diff.	SMM	Diff.	SMM	Diff.	SMM	Diff.
36	36 288	35	30	22	18	31	27	24	20
72	70 000	34	29	21	18	33	29	25	21
144	145 152	35	30	22	18	31	28	26	21
288	285 628	36	29	22	18	31	31	26	21
576	580 608	37	29	22	18	32	29	26	22
864	867 328	36	29	22	18	31	29	26	21
1152	1 153 852	38	30	23	18	35	29	27	22

Chapter 9

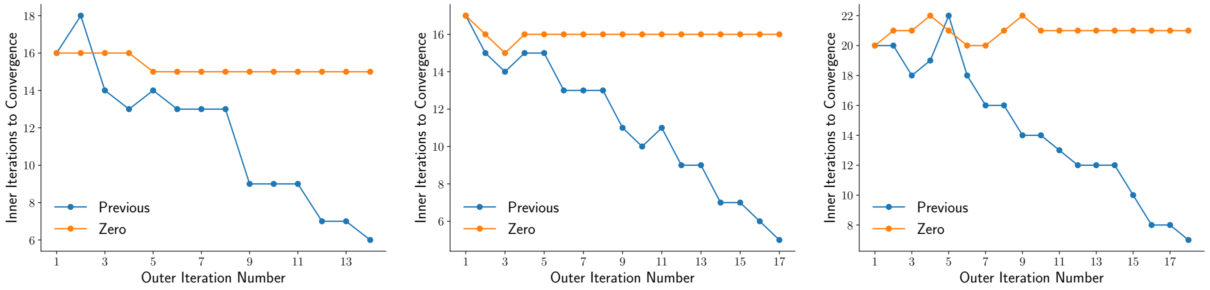
Additional Results and Discussion

The purpose of this chapter is to provide additional results that apply to all of the moment methods presented here and to compare the methods developed in previous chapters. In particular, we use the IP VEF method to demonstrate an effect of the choice of the initial guess for the inner preconditioned iterative solver and investigate iterative convergence on a mesh with reentrant faces. We then concatenate the results from Chapters 6, 7, and 8 into common tables and discuss their relative performance on the benchmarks used in this dissertation.

9.1 Previous Outer Iteration as Initial Guess for Inner Solver

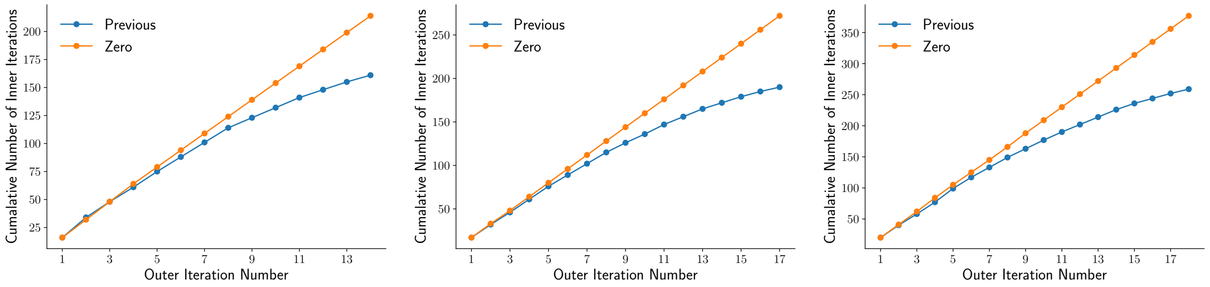
Here, we discuss the use of the previous fixed-point iteration's solution as the initial guess for the inner linear solve. As the outer iteration converges, it will provide a better and better initial guess. This approach is compared to a simple initial guess of zero for each inner solve. This comparison is performed using the interior penalty VEF method applied to the crooked pipe problem discussed in Section 6.7.3. The inner solver was BiCGStab with a tolerance of 10^{-8}. The outer solver used fixed-point iteration with two Anderson vectors. The outer tolerance was 10^{-6}.

Figure 9.1 shows the number of inner iterations required for convergence at each outer iteration for each of the initial guess strategies. These results were generated for each polynomial order using the crooked pipe mesh with 7168 elements. Aside from two cases, the non-zero initial guess required the same or fewer iterations to converge as the solve that used an initial guess of zero. In addition, as the outer solve progresses the number of inner iterations to convergence decreases. The cumulative number of inner iterations as the outer iteration progresses is shown in Fig. 9.2. Using the previous outer iteration as an initial guess led to a reduction in the total number of inner iterations required to solve the problem of 25%, 30%, and 32% for $p = 1$, $p = 2$, and $p = 3$, respectively, when compared to using a zero initial guess for each inner solve.



(a) (b) (c)

Figure 9.1: A comparison of the number of inner iterations to convergence at each outer iteration when the inner solver used the previous outer iteration as the initial guess and when an initial guess of zero was used for each iteration. The most refined mesh for polynomials orders (a) 1, (b), 2, and (c) 3 are shown. Using the previous outer iteration as an initial guess reduces the total number of inner iterations required to solve the problem.



(a) (b) (c)

Figure 9.2: A comparison of the cumulative number of inner iterations to convergence as the outer iteration progresses when the inner solver used the previous outer iteration as the initial guess and when an initial guess of zero was used for each iteration. The most refined mesh for polynomials orders (a) 1, (b), 2, and (c) 3 are shown. Using the previous outer iteration as the initial guess reduced the total number of inner iterations to solve the problem by 53, 82, and 118, respectively. This represents a reduction in inner iterations of 25%, 30%, and 31%, respectively.

Table 9.1: The number of Anderson-accelerated fixed-point iterations and the maximum, minimum, and total number of inner iterations performed across all outer iterations for the IP VEF method on the crooked pipe problem refined in h and p. An Anderson space of size two is used. The effect of using the previous outer iteration's solution as an initial guess for the inner solver is compared to using an initial guess of zero at each inner iteration.

		Outer		Max Inner		Min Inner		Total Inner	
	N_e	Previous	Zero	Previous	Zero	Previous	Zero	Previous	Zero
$p = 1$	112	10	10	16	17	6	15	115	166
	448	11	11	17	17	7	16	132	186
	1792	13	13	18	17	4	15	146	215
	7168	14	14	18	16	6	15	161	214
$p = 2$	112	13	13	16	16	5	14	139	203
	448	15	15	17	17	5	14	168	226
	1792	16	16	17	17	4	14	178	243
	7168	17	17	17	17	5	15	190	272
$p = 3$	112	15	15	19	20	5	17	187	284
	448	16	16	22	22	7	19	224	347
	1792	17	17	22	22	6	20	239	369
	7168	18	18	22	22	7	20	259	377

The crooked pipe hp scaling is repeated in Table 9.1. Here, it is seen that the initial guess scheme for the inner iteration does not affect the convergence of the outer iteration. The maximum number of inner iterations across all the outer iterations were within two iterations of each other for both initial guess schemes. However, the minimum number of iterations was typically 30%-50% lower when the previous outer iteration's solution was used as an initial guess. This behavior led to fewer total number of inner iterations required to solve the problem.

9.2 Acceleration of Inexact Sweeps on the Triple Point Mesh

On a high-order mesh with reentrant faces, the transport equation is only approximately inverted at each outer iteration (see the discussion in Section 5.4). This means the VEF data are computed from an angular flux possessing errors due to lagging the inflow data on reentrant faces. While VEF converged robustly in the thick diffusion limit on a mesh with reentrant faces, degraded performance for optically thin problems was observed as compared

Table 9.2: The number of fixed-point iterations to convergence on the triple point mesh in the thick diffusion limit. On the triple point mesh, the transport equation is not inverted exactly at each iteration due to the presence of re-entrant faces. The performance of the IP VEF method with $p = 2$ is compared when 1, 2, and 3 partial inversions of the transport equation are performed at each fixed-point iteration. More transport inversions leads to faster convergence but not to the point that fewer total inversions are performed.

	Outer Its.			Total Sweeps		
ϵ	1	2	3	1	2	3
10^{-1}	19	11	10	19	22	30
10^{-2}	11	8	7	11	16	21
10^{-3}	8	5	4	8	10	12
10^{-4}	6	4	3	6	8	9

to an equivalent mesh without reentrant faces. Here, we demonstrate two schemes to produce more robust behavior for optically thin problems on curved meshes. The thick diffusion limit problem scales the material data according to

$$\sigma_t = \frac{1}{\epsilon}\,, \quad \sigma_s = \frac{1}{\epsilon} - \epsilon\,, \quad q = \epsilon\,. \tag{9.1}$$

The interior penalty VEF method with $p = 2$ is used.

First, we show the effect of using multiple partial transport inversions per outer iteration. That is, at each outer iteration, the approximate inversion of the streaming and collision operator is iterated in order to produce an angular flux that is closer to the angular flux computed with a direct method. Table 9.2 shows the number of outer iterations required to solve the problem to a tolerance of 10^{-6} when one, two, and three partial inversions of the transport problem are performed as $\epsilon \to 0$. For each value of ϵ, performing more partial sweeps per iteration reduces the total number of iterations to convergence. In fact, for three sweeps, the iterative performance is within two iterations of an equivalent problem on an orthogonal mesh (e.g. Table 6.3). However, the iterative performance is not improved to the point that fewer total sweeps are performed. While the three sweep option requires the fewest outer iterations, it requires the most sweeps. This suggests that improving robustness to reentrant faces through increasing the number of sweeps per outer iteration comes at a significant cost. A compromise between these two ideas would be to solve the streaming and collision operators to a certain tolerance. This would provide more robustness while potentially reducing the total number of sweeps.

Next, we discuss application of Anderson acceleration to accelerate the outer iteration on a high-order mesh. We compare building the Anderson space from previous iterates of the scalar flux only and from the scalar and angular flux. These are referred to as the "low memory" and "augmented" options. Note that we store the entire angular flux in the

Table 9.3: The number of Anderson-accelerated fixed-point iterations to solve the thick diffusion limit problem on the triple point mesh. Fixed-point iteration is compared to Anderson-accelerated fixed-point iteration with an Anderson space of five scalar flux solution vectors (Low Memory) and five scalar and angular flux solution vectors (Augmented). The slowdown of the Low Memory option indicates Anderson cannot accelerate the slowdown from inexact sweeps when the angular flux is not included in the Anderson space.

ϵ	Fixed Point	Low Memory	Augmented
10^{-1}	18	18	15
10^{-2}	11	13	11
10^{-3}	8	12	8
10^{-4}	6	6	6

augmented space for implementational ease only. It is possible to store only the degrees of freedom corresponding to reentrant faces. This would be a nonlinear analog of the ideas used for Krylov-accelerated source iteration [62]. Fixed-point iteration is compared to the two Anderson acceleration schemes in the thick diffusion limit on the triple point mesh in Table 9.3. The low memory option takes the same or more iterations as fixed-point iteration. This suggests that Anderson applied to the scalar flux only cannot accelerate inexact sweeps due to reentrant faces. The augmented Anderson space resulted in identical convergence for $\epsilon = 10^{-2}$, 10^{-3}, and 10^{-4}. For $\epsilon = 10^{-1}$, the augmented Anderson scheme converged three iterations faster than fixed-point iteration. This suggests that Anderson acceleration can increase iterative efficiency on thin problems but only when angular flux information is included in the Anderson space.

