Mixed-Cell Methods for Diffusion Problems in Multiphase Systems

by

Nick Patterson

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Applied Physics)
in The University of Michigan
2014

Doctoral Committee:

Professor Katsuyo S. Thornton, Chair
Professor R. Paul Drake
Professor Krzysztof J. Fidkowski
Professor Edward W. Larsen
Professor William R. Martin

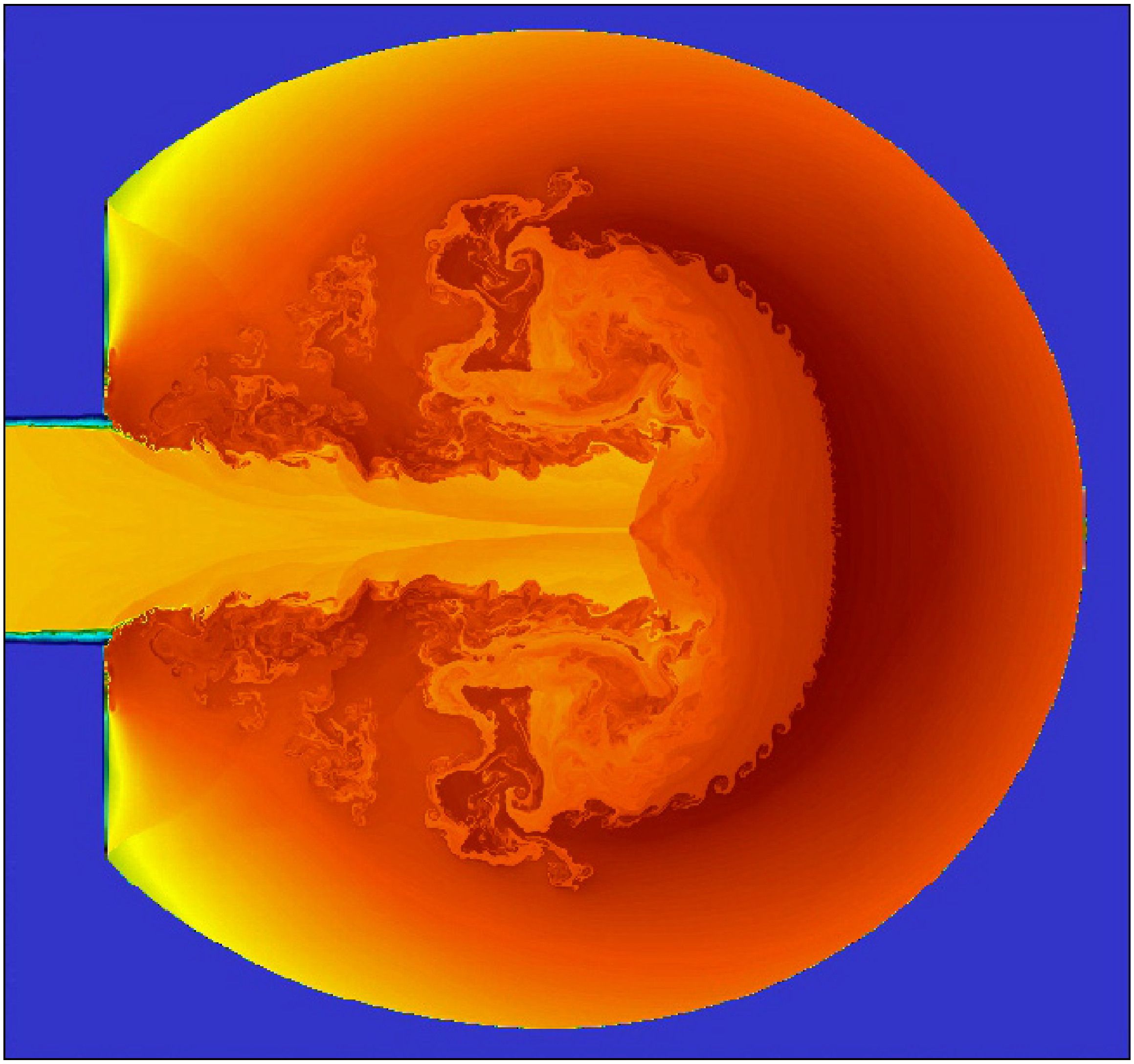


Image of a shock wave I created during my first year of grad school. It simulates a laser hitting an enclosed box of gas from the left. The colors represent temperature on a logarithmic scale, where blue is coldest and red is hottest. The simulation was created using FLASH, a hydrodynamic code, and the figure was made using VisIt.

ACKNOWLEDGEMENTS

First, I would like to thank my lovely Katie for taking care of me and putting up with my work schedule during the painful process of writing this thesis. Also, I want to acknowledge all of the grammatical editing from her and her mother.

Second, I would like to thank Eric Myra (research scientist at the University of Michigan) and my committee for all of their edits, inputs, suggestions, questions, and time. Katsuyo spent an inordinate amount of her time carefully reading every word and painstakingly ensuring that each verb and sentence added to a logical picture. Also thanks to the many group members in Thornton lab who helped edit sections of my thesis and provided input for my oral defense.

Third, I would like to thank Scott Runnels at Sandia Labs and Jim Morel at Texas A&M for their private discussions regarding the Support-Operators Method. Jim was able to give me some insight on the general functioning of the method while Scott was able to elucidate some of the mathematical and computational steps that were troubling me. Their combined input enabled me to write my own version of the SOM and to extend the method to better account for mixed cells.

Fourth, I would like to thank Chong (C. H. Chang) and Tony (A. J. Scannapieco) at Los Alamos National Laboratories (LANL) for their collaborations in making the multi-T model. Their suggestions of test problems were instrumental; Chapters 7 and 8 are based on their recommendations. I would also like to acknowledge the Predictive Sciences Academic Alliances Program (PSAAP) in NNSA. The PSAAP program funded my visit to Los Alamos which made the multi-temperature work possible and led to the mathematically sound testing used for all the mixed-cell models presented in this thesis.

This research was supported in part through computational resources and services provided by Advanced Research Computing at the University of Michigan, Ann Arbor. This work was funded in part by the University of Michigan and by the Predictive Sciences Academic Alliances Program in NNSA-ASC via grant DEFC52-08NA28616.

TABLE OF CONTENTS

CHAPTER

LIST OF FIGURES

Figure

LIST OF TABLES

<u>**Table**</u>

LIST OF APPENDICES

Appendix

LIST OF ABBREVIATIONS

ALE Arbitrary Lagrangian Eulerian

AMR Adaptive Mesh Refinement

CRASH Center for RAdiative Shock Hydrodynamics

CFD Computational Fluid Dynamics

DNS Direct Numerical Simulation

EoS Equation of State

FDM Finite Difference Methods

FEM Finite Element Methods

FLD Flux Limited Diffusion

FVM Finite Volume Methods

IIM Immersed Interface Method

IMC Implicit Monte Carlo

IMPES Implicit Pressure and Explicit Saturation

LES Large Eddy Simulation

LHS Left-Hand Side

LSM Level Set Methods

LTE Local Thermodynamic Equilibrium

MC Monte Carlo

MES Method of Exact Solutions

MFD Mimetic Finite Difference methods

MFP Mean Free Path

MMS Method of Manufactured Solutions

MoF Moment of Fluid

NS Navier-Stokes

RANS Reynolds Averaged Navier-Stokes

RHS Right-Hand Side

SOM Support-Operators Method

SoV Separation of Variables

SPD Symmetric Positive-Definite

VoF Volume of Fluid

ABSTRACT

Mixed-Cell Methods for Diffusion Problems in Multiphase Systems
by
Nick Patterson

Chair: Katsuyo Thornton

We simulate diffusion in multimaterial systems with a cell-centered Eulerian mesh in two dimensions. A system with immiscible fluids contains sharp interfaces. An Eulerian mesh is fixed in space and does not move with the material. Therefore, cells with an interface contain multiple fluids; these are known as mixed cells. The treatment of mixed cells can vary in computational cost and accuracy. In some cases, the primary source of inaccuracy can be attributed to approximations made in modeling the mixed cells. This thesis focuses on the treatment of mixed cells based on the diffusion approximation of the transport equation. We introduce five subgrid, mixed-cell models. Two models have a single temperature for each cell, while the other three allow a separate temperature for each phase. The single-temperature models are implemented using the Support-Operators Method, which is derived herein. The first single-temperature model utilizes an effective tensor diffusivity that distinguishes diffusion tangent and normal to the interface. The second single-temperature model specifies a unique diffusivity in each corner of a mixed cell, which is effectively a mesh refinement of the mixed cell. The three multi-temperature models have increasingly accurate levels of approximation of the flux: (i) flux is calculated between cell-centers for each phase, (ii) flux is calculated between the centroid of each phase, and (iii) flux normal to an interface is calculated between centroids of each phase. The physical interpretations of these models are: (i) each phase occupies the entire cell, (ii) oblique flux is continuous, (iii) only normal flux is continuous. The standard approximation, using the harmonic mean of the diffusivities present in a mixed cell as an effective diffusivity, is also tested for comparison. We also derive two time-dependent analytical solutions for diffusion in a two-phase system, in both one and two dimensions. With the standard model as a reference point, the accuracy of the new models is quantified, and the convergence rates of the error are determined between pairs of spatial resolutions for the two problems with analytical solutions. Simulations of multiphysics and multimaterial phenomenon may benefit from increased mixed-cell fidelity achieved in this dissertation.

CHAPTER I

Introduction

In this thesis, we study mixed cells, computational cells containing more than one phase, in the context of cell-centered diffusion in two dimensions (2D). Mixed cells occur when the interfaces do not align with the computational grid boundaries. This is common in multiphase flow simulations (discussed in the next section). Mixed-cell inaccuracy often dominates the error of simulation results since each phase has its own diffusivity, which violates the discretization assumption (values in a cell are constant or smooth), leading to a jump discontinuity of diffusivities at the interface within the cell. A multiphysics, multiphase computational scheme that is second-order accurate could lose its accuracy to first-order or lower due to mixed-cell error. In other words, a higher-order method may become limited to the accuracy of numerical solution in the mixed cells, essentially wasting the computational effort that is providing higher accuracy away from interfaces. Furthermore, the values of the solution at the interfaces are sometimes more important than in the bulk in many multiphase flow problems, such as when reactions or energy exchange occurs along the interface. Mixed cells are thus an important research problem; an increase in the accuracy of mixed cells would enable multiphase computational schemes to regain the theoretical order of accuracy of the discretization.

The goal of this thesis is to improve mixed-cell accuracy, ideally obtaining a mixed-cell model that is second-order accurate in various p-norms. E_∞, the ∞-norm of the error as defined in Section 5.2, is the best measure for mixed-cell accuracy because it finds only the maximum error, which typically occurs along the interfaces. Having second-order convergence of E_∞ is cogent evidence of an accurate treatment of mixed cells. An alternative means to measure the accuracy of a method is if a model consistently yields lower error at coarser grids compared to a standard method. A test case or known solution is needed to compute the errors, but no standard test problem exists against which various mixed-cell models have been benchmarked. Another goal of this thesis is to select and evaluate time-dependent, multiphase analytical solutions that can handle an arbitrary diffusivity ratio to be used as a test problem to quantify the accuracy of mixed-cell models.

1.1 Multiphase Flow

This work is motivated by the physics of fluid dynamics with multiple phases or materials present in the system. Multiphase flows occur in the context of transport phenomena in many diffusion related processes in physics and industrial applications. Porous-media flow is a far-reaching field with numerous examples of multiphase flow, including groundwater and contaminant flow through reservoirs. Oil and fluid flow is important for petroleum engineering, such as oil extraction from reservoirs, which also involves porous-media flow. Heat conduction in heterogeneous phases is a critical phenomenon for industrial applications, such as heat dissipation that occurs with solidification during a casting, as well as the cooling of engines or turbines. Fluid mixing that occurs with chemical reactions at fluid interfaces is another example that is found in many industrial applications. Radiation transport has many examples in which multiphase flow is important, such as plasmas in a nebula, supernovae, inertial confinement fusion, and high-energy density laboratory astrophysics.

These are just some of the multifarious examples that may use computational modeling to aid in the research of basic phenomena and the development of industrial applications. Interfaces are the boundaries between phases in multiphase systems. We use ‘phases’ throughout this thesis to indicate different states of the same material or different materials that are immiscible. Multiphase, in this context, refers to a system with more than one identifiable phase. A system with two materials that is finely mixed or homogeneous would not be considered a multiphase system. For problems examined in this work, we assume a sharp interface, i.e., an interface has no width.

1.2 Description of Dynamics

Fluid motion can be described from one of two points of view [117]: the Lagrangian description or the Eulerian description, which are discussed below. (In the forthcoming terminology, mesh and grid are equivalent, and a cell and an element are used interchangeably.)

1.2.1 Lagrangian Description

The Lagrangian description follows individual fluid particles/parcels as they move along the system [117]. The mesh for a numerical scheme using the Lagrangian viewpoint is attached to the fluid and moves as the system flows, making Lagrangian meshes an option for computational hydrodynamics. The primary benefit of a Lagrangian method for interfaces is that the interface position is naturally maintained. The interface position is known and remains sharp. The moving mesh leads to so-called 'body-fitted grids,' where the grid (or mesh) is shaped such that the interfaces occur only along the boundaries of the cells adjacent to interfaces. Each cell represents an arbitrarily shaped 'particle' or parcel of fluid that deforms with the phase from its original shape. Accordingly, every cell is a *pure cell*, which is a cell containing only one phase. Absence of mixed cells is advantageous because the physics and numerical methods are well developed for pure (single-phase) cells, but not for mixed cells. Additionally, the discretization process assumes that all variables and parameters are smooth or constant within each cell, which pure cells generally follow, while multiphase cells do not.

There are inherent disadvantages to Lagrangian methods. For example, interpolation of the discrete data is required to compute the solutions. There is a high computational cost in tracking the interfaces and moving the mesh, and more complicated interfaces increase the computational demand. The dynamic mesh leads to skewed and non-uniform cells, which may reduce the overall accuracy as compared to a uniform or smooth mesh. Furthermore, three-dimensional implementation is especially difficult in a Lagrangian method. Under most circumstances, these are acceptable drawbacks. However, some of the more challenging issues with Lagrangian techniques occur when the interfaces evolve quickly, such as with high velocity flows, systems with shock waves, or processes involving rapid phase changes. The costly mesh regeneration and interface updates may become prohibitively expensive when the interface evolves rapidly. Additionally, the interfaces may move so quickly that catastrophic failure occurs, where the mesh becomes entangled. A significant vorticity can also lead to mesh entanglement. These problems are sometimes remedied by remapping, in which the grid is projected back onto a simple rectangular mesh. When remapping a Lagrangian method, the position of the interface must be estimated, resulting in an additional source of error. Additionally, remapping systems with significant vorticity yields unphysical results due to spurious vorticity dissipation. A Lagrangian technique may incur mixed cells after such a remap, making the mixed-cell approaches in this thesis applicable in such a case.

1.2.2 Eulerian Description

The Eulerian description observes the state of a fluid at given locations [117]. Thus, an Eulerian method maintains a stationary grid. Rather than moving the mesh with the flow, the flow is tracked by the density, velocity, and other state variables that describe the flow. Unlike Lagrangian methods, Eulerian methods do not encounter mesh entanglement because the mesh is stationary, even when the interfacial geometry is complicated or flow distortions are large. Eulerian methods are simpler to implement computationally in both two dimensions and three dimensions (3D).

Since the interface lies within an Eulerian mesh, Eulerian methods require a method to determine interfacial positions; see Kothe and Rider [116] for a comparison of various methods. We provide two examples for such methods: level set methods (LSM) and volume of fluid (VoF) methods [68], both of which handle interface merging and splitting [52]. VoF methods and LSM are tools to capture/reconstruct the interface in an Eulerian mesh, and they must be used in conjunction a scheme that calculates the fluid field equations.

Level Set Methods. Level set methods (LSM), which capture the interface location via a level-set function, were introduced by Osher and Sethian [165]. A level-set function takes a value, typically zero, on the interface; as the interface moves, the zero point of the level-set function also moves. The sign of the level-set function, positive or negative, distinguishes the two phases. The signed distance function, whose absolute value is equal to the distance to the nearest interface, is often used as the level-set function.

Advantages of LSM include well-defined topological merging and breaking [116], comparable complexity in 3D as in 2D [52], and the ability to treat interfaces with curvature. The interface normal and curvature can be found from the gradient of the level-set function [52]. The interfacial positions are not explicitly evolved in this method [204]. While LSM naturally treat topological changes of the interface, the cost to update the distance function can be expensive for rapidly evolving interfaces [116]. Interfaces tend to be smoothed in LSM [52], which leads to mass loss or gain; this is the primary disadvantage of LSM [116]. Modifications and variations of LSM have been proposed to correct the mass conservation issue [18, 201]. Current methods are not locally conservative but are effectively conservative globally [18].