9.3 Comparison of Methods

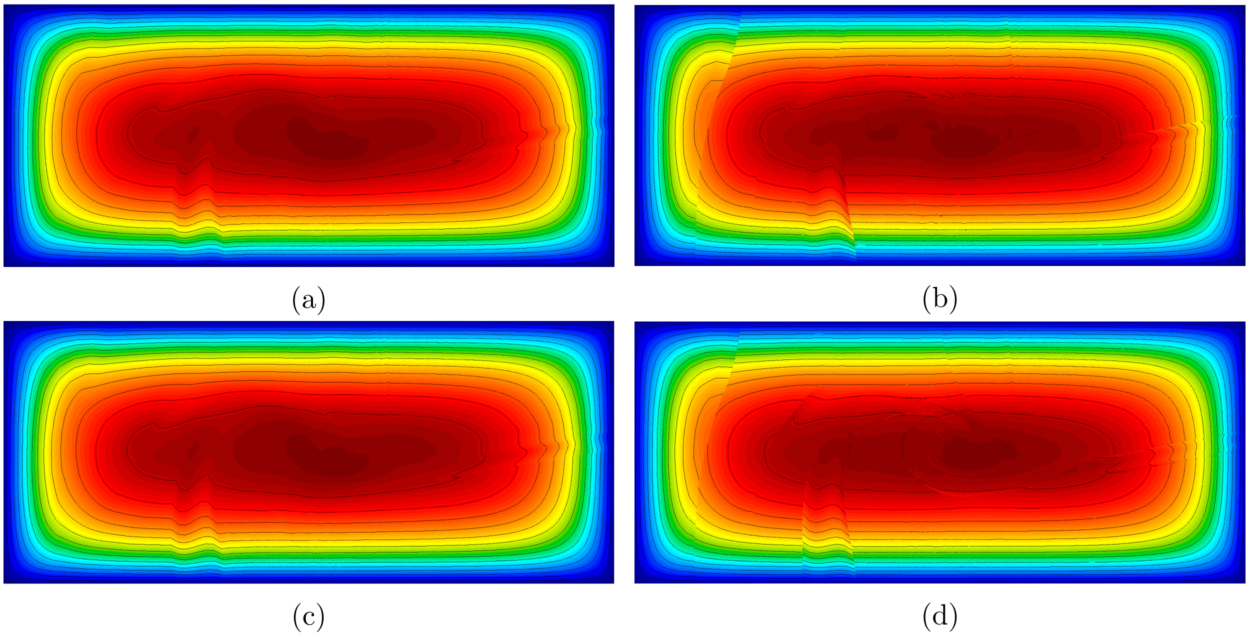
Here, we attempt to provide a coherent, unified discussion of the results presented in Chapters 6, 7, and 8. Since the H1 method from Chapter 7 could not be efficiently solved, we do not consider it in the discussions presented here.

9.3.1 Solution Quality on the Triple Point Mesh

Here, we compare the sensitivity to mesh distortion of the VEF methods presented in Chapters 6 and 7. The thick diffusion limit problem such that

$$\sigma_t = \frac{1}{\epsilon}\,, \quad \sigma_s = \frac{1}{\epsilon} - \epsilon\,, \quad q = \epsilon\,. \tag{9.2}$$

with $\epsilon = 10^{-2}$ is used to test solution quality on the triple point mesh. This problem should have a monotonically increasing solution with smooth contours. Deviations from this



(a)

(b)

(c)

(d)

Figure 9.3: Plots of the scalar flux from the thick diffusion limit problem with $\epsilon = 10^{-2}$ on the triple point mesh for the (a) interior penalty, (b) MDLDG, (c) CG, and (d) HRT VEF methods. The solution should be smoothly varying and obey a maximum principle. Deviations from this behavior are due to mesh imprinting. These plots show that the IP and CG methods have nearly identical solution quality and in particular have continuous contour lines. The MDLDG and HRT methods produced solutions with discontinuous contour lines.

behavior are due to mesh imprinting from the poor approximation properties of severely distorted elements. The scalar flux solutions for the IP, MDLDG, CG, and HRT VEF methods with $p = 2$ are compared in Fig. 9.3. Observe that all methods show non-monotonic behavior suggesting all of the methods are sensitive to mesh imprinting. It is interesting to note which methods exhibit continuous contour lines for the solution. The IP method, a DG method that is stabilized through the use of a penalty term, appears to have very similar solution quality to the CG method which solves for the scalar flux in a continuous finite element space. This indicates the effect of the penalty parameter regularizing toward continuous solutions. By contrast, the MDLDG method, a DG method where a penalty term is not used, has discontinuous contours. The HRT method appears to be the most sensitive to mesh distortion as the “swirl” in the center of the domain appears the most pronounced. This may be due to the first-order form’s weaker solution requirements. In addition, the gradient of the Piola transform required for the Raviart Thomas methods may allow mesh distortion to affect the solution more. This comparison may indicate that the IP and CG methods are more numerically diffusive than the MDLDG and HRT methods.

Table 9.4: The number of iterations until convergence in the thick diffusion limit for all the methods presented in this dissertation for $p = 2$. An 8×8 orthogonal mesh is used with S_4 angular quadrature. The fixed-point tolerance was 10^{-6}.

	VEF						SMM			
ϵ	IP	BR2	MDLDG	CG	RT	HRT	IP	CG	RT	HRT
10^{-1}	8	8	8	8	8	8	11	11	10	10
10^{-2}	6	6	6	6	6	6	7	7	7	7
10^{-3}	4	4	4	4	4	4	5	5	5	5
10^{-4}	3	3	3	3	3	3	5	4	4	4

9.3.2 Thick Diffusion Limit

Table 9.4 shows the number of fixed-point iterations to converge the VEF and SMM algorithms in the thick diffusion limit. This problem is characterized by

$$\sigma_t = \frac{1}{\epsilon}\,, \quad \sigma_s = \frac{1}{\epsilon} - \epsilon\,, \quad q = \epsilon\,. \tag{9.3}$$

with the thick diffusion limit corresponding to $\epsilon \to 0$. A coarse 8×8 mesh is used with S_4 angular quadrature. The fixed-point tolerance was 10^{-6}. Observe that all the VEF methods converged equivalently for each value of ϵ tested. The SMM algorithms converged within one iteration of each other. SMM converged 1-3 iterations slower than VEF. On this idealized problem where negativities are unlikely to occur all methods performed very similarly.

The thick diffusion limit problem was repeated on the triple point mesh in Table 9.5. Here, the presence of reentrant faces prevents the ability to solve the transport equation exactly with an element-by-element sweep. The transport equation is instead solved with an approximate sweep that lags the incoming angular flux data corresponding to reentrant faces. This means the transport equation is not exactly inverted at each fixed-point iteration, slowing convergence of the VEF and SMM algorithms as compared to the orthogonal mesh problem. In addition, the severely distorted elements have poor approximation ability. This problem represents a difficult stress test. These results indicate that all VEF and SMM methods are robust to reentrant faces. However, it is seen that the MDLDG, RT, and HRT VEF methods are slower to converge compared to the IP, BR2, and CG VEF methods. This is likely due to an increase in negativities for the methods that are less numerically diffusive. A similar behavior is seen for the RT and HRT SMMs as compared to the IP and CG SMMs. Again, VEF generally converged a few iterations faster than SMM.

9.3.3 Crooked Pipe

Next, we compare performance on the multi-material crooked pipe problem. This problem is defined in Sections 6.7.3, 7.5.4, and 8.2.3. The problem consists of two materials with

Table 9.5: The number of iterations until convergence in the thick diffusion limit for all the methods presented in this dissertation for $p = 2$. The triple point mesh is used with S_4 angular quadrature. The fixed-point tolerance was 10^{-6}.

	VEF						SMM			
ϵ	IP	BR2	MDLDG	CG	RT	HRT	IP	CG	RT	HRT
10^{-1}	19	19	23	19	21	21	21	21	20	20
10^{-2}	11	11	19	11	19	19	11	11	15	16
10^{-3}	8	8	9	8	13	13	8	8	12	12
10^{-4}	6	6	6	6	8	8	6	6	8	8

Table 9.6: The number of Anderson-accelerated fixed-point iterations until convergence on the crooked pipe problem for all the methods presented in this dissertation. An Anderson space of size two was used. The iterative tolerance was 10^{-6}.

		VEF						SMM			
	N_e	IP	BR2	MDLDG	CG	RT	HRT	IP	CG	RT	HRT
$p = 1$	112	10	10	14	10	13	13	11	11	13	13
	448	11	11	16	12	13	13	12	12	14	14
	1792	13	13	16	13	16	16	14	14	15	15
	7168	14	14	18	14	16	16	15	14	17	17
$p = 2$	112	13	13	16	13	15	15	14	15	15	15
	448	15	15	18	15	16	16	15	15	16	16
	1792	16	16	18	16	18	17	16	16	17	17
	7168	17	17	19	17	17	17	16	16	18	18
$p = 3$	112	15	15	17	15	16	16	16	15	15	15
	448	16	16	18	16	17	17	15	15	16	17
	1792	17	17	19	17	17	17	18	18	18	18
	7168	18	18	19	18	19	19	20	20	18	18

Table 9.7: The average number of inner iterations until convergence across all the outer iterations on the crooked pipe problem.

		VEF						SMM			
	N_e	IP	BR2	MDLDG	CG	RT	HRT	IP	CG	RT	HRT
$p = 1$	112	11.50	11.20	6.43	4.90	6.54	4.54	16.73	7.18	7.69	8.08
	448	12.00	11.18	6.94	4.75	8.31	5.31	17.25	8.25	9.07	9.00
	1792	11.23	11.00	7.38	4.85	8.12	5.19	17.43	9.00	9.07	9.07
	7168	11.50	11.21	6.94	5.00	8.50	5.50	17.40	9.29	9.00	9.06
$p = 2$	112	10.69	10.85	7.62	6.23	13.53	6.53	17.50	9.73	13.20	11.27
	448	11.20	10.80	7.94	6.07	14.19	7.00	17.33	10.40	14.56	11.69
	1792	11.12	11.12	9.11	6.44	14.22	7.53	17.06	9.81	15.59	13.47
	7168	11.18	11.35	9.32	6.41	15.59	8.18	18.38	10.81	15.56	14.17
$p = 3$	112	12.47	12.20	12.29	7.53	11.69	6.12	22.12	12.73	12.67	11.20
	448	14.00	13.19	11.61	8.56	12.23	6.24	23.67	13.40	13.12	10.88
	1792	14.06	14.35	12.58	8.71	13.41	6.18	23.67	13.33	13.28	10.89
	7168	14.39	14.17	12.95	9.11	13.21	6.26	23.20	13.45	13.39	12.06

an 1000x difference in total interaction cross section and corresponds to the first Newton iteration of a time-dependent TRT algorithm. A comparison of all the VEF and SMM methods presented here is provided in Table 9.6. Anderson-accelerated fixed-point iteration is used with a tolerance of 10^{-6}. Two Anderson vectors are used. The zero and scale negative flux fixup was used for the DG VEF methods. The mixed finite element VEF methods and all the SMMs used the quadratic programming negative flux fixup. We again see that the less numerically diffusive methods required 1-3 more iterations to converge than the numerically diffusive methods. This is true for both the VEF and SMM methods. All methods show a slight increase in iterations as the polynomial order is refined. This is likely due to the increased reliance on the negative flux fixup for higher polynomial orders. Even on this difficult problem, SMM converged only slightly slower than the corresponding VEF method.

Table 9.7 shows the average number of inner preconditioned linear solver iterations until a convergence of 10^{-8} across all the outer iterations on the crooked pipe problem. All of the VEF methods and the RT SMM used BiCGStab. The IP, CG, and HRT SMMs used the conjugate gradient solver. Methods that used BiCGStab are twice as expensive per iteration compared to methods that used conjugate gradient. Here, we attempt only to show that all of the methods can be scalably solved with respect to both the mesh size and the polynomial order. A comparison of performance would require timing data.

Table 9.8: A weak scaling study of the inner solve on the first iteration of the crooked pipe for all the VEF methods and SMMs presented in this dissertation. The inner solver tolerance was 10^{-8}.

		VEF				SMM			
Processors	N_e	IP	CG	RT	HRT	IP	CG	RT	HRT
36	36 288	20	14	29	14	35	22	31	24
72	70 000	22	13	34	15	34	21	33	25
144	145 152	23	15	31	15	35	22	31	26
288	285 628	23	17	33	16	36	22	31	26
576	580 608	24	15	32	15	37	22	32	26
864	867 328	26	17	32	18	36	22	31	26
1152	1 153 852	26	16	33	16	38	23	35	27

9.3.4 Weak Scaling

Table 9.8 summarizes the weak scaling performance of all the methods presented here. The data is taken from Sections 6.7.4, 7.5.6, and 8.2.4. The first iteration of the crooked pipe problem is used to demonstrate the convergence of the inner solve. The tolerance for the preconditioned linear solvers was 10^{-8}. The VEF methods used BiCGStab while the IP, CG, and HRT SMMs used conjugate gradient. The RT SMM used BiCGStab. Both IP methods used the uniform subspace correction preconditioner with one Jacobi iteration and one AMG V-cycle per iteration. The CG and HRT VEF methods and SMMs used one AMG V-cycle as a preconditioner. Both RT methods used a lower block triangular preconditioner that used one iteration of Gauss-Seidel to approximate the inverse of the total interaction mass matrix and one AMG V-cycle to approximate the inverse of the lumped Schur complement.

Observe that all methods can be solved efficiently in parallel on a mesh of over a million elements. For both VEF and SMM, the CG and HRT solvers converged similarly. The RT and IP methods generally required the most iterations. Comparing VEF to SMM, it is seen that the SMM solvers require more iterations. This is due to the use of conjugate gradient instead of BiCGStab for the IP, CG, and HRT methods. For the case of RT VEF and SMM where BiCGStab was used for both types, the iteration counts are very close with both methods converging within ±2 iterations of each other.

9.3.5 A Qualitative Discussion of Cost

The primary costs in each iteration of a VEF or SMM algorithm are: the transport sweep, computing the closures for the moment system, and forming and solving the moment system. All methods share the costs of the transport sweep and computing the closures. Furthermore, computation of the VEF and SMM data have a similar cost. For example, computing the Eddington tensor requires using angular quadrature to compute the second and zeroth moment of the discrete angular flux and forming their ratio. The SMM correction tensor also computes the second and zeroth moment of the discrete angular flux but instead subtracts them. Thus, the methods are differentiated by the cost of forming and inverting the system.

SMMs have a distinct advantage over VEF in the cost of forming their associated moment systems. Since the left hand side of the SMM system is simply radiation diffusion, it is independent of the angular flux and thus does not change from iteration to iteration. This means the left hand side matrix can be constructed once and reused as the iteration progresses. On the other hand, the left hand side of the VEF moment system is dependent on the angular flux and therefore must be updated at each iteration. Since it is much cheaper to form a vector than a global sparse matrix, SMMs have lower assembly costs than VEF methods. In addition, since the left hand side operator is iteratively fixed, the setup costs associated with preconditioned iterative solvers can also be amortized. For example, the AMG setup phase need only be performed once for the SMM moment system whereas in VEF the setup phase must be performed at each iteration due to the changing left hand side operator. Since VEF is non-symmetric, more expensive iterative solvers must be used. In the case of GMRES, non-symmetry also invites an increase in memory requirements in order to store the Krylov space. Thus, independent of the discretization used, SMMs are expected to be cheaper per iteration than an analogous VEF method.

We now compare the DG, CG, and mixed finite element discretization methods in terms of cost. The mixed finite element techniques solve for both the scalar flux and the current whereas the DG and CG techniques only solve for the scalar flux. This means the mixed finite element systems will be significantly larger or, in the case of a hybridized method, will require significantly higher setup costs. Due to this, the mixed finite element methods will both be more expensive to form and invert. The DG methods have immediate compatibility with the hydrodynamics framework of [12]. However, DG methods have more unknowns than the CG methods and also require more complicated preconditioners. Thus, DG methods are more costly than the CG methods.

In light of these arguments, the CG SMM is expected to have the lowest cost per iteration of all the VEF methods and SMMs presented here.

Chapter 10

Conclusions

This dissertation has developed efficient, high-order finite element radiation transport methods that are compatible with curved meshes. Variable Eddington Factor (VEF) methods and Second Moment Methods (SMMs) were developed for the steady-state, mono-energetic, Boltzmann transport equation with isotropic scattering. This model problem emulates the requirements of a single time step of a more complicated thermal radiative transfer (TRT) algorithm and has been demonstrated to be an effective proxy for mathematical research into the design of TRT algorithms in Yee *et al.* [34]. We developed two classes of discretization techniques for the moment systems arising in both VEF and SMM. This led to a total of 10 novel moment methods each with

1. high-order accuracy,
2. compatibility with curved meshes, and
3. efficient preconditioned iterative solvers.

These methods are the first to have *any* of the above properties. The methods are based on an independent discretization approach where the transport and moment equations are discretized separately. This allows the flexibility to design moment algorithms where the transport and moment discretizations are in some sense optimal for their intended uses. In our case, we elected to choose the discretization for the moment system to be able to leverage existing linear solver technology and to have multiphysics compatibility with the hydrodynamics framework of [12]. The discretizations for the moment system were paired with a Discontinuous Galerkin (DG) discretization of the Discrete Ordinates (S_N) transport equations to form effective linear transport algorithms.

In this chapter, we summarize the results from Chapters 6 – 9 and outline future directions for the methods presented here.

10.1 Discontinuous Galerkin VEF

In Chapter 6, the unified framework for DG methods presented by Arnold *et al.* [58] was extended to the non-symmetric VEF equations. In this framework, a clever choice of numerical flux allows a local elimination of the vector-valued, first moment equation. This allows formation of a discretization for the second-order form of the VEF equations. The local elimination of the first moment is stabilized with penalization terms and/or the so-called lifting operators. Such terms ensure the resulting algebraic system will be positive definite.

Using this extended framework, we derived analogs of the Interior Penalty (IP), Second Method of Bassi and Rebay (BR2), Minimal Dissipation Local Discontinuous Galerkin (MDLDG), and continuous finite element (CG) methods. The CG method represents an extension to multiple dimensions, high-order, curved meshes, and efficient solvers of the method of Warsa and Anistratov [22]. The recently developed Uniform Subspace Correction (USC) preconditioning technique from Pazner and Kolev [59] was extended to the IP and BR2 methods. This preconditioner uses a decomposition of the DG finite element space into a CG and DG space. Algebraic Multigrid (AMG) is applied to an operator corresponding to the CG part of the space. A simple Jacobi iteration is used to approximate the inverse of the operator associated with the DG part. This allows leveraging the efficiency of AMG as applied to CG elliptic operators. When combined with the Stabilized Bi-Conjugate Gradient Method (BiCGStab), this preconditioning technique led to iteration counts that were independent of the mesh size, polynomial order, and penalty parameter. The MDLDG and CG techniques were effectively preconditioned by AMG. A Method of Manufactured Solutions (MMS) problem demonstrated that all methods converged optimally on curved meshes for a transport solution that was quadratically anisotropic in angle.

The DG discretizations of the VEF equations were combined with a DG discretization of the S_N transport equations to form efficient linear transport algorithms. The methods were demonstrated to be efficient in both outer fixed-point iterations and inner preconditioned iterative solver iterations on single and multi-material problems. Convergence was nearly identical for all the VEF methods: the IP, BR2, and CG methods converged within 1 iteration of each other with MDLDG requiring at most 3 more iterations than the IP, BR2, and CG methods. This indicates that a continuous finite element method can be efficiently and robustly paired to a discontinuous finite element transport discretization. The inner preconditioned iterative solvers were all shown to be scalable in h and p.

Finally, we tested the IP and CG methods in a weak scaling study. The VEF moment systems were solved in parallel out to 1152 processors and over 10 million scalar flux unknowns. A non-physically difficult mock problem designed to stress the linear solver was found to cause non-convergence of the USC preconditioner applied to the IP discretization. It was found that AMG was struggling to precondition the CG operator. Uniform convergence was recovered by applying AMG to a symmetrized version of the CG operator. The weak scaling study was repeated on a physically realistic problem where the standard USC method converged optimally. AMG was effective for the CG discretization on this problem.

These results indicate that the choice of the VEF discretization does not significantly impact the convergence of the VEF algorithm. Thus, it is recommended to choose the discretization based on other aspects of the algorithm such as ease of implementation, assembly cost, or the expense of the preconditioned iterative solver. We note that the MDLDG method has an increased assembly cost due to its non-local stencil that is formed using sparse matrix multiplication. The BR2 method is more complicated to implement than the IP method but avoids the need to tune the IP method's penalty parameter. Finally, given that the CG method converged equivalently to the IP and BR2 methods, this study indicates there is no added benefit to using the more complicated and expensive DG techniques.

10.2 Mixed Finite Element VEF

Mixed finite element discretizations of the VEF moment system were pursued in order to match as closely as possible the Raviart Thomas (RT)-based radiation diffusion methods used in the hydrodynamics framework of [12]. Such methods were described by Maginot and Brunner [106]. However, the presence of the Eddington tensor in the VEF equations precludes the use of the standard RT discretization. In Chapter 7, we investigated three approaches each of which produces a scalar flux in the DG finite element space expected for the thermodynamic variables of [12]. The methods were differentiated by the choice of finite element space used to approximate the current. We considered: 1) a discrete subspace of $[H^1(\mathcal{D})]^2$ where each component is represented with continuous finite elements, 2) a method that uses the RT space along with DG-like numerical fluxes to treat the discontinuities arising from the presence of the Eddington tensor in the VEF first moment equation, and 3) a Hybridized Raviart Thomas (HRT) method where continuity of the normal component of the current is enforced weakly with a Lagrange multiplier. These methods are referred to as H1, RT, and HRT, respectively. The VEF discretizations were paired with a high-order DG discretization of the S_N transport equations to solve problems from linear transport.

A series of MMS problems demonstrated that all three methods have the optimal order of accuracy for the the scalar flux. However, the optimality of the approximation for the current was problem dependent. On diffusive problems, the current converged optimally. On a quadratically anisotropic transport problem, the current converged suboptimally for some polynomial orders. In particular, the H1 method was optimal for odd polynomial orders but was suboptimal by half an order of h for even. The RT and HRT methods showed a loss of one order for odd polynomial orders and a loss of half an order for even.

All three methods showed rapid and robust convergence on a single-material thick diffusion limit test problem on both a simple orthogonal mesh and a severely distorted third-order mesh generated with a Lagrangian hydrodynamics code. The methods were tested on mesh and polynomial order refinements of a two-material linearized crooked pipe problem that had a 1000x difference in total cross section. Fixed-point convergence was robust for all three methods with RT and HRT converging equivalently. The H1 method converged slower, requiring $\approx$1.75x more iterations than RT and HRT on the largest mesh for each polynomial order.

We also investigated preconditioned iterative solvers for the H1, RT, and HRT methods. Lower block triangular preconditioners were used for the H1 and RT methods that employ Jacobi smoothing on the total interaction mass matrix and AMG on the lumped Schur complement. The solvers for the HRT method leveraged the element-by-element block structure generated by discontinuous approximations to form a reduced system for the Lagrange multiplier only, leading to fewer globally coupled unknowns than in the H1 or RT methods. AMG was applied directly to the reduced problem. The preconditioned iterative solvers were tested on a series of increasingly distorted meshes to test their robustness. The H1 and HRT methods converged for all distortions but the RT method failed to converge once the mesh became too distorted. The RT and HRT methods were shown to have scalable solvers in both h and p on the linearized crooked pipe problem. However, the solvers for H1 were not scalable. It was found that AMG was struggling to adequately precondition the lumped Schur complement due to the presence of highly oscillatory, slowly decaying modes. These modes are a consequence of the mismatch between the finite element spaces used to approximate the VEF scalar flux and current and were present even on a simple Poisson eigenvalue problem. Finally, a weak scaling study demonstrated that the RT and HRT methods can be scalably solved out to 1152 processors and over 10 million VEF scalar flux unknowns. Compared to solving radiation diffusion, solving the non-symmetric VEF equations required at most 9 and 5 more iterations for the RT and HRT methods, respectively.

These results indicate the combination of a DG S_N discretization and the RT or HRT VEF discretizations form an effective high-order method for linear transport problems. Both the RT and HRT discretizations of the VEF equations have high-order accuracy, compatibility with curved meshes, and robust and scalable convergence in both outer fixed-point iterations and inner preconditioned linear solver iterations. The performance of the methods are differentiated only in the presence of severely distorted meshes. In such case, the preconditioned iterative solver for the HRT method was robust to mesh distortion whereas the solver for the RT method was not. The H1 method is not recommended for use in a production code due to the lack of scalable iterative solvers and its slower fixed-point iteration convergence rate on the linearized crooked pipe problem as compared to the RT and HRT methods.

In radiation-hydrodynamics calculations, the scalar flux and current are coupled to the hydrodynamics' energy balance and momentum equations, respectively. Due to the suboptimal accuracy of the VEF current on transport problems, it is unclear whether the mixed finite element methods presented here would yield improvements in physics fidelity commensurate with the increased cost of solving for both the VEF scalar flux and current.