Volume of Fluid. The volume of fluid (VoF) methods, introduced by Hirt and

Nichols [94] and refined by Youngs [220], advect the fluids in one step and reconstruct the interface in the second step [68]. The volume fraction, a scalar function, gives the percentage or fraction of each fluid in a cell. A mixed cell has a volume fraction between zero or one, while a pure cell has a volume fraction equal to zero or unity [183]. The interface advection in VoF methods leads to numerical diffusion since the exact interface position is not tracked. After every volume fraction update, the interface is reconstructed based on the volume fraction of a cell and its neighbors [183]. There are a variety of ways to approximate the interfaces and calculate interface normals.

VoF methods are simple, flexible, and economical [116]. Large flow distortions and interfacial motions are treated easily, and topological changes do not increase the complexity of the algorithm. Perhaps the most important feature of the VoF methods is that they conserve the mass of each fluid component [70]. VoF methods are desirable for rapidly moving flows (where Lagrangian methods and LSM have difficulties) due to their stability, robustness, and conservative nature.

The primary disadvantage of VoF methods is the loss of exact interfacial position; only discrete volume data is retained instead, and volume fractions alone do not guarantee unique interface reconstruction. Additionally, unlike LSM, VoF methods do not naturally provide information required to compute local curvature [52]. Additional improvements such as the Moment of Fluid (MoF) method [70], which tracks the centroids of the phase volumes, allows the interface to be reconstructed uniquely and remedies these issues.

1.3 Numerical Methods

1.3.1 Finite Volume Methods and Finite Element Methods

Finite volume methods (FVM) are the most widely applied scheme in computational fluid dynamics (CFD) [93]. FVM use a conservative discretization, where the integral conservation laws are applied to each control volume. 'Finite volume' refers to this control volume, which is a cell in the mesh for this system. The cells in the mesh are connected through surface fluxes, which arise from converting a divergence integral to a surface integral (via the divergence theorem). FVM can be used on any grid type, including unstructured grids. FVM may employ vertex-based or cell-centered elements. The elements may change shape or position, allowing them to be used with Lagrangian meshes. However, FVM are also frequently used with Eulerian meshes as well. By discretizing the conservation equation without requiring basis functions, the FVM are conceptually simpler than finite element methods.

Finite element methods (FEM) [93, 117] are similar to FVM: they discretize the mesh into cells or elements, they use an integral governing equation, and they can use arbitrary meshes. FVM use cell-centered variables, while FEM are usually vertex-based, as opposed to face-centered or cell-centered. The solution is represented as a sum of basis functions, also known as shape functions, interpolation functions, and trial functions. The performance of the code depends on the choice of the basis functions, which should be chosen appropriately for a given set of equations and boundary conditions. FEM are generally more numerically stable and accurate than FVM. However, FEM are more computationally expensive, more difficult to code, and are less intuitive than finite volume or finite difference methods (FDM) (discussed in the next section). FEM are commonly used in structural engineering, but are also used in fluid dynamics [93, 117]. For CFD, the Galerkin formulation of FEM and their many variants are the most successful class of FEM [117, 151, 152].

1.3.2 Spectral Methods

Spectral methods are related to FEM because they both represent the solutions using a set of basis functions. Spectral methods can be considered *global* methods because each basis function is defined over the entire computational domain, while FEM can be considered *local* methods because the basis functions are nonzero only in a small area of the computational domain (an element). Spectral methods have superb, often exponential, convergence rates (compared to FEM and FVM which are often second-order). While less expensive than FEM, the global nature of spectral methods leads to difficulties when the boundary has complex geometries or when the phase coefficients are discontinuous.

1.3.3 Finite Difference Methods

FDM are based on Taylor expansions of functions and can be formulated for an arbitrary order of accuracy based on the number of terms used in the expansion [93]. The governing equations are discretized directly, unlike FVM that first integrate the equations over volume or FEM that define the solution in terms of basis functions. As opposed the cells in FVM and FEM, FDM use a collection of individual points in space. Thus, the mesh is not partitioned, and a solution must be interpolated to provide values between grid points. FDM are generally simple to code, especially for uniform grids, but require structured meshes [93]. The restriction on grid types makes FDM more suited for Eulerian grids.

1.4 Volume Fractions & Mixed Cells

1.4.1 Volume Fraction

The volume fraction is an important parameter that characterizes a mixed cell, and therefore, we here provide a discussion. Volume fractions appear not only in Eulerian VoF methods but also, in fact, in a variety of methods as an intermediate step. The difference is that VoF methods must average the phase properties within a mixed cell and advect the interface, while other methods that use volume fraction may assume immiscible phases, track explicit interfacial positions, and/or update the interface more precisely. LSM, for example, maintain knowledge of the interface location via the level-set function, but may be coupled with a method that uses volume fractions in order to compute the flow field. Another common technique that may use volume fractions is an Arbitrary Lagrangian Eulerian (ALE) method, where the interface is updated using a Lagrangian method, allowing the cells to flow along with the fluid. A remapping step then occurs in which the new Lagrangian mesh is mapped to a fixed (Eulerian) grid. Field calculations are then performed on this Eulerian grid, using volume fractions. The results are then used for the next Lagrangian interface update. ALE methods reduce the numerical diffusion issues that occur in VoF methods and do not eliminate the problem of vorticity dissipation.

1.4.2 Mixed Cells

This dissertation focuses on the treatment of mixed cells, and the exposition is given under the assumption that only volume fractions are known (as in a VoF algorithm). However, as discussed above, the application of an advanced mixed-cell treatment has a larger scope than just to a VoF method, pertinent to any scheme that involves mixed cells, including some LSM and Lagrangian mesh techniques. This work focuses on the method to account for the effect of mixed cells, not on following interface positions. Consequently, the problems solved in this thesis have stationary interfaces in order to focus exclusively on the treatment of mixed cells. This is a rational simplification because the subcell models use only volume fractions and interface normals, allowing this work to apply to a variety of multiphase methods. Moreover many multiphase codes use operator splitting, updating the interface position in a hydrodynamics portion of the code and then updating the flow properties in another portion of the code. We concentrate our study on cell-centered diffusion solvers because cell-centered methods are more easily integrated with hydrodynamics solvers used, for example, in radiation hydrodynamics.

Mixed cells are a large source of numerical error. The error in mixed cells is larger than the rest of the domain, typically resulting in first-order accuracy near an interface and higher-order elsewhere. Therefore, mixed cells dominate the error in simulations, requiring attention (e.g., mesh adaptivity) to maintain overall computational accuracy. Since mixed cells have the potential to reduce a high-order scheme to first-order, mixed cell accuracy is an important research topic.

The most common, and perhaps most accurate, technique to study multiphase flow is to ensure that all cells are pure. This is typically achieved by aligning the interfaces with the grid boundaries. This is straightforward when the interfacial geometries are simple but becomes problematic when they are complex. Body-fitted grids (usually with FEM or FVM) are used to align element faces with interfaces. Body-fitted grids should be used whenever practical if phase discontinuities are a significant source of error. However, in some cases, the cost of body-fitted grids outweighs their benefit. If the interface is stationary, the generation of a body-fitted mesh is needed only once, and the cost is justified. If the interface is time-dependent, moving the mesh at each time step represents a large computational cost [215, 216, 218], and simply allowing the interface to move relative to a uniform grid may be more appropriate [123]. For example, Lagrangian methods may be too costly or unstable for problems involving high velocity flows or shock waves. ALE codes represent a middle ground between body-fitted grids (a Lagrangian technique) and stationary grids (an Eulerian technique). ALE methods still incur mixed-cell effects on the Eulerian steps, while potential grid distortions are also possible during the Lagrangian step. Our analysis is applicable to systems in which mixed cells cannot be completely eliminated, including Eulerian, ALE, and some Lagrangian methods.

Cutting a mixed cell into two or more pure cells is a logical strategy for handling mixed cells. While cutting the cell can be effective, it leads to non-uniform grids and cells of complex shape. For the same reasons above, we do not consider this case, although the mixed-cell models presented herein do *effectively* cut the mixed cells. We restrict our analysis to a Cartesian grid but note that a logically rectangular, AMR-generated, or unstructured grid would be able to utilize a similar algorithm, with modifications to account for the neighbor sizes and positions.

When mixed cells are not avoided or cut, the phase properties are typically homogenized (averaged) over each cell. The three most common averages are the arithmetic, geometric, and harmonic mean. For diffusivity, the harmonic mean is the standard approach. Homogenization leads to artificial mixing that potentially results in phase properties that do not reflect the physical system. This occurs, for example, with opacities in radiation transport problems that may vary by greater than six orders of magnitude. A simple average of opacities of an opaque medium and a transparent medium does not reflect the radiation transport in the combined two-phase medium. Homogenization often leads to inaccuracies, with an error larger than the discretization error of the rest of the domain. Since the harmonic mean for diffusivity is the standard technique, we use it as a reference point to which to compare our new mixed-cell models.

As an alternative to viewing a mixed cell as a new phase with intermediate values, a mixed cell can instead be considered as a cell containing immiscible phases. In this interpretation of a mixed cell, an interface thus changes from smeared to sharp. A mixed cell would then maintain the properties of each phase, where appropriate. However, by considering the phases separately, we require a subcell or subgrid model, in which structures below the smallest element are considered.

Adaptive mesh refinement (AMR) algorithms, in which a cell is partitioned into multiple smaller cells in regions that are determined to require a higher resolution, may be used to increase the grid resolution of interfacial regions. While this approach reduces the total volume of mixed cells, the effect of mixed cells is not eliminated, only abated, degree of which depends on the level of refinement used. Thus, mixed cells are still relevant to an AMR code, just at a smaller extent. Furthermore, AMR algorithms introduce substantial complexity, such as requiring varying time step sizes for differently sized cells and distorting the band structure of the coefficient matrix, and thus are not always advantageous.

1.5 Contributions

This dissertation focuses on subgrid models for mixed cells. We introduce five new models that are compared with each other as well as with a standard treatment (harmonic mean for diffusivity). Comparisons are quantified using convergence rates of each model on four test problems: a 1D steady-state manufactured solution, a 2D steady-state manufactured solution, a 1D dynamic boundary value problem, and a 2D dynamic boundary value problem.

We introduce five new mixed-cell models, two of which are single-temperature models and three of which are multi-temperature models. We include the standard method (harmonic mean of diffusivities) as the first single-temperature method, denoting it as the S1 model. This is used as a reference against which to measure the benefit of the new models.

The S2 model approximates the diffusive flux along an interface with an effective diffusivity tensor. The flux perpendicular to an interface is modeled with the harmonic mean, while the flux parallel to an interface uses the arithmetic mean. A tensor diffusivity contains the two values, rotated according to the interface orientation. This is physically reasonable because the flux normal to an interface between an opaque and transparent medium is limited by the opaque medium, and the harmonic mean gives a diffusivity that is close to that of the opaque material. The flux tangent to an interface flows freely through the transparent medium but is limited in the opaque medium. The arithmetic mean accounts for the volume ratio of the transparent medium and the opaque medium.

The S3 model takes advantage of a particular numerical implementation (Support-Operators Method), in which the diffusivity can be defined separately in each corner (vertex) of a cell, while still utilizing a stable, cell-centered method. The flexibility of having up to four different diffusivities in a 2D mixed cell allows for an effective mesh refinement, similar to AMR or cutting the cell. However, unlike those methods, the S3 model does not increase the algorithm complexity, lead to nonrectangular cells, or disrupt the band structure of the coefficient matrix. By splitting the cell into pure cells, we avoid any phase homogenization, allowing for a more physically accurate computation of diffusion. This model is easily extrapolated to 3D.

The multi-temperature models assume that each phase in a mixed cell has a distinct temperature. The M1 model uses a rough approximation in which the distances *between* adjacent cells are set to fixed values (Δx for left and right, Δy for up and down), while the separation between distinct phases *within* a mixed cell is set to about half the cell width. This subcell model introduces a more physically accurate model of separate temperatures for distinct phases while adding little computational complexity. This model avoids complicated interface reconstruction and, therefore, is applicable to 3D as well, but it involves a significant simplification.

The M2 model refines the M1 model by computing distances more precisely. Instead of assuming a uniform spacing between phases, the centroid of each phase existing within every cell is determined. The distances are then computed as centroid-to-centroid distances, which is physically more accurate. The disadvantage of this method is that the interface must be reconstructed, and centroids must be located. However, this model could be seamlessly integrated with a larger computational scheme that utilize interface reconstruction methods that track both volume and centroid location, such as the MoF, which determines centroids. This method can be extrapolated to 3D, provided the interface reconstruction methods employed are capable of providing the centroids.

The M3 model has the highest refinement among the multi-temperature methods we examine. Similar to the M2 model, centroids are determined based on a linear interface reconstruction. However, since centroid-to-centroid distances may be oblique to the interfaces separating cells and/or phases, the M3 model computes the normal distance to each face from each centroid. Thus, the M3 model makes an improvement to the M2 model to enhance the accuracy by accounting for the obliqueness of the vector between two centroids with respect to the face normal. As with the M2 model, the M3 model can be expanded to 3D with interface reconstruction methods that provide the centroid information.

1.6 Synopsis

The remainder of this thesis is organized into the following chapters.

Chapter II: Background and History
This chapter provides background information on fluid dynamics and diffusion. Applications of diffusion and multiphase flow are discussed. Radiation transport is described at length, including a discussion of some of the most common computational methods as well as a derivation of the diffusion approximation. The motivating problem is then described, followed by a description of the problem this dissertation addresses: heat conduction. While the language of the problem and subsequent chapters is in terms of heat conduction, this work applies to other problems that can be formulated in the form of the diffusion equation, such as porous media flow and diffusion-approximated radiative transfer. Previous work related to or involving mixed cells is then reviewed. The chapter concludes with a discussion of iterative solution methods, in particular, the conjugate gradient solver.

Chapter III: Single-Temperature Methods
We introduce three single-temperature models (one existing and two newly developed models). All three models are implemented using the Support-Operators Method (SOM), a mimetic approach to FDM. While only one of the three models requires this particular implementation, we use it for all three for a consistent means to compare each model. SOM is an advanced discretization technique that treats discontinuous diffusivities in a stable and accurate manner. One peripheral benefit of this section is a coherent derivation and explanation of SOM. SOM have the potential to be used in place of FVM and FEM in many instances, therefore the clear presentation of this

method is useful for future work.

Chapter IV: Multi-Temperature Methods
This chapter presents three new multi-temperature models we developed, based on volume averaging of the diffusion equation over a cell. Volume averaging leads to a volume-fraction-dependent equation, with an exchange term for mixed cells. The derivation and discretization of the three models are presented.

Chapter V: Numerical Results: Code Verification
This chapter discusses code verification. We specify means to quantify the accuracy of a numerical method through error analyses. This is followed by four numerical problems that test various aspects of our single- and multi-temperature codes. These tests focus on pure-cell cases to verify the codes.

Chapter VI: Numerical Results: 1D & 2D Manufactured Solution
This chapter discusses the method of manufactured solutions (MMS) as a means to create test problems. We create two mixed-cell test problems using the MMS method, one each in 1D and 2D. We then test each mixed-cell model on this analytic solution.

Chapter VII: Numerical Results: 1D Test
This chapter benchmarks the mixed-cell models against a 1D analytical solution. The analytical solution describes a two-phase composite-medium problem. This is an eigenvalue problem with a Fourier-series solution. The simulations are performed for various diffusivities and interface configurations. Convergence rates for each model in three p-norms are given for each test case.

Chapter VIII: Numerical Results: 2D Test
This chapter benchmarks the mixed-cell models against a 2D analytical solution. The analytical solution is a 1D radial solution in cylindrical coordinates. This radial solution is then projected onto a 2D grid, resulting in mixed cells at any resolution because a Cartesian mesh cannot resolve a circular interface. The mixed-cell models are compared for three different diffusivity ratios, and convergence rates for each model in three p-norms are given for each test case.

Chapter IX: Summary
The last chapter summarizes the new models and their performance. Additionally, several specific steps for future work are recommended.

CHAPTER II

Background and History

This chapter provides a background on multiphase flow and diffusion, as well as the approximations and numerical methods commonly implemented in the literature. We begin our discussion in Section 2.1 with an introduction to diffusion and fluid dynamics. We review some of the applications of diffusion, especially those that also involve multi-fluid flow in Section 2.2. Radiation transport is the motivation and key application of this work and is discussed in detail in Section 2.3. After a brief description of radiation transport phenomenon, we summarize a stochastic computational method to model it. We then introduce a deterministic, continuum model for radiation transport, followed by an overview of approximations to this model, including the diffusion approximation. After an introduction to diffusion, fluids, and radiation transport is completed, we focus on interfaces.

The goal of this thesis is to examine how various methods of treating interfaces perform numerically. Section 2.4 provides a motivation for considering interfaces as an interesting research topic. We show that radiation transport and phase heat conduction can both be modeled with a similar mathematical formulation, allowing the mixed-cell models presented in Chapters III-IV to be applicable to both phenomena. After presenting the simplified problem, we provide a thorough overview of previous work on interface problems. We conclude the chapter with a discussion of iterative methods and efficient sparse matrix solvers in Section 2.5.