10.3 Second Moment Methods

We also investigated a moment method closely related to the VEF method known as the SMM. Where VEF uses nonlinear, multiplicative closures, SMM uses linear, additive closures. This allows design of an iterative scheme that only requires the inversion of a simple radiation diffusion system at each iteration. Such a scheme avoids the difficulty of inverting the non-symmetric VEF system required by VEF methods. Due to this, simpler discretizations, preconditioners, and solvers can be applied to the SMM moment system. Thus, each iteration of the SMM is expected to be much cheaper than an analogous VEF method.

Chapter 3 defined the SMM algorithm and established the close connection between SMM and VEF. Following Cefus and Larsen [54], we showed that the SMM algorithm is equivalent to a VEF algorithm linearized about a diffusion solution. Thus, the SMM algorithm can be viewed as both a moment method with additive closures and a linearization of the VEF method. This connection was used to systematically convert the discrete VEF algorithms developed in Chapters 6 and 7 to the corresponding discrete SMMs. This includes the IP, CG, RT, and HRT methods. The SMM moment systems were preconditioned and solved using the standard techniques developed for the IP, CG, RT, and HRT discretization of radiation diffusion, respectively. This included the use of the conjugate gradient method to solve the IP, CG, and HRT methods.

An MMS problem was used to demonstrate the accuracy of the IP, CG, RT, and HRT SMMs. All methods converged in an analogous manner to their associated VEF method. This includes the lower order of accuracy for the current seen in the RT and HRT methods. The methods were tested on both single and multi-material problems. Convergence on these problems was very close to the corresponding VEF methods with the SMM variant converging only a few iterations slower. All methods showed uniform convergence in inner preconditioned linear solver iterations. Finally, a weak scaling study was conducted to show that the SMM moment system can be scalably solved in parallel.

10.4 Generalities of the VEF Method

In Chapter 9, the IP VEF method was used to demonstrate properties of the VEF algorithm in general. We showed that the use of the previous outer iteration as an initial guess for the inner preconditioned inner solver led to a reduction of $\sim 30\%$ in the total number of inner iterations to solve the problem when compared to the use of a zero initial guess. As the outer iteration converges, it provides an increasingly more accurate initial guess to the inner solver. This meant that the number of inner iterations at each outer iteration decreased as the outer iteration progressed. For the zero initial guess, the number of inner iterations per outer iteration remained constant. The choice of the initial guess did not affect the convergence of the outer iteration. Thus, the use of the previous outer iteration's solution as an initial guess for the inner solver reduced the cost of the overall algorithm.

We also presented two schemes to provide increased robustness and convergence on curved meshes with reentrant faces. On such a mesh, the transport equation cannot be reordered to be block lower triangular by element precluding the use of the traditional transport sweep. Instead, the pseudo-optimal reordering of the elements presented in Haut *et al.* [15] is used so that the upper triangular portion of the transport system is as small possible. The resulting transport sweep lags incoming angular flux information on reentrant faces meaning the transport equation is only partially inverted at each fixed-point iteration of the VEF algorithm. Compared to an orthogonal mesh problem, the presence of reentrant faces was seen to increase the number of iterations required to converge the VEF algorithm. This was especially true for optically thin problems where lagging angular fluxes has a more global effect due to the larger mean free path in the problem.

We investigated the effect of performing multiple partial transport inversions per fixed-point iteration. We observed a reduction in the number of fixed-point iterations to convergence that approached the convergence rates seen on the orthogonal mesh problem. However, this reduction in fixed-point iterations was not enough to reduce the total number of partial transport inversions performed. That is, when three partial inversions were performed at each outer iteration, the number of outer iterations until convergence decreased by less than 3x meaning more total partial transport inversions were performed in the simulation. This result indicates multiple partial inversions of the transport equation could provide robustness to severe mesh distortions but is not likely to be useful in reducing the cost of the simulation.

We also investigated the use of Anderson acceleration to mitigate the slowdown associated with reentrant faces. We compared two options. The first stored previous iterates of the scalar flux only and the second stored both the scalar and angular fluxes. These schemes were referred to as “low memory” and “augmented”, respectively. It was seen that the low memory option converged a few iterations slower than unaccelerated fixed-point iteration. The augmented scheme performed equivalently to fixed-point iteration on thick problems where lagging incoming angular fluxes on reentrant faces has a reduced impact. On a thin problem, Anderson acceleration with the augmented Anderson space converged a few iterations faster than fixed-point iteration. This suggests that augmenting the Anderson space with the angular flux is required to increase robustness and iterative efficiency on curved meshes.

10.5 Comparison of Methods

Section 9.3 provided a comparison of all the methods presented in this dissertation on the thick diffusion limit, crooked pipe, and weak scaling problems. We omit the $[H^1(\mathcal{D})]^2 \times L^2(\mathcal{D})$ from the following discussion as it could not be scalably solved and is thus not recommended for use in a production setting. It was seen that the mixed finite element-based VEF methods were qualitatively the most sensitive to severe mesh distortion. This may be due to the decreased solution regularity allowed by mixed methods or a result of the gradient of the Piola transformation allowing mesh imprinting to impact solution quality. Additionally, we showed that the solution quality of the IP and CG VEF methods were visually close. This suggests that the penalty parameter used in the IP method regularizes the solution to that of the solution of the CG VEF method.

On the crooked pipe problem, the IP, BR2, MDLDG, CG, RT, and HRT methods used in both the VEF and SMM algorithm converged nearly identically in outer Anderson-accelerated fixed-point iterations: all methods were equivalent to ±4 iterations. This result indicates that the moment algorithm is insensitive to both the choice of the closure used to define the moment system and the discretization used to approximate the moment system. This means the closures and discretization techniques used to form a moment method can be entirely chosen by the larger algorithmic requirements such as implementational ease, computational efficiency, and multiphysics compatibility. We note that the methods compared here all have scalable preconditioned iterative solvers. From the discussion provided in Section 9.3.5, we expect the SMMs to have lower cost than the analogous VEF methods and that the CG-based methods will have the lowest cost followed by DG and then mixed finite element. Given that the solution quality between the CG and IP VEF methods was so similar, this study suggests the added degrees of freedom and more complicated preconditioner associated with the IP method are not warranted. Of the 10 moment methods presented here with efficient solvers, we recommend the CG SMM for implementation in production due to its expected low cost per iteration.

10.6 Future Work

The obvious next step for this research is extending the methods presented here to a full radiation-hydrodynamics algorithm. Such an algorithm would include frequency and time dependence as well as the coupling between the transport equation and the hydrodynamics' energy balance and momentum equations. A one-dimensional version of the H1 algorithm presented here was used to form efficient algorithms for the gray TRT and radiation-hydrodynamics problems in Lou *et al.* [41] and Lou and Morel [42], respectively. Yee *et al.* [34] presents a multi-dimensional gray TRT algorithm also based on the H1 method. The efficiency of these algorithms would be greatly improved by the use of any one of the VEF or SMM discretizations presented here that have efficient solvers (i.e. using the RT or HRT methods instead of H1). The primary remaining research topic is the extension to the frequency-dependent case. Such an extension is presented in one spatial dimension by Anistratov [44, 45] where an efficient frequency-dependent algorithm is formulated using a multi-level approach. A promising path toward a full radiation-hydrodynamics algorithm would be combining the one-dimensional radiation-hydrodynamics algorithm from Lou and Morel [42], the efficient multi-dimensional moment discretizations presented here, and the frequency-dependent algorithm of Anistratov [44, 45].

From the discussion of cost in Section 9.3.5, the CG SMM algorithm is recommended as it has the fewest number of unknowns and is the simplest to implement and precondition effectively. However, it is unknown whether using a continuous finite element space for the radiation energy density would negatively impact robustness and stability in the larger multiphysics setting. There may also be computational aspects associated with the use of a representation for the energy density that is globally coupled. That is, a continuous finite element representation would not allow the beneficial property of element-local coupling between the moment system and the hydrodynamic energy balance equation that discontinuous representations enjoy.

Another research task is to verify the efficacy of the SMM for TRT and radiation-hydrodynamics. An RT-based SMM radiation-hydrodynamics algorithm is particularly attractive within the hydrodynamics framework of [12]. We have designed the RT SMM moment discretization so that the left hand side operator exactly matches that of the diffusion operator used by [12]. Since the coupling to the transport solver occurs only in the right hand side terms, the linear and nonlinear solvers already in place for radiation diffusion can be reused in an SMM algorithm by simply including the SMM correction sources. As with the VEF extension to radiation-hydrodynamics, a treatment for the frequency-dependent case must be investigated. Finally, suboptimal orders of accuracy for the current were observed for the RT and HRT SMM and VEF methods. The impact of this reduced accuracy on the coupling of the moment system to the hydrodynamics' momentum equation should be investigated.

10.7 Coda

This dissertation demonstrated that efficient, independent moment methods can be systematically generated by extending existing finite element discretization techniques and their associated scalable preconditioned iterative solvers to the VEF and SMM moment systems. Aside from the $[H^1(\mathcal{D})]^2 \times L^2(\mathcal{D})$ mixed finite element-based methods, all of the moment methods presented here are strong candidates for implementation in a production radiation-hydrodynamics algorithm. The methods were robust to the thick diffusion limit, inexact sweeps from reentrant faces, and strongly heterogeneous materials. The convergence of the moment algorithm was insensitive to both the choice of the moment system (e.g. VEF vs. SMM) and the discretization technique used for the moment system. Thus, this study indicates that moment algorithms can instead be designed to satisfy the requirements of the larger algorithm such as multiphysics compatibility and computational efficiency.

Bibliography

[1] J. I. Castor, *Radiation Hydrodynamics*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511536182.

[2] D. Mihalas and B. W. Mihalas, *Foundations of Radiation Hydrodynamics*. Oxford University Press, New York, 1984.

[3] R. Town, "Overview of laser-plasma interaction codes used for icf research at llnl," Jul. 2018. DOI: 10.2172/1460933.

[4] L. Chacón *et al.*, "Multiscale high-order/low-order (HOLO) algorithms and applications," *Journal of Computational Physics*, vol. 330, pp. 21–45, 2017. DOI: 10.1016/j.jcp.2016.10.069.

[5] A. B. Wollaber, H. Park, R. B. Lowrie, R. M. Rauenzahn, and M. A. Cleveland, "Multigroup radiation hydrodynamics with a high-order–low-order method," *Nuclear Science and Engineering*, vol. 185, no. 1, pp. 117–129, 2017. DOI: 10.13182/NSE16-45.

[6] A. B. Zylstra *et al.*, "Burning plasma achieved in inertial fusion," *Nature*, vol. 601, no. 7894, pp. 542–548, 2022. DOI: 10.1038/s41586-021-04281-w.

[7] F. F. Chen, *Introduction to Plasma Physics and Controlled Fusion*, Third. Springer Cham, 2016. DOI: 10.1007/978-3-319-22309-4.

[8] G. Pomraning, *The Equations of Radiation Hydrodynamics*. Dover Publications, 1973.

[9] D. S. Kershaw, M. K. Prasad, and J. D. Beason, "A simple and fast method for computing the relativistic Compton scattering kernel for radiative transfer.," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 36, pp. 273–282, Oct. 1986. DOI: 10.1016/0022-4073(86)90050-6.

[10] M. L. Adams and E. W. Larsen, "Fast iterative methods for discrete-ordinates particle transport calculations," *Progress in Nuclear Energy*, vol. 40, no. 1, pp. 3–159, 2002. DOI: https://doi.org/10.1016/S0149-1970(01)00023-3.

[11] N. A. Gentile, "Implicit Monte Carlo Diffusion—An Acceleration Method for Monte Carlo Time-Dependent Radiative Transfer Simulations," *Journal of Computational Physics*, vol. 172, no. 2, pp. 543–571, Sep. 2001. DOI: 10.1006/jcph.2001.6836.

[12] V. A. Dobrev, T. V. Kolev, and R. N. Rieben, "High-order curvilinear finite element methods for Lagrangian hydrodynamics," *SIAM Journal on Scientific Computing*, vol. 34, no. 5, B606–B641, 2012. DOI: 10.1137/120864672.

[13] V. A. Dobrev, T. E. Ellis, T. V. Kolev, and R. N. Rieben, "High-order curvilinear finite elements for axisymmetric Lagrangian hydrodynamics," *Computers & Fluids*, vol. 83, pp. 58–69, 2013. DOI: 10.1016/j.compfluid.2012.06.004.

[14] R. W. Anderson, V. A. Dobrev, T. V. Kolev, R. N. Rieben, and V. Z. Tomov, "High-order multi-material ALE hydrodynamics," *SIAM Journal on Scientific Computing*, vol. 40, no. 1, B32–B58, 2018. DOI: 10.1137/17M1116453.

[15] T. S. Haut, P. G. Maginot, V. Z. Tomov, B. S. Southworth, T. A. Brunner, and T. S. Bailey, "An efficient sweep-based solver for the S_N equations on high-order meshes," *Nuclear Science and Engineering*, 2019. DOI: 10.1080/00295639.2018.1562778.

[16] D. Woods, "Discrete ordinates radiation transport using high-order finite element spatial discretizations on meshes with curved surfaces," Ph.D. dissertation, Oregon State University, 2018.

[17] T. S. Haut, B. S. Southworth, P. G. Maginot, and V. Z. Tomov, "Diffusion synthetic acceleration preconditioning for discontinuous Galerkin discretizations of S_N transport on high-order curved meshes," *SIAM Journal on Scientific Computing*, vol. 42, no. 5, B1271–B1301, 2020. DOI: 10.1137/19M124993X.

[18] B. S. Southworth, M. Holec, and T. S. Haut, "Diffusion synthetic acceleration for heterogeneous domains, compatible with voids," *Nuclear Science and Engineering*, vol. 195, no. 2, pp. 119–136, 2021. DOI: 10.1080/00295639.2020.1799603.

[19] V. Ya. Gol'din, "A quasi-diffusion method of solving the kinetic equation," *USSR Computational Mathematics and Mathematical Physics*, vol. 4, pp. 136–149, 1964.

[20] R. J. Mason, "Implicit moment particle simulation of plasmas," *Journal of Computational Physics*, vol. 41, no. 2, pp. 233–244, 1981. DOI: https://doi.org/10.1016/0021-9991(81)90094-2.

[21] C. Newman, G. Womeldorff, L. Chacón, and D. A. Knoll, "High-order/low-order methods for ocean modeling," *Procedia Computational Science*, vol. 51, no. C, 2086–2096, 2015. DOI: 10.1016/j.procs.2015.05.477.

[22] J. Warsa and D. Anistratov, "Two-level transport methods with independent discretization," *Journal of Computational and Theoretical Transport*, vol. 47, no. 4-6, pp. 424–450, 2018. DOI: 10.1080/23324309.2018.1497991.

[23] R. Alcouffe, "Diffusion synthetic acceleration methods for the diamond-differenced discrete-ordinates equations," *Nuclear Science and Engineering*, vol. 64, pp. 344–355, 1977.

[24] D. Y. Anistratov and V. Y. Gol'din, "Nonlinear methods for solving particle transport problems," *Transport Theory and Statistical Physics*, vol. 22, no. 2-3, pp. 125–163, 1993. DOI: 10.1080/00411459308203810.

[25] E. W. Larsen, J. E. Morel, and W. F. Miller Jr., "Asymptotic solutions of numerical transport problems in optically thick, diffusive regimes," *Journal of Computational Physics*, vol. 69, no. 2, pp. 283–324, 1987. DOI: https://doi.org/10.1016/0021-9991(87)90170-7.

[26] D. Mihalas, *Stellar Atmospheres*. W. H. Freeman and Co, 1978.

[27] L. H. Auer and D. Mihalas, "On the use of variable Eddington factors in non-LTE stellar atmospheres computations," *Monthly Notices of the Royal Astronimical Society*, vol. 149, pp. 65–74, 1970. DOI: 10.1093/mnras/149.1.65.

[28] E. Aristova and D. Baydin, "Implementation of the quasidiffusion method for calculating the critical parameters of a fast neutron reactor in 3D hexagonal geometry," *Mathematical Models and Computer Simulations*, vol. 5, pp. 145–155, 2013.

[29] A. Tamang and D. Y. Anistratov, "A multilevel projective method for solving the space-time multigroup neutron kinetics equations coupled with the heat transfer equation," *Nuclear Science and Engineering*, vol. 177, no. 1, pp. 1–18, 2014. DOI: 10.13182/NSE13-42.

[30] D. Y. Anistratov, E. N. Aristova, and V. Y. Gol'din, "A nonlinear method for solving the problems of radiation transfer in medium," *Matematicheskoe modelirovanie*, vol. 8, no. 12, pp. 3–28, 1996.

[31] D. Y. Anistratov and V. Y. Gol'din, "Comparison of difference schemes for the quasidiffusion method for solving the transport equation," *Problems of Atomic Science and Engineering: Method and Codes for Numerical Solution Mathematical Physics Problems*, vol. 2, pp. 17–23, 1986.

[32] E. Aristova and V. Gol'din, "Computation of anisotropy scattering of solar radiation in atmosphere (monoenergetic case)," *Journal of Quantitative Spectroscopy and Radiative Transfer*, vol. 67, no. 2, pp. 139–157, 2000. DOI: https://doi.org/10.1016/S0022-4073(99)00201-0.

[33] P. Ghassemi and D. Y. Anistratov, "Multilevel quasidiffusion method with mixed-order time discretization for multigroup thermal radiative transfer problems," *Journal of Computational Physics*, vol. 409, p. 109 315, 2020. DOI: https://doi.org/10.1016/j.jcp.2020.109315.

[34] B. Yee, S. Olivier, B. Southworth, M. Holec, and T. Haut, "A new scheme for solving high-order DG discretizations of thermal radiative transfer using the variable Eddington factor method," in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2021)*, 2021.

[35] D. Y. Anistratov and J. M. Coale, *Implicit methods with reduced memory for thermal radiative transfer*, 2021. DOI: 10.48550/ARXIV.2103.02726.

[36] Y.-F. Jiang, J. M. Stone, and S. W. Davis, "A Godunov method for multidimensional radiation magnetohydrodynamics based on a variable Eddington tensor," *The Astrophysical Journal Supplement Series*, vol. 199, no. 1, p. 14, 2012. DOI: 10.1088/0067-0049/199/1/14.

[37] N. Y. Gnedin and T. Abel, "Multi-dimensional cosmological radiative transfer with a variable Eddington tensor formalism," *New Astronomy*, vol. 6, no. 7, pp. 437–455, 2001. DOI: https://doi.org/10.1016/S1384-1076(01)00068-9.

[38] M. Gehmeyr and D. Mihalas, "Adaptive grid radiation hydrodynamics with TITAN," *Physica D: Nonlinear Phenomena*, vol. 77, no. 1, pp. 320–341, 1994. DOI: https://doi.org/10.1016/0167-2789(94)90143-0.

[39] S. W. Davis, J. M. Stone, and Y.-F. Jiang, "A radiation transfer solver for Athena using short characteristics," *The Astrophysical Journal Supplement Series*, vol. 199, no. 1, p. 9, 2012. DOI: 10.1088/0067-0049/199/1/9.

[40] S. S. Olivier and J. E. Morel, "Variable Eddington factor method for the S_N equations with lumped discontinuous Galerkin spatial discretization coupled to a drift-diffusion acceleration equation with mixed finite-element discretization," *Journal of Computational and Theoretical Transport*, vol. 46, no. 6-7, pp. 480–496, 2017. DOI: 10.1080/23324309.2017.1418378.

[41] J. Lou, J. E. Morel, and N. Gentile, "A variable Eddington factor method for the 1-D grey radiative transfer equations with discontinuous Galerkin and mixed finite-element spatial differencing," *Journal of Computational Physics*, vol. 393, pp. 258–277, 2019. DOI: https://doi.org/10.1016/j.jcp.2019.05.012.

[42] J. Lou and J. E. Morel, "A variable eddington factor method with different spatial discretizations for the radiative transfer equation and the hydrodynamics/radiation-moment equations," *Journal of Computational Physics*, vol. 439, p. 110 393, 2021. DOI: https://doi.org/10.1016/j.jcp.2021.110393.

[43] B. C. Yee, S. S. Olivier, T. S. Haut, M. Holec, V. Z. Tomov, and P. G. Maginot, "A quadratic programming flux correction method for high-order DG discretizations of S_N transport," *Journal of Computational Physics*, vol. 419, p. 109 696, 2020. DOI: https://doi.org/10.1016/j.jcp.2020.109696.

[44] D. Y. Anistratov, "Nonlinear iterative projection methods with multigrid in photon frequency for thermal radiative transfer," *Journal of Computational Physics*, vol. 444, p. 110 568, 2021. DOI: https://doi.org/10.1016/j.jcp.2021.110568.

[45] ——, "Stability analysis of a multilevel quasidiffusion method for thermal radiative transfer problems," *Journal of Computational Physics*, vol. 376, pp. 186–209, 2019. DOI: https://doi.org/10.1016/j.jcp.2018.09.034.

[46] M. M. Miften and E. W. Larsen, "The quasi-diffusion method for solving transport problems in planar and spherical geometries," *Transport Theory and Statistical Physics*, vol. 22, no. 2-3, pp. 165–186, 1993. DOI: 10.1080/00411459308203811.

[47] J. P. Jones, "The quasidiffusion method for solving radiation transport problems on arbitrary quadrilateral meshes in 2D r-z geometry.," Ph.D. dissertation, North Carolina State University, 2019.

[48] W. A. Wieselquist, D. Y. Anistratov, and J. E. Morel, "A cell-local finite difference discretization of the low-order quasidiffusion equations for neutral particle transport on unstructured quadrilateral meshes," *Journal of Computational Physics*, vol. 273, pp. 343–357, 2014. DOI: https://doi.org/10.1016/j.jcp.2014.05.011.

[49] N. D. Vallette, "Discretization and solution of quasi-diffusion equations," M.S. thesis, Texas A&M University, 2002.

[50] S. Olivier, P. Maginot, and T. Haut, "High order mixed finite element discretization for the variable Eddington factor equations," in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2019)*, 2019.

[51] W.A. Wieselquist, "A low-order quasidiffusion discretization via linear-continuous finite-elements on unstructured triangular meshes," in *Proceedings of PHYSOR 2010: Advances in Reactor Physics to Power the Nuclear Renaissance*, The American Nuclear Society, 2010.

[52] D. Y. Anistratov and J. S. Warsa, "Discontinuous finite element quasi-diffusion methods," *Nuclear Science and Engineering*, vol. 191, no. 2, pp. 105–120, 2018. DOI: 10.1080/00295639.2018.1450013.

[53] E. Lewis and W. Miller Jr., "A comparison of p1 synthetic acceleration techniques," in *Transations of the American Nuclear Society 23*, 1976.

[54] G. R. Cefus and E. W. Larsen, "Stability analysis of the quasideffusion and second moment methods for iteratively solving discrete-ordinates problems," *Transport Theory and Statistical Physics*, vol. 18, no. 5-6, pp. 493–511, 1989. DOI: 10.1080/00411458908204700.

[55] N. D. Stehle, D. Y. Anistratov, and M. L. Adams, "A hybrid transport-diffusion method for 2d transport problems with diffusive subdomains," *Journal of Computational Physics*, vol. 270, pp. 325–344, 2014. DOI: https://doi.org/10.1016/j.jcp.2014.03.056.

[56] D. Y. Anistratov, J. M. Coale, J. S. Warsa, and J. H. Chang, "Multilevel second-moment methods with group decomposition for multigroup transport problems," in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2021)*, 2021. DOI: 10.13182/M&C21-33798.

[57] P. G. Maginot, P. F. Nowak, and M. L. Adams, “A review of the upstream corner balance spatial discretization,” in *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering, Jeju, Korea (M&C 2017)*, 2017.

[58] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini, “Unified analysis of discontinuous Galerkin methods for elliptic problems,” *SIAM Journal on Numerical Analysis*, vol. 39, no. 5, pp. 1749–1779, 2002. DOI: `10.1137/S0036142901384162`.

[59] W. Pazner and T. Kolev, “Uniform subspace correction preconditioners for discontinuous Galerkin methods with *hp*-refinement,” *Communications on Applied Mathematics and Computation*, Jul. 2021. DOI: `10.1007/s42967-021-00136-3`.

[60] E. E. Lewis and W. F. Miller Jr., *Computational Methods of Neutron Transport*. American Nuclear Society, 1993.

[61] R. G. McClarren, “Spherical harmonics methods for thermal radiation transport,” Ph.D. dissertation, The University of Michigan, 2006.