2.1 Introduction to Diffusion and Fluid Dynamics

2.1.1 Diffusion

Any phenomenon that can be described as a *random walk* can be modeled by diffusion. A random walk can be illustrated by the drunken-walk analogy: an inebriated bar patron attempts to walk home after imbibing to excess. He takes one step and falls. He stands up, facing a direction independent of how he fell. He takes one step forward in this random direction and falls again. This process repeats with a constant step size. To determine the inebriated man's progress, one can calculate the expected value of distance from the initial position as a function of the number of steps. The average of many instances of possible walks results in an expected distance, although a single particular walk may have a distance much longer or shorter than the expected value. Such a process describes two-dimensional (2D) diffusion, but can be extended to lower or higher dimensions. Diffusion processes describe many everyday phenomena, such as how a drop of dye spreads in still water to become turbid, or how heat conducts from the stovetop through a cast-iron pan. Even the movement of crowds of people, each making individual decisions, can be modeled with diffusion, such as traffic patterns or pedestrians leaving a stadium. The financial industry uses diffusion for pricing and market predictions [90]. Perhaps the most sensational application of diffusion was by Albert Einstein [76], who explained that the random motion, called Brownian motion, of pollen suspended in water is due to collisions with individual water molecules. This work was extraordinary because it changed the minds of many of the members of the scientific community who had previously rejected the existence of the atom [180]. Although microscopic or molecular motions physically cause diffusion, diffusion can be described mathematically at the continuum level. For this reason, diffusion is a prevalent approximation and, when applied to appropriate systems, is highly accurate without the need to model the dynamics of individual atoms or molecules.

Fick [82] empirically derived diffusion by measuring the concentration of salt in tubes of water of varying lengths. The governing equations of diffusion are still known as Fick's Laws, which are comprised of two equations,

$$\vec{F} = -D\vec{\nabla}U \tag{2.1a}$$

$$\frac{\partial U}{\partial t} = -\vec{\nabla}\cdot\vec{F} + Q, \tag{2.1b}$$

where U is a scalar field that can represent the density of a variety of conserved quantities, such as energy or the number of particles. We shall simply refer to U as concentration without specifying the species or type of conserved quantity. $\vec{F}$ is the flux or current (amount of the conserved quantity crossing a unit surface area per unit time), D is the diffusion coefficient or diffusivity, and Q is a source or sink of the species per unit volume. The negative sign in Equation (2.1a) shows that the flux is directed from higher to lower concentration.

The diffusion equation, Equation (2.1), describes the change in concentration. The diffusion coefficient (D), amongst its many names and roles, can be summed up as the mobility of the diffusive species to transport in response to a driving force. The situation in which the species diffuses independently of direction is called isotropic diffusion; here, D is a scalar. If, however, there is a directional dependence, $\mathbf{D}$ is a tensor of size 2×2 in 2D and 3×3 in 3D to reflect the anisotropy. The literature makes a distinction for two different cases of tensor diffusion: diagonal tensor and full tensor. A diagonal tensor has zeros for all off-diagonal terms. The most general case, the full tensor problem, occurs when any of the off-diagonal values are nonzero. Diagonal tensor problems are often treated separately because they are less complicated to implement numerically, requiring, for example, only a five-point rather than nine-point stencil in 2D [7].

The left-hand side (LHS) of Equation (2.1b) represents the rate of change in concentration with time at a fixed point. The first term on the right-hand side (RHS) represents the rate of net flow into an infinitesimal volume element. The last term is a source (Q), which could originate from an endothermic or exothermic chemical reaction in a heat diffusion model.

Equations similar to the diffusion equation appear in other contexts. In slow, viscous fluid flow, Darcy's law states that

$$\vec{F} = -k\vec{\nabla}P,$$

where $\vec{F}$ here is the fluid flux, P is the pressure, and k is the permeability of the material through which the fluid is flowing. Fourier's law for heat conduction states that

$$\vec{F} = -k\vec{\nabla}T,$$

where $\vec{F}$ here is the heat flux, T is the temperature, and k is thermal conductivity of the material through which the thermal energy is flowing. In electromagnetics, the generalized Ohm's law states that

$$\vec{J} = -\sigma\vec{\nabla}\Phi,$$

where $\vec{J}$ is the electric current, Φ is the electric potential, and σ is the electric connectivity of the material through which the current is flowing. All three of these cases have a form identical to Fick's first law: flux of some carrier (fluid, thermal energy, charge) is equal to a carrier-dependent constant multiplied with the gradient based 'driving force.' Note that the material constant can vary with the local conditions, giving rise to nonlinearities. All of these cases can be solved numerically using similar techniques. Consequently, diffusion models and solvers are powerful tools that can be applied to many different phenomena.

2.1.2 Fluid Dynamics

The field dedicated to the study of fluid flow is hydrodynamics. The description in this section is based on Hirsch [93], who gives a thorough introduction to fluid dynamics. Conservation laws for three quantities define the evolution of fluid flow: mass, momentum, and energy. The conservation law formulation of the fluid equations is powerful because, when properly discretized, the equations remain conservative.

Conservation laws. The conservation laws for mass, momentum, and energy can be written in integral or differential form. Conservation, in its most general form, applies to all three of these quantities. Consider an arbitrary volume Ω bounded by surface S that is fixed in space and crossed by the fluid flow. Let A represent a conserved flow quantity which could be, for example, energy, mass, number of particles, etc. A is the total amount of the quantity inside the volume Ω,

$$A = \int_\Omega U d\Omega, \tag{2.2}$$

where U is the density of the quantity. The quantity A changes as a result of fluxes — vector quantities measuring the amount of A crossing a unit area per unit time, into or out of Ω. The flux tangential to a surface does not change the amount of A in the domain; only the component of the flux normal to the surface S contributes to the change. Sources account for any other contributions to the change of A, and these sources are divided into volume (Q_V) and surface ($\vec{Q}_S$) sources, although surface sources tend to be enforced on boundaries or as an effective flux. The general form of a conservation law in integral form is then

$$\frac{\partial}{\partial t} \int_\Omega U d\Omega + \oint_S \vec{F} \cdot d\vec{S} = \int_\Omega Q_V d\Omega + \oint_S \vec{Q}_S \cdot d\vec{S}. \tag{2.3}$$

This is valid for any fixed volume Ω and its bounding surface S. In addition, the flux may be discontinuous (such as with shock waves) in the integral form because derivatives of the flux are not required.

A local differential form of the conservation law can be derived by transforming the surface integrals into volumetric integrals via the Divergence theorem, provided the surface sources and fluxes are continuous. Thereby, the differential conservation law can be obtained as

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot \vec{F} = Q_V + \vec{\nabla} \cdot \vec{Q}_S, \tag{2.4}$$

where surface sources have the same effect as a flux and could instead be written as an effective flux, $(\vec{F} - \vec{Q}_S)$. Hirsch prefers to define the surface sources and fluxes separately to delineate the physical origins of these terms [93]. Equation (2.4), the differential conservation law, is more restrictive than the integral conservation law, Equation (2.3), because it requires the fluxes (and surface sources) to be differentiable, which is not the case in some systems (such as those with shock waves). Similar integral and differential equations can be written for the conservation of a vector quantity, such as momentum, where the conservation law applies to each component of the vector quantity.

The flux, $\vec{F}$, in a fluid is produced by two physically distinct effects: advection and diffusion. Advection is the amount of the quantity A that is transported in the direction of the flow, $\vec{F}_{\text{adv}} = U\vec{v}$, where $\vec{v}$ is the flow velocity. Diffusion describes the tendency of the molecular motions of a fluid to reduce any nonhomogeneity in the concentration, $\vec{F}_{\text{dif}} = -D\vec{\nabla}U$. Advection is facilitated by velocity and does not occur in a stationary fluid. Diffusion can be present in a stationary or moving fluid, but only occurs if U is non-uniform ($\vec{\nabla}U \neq 0$). The flow advects all quantities while only some quantities diffuse. The Peclet number is a measure of the ratio of the advective and diffusive fluxes. Inserting these fluxes into Equation (2.4) gives the general conservative form of the transport equation,

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot (\vec{v}U) = \vec{\nabla} \cdot (D\vec{\nabla}U) + Q_V + \vec{\nabla} \cdot \vec{Q}_S, \tag{2.5}$$

which is also known as the advection-diffusion equation.