[62] J. S. Warsa, T. A. Wareing, and J. E. Morel, “Krylov iterative methods and the degraded effectiveness of diffusion synthetic acceleration for multidimensional S_N calculations in problems with material discontinuities,” *Nuclear Science and Engineering*, vol. 147, no. 3, pp. 218–248, 2004. DOI: `10.13182/NSE02-14`.

[63] M. Vaĭnberg, *Variational methods for the study of nonlinear operators [by] M. M. Vainberg. With a chapter on Newton’s method by L. V. Kantorovich and G. P. Akilov. Translated and supplemented by Amiel Feinstein.* eng, ser. Holden-Day series in mathematical physics. San Francisco: Holden-Day, 1964.

[64] P. Ciarlet, *The Finite Element Method for Elliptic Problems*. Elsevier Science, 1978.

[65] S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, ser. Texts in Applied Mathematics. Springer New York, NY, 2008.

[66] A. Quarteroni and A. Valli, *Numerical Approximation of Partial Differential Equations*. Springer, Berlin, Heidelberg, 1994. DOI: `10.1007/978-3-540-85268-1`.

[67] D. Boffi, F. Brezzi, and M. Fortin, *Mixed Finite Element Methods and Applications*. Springer, Berlin, Heidelberg, 2013. DOI: `10.1007/978-3-642-36519-5`.

[68] A. Ern and J.-L. Guermond, *Theory and Practice of Finite Elements*, ser. Applied Mathematical Sciences. Springer New York, NY, 2004.

[69] T. I. Zhodi, *A Finite Element Primer for Beginners: The Basics*. Springer Cham, 2018.

[70] M. Reed and B. Simon, *I: Functional Analysis*, ser. Methods of Modern Mathematical Physics. Elsevier Science, 1981.

[71] L. Evans, *Partial Differential Equations*, ser. Graduate studies in mathematics. American Mathematical Society, 2010.

[72] R. C. Kirby, "From functional analysis to iterative methods," *SIAM Review*, vol. 52, no. 2, pp. 269–293, 2010. DOI: `10.1137/070706914`.

[73] C. Canuto, M. Y. Hussaini, A. Quateroni, and T. A. Zang, *Spectral Methods, Fundamentals in Single Domains*, ser. Scientific Computation. Springer, Berlin, Heidelberg, 2006.

[74] L. N. Trefethen, *Approximation Theory and Approximation Practice*, ser. Other Titles in Applied Mathematics. SIAM, 2019.

[75] P. Ciarlet, *Three-Dimensional Elasticity*, ser. Mathematical Elasticity. Elsevier Science, 1988.

[76] J. Oden, *Finite Elements of Nonlinear Continua*, ser. Advanced engineering series. McGraw-Hill, 1971.

[77] M. E. Rognes, R. C. Kirby, and A. Logg, "Efficient assembly of $H(\mathrm{div})$ and $H(\mathrm{curl})$ conforming finite elements," *SIAM Journal on Scientific Computing*, vol. 31, no. 6, pp. 4130–4151, 2010. DOI: `10.1137/08073901X`.

[78] Raviart, P. A. and Thomas, J. M., "A mixed finite element method for 2-nd order elliptic problems," in *Mathematical Aspects of Finite Element Methods: Proceedings of the Conference Held in Rome, December 10–12, 1975.* Berlin, Heidelberg: Springer Berlin Heidelberg, 1977, pp. 292–315.

[79] P. Raviart and J. Thomas, "Dual finite element models for second order elliptic problems," *Energy Methods in Finite Element Analysis*, pp. 175–191,

[80] F. Brezzi and M. Fortin, *Mixed and Hybrid Finite Element Methods.* Springer, Berlin, Heidelberg, 1991.

[81] X. S. Li and J. W. Demmel, "SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems," *ACM Trans. Mathematical Software*, vol. 29, no. 2, pp. 110–140, 2003.

[82] M. R. Hestenes and E. Stiefel, "Methods of conjugate gradients for solving linear systems," *Journal of Research of the National Bureau of Standards*, vol. 49, no. 6, pp. 409–436, 1952.

[83] M. R. Hestenes, *Conjugate Direction Methods in Optimization.* Springer-Verlag, New York, 1980.

[84] J. R. Shewchuk, "An introduction to the conjugate gradient method without the agonizing pain," USA, Tech. Rep., 1994.

[85] C. Lanczos, "Solution of systems of linear equations by minimized iterations," *Journal of Research of the National Bureau of Standards*, vol. 49, pp. 33–53, 1952.

[86] R. Fletcher, "Conjugate gradient methods for indefinite systems," in *Numerical Analysis*, G. A. Watson, Ed., Berlin, Heidelberg: Springer Berlin Heidelberg, 1976, pp. 73–89.

[87] Y. Saad, *Iterative Methods for Sparse Linear Systems*, ser. Other Titles in Applied Mathematics. SIAM, 2003.

[88] H. A. van der Vorst, "Bi-CGStab: A fast and smoothly converging variant of Bi-CG for the solution of nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 13, no. 2, pp. 631–644, 1992. DOI: `10.1137/0913035`.

[89] *Templates for the Solution of Linear Systems, Building Blocks for Iterative Methods*, ser. Other Titles in Applied Mathematics. SIAM, 1994.

[90] R. Anderson *et al.*, "MFEM: A modular finite element methods library," *Computers & Mathematics with Applications*, Jul. 2020. DOI: `10.1016/j.camwa.2020.06.009`.

[91] Y. Saad and M. H. Schultz, "GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems," *SIAM Journal on Scientific and Statistical Computing*, vol. 7, pp. 856–869, 1986.

[92] W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, ser. Other Titles in Applied Mathematics. SIAM, 2000.

[93] A. Brandt, S. McCormick, and J. Ruge, "Algebraic multigrid (AMG) for sparse matrix equations," *Sparsity and Its Applications*, pp. 257–284, 1984.

[94] R. D. Falgout and U. M. Yang, "Hypre: A library of high performance preconditioners," in *Proceedings of the International Conference on Computational Science-Part III*, ser. ICCS '02, Berlin, Heidelberg: Springer-Verlag, 2002, 632–641.

[95] W. Pazner and T. Haut, "A short note on the accuracy of the discontinuous galerkin method with reentrant faces," *Journal of Computational Physics*, vol. 443, p. 110 448, 2021. DOI: `https://doi.org/10.1016/j.jcp.2021.110448`.

[96] K. Lathrop, "Spatial differencing of the transport equation: Positivity vs. accuracy," *Journal of Computational Physics*, vol. 4, no. 4, pp. 475–498, 1969. DOI: `https://doi.org/10.1016/0021-9991(69)90015-1`.

[97] S. K. Godunov, "Different methods for shock waves," Ph.D. dissertation, Moscow State University, 1954.

[98] S. Hamilton, M. Benzi, and J. Warsa, "Negative flux fixups in discontinuous finite element S_N transport," in *International Conference on Mathematics, Computational Methods and Reactor Physics (M&C 2009), American Nuclear Society, LaGrange Park, Illinois, USA*, Citeseer, 2009.

[99] M. Ainsworth, G. Andriamaro, and O. Davydov, "Bernstein–bézier finite elements of arbitrary order and optimal assembly procedures," *SIAM Journal on Scientific Computing*, vol. 33, no. 6, pp. 3087–3109, 2011. DOI: `10.1137/11082539X`.

[100] B. Cockburn and B. Dong, "An analysis of the minimal dissipation local discontinuous Galerkin method for convection–diffusion problems," *Journal of Scientific Computing*, vol. 32, no. 2, 233–262, Aug. 2007. DOI: `10.1007/s10915-007-9130-3`.

[101] R. Anderson *et al.*, “MFEM: A modular finite element methods library,” *Computers & Mathematics with Applications*, Jul. 2020. DOI: 10.1016/j.camwa.2020.06.009.

[102] *MFEM: Modular finite element methods [Software]*, https://mfem.org, 2010. DOI: 10.11578/dc.20171025.1248.

[103] A. C. Hindmarsh *et al.*, “SUNDIALS: Suite of nonlinear and differential/algebraic equation solvers,” *ACM Transactions on Mathematical Software (TOMS)*, vol. 31, no. 3, pp. 363–396, 2005.

[104] O. C. Zienkiewicz, “Displacement and equilibrium models in the finite element method by b. fraeijs de veubeke, chapter 9, pages 145–197 of stress analysis, edited by o. c. zienkiewicz and g. s. holister, published by john wiley & sons, 1965,” *International Journal for Numerical Methods in Engineering*, vol. 52, no. 3, pp. 287–342, 2001. DOI: https://doi.org/10.1002/nme.339.

[105] V. Dobrev, T. Kolev, C. S. Lee, V. Tomov, and P. S. Vassilevski, “Algebraic hybridization and static condensation with application to scalable h(div) preconditioning,” *SIAM Journal on Scientific Computing*, vol. 41, no. 3, B425–B447, 2019. DOI: 10.1137/17M1132562.

[106] P. G. Maginot and T. A. Brunner, “Lumping techniques for mixed finite element diffusion discretizations,” *Journal of Computational and Theoretical Transport*, vol. 47, no. 4-6, pp. 301–325, 2018. DOI: 10.1080/23324309.2018.1461653.

[107] J. Lautard and F. Moreau, “A fast 3-d parallel diffusion solver based on a mixed-dual finite element approximation,” in *Proceedings of the American Nuclear Society Topical Meeting: Mathematical Methods and Supercomputing in Nuclear Applications Karlsruhe, Germany*, 1993.

[108] J. P. Hennart, E. H. Mund, and E. D. Valle, “A composite nodal finite element for hexagons,” *Nuclear Science and Engineering*, vol. 127, no. 2, pp. 139–153, 1997. DOI: 10.13182/NSE97-A28593.

[109] A.-M. Baudron and J.-J. Lautard, “Minos: A simplified pn solver for core calculation,” *Nuclear Science and Engineering*, vol. 155, no. 2, pp. 250–263, 2007. DOI: 10.13182/NSE07-A2660.

[110] F. Brezzi, “Stability of saddle-points in finite dimensions,” in *Frontiers in Numerical Analysis: Durham 2002*, J. F. Blowey, A. W. Craig, and T. Shardlow, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 17–61. DOI: 10.1007/978-3-642-55692-0_2.

[111] H. C. Elman, D. J. Silvester, and A. J. Wathen, *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*, Second, ser. Numerical Mathematics and Scientific Computation. Oxford University Press, Oxford, 2014. DOI: 10.1093/acprof:oso/9780199678792.001.0001.

[112] M. Benzi, G. H. Golub, and J. Liesen, “Numerical solution of saddle point problems,” *Acta Numerica*, vol. 14, 1–137, 2005. DOI: 10.1017/S0962492904000212.

[113] J. S. Warsa, T. A. Wareing, and J. E. Morel, "Fully consistent diffusion synthetic acceleration of linear discontinuous SN transport discretizations on unstructured tetrahedral meshes," *Nuclear Science and Engineering*, vol. 141, no. 3, pp. 236–251, 2002. DOI: 10.13182/NSE141-236.

[114] M. L. Adams and W. R. Martin, "Diffusion synthetic acceleration of discontinuous finite element transport iterations," *Nuclear Science and Engineering*, vol. 111, no. 2, pp. 145–167, 1992. DOI: 10.13182/NSE92-A23930.

[115] Y. Wang and J. C. Ragusa, "Diffusion synthetic acceleration for high-order discontinuous finite element S_N transport schemes and application to locally refined unstructured meshes," *Nuclear Science and Engineering*, vol. 166, no. 2, pp. 145–166, 2010. DOI: 10.13182/NSE09-46.

Appendix A

Method of Manufactured Solutions Supplemental Data

Table A.1: Error values for the VEF method on an isotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the VEF data are constant in space and thus are represented exactly.

		$\|\varphi - \varphi_{\text{ex}}\|$			$\|\varphi - \Pi\varphi_{\text{ex}}\|$			$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$		
p	h	H1	RT	HRT	H1	RT	HRT	H1	RT	HRT
1	3.994×10^{-2}	4.160×10^{-4}	4.161×10^{-4}	4.161×10^{-4}	9.853×10^{-6}	1.067×10^{-5}	1.067×10^{-5}	5.605×10^{-4}	1.251×10^{-3}	1.251×10^{-3}
	1.997×10^{-2}	1.040×10^{-4}	1.040×10^{-4}	1.040×10^{-4}	1.209×10^{-6}	1.260×10^{-6}	1.260×10^{-6}	1.399×10^{-4}	3.125×10^{-4}	3.125×10^{-4}
	1.331×10^{-2}	4.624×10^{-5}	4.624×10^{-5}	4.624×10^{-5}	3.570×10^{-7}	3.693×10^{-7}	3.693×10^{-7}	6.217×10^{-5}	1.389×10^{-4}	1.389×10^{-4}
	9.985×10^{-3}	2.601×10^{-5}	2.601×10^{-5}	2.601×10^{-5}	1.505×10^{-7}	1.552×10^{-7}	1.552×10^{-7}	3.497×10^{-5}	7.812×10^{-5}	7.812×10^{-5}
2	5.874×10^{-2}	1.407×10^{-5}	1.411×10^{-5}	1.411×10^{-5}	7.222×10^{-7}	1.301×10^{-6}	1.301×10^{-6}	2.718×10^{-5}	1.629×10^{-4}	1.629×10^{-4}
	3.026×10^{-2}	1.921×10^{-6}	1.922×10^{-6}	1.922×10^{-6}	4.412×10^{-8}	8.331×10^{-8}	8.331×10^{-8}	3.236×10^{-6}	2.254×10^{-5}	2.254×10^{-5}
	1.997×10^{-2}	5.521×10^{-7}	5.523×10^{-7}	5.523×10^{-7}	8.065×10^{-9}	1.542×10^{-8}	1.542×10^{-8}	8.901×10^{-7}	6.495×10^{-6}	6.495×10^{-6}
	1.490×10^{-2}	2.295×10^{-7}	2.295×10^{-7}	2.295×10^{-7}	2.466×10^{-9}	4.740×10^{-9}	4.740×10^{-9}	3.626×10^{-7}	2.701×10^{-6}	2.701×10^{-6}
3	7.681×10^{-2}	9.628×10^{-7}	9.905×10^{-7}	9.905×10^{-7}	9.972×10^{-8}	2.604×10^{-7}	2.604×10^{-7}	3.951×10^{-6}	3.112×10^{-5}	3.112×10^{-5}
	3.994×10^{-2}	7.071×10^{-8}	7.112×10^{-8}	7.112×10^{-8}	3.471×10^{-9}	8.727×10^{-9}	8.727×10^{-9}	2.818×10^{-7}	2.231×10^{-6}	2.231×10^{-6}
	2.628×10^{-2}	1.326×10^{-8}	1.329×10^{-8}	1.329×10^{-8}	4.149×10^{-10}	1.043×10^{-9}	1.043×10^{-9}	5.275×10^{-8}	4.175×10^{-7}	4.175×10^{-7}
	1.997×10^{-2}	4.426×10^{-9}	4.432×10^{-9}	4.432×10^{-9}	1.041×10^{-10}	2.619×10^{-10}	2.619×10^{-10}	1.762×10^{-8}	1.393×10^{-7}	1.393×10^{-7}
4	9.986×10^{-2}	3.563×10^{-7}	3.566×10^{-7}	3.566×10^{-7}	4.665×10^{-8}	4.712×10^{-8}	4.712×10^{-8}	1.261×10^{-6}	4.258×10^{-6}	4.258×10^{-6}
	4.993×10^{-2}	1.155×10^{-8}	1.157×10^{-8}	1.157×10^{-8}	7.186×10^{-10}	8.170×10^{-10}	8.170×10^{-10}	3.640×10^{-8}	2.027×10^{-7}	2.027×10^{-7}
	3.328×10^{-2}	1.524×10^{-9}	1.525×10^{-9}	1.525×10^{-9}	6.290×10^{-11}	6.896×10^{-11}	6.896×10^{-11}	4.554×10^{-9}	2.712×10^{-8}	2.712×10^{-8}
	2.496×10^{-2}	3.619×10^{-10}	3.619×10^{-10}	3.619×10^{-10}	1.119×10^{-11}	1.209×10^{-11}	1.210×10^{-11}	1.089×10^{-9}	6.473×10^{-9}	6.473×10^{-9}

Table A.2: Error values for the VEF method on a quadratically anisotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the angular flux used to calculate the VEF data is represented with Y_p. Due to this, the maximum accuracy expected is order $p + 1$.

		$\lVert \varphi - \varphi_{\text{ex}} \rVert$			$\lVert \varphi - \Pi\varphi_{\text{ex}} \rVert$			$\lVert \boldsymbol{J} - \boldsymbol{J}_{\text{ex}} \rVert$		
p	h	H1	RT	HRT	H1	RT	HRT	H1	RT	HRT
1	3.994×10^{-2}	1.891×10^{-3}	1.891×10^{-3}	1.890×10^{-3}	2.589×10^{-4}	2.725×10^{-4}	2.638×10^{-4}	7.454×10^{-3}	1.703×10^{-2}	1.499×10^{-2}
	1.997×10^{-2}	4.707×10^{-4}	4.708×10^{-4}	4.707×10^{-4}	4.915×10^{-5}	5.110×10^{-5}	4.979×10^{-5}	1.925×10^{-3}	8.746×10^{-3}	7.651×10^{-3}
	1.331×10^{-2}	2.090×10^{-4}	2.090×10^{-4}	2.090×10^{-4}	1.980×10^{-5}	2.035×10^{-5}	2.004×10^{-5}	8.786×10^{-4}	5.866×10^{-3}	5.124×10^{-3}
	9.985×10^{-3}	1.175×10^{-4}	1.175×10^{-4}	1.175×10^{-4}	1.061×10^{-5}	1.082×10^{-5}	1.066×10^{-5}	5.077×10^{-4}	4.410×10^{-3}	3.850×10^{-3}
2	5.874×10^{-2}	2.793×10^{-4}	2.787×10^{-4}	2.826×10^{-4}	7.262×10^{-5}	7.092×10^{-5}	8.536×10^{-5}	1.373×10^{-3}	1.125×10^{-3}	1.165×10^{-3}
	3.026×10^{-2}	3.994×10^{-5}	3.992×10^{-5}	4.028×10^{-5}	9.827×10^{-6}	9.746×10^{-6}	1.114×10^{-5}	2.745×10^{-4}	2.032×10^{-4}	2.165×10^{-4}
	1.997×10^{-2}	1.158×10^{-5}	1.157×10^{-5}	1.160×10^{-5}	2.853×10^{-6}	2.843×10^{-6}	2.931×10^{-6}	9.552×10^{-5}	7.032×10^{-5}	7.532×10^{-5}
	1.490×10^{-2}	4.826×10^{-6}	4.825×10^{-6}	4.864×10^{-6}	1.194×10^{-6}	1.192×10^{-6}	1.343×10^{-6}	4.534×10^{-5}	3.348×10^{-5}	3.690×10^{-5}
3	7.681×10^{-2}	8.143×10^{-5}	8.129×10^{-5}	8.124×10^{-5}	1.357×10^{-5}	1.308×10^{-5}	1.261×10^{-5}	4.150×10^{-4}	3.580×10^{-4}	3.349×10^{-4}
	3.994×10^{-2}	5.639×10^{-6}	5.638×10^{-6}	5.638×10^{-6}	7.684×10^{-7}	7.765×10^{-7}	7.752×10^{-7}	3.053×10^{-5}	5.762×10^{-5}	5.128×10^{-5}
	2.628×10^{-2}	1.050×10^{-6}	1.050×10^{-6}	1.051×10^{-6}	1.256×10^{-7}	1.270×10^{-7}	1.310×10^{-7}	5.681×10^{-6}	1.708×10^{-5}	1.505×10^{-5}
	1.997×10^{-2}	3.498×10^{-7}	3.498×10^{-7}	3.499×10^{-7}	3.895×10^{-8}	3.934×10^{-8}	4.032×10^{-8}	1.888×10^{-6}	7.609×10^{-6}	6.672×10^{-6}
4	9.986×10^{-2}	1.468×10^{-5}	1.464×10^{-5}	1.473×10^{-5}	4.023×10^{-6}	3.433×10^{-6}	3.797×10^{-6}	5.720×10^{-5}	5.987×10^{-5}	6.984×10^{-5}
	4.993×10^{-2}	5.743×10^{-7}	5.738×10^{-7}	5.753×10^{-7}	1.097×10^{-7}	1.077×10^{-7}	1.163×10^{-7}	3.569×10^{-6}	3.667×10^{-6}	3.399×10^{-6}
	3.328×10^{-2}	7.940×10^{-8}	7.937×10^{-8}	8.009×10^{-8}	1.530×10^{-8}	1.514×10^{-8}	1.858×10^{-8}	5.995×10^{-7}	5.478×10^{-7}	5.495×10^{-7}
	2.496×10^{-2}	1.911×10^{-8}	1.910×10^{-8}	1.926×10^{-8}	3.758×10^{-9}	3.737×10^{-9}	4.489×10^{-9}	1.618×10^{-7}	1.428×10^{-7}	1.432×10^{-7}

Table A.3: Error values for the VEF method on a quadratically anisotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the angular flux used to calculate the VEF data is represented with Y_{p+1}. Due to this, the maximum accuracy expected is order $p + 2$.

		$\lVert \varphi - \varphi_{\text{ex}} \rVert$			$\lVert \varphi - \Pi\varphi_{\text{ex}} \rVert$			$\lVert \boldsymbol{J} - \boldsymbol{J}_{\text{ex}} \rVert$		
p	h	H1	RT	HRT	H1	RT	HRT	H1	RT	HRT
0	1.997×10^{-2}	1.564×10^{-2}	1.564×10^{-2}	1.564×10^{-2}	5.329×10^{-4}	5.301×10^{-4}	5.172×10^{-4}	7.910×10^{-3}	1.028×10^{-2}	1.028×10^{-2}
	9.985×10^{-3}	7.827×10^{-3}	7.827×10^{-3}	7.827×10^{-3}	1.309×10^{-4}	1.323×10^{-4}	1.291×10^{-4}	2.849×10^{-3}	5.138×10^{-3}	5.138×10^{-3}
	6.657×10^{-3}	5.219×10^{-3}	5.219×10^{-3}	5.219×10^{-3}	5.784×10^{-5}	5.878×10^{-5}	5.737×10^{-5}	1.564×10^{-3}	3.425×10^{-3}	3.424×10^{-3}
	4.993×10^{-3}	3.915×10^{-3}	3.914×10^{-3}	3.914×10^{-3}	3.244×10^{-5}	3.305×10^{-5}	3.226×10^{-5}	1.021×10^{-3}	2.568×10^{-3}	2.568×10^{-3}
1	3.994×10^{-2}	1.876×10^{-3}	1.875×10^{-3}	1.875×10^{-3}	1.032×10^{-4}	1.097×10^{-4}	9.773×10^{-5}	4.561×10^{-3}	3.443×10^{-3}	1.350×10^{-3}
	1.997×10^{-2}	4.683×10^{-4}	4.683×10^{-4}	4.683×10^{-4}	1.275×10^{-5}	1.422×10^{-5}	1.256×10^{-5}	1.210×10^{-3}	1.741×10^{-3}	3.569×10^{-4}
	1.331×10^{-2}	2.081×10^{-4}	2.081×10^{-4}	2.081×10^{-4}	3.764×10^{-6}	4.275×10^{-6}	3.754×10^{-6}	5.459×10^{-4}	1.166×10^{-3}	1.675×10^{-4}
	9.985×10^{-3}	1.171×10^{-4}	1.171×10^{-4}	1.171×10^{-4}	1.586×10^{-6}	1.826×10^{-6}	1.594×10^{-6}	3.090×10^{-4}	8.758×10^{-4}	9.902×10^{-5}
2	5.874×10^{-2}	2.717×10^{-4}	2.714×10^{-4}	2.714×10^{-4}	2.914×10^{-5}	2.706×10^{-5}	2.750×10^{-5}	5.506×10^{-4}	2.855×10^{-4}	2.109×10^{-4}
	3.026×10^{-2}	3.878×10^{-5}	3.877×10^{-5}	3.877×10^{-5}	2.075×10^{-6}	1.847×10^{-6}	1.917×10^{-6}	7.526×10^{-5}	3.918×10^{-5}	3.010×10^{-5}
	1.997×10^{-2}	1.123×10^{-5}	1.123×10^{-5}	1.123×10^{-5}	3.945×10^{-7}	3.488×10^{-7}	3.639×10^{-7}	2.050×10^{-5}	1.148×10^{-5}	9.074×10^{-6}
	1.490×10^{-2}	4.677×10^{-6}	4.677×10^{-6}	4.677×10^{-6}	1.224×10^{-7}	1.081×10^{-7}	1.130×10^{-7}	8.233×10^{-6}	4.894×10^{-6}	3.947×10^{-6}
3	7.681×10^{-2}	8.049×10^{-5}	8.039×10^{-5}	8.038×10^{-5}	5.467×10^{-6}	4.567×10^{-6}	4.109×10^{-6}	2.767×10^{-4}	9.063×10^{-5}	4.817×10^{-5}
	3.994×10^{-2}	5.591×10^{-6}	5.589×10^{-6}	5.589×10^{-6}	2.175×10^{-7}	2.337×10^{-7}	2.064×10^{-7}	2.136×10^{-5}	1.636×10^{-5}	4.094×10^{-6}
	2.628×10^{-2}	1.043×10^{-6}	1.043×10^{-6}	1.043×10^{-6}	2.684×10^{-8}	3.006×10^{-8}	2.624×10^{-8}	4.109×10^{-6}	5.061×10^{-6}	9.155×10^{-7}
	1.997×10^{-2}	3.477×10^{-7}	3.477×10^{-7}	3.477×10^{-7}	6.806×10^{-9}	7.758×10^{-9}	6.731×10^{-9}	1.386×10^{-6}	2.288×10^{-6}	3.453×10^{-7}

Table A.4: Error values for the SMM method on an isotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the SMM correction sources aree constant in space and thus are represented exactly.