Regimes of fluid dynamics. It is important to distinguish three bounding sets of regimes for the coupled set of equations resulting from conservation of mass, momentum, and energy: viscous and inviscid; compressible and incompressible; laminar and turbulent. Viscosity is a measure of a fluid's internal friction of fluid layers against each other. A fluid with no viscosity is an inviscid fluid, and its momentum equation simplifies because the viscosity term vanishes. The system of equations for an inviscid fluid is known as the Euler equations. A compressible fluid may change volume for a given amount of mass, meaning the material derivative ($\frac{\partial}{\partial t} + \vec{v} \cdot \vec{\nabla}$) of the density is nonzero. All fluids are compressible to some degree, but some fluids can be approximated as incompressible because the amount of compression is negligible for the given conditions. The mass conservation equation reduces to a simpler form for the case of an incompressible fluid because the material derivative of density is zero. The system of equations for an incompressible fluid from conservation laws form a system known as the Navier-Stokes (NS) equations [93]. Laminar flow occurs below a critical velocity where the flow has well defined streamlines. Flows become turbulent above this critical velocity, where the flow is chaotic and variables fluctuate around a mean value. These fluctuations are strongly nonlinear and cannot be described deterministically, but may be solved numerically [93].

The turbulent nature of the coupled and nonlinear NS equations is difficult to simulate, spawning a variety of numerical techniques to solve or approximate the equations. A Direct Numerical Simulation (DNS) solves the NS equations at all relevant length scales without any approximations [64, 93]. This gives exact information of local values of conserved quantities (energy, mass, momentum). Since turbulence occurs over a range of scales, large 3D meshes are needed in a DNS. Greater velocities cause more turbulence, and thus higher mesh resolution is required. A DNS is limited by mesh size and computation time. The study of turbulent flow using a DNS for industrial purposes is beyond computational capabilities for the foreseeable future [64, 93]. An alternative, the Reynolds Averaged Navier-Stokes (RANS) approach, is the most widely used turbulence model in computational fluid dynamics [93]. The RANS method handles turbulence by taking a statistical average of the equations. This averaging process creates a turbulent correlation function that requires a closure model. This reduces the number of degrees of freedom, decreasing the computational cost. RANS methods tend to require empirically derived closure models for each individual system and, hence, lack universality [64]. The most accurate approximation to turbulent flow is known as the Large Eddy Simulation (LES). In LES, a sub-filter model below a threshold length scale approximates turbulence, while it is calculated exactly (as in DNS) above the threshold. The LES can be viewed as a hybrid method of the full calculation of the DNS and the fully averaged calculation of the RANS approach [64]. There are many more approaches for simplifying the NS equations, such as the thin shear layer approximation and the boundary layer approximation [93], but delving deeper into these approximations is outside the scope of this overview.

2.2 Relevance of Diffusion & Multiphase Flow in Applied Problems

Diffusion and multiphase flow are relevant in many scientific and technological problems. While both of these fields are vast, we focus on situations that involve interfaces between two phases, for which their modeling requires mixed cells. While it is beyond the scope of this document to catalog the numerous computational applications of fluid flow coupled with diffusion, we describe a few of the key topics in this section. The discussion on radiation transport is more detailed than the other sections due to its substantial relation to this work. In addition, we discuss previous work for mixed-cell problems in Section 2.4.3.

Porous media flow. Fluid flow through porous media is one of the phenomena described by a mathematical framework similar to Fick's law of diffusion. The petroleum industry models flow of oil through porous rock systems [53, 56, 51, 73, 72, 75, 123, 187, 194, 199, 213], often referred to as reservoir simulations. These types of simulations are also referred to by the acronym IMPES, for IMplicit Pressure and Explicit Saturation [14, 169]. Mixed finite element methods (FEM), and their concomitant finite difference methods (FDM) form, have been used in petroleum reservoir engineering for the last 60 years [14, 167, 213]. One common application is fluid injection for oil recovery, where the resident fluid, oil, is extracted from a rocky soil system by pumping a displacement liquid into the ground [199]. The displacement fluid flows more easily than the resident fluid, forcing the oil out of the ground through extraction points. This same process is also studied for the injection of hazardous waste deep underground [187].

A closely related topic to porous media flow is the flow of groundwater [7, 25, 43, 59, 115, 129, 195], which is useful for hydrologists. Likewise, waste or contaminant flow through porous media can be examined [28, 55, 59, 138, 195], which is of importance to environmental protection and nuclear waste storage [78, 156, 187]. Flow through fractured porous media, which contain larger voids within them, are also studied in the context of geothermal reservoirs [77, 162, 187]. When a Cartesian mesh is used in modeling such a system, the heterogeneous nature of these reservoirs often leads to features not aligned with the grid, resulting in mixed cells and/or tensor permeabilities. Reservoir systems often use the harmonic mean of permeabilities to account for the heterogeneity [14, 73].

Solidification. Another noteworthy application of diffusion is in the study of solidification for casting [129, 157, 192, 210], during which a molten liquid (typically metal) freezes inside a mold. Interfacial heat transfer conditions are especially important in many different types of casting, and interfacial heat transfer (along with phase changes) is one of the most critical phenomena to produce accurate solidification models [192]. Casting models are used in industry to predict and visualize the solidification structure, which aids in identifying defects, exploring effects of using different processing temperatures or cooling times, and reducing trial stage of developing new casts and methods.

Other applications. We list a few of the many other applications of multiphase diffusion techniques. These techniques are relevant to electrostatics and potential theory [56, 129, 211] since both have a similar mathematical framework as diffusion. Bubble dynamics [129, 218] are multiphase problems with large density jumps, having applications to chemical, electronic, petroleum, and power industries. Reference [50], which examines mixing of multiphase flows, notes that chaotic mixing from turbulent enhanced diffusion has applications to pollutant transport and weather patterns. Magnetohydrodynamic turbulence simulations may involve diffusion [130, 155, 177], as in Ramshaw and Chang [177] where they simulate multicomponent (different ions and neutral particles) plasma diffusion at non-equilibrium temperatures. Biomedical applications include the problem of blood flow through the moving boundaries of a beating heart [124, 171, 172, 173]. Anisotropic diffusion is used to model light propagation and distribution through human tissue [13, 15, 79, 84, 106, 108, 154, 208], which has medical applications for skin diagnostics and laser therapy.

Transport. The diffusion approximation is a prevalent treatment of transport theory. First derived for neutral particles in a homogeneous medium [15, 46, 60, 122], diffusion transport was then applied to neutrons in heterogeneous domains [4, 5, 15, 19, 121, 120, 193]. Diffusion is frequently used to approximate radiative transfer, as in References [113, 114, 161, 176, 181, 182, 184]. The diffusion approximation of radiative transfer is part of a class of methods using a continuum model of the radiation field to approximate the transport equation. Another class of methods is more accurate and uses a discrete (particle-based) model to solve the full transport equation.

There are many models to solve radiative transfer, each yielding different levels of fidelity. Olson *et al.* [161] compares many of the continuum and deterministic methods, including diffusion, flux-limited diffusion (FLD), and spherical harmonics (P_N). FLD is one of the most common, yet least accurate, approximations used [161]. The discrete-ordinate method is another important model for radiative transfer [17, 102, 109, 131, 200]. One technique to model radiative transfer is to combine a discrete transport method with the diffusion approximation, a continuum method, such as in Pomraning and Foglesong [176]. This approach has the advantage of only using the costly (transport) methods where they are needed while using inexpensive (diffusion) methods where they are sufficiently accurate. This type of hybrid method provides a balance between accuracy and computational cost. However, the coupling of two methods generates other numerical issues, particularly in transitional regions. For more details on many of these methods; see Section 2.3.3.

In the following section, Section 2.3, we discuss the phenomenon of radiation transport and means to model it computationally with either stochastic or deterministic approaches.

2.3 Overview of Radiative Transfer

Radiation transport is the movement of energy through (and interactions with) a physical system by radiation [65]. The radiation field can be described as an ensemble of photons, each with a specific frequency ν and traveling in a direction $\vec{\Omega}$. Photon-photon interactions are negligible due to the small scattering cross-section of photons relative to other particles of interest, such as ions and electrons. Photons propagating through a medium can interact with that matter in three ways: emission, absorption, and scattering. Photons can be absorbed by matter, removing that energy from the radiation field while increasing the matter's energy and temperature. Absorbed radiation adds to the energy and momentum density of the medium [146]. Emission, the reverse process of absorption, is the ejection of a photon from matter, decreasing the internal energy of the matter while increasing the energy of the radiation field. The amount of energy lost/gained by the radiation field is exactly equal to the energy gained/lost by the matter. Scattering is another form of light-matter interaction. In scattering, the light is not simply absorbed and then re-emitted, but rather it effectively 'collides' with matter and 'ricochets' in another direction. A scattered photon changes directions after interaction with a scattering center, and may have its frequency modified [146]. When matter-matter interactions are also important (as in hydrodynamics), they should also be considered, leading to a coupled physics problem (i.e., radiation hydrodynamics).

The remainder of this section is outlined as follows: in Section 2.3.1 we discuss a stochastic method to simulate radiative transfer, then in Section 2.3.2 we describe a deterministic model for radiation transport, and we conclude in Section 2.3.3 with methods to approximate this model. Note that we use the term 'material' instead of 'phase' in the context of radiation transport to conform to the standard terminology.

2.3.1 Monte Carlo & Implicit Monte Carlo Methods

Monte Carlo (MC) methods are stochastic approaches for transport problems. The random (probabilistic) nature of MC methods means that multiple runs with identical initial conditions and number of particles may give different results, although results should converge with increasing numbers of particles. This stochastic nature is in contrast to deterministic methods, which yield consistent and identical results for a given initial condition. MC methods determine the average behavior of a system by tracking many individual particles. Many applications of MC involve linear phenomena, where the action of each particle is independent of other particles. MC methods are computationally intensive, but the independent nature of each particle allows solution via parallel computing. However, for radiation transport, the nonlinear nature of the problem, particularly how the absorption and emission from the material affects subsequent absorption and emission events, couples the particles to some extent.

In a MC method for radiative transfer, a probability (estimated from experiments) is assigned for each possible photon event (emission, absorption, scattering), and a pseudo-random number is used concomitantly with the probabilities to determine the fate of a photon for each interaction with matter [42]. Each photon is followed separately for its entire existence, from emission until it exits the domain or is absorbed. Memory is reserved to describe the state of each photon, and this memory is freed once a photon is removed from the system. The material energy increases for each absorbed photon and decreases for each emitted photon.

An explicit time differencing scheme would use the material temperature at the beginning of each time step to set the material state, which governs the probability of emission and absorption events. In situations where the amount of energy exchanged between the matter and radiation in a time step is enough to significantly change the material temperature, instabilities may occur. However, an explicit time step in such systems that prevents these (nonlinear) instabilities may be too small for practical use. Using a larger time step would determine the emission probabilities at the beginning of the time step, but this would prevent the matter from reradiating energy it absorbed during the time step [87]. One possible solution to this absorptions/re-emission issue is to use implicit time dependence.

Implicit Monte Carlo (IMC) was the first successful method for simulating radiative transfer and remains a prevalent method [83, 103, 217]. IMC treats the nonlinearities of transport equations using a linear approximation and small time steps [103]. The IMC method is distinguished from standard MC methods in the manner the time dependence is treated [42]. The implicit method estimates the material temperature (using the material energy equation) at the end of the time step to compute the material state and determine emission and absorption probabilities [87]. The end result for the IMC method is that absorption and rapid re-emission are modeled as isotropic 'pseudo-scattering' or as an effective scattering term [42, 87, 103].

When a sufficiently large number of particles is used, the IMC method offers excellent accuracy for transport simulations. The price for such accuracy is considerable cost in terms of computer runtime and memory needs. The error for MC methods is lowest where the particle concentration is highest, allowing one to raise the number density in regions of interest to improve accuracy without significantly increasing the total number of particles. The error for IMC simulations scales approximately with $\alpha/\sqrt{N}$, where α is a scaling constant that depends only on the implementation, and N is the number of particles simulated [42, 103]. Since the error is roughly inversely proportional to the square root of the number of particles, a factor of ten increase in accuracy requires about one hundred times as many particles in the simulation. In addition to statistical noise caused by limited numbers of particles, the IMC method introduces truncation error for both the spatial and temporal discretization [103]. Despite these limitations, the accuracy of the IMC methods leads to their use in radiative transfer as a benchmarking tool to gauge less accurate methods [103]. Also note that there are other MC methods besides IMC to handle the nonlinear radiative transfer problems [103], such as in Reference [63].

2.3.2 Gray Radiation Transport Equation

Deterministic models describe the radiation field via the density or intensity of photons. The photon number density per unit volume per unit solid angle per unit frequency is $N(\vec{x}, \vec{\Omega}, \nu, t)$, where $\vec{x}$ represents the spatial position, $\vec{\Omega}$ is the direction of travel, ν is the frequency of the photons, and t is the time. In the following discussion, we assume the material is at rest. The number of photons inside a volume dV around point $\vec{x}$, traveling in direction $d\Omega$ around $\vec{\Omega}$, in a frequency range $d\nu$ around ν, at time t [103], is given by

$$N(\vec{x}, \vec{\Omega}, \nu, t) dV d\vec{\Omega} d\nu. \tag{2.6}$$

Since a photon energy is specified by $E = h\nu$ and has a known velocity c, the radiation field is usually expressed in terms of the radiation intensity,

$$I(\vec{x}, \vec{\Omega}, \nu, t) = ch\nu N(\vec{x}, \vec{\Omega}, \nu, t), \tag{2.7}$$

which has units of energy per unit area per unit time per unit frequency per unit solid angle. The radiation intensity, I, gives the amount of photon energy in a volume at a given time, where the volume is of a phase-space of three space variables ($\vec{x}$), three momentum variables (a two component direction vector $\vec{\Omega}$ and the frequency ν), as well as time (t). Therefore, two photons at the same position and time, traveling in different directions, occupy separate volumes in this phase-space, as would two photons of different frequency traveling in the same direction at the same time.

The radiation intensity can be integrated over frequency to yield the gray radiation intensity,

$$I(\vec{x}, \vec{\Omega}, t) = \int_0^\infty ch\nu N(\vec{x}, \vec{\Omega}, \nu, t) d\nu. \tag{2.8}$$

However, treatments that are more accurate usually require a multigroup technique to express the radiation intensity. In a multigroup method, instead of integrating intensity over all frequencies, a collection of frequency-dependent intensities is calculated by integrating over a frequency interval. This allows for a more physically accurate description of radiation transport, including inelastic scattering and other frequency-dependent processes. However, we use the gray approximation for the remainder of this section for simplicity, and 'radiation intensity' refers only to the gray radiation intensity.

We now examine the equations governing radiation transport (see References [49, 65, 146, 175] for more details). The radiation transport equation (RTE) accounts for five ways that a photon can enter or exit a volume in the phase-space:

streaming: travels unimpeded into or out of the phase-space volume

absorption: captured by matter

emission: emitted from matter

out-scattering: scatters from inside the phase-space volume out of it

in-scattering: scatters from outside of the phase-space volume into it .

The gray RTE, in an inertial (lab) frame, can be written as [42, 49, 65, 87, 103, 146]

$$\frac{1}{c}\frac{\partial I}{\partial t} + \vec{\Omega} \cdot \vec{\nabla} I = -(\sigma_a + \sigma_s) I + \int_{4\pi} \frac{\sigma_s}{4\pi} I d\Omega + \eta + \frac{Q_r}{4\pi}, \tag{2.9}$$

where σ_a is the absorption opacity, σ_s is the scattering opacity, η is the emission term, and Q_r is a radiation source term. The units of each term in the equation are power density over a sphere (energy per unit time per unit volume per unit solid angle). The two terms on the LHS together describe the transport of radiation (i.e., the streaming of photons), which is zero in the absence of any sources or sinks. The first term on the RHS is a combination of absorption and out-scattering rates, which are negative because both processes remove radiation from the volume. The second term describes in-scattering, where the intensity is integrated over all angles to account for in-scattering from any direction. The third term on the RHS is the emission rate, which is positive because it adds radiation to the field. The last term is an arbitrary radiation source.

Note that this is a classical description of the radiation field, and thus quantum effects are disregarded [65]. The uncertainty principle is not considered, source terms have no quantum degeneracy, and photons are treated as point particles. Polarization, spin, interference, refraction, and diffraction are ignored. Even without the quantum effects, the radiation intensity as described achieves accurate results in practice and is a useful approximation for many systems.

The radiation field is coupled to the material through photon emission and absorption. There are many types of emission, such as bound-bound emission, which occurs when an electron in an atom drops from a higher to a lower energetic quantum state, releasing a photon equal to the energy loss. All matter, regardless of its composition or configuration, also emits photons because of thermal motions [219]; this radiated energy depends only on temperature and is called *thermal radiation*. Blackbody radiation (also known as Planck's law of radiation) or its frequency-integrated form (the Stefan-Boltzmann law), Equation (2.10), describes the thermal emission of a blackbody. A blackbody is an idealized form of matter, one that perfectly absorbs and emits at all frequencies.

We can assume blackbody emission if the system is in local thermodynamic equilibrium (LTE). The meaning of LTE, in this case, is that the material can be given a temperature and treated as a blackbody source at this temperature, although the radiation is not in equilibrium with the material [144] (if it were, the emission and absorption terms would identically cancel). In other words, by LTE we mean that the material is in equilibrium with itself. The blackbody approximation must be made in the rest frame of the fluid, where thermal emission is isotropic. However, we make a stricter assumption of a stationary material. Integration of the Planck function over all angles and frequencies gives [42, 144, 209],

$$\begin{aligned} B(T) &= \int_0^\infty \int_{4\pi} B(T, \nu') d\nu' d\vec{\Omega}' \\ &= \int_0^\infty \int_{4\pi} \frac{2h\nu'^3}{c^2} \frac{1}{\exp \frac{h\nu'}{kT} - 1} d\nu' d\vec{\Omega}' \\ &= \frac{8\pi^5 k^4}{15h^3c^2} T^4 = acT^4, \end{aligned} \tag{2.10}$$

where the radiation constant (a) is

$$a = \frac{8\pi^5 k^4}{15h^3c^3}, \tag{2.11}$$

with units of energy per unit area per unit time per temperature to the fourth power. $B(T)$ is the power radiated in all directions (energy per unit area per unit time); hence the corresponding intensity is $\frac{1}{4\pi} acT^4$ (energy per unit area per unit time per unit solid angle) [144]. The rate of absorption is the product of the blackbody intensity with absorption opacity (σ_a); this is also the rate of emission due to the assumption of LTE. Accordingly, for our system,

$$\eta = \sigma_a \frac{1}{4\pi} acT^4, \tag{2.12}$$

and the gray RTE becomes

$$\frac{1}{c}\frac{\partial I}{\partial t} + \vec{\Omega} \cdot \vec{\nabla} I = -(\sigma_a + \sigma_s) I + \int_{4\pi} \frac{\sigma_s}{4\pi} I d\Omega + \sigma_a \frac{1}{4\pi} acT^4 + \frac{Q_r}{4\pi}. \tag{2.13}$$

The absorption opacity combines many of the complications of the absorption of light by matter, such as the material's reflectivity, the material's thickness versus the frequency of light, etc.

The material energy equation, with assumptions of LTE of the matter with itself and conservative or elastic scattering (where a photon can change direction but not frequency), can be expressed as [87, 103, 144]

$$\rho(T) c_v(T) \frac{\partial T}{\partial t} = \sigma_a \int_{4\pi} I d\Omega - \sigma_a acT^4 + Q_m, \tag{2.14}$$

where Q_m is a material energy source (not necessarily related to the radiation source Q_r), ρ is the material density, and c_v is the specific heat capacity of the material. Note the units of Equation (2.14) are power density (energy per time per volume), and the units of the heat capacity are energy per mass per temperature. Also, this equation makes the assumption that the material energy (U_m) depends on heat capacity and temperature as

$$U_m = \int_0^T \rho(T) c_v(T) dT. \tag{2.15}$$

The LHS of Equation (2.14) represents the total change of energy with time, while the RHS gives the specific sources with which the energy can change. We are assuming a stationary material, so radiation is the dominant form of energy exchange. The first term on the RHS represents any radiation absorbed by the material, which is integrated over all angles to account for absorption from any incoming direction. The second term on the RHS is the energy emitted by the material due to blackbody radiation; this term is negative because this energy is lost from the material. The final term accounts for any other heat source or sink within the material.

Equations (2.13) and (2.14) are coupled equations that must be solved in order to compute the radiation transport. Even when approximated, these are difficult to solve due to the large phase-space (seven variables) as well as the nonlinearity. Various methods exist to deal with nonlinearities [40, 41, 42, 85, 112, 113, 160, 184], while References [49, 125, 146, 175] are good sources for more information on radiation transport.

2.3.3 Approximations to the Radiative Transport Equation

We now describe specific methods of solving the RTE following a seminal review article [42] as well as a comprehensive thesis [103]. The main methods explored are discrete ordinates, spherical harmonics, diffusion, and flux-limited diffusion (FLD). These methods are deterministic, in contrast to MC methods. The discretized RTE forms a large system of linear equations that is often solved iteratively. Deterministic methods converge more quickly than MC methods: a factor of four increase in simulation size will decrease the error of a second-order deterministic method by a factor of sixteen while error for MC methods will decrease by about a factor of two.

Solving the RTE is complicated because of the large number of dimensions involved, and therefore various simplifications must be made. These simplifications mostly involve the angular dependence of the radiation, either in the angular discretization or by taking angular moments. An angular moment of some arbitrary function $f(\vec{x}, \vec{\Omega}, \nu, t)$ is defined as follows:

0^{th} moment: $\int_{4\pi} f d\Omega$

1^{st} moment: $\int_{4\pi} \vec{\Omega} f d\Omega$

2^{nd} moment: $\int_{4\pi} \vec{\Omega}\vec{\Omega} f d\Omega$,

where $\vec{\Omega}\vec{\Omega} = \vec{\Omega} \otimes \vec{\Omega}$. The symbol ⊗ is the outer product (also known as the tensor product and the dyadic product), which changes vectors into tensors, in contrast to the inner product, which converts vectors into scalars. The number of the moment is determined by how many factors of $\vec{\Omega}$ occur in the integral. Note that infinitely many moments can be taken, moments may be taken with respect to quantities other than angle, and moments can be taken of entire equations as well as of a single function.

We next discuss four of the most common approximations to the RTE: discrete ordinates, spherical harmonics, diffusion, and flux-limited diffusion. Each method has a spatial discretization that works best with its approximation for the angular description, while the temporal discretization is typically independent of the method used [103].

2.3.3.1 Discrete Ordinates

One of the most frequently used discretization methods for the RTE is the discrete ordinate method [103]. This method assumes that photons can travel along only a finite set of directions [42]. Otherwise stated, the radiation intensity (I) is represented by a sum of intensities (I_m), corresponding to a discrete set of directions ($\vec{\Omega}_m$) [42],

$$I(\vec{x}, \vec{\Omega}, t) = \sum_{m=1}^{M} I_m(\vec{x}, t)\delta(\vec{\Omega} - \vec{\Omega}_m), \tag{2.16}$$

where I_m is the intensity I averaged over a definite range of $d\Omega$ about Ω_m [146]. This method is referred to as the S_N method, where the subscript N is related to the number of discrete directions (ordinates) used [42, 128]. A quadrature set, w_m, is a weighted set that is used to evaluate the integral of the intensity [103, 144],

$$\int_{4\pi} I(\vec{x}, \vec{\Omega}', t) d\Omega' \approx \sum_{m=1}^{M} w_m I_m, \tag{2.17}$$

where $\sum_{m=1}^{M} w_m = 1$.

The RTE for the discrete ordinate method is written as,

$$\frac{1}{c}\frac{\partial I_m}{\partial t} + \vec{\Omega}_m \cdot \vec{\nabla} I_m = -(\sigma_a + \sigma_s) I_m + \frac{\sigma_s}{4\pi}\sum_{m=1}^{M} w_m I_m + \frac{\sigma_a a c T^4}{4\pi} + \frac{Q_r}{4\pi}. \tag{2.18}$$

Note that each direction m is coupled to all the other directions through the scattering term, and the directions are also coupled to each other though the material energy equation via the temperature.

Increasing the number of ordinates increases the accuracy as well as the computational cost. Since a limited number of ordinates must be used in practice, some areas of the domain do not receive the accurate amount of radiation that they would with higher resolution. This leads to the defect called *ray effects* [39, 42, 128, 148], which can cause large spatial oscillations in the energy density. One attempt to counter ray effects is to introduce an extra scattering term to smooth the variation [38, 42, 104, 179].

A highly scattering system can take a large number of iterations to converge. This can be countered by multigrid methods, Krylov methods, or other forms of acceleration [103]. However, the diffusion approximation may be more appropriate for such a system, providing sufficient accuracy with reduced computational cost.

2.3.3.2 Spherical Harmonics

The spherical harmonic method takes a series of moments of the intensity to approximate the angular dependence. In contrast to the discrete ordinate method of discretizing the angular dependence into a finite number of directions, the spherical harmonics method takes a finite number of angular moments of the RTE to build an orthonormal basis for the solution. A solution is constructed as a linear combination of basis functions. The standard notation for the spherical harmonics method is the P_N method.

The approximation of the radiation intensity is written in terms of spherical harmonic functions, $Y_{l,m}$, which are defined as

$$Y_{l,m}(\theta, \phi) = \sqrt{\frac{2l+1}{4\pi}\frac{(l-m)!}{(l+m)!}} P_l^m(\cos\theta) e^{im\phi}, \tag{2.19}$$

where $P_l^m(\cos\theta)$ are the Legendre polynomials and the angles θ and ϕ