		$\|\varphi - \varphi_{\text{ex}}\|$		$\|\varphi - \Pi\varphi_{\text{ex}}\|$		$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$	
p	h	RT	HRT	RT	HRT	RT	HRT
1	3.994×10^{-2}	4.161×10^{-4}	4.161×10^{-4}	1.067×10^{-5}	1.067×10^{-5}	1.251×10^{-3}	1.251×10^{-3}
	1.997×10^{-2}	1.040×10^{-4}	1.040×10^{-4}	1.260×10^{-6}	1.260×10^{-6}	3.125×10^{-4}	3.125×10^{-4}
	1.331×10^{-2}	4.624×10^{-5}	4.624×10^{-5}	3.693×10^{-7}	3.693×10^{-7}	1.389×10^{-4}	1.389×10^{-4}
	9.985×10^{-3}	2.601×10^{-5}	2.601×10^{-5}	1.552×10^{-7}	1.552×10^{-7}	7.812×10^{-5}	7.812×10^{-5}
2	5.874×10^{-2}	1.411×10^{-5}	1.411×10^{-5}	1.301×10^{-6}	1.301×10^{-6}	1.629×10^{-4}	1.629×10^{-4}
	3.026×10^{-2}	1.922×10^{-6}	1.922×10^{-6}	8.331×10^{-8}	8.331×10^{-8}	2.254×10^{-5}	2.254×10^{-5}
	1.997×10^{-2}	5.523×10^{-7}	5.523×10^{-7}	1.542×10^{-8}	1.542×10^{-8}	6.495×10^{-6}	6.495×10^{-6}
	1.490×10^{-2}	2.295×10^{-7}	2.295×10^{-7}	4.740×10^{-9}	4.740×10^{-9}	2.701×10^{-6}	2.701×10^{-6}
3	7.681×10^{-2}	9.905×10^{-7}	9.905×10^{-7}	2.604×10^{-7}	2.604×10^{-7}	3.112×10^{-5}	3.112×10^{-5}
	3.994×10^{-2}	7.112×10^{-8}	7.112×10^{-8}	8.727×10^{-9}	8.727×10^{-9}	2.231×10^{-6}	2.231×10^{-6}
	2.628×10^{-2}	1.329×10^{-8}	1.329×10^{-8}	1.043×10^{-9}	1.043×10^{-9}	4.175×10^{-7}	4.175×10^{-7}
	1.997×10^{-2}	4.432×10^{-9}	4.432×10^{-9}	2.619×10^{-10}	2.619×10^{-10}	1.393×10^{-7}	1.393×10^{-7}
4	9.986×10^{-2}	3.566×10^{-7}	3.566×10^{-7}	4.712×10^{-8}	4.712×10^{-8}	4.258×10^{-6}	4.258×10^{-6}
	4.993×10^{-2}	1.157×10^{-8}	1.157×10^{-8}	8.170×10^{-10}	8.170×10^{-10}	2.027×10^{-7}	2.027×10^{-7}
	3.328×10^{-2}	1.525×10^{-9}	1.525×10^{-9}	6.896×10^{-11}	6.896×10^{-11}	2.712×10^{-8}	2.712×10^{-8}
	2.496×10^{-2}	3.619×10^{-10}	3.619×10^{-10}	1.209×10^{-11}	1.209×10^{-11}	6.473×10^{-9}	6.473×10^{-9}

Table A.5: Error values for the SMM method on a quadratically anisotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the angular flux used to calculate the SMM correction sources are represented with Y_p. Due to this, the maximum accuracy expected is order $p+1$.

		$\|\varphi - \varphi_{\text{ex}}\|$		$\|\varphi - \Pi\varphi_{\text{ex}}\|$		$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$	
p	h	RT	HRT	RT	HRT	RT	HRT
1	3.994×10^{-2}	1.898×10^{-3}	1.897×10^{-3}	3.177×10^{-4}	3.169×10^{-4}	1.717×10^{-2}	1.719×10^{-2}
	1.997×10^{-2}	4.721×10^{-4}	4.720×10^{-4}	6.213×10^{-5}	6.141×10^{-5}	8.772×10^{-3}	8.773×10^{-3}
	1.331×10^{-2}	2.096×10^{-4}	2.096×10^{-4}	2.522×10^{-5}	2.511×10^{-5}	5.876×10^{-3}	5.876×10^{-3}
	9.985×10^{-3}	1.178×10^{-4}	1.178×10^{-4}	1.355×10^{-5}	1.344×10^{-5}	4.415×10^{-3}	4.415×10^{-3}
2	5.874×10^{-2}	2.799×10^{-4}	2.827×10^{-4}	7.620×10^{-5}	8.578×10^{-5}	1.162×10^{-3}	1.274×10^{-3}
	3.026×10^{-2}	4.020×10^{-5}	4.036×10^{-5}	1.085×10^{-5}	1.144×10^{-5}	2.172×10^{-4}	2.423×10^{-4}
	1.997×10^{-2}	1.167×10^{-5}	1.167×10^{-5}	3.200×10^{-6}	3.228×10^{-6}	7.629×10^{-5}	8.254×10^{-5}
	1.490×10^{-2}	4.866×10^{-6}	4.894×10^{-6}	1.348×10^{-6}	1.448×10^{-6}	3.660×10^{-5}	4.013×10^{-5}
3	7.681×10^{-2}	8.182×10^{-5}	8.174×10^{-5}	1.608×10^{-5}	1.564×10^{-5}	3.797×10^{-4}	3.852×10^{-4}
	3.994×10^{-2}	5.670×10^{-6}	5.671×10^{-6}	9.841×10^{-7}	9.923×10^{-7}	5.892×10^{-5}	5.896×10^{-5}
	2.628×10^{-2}	1.055×10^{-6}	1.056×10^{-6}	1.638×10^{-7}	1.683×10^{-7}	1.728×10^{-5}	1.729×10^{-5}
	1.997×10^{-2}	3.513×10^{-7}	3.515×10^{-7}	5.118×10^{-8}	5.238×10^{-8}	7.665×10^{-6}	7.666×10^{-6}
4	9.986×10^{-2}	1.462×10^{-5}	1.471×10^{-5}	3.481×10^{-6}	3.833×10^{-6}	5.630×10^{-5}	5.975×10^{-5}
	4.993×10^{-2}	5.757×10^{-7}	5.776×10^{-7}	1.182×10^{-7}	1.273×10^{-7}	3.765×10^{-6}	3.797×10^{-6}
	3.328×10^{-2}	7.983×10^{-8}	8.056×10^{-8}	1.742×10^{-8}	2.052×10^{-8}	5.852×10^{-7}	6.179×10^{-7}
	2.496×10^{-2}	1.924×10^{-8}	1.943×10^{-8}	4.380×10^{-9}	5.160×10^{-9}	1.565×10^{-7}	1.638×10^{-7}

Table A.6: Error values for the SMM method on a quadratically anisotropic MMS test problem. The H1, RT, and HRT columns refer to the $Y_p \times W_{p+1}$, $Y_p \times RT_p$, and hybridized $Y_p \times RT_p$ discretizations, respectively. The error in the scalar flux, the error in the scalar flux when the exact solution is first projected onto Y_p, and the error in the current are presented for each method over a range of values of p. Here, the angular flux used to calculate the SMM correction sources are represented with Y_{p+1}. Due to this, the maximum accuracy expected is order $p+2$.

		$\|\varphi - \varphi_{\text{ex}}\|$		$\|\varphi - \Pi\varphi_{\text{ex}}\|$		$\|\boldsymbol{J} - \boldsymbol{J}_{\text{ex}}\|$	
p	h	RT	HRT	RT	HRT	RT	HRT
0	1.997×10^{-2}	1.564×10^{-2}	1.564×10^{-2}	5.301×10^{-4}	5.172×10^{-4}	1.028×10^{-2}	1.028×10^{-2}
	9.985×10^{-3}	7.827×10^{-3}	7.827×10^{-3}	1.323×10^{-4}	1.291×10^{-4}	5.138×10^{-3}	5.138×10^{-3}
	6.657×10^{-3}	5.219×10^{-3}	5.219×10^{-3}	5.878×10^{-5}	5.737×10^{-5}	3.425×10^{-3}	3.424×10^{-3}
	4.993×10^{-3}	3.914×10^{-3}	3.914×10^{-3}	3.305×10^{-5}	3.226×10^{-5}	2.568×10^{-3}	2.568×10^{-3}
1	3.994×10^{-2}	1.875×10^{-3}	1.875×10^{-3}	1.097×10^{-4}	9.773×10^{-5}	3.443×10^{-3}	1.350×10^{-3}
	1.997×10^{-2}	4.683×10^{-4}	4.683×10^{-4}	1.422×10^{-5}	1.256×10^{-5}	1.741×10^{-3}	3.569×10^{-4}
	1.331×10^{-2}	2.081×10^{-4}	2.081×10^{-4}	4.275×10^{-6}	3.754×10^{-6}	1.166×10^{-3}	1.675×10^{-4}
	9.985×10^{-3}	1.171×10^{-4}	1.171×10^{-4}	1.826×10^{-6}	1.594×10^{-6}	8.758×10^{-4}	9.902×10^{-5}
2	5.874×10^{-2}	2.714×10^{-4}	2.714×10^{-4}	2.706×10^{-5}	2.750×10^{-5}	2.855×10^{-4}	2.109×10^{-4}
	3.026×10^{-2}	3.877×10^{-5}	3.877×10^{-5}	1.847×10^{-6}	1.917×10^{-6}	3.918×10^{-5}	3.010×10^{-5}
	1.997×10^{-2}	1.123×10^{-5}	1.123×10^{-5}	3.488×10^{-7}	3.639×10^{-7}	1.148×10^{-5}	9.074×10^{-6}
	1.490×10^{-2}	4.677×10^{-6}	4.677×10^{-6}	1.081×10^{-7}	1.130×10^{-7}	4.894×10^{-6}	3.947×10^{-6}
3	7.681×10^{-2}	8.039×10^{-5}	8.038×10^{-5}	4.567×10^{-6}	4.109×10^{-6}	9.063×10^{-5}	4.817×10^{-5}
	3.994×10^{-2}	5.589×10^{-6}	5.589×10^{-6}	2.337×10^{-7}	2.064×10^{-7}	1.636×10^{-5}	4.094×10^{-6}
	2.628×10^{-2}	1.043×10^{-6}	1.043×10^{-6}	3.006×10^{-8}	2.624×10^{-8}	5.061×10^{-6}	9.155×10^{-7}
	1.997×10^{-2}	3.477×10^{-7}	3.477×10^{-7}	7.758×10^{-9}	6.731×10^{-9}	2.288×10^{-6}	3.453×10^{-7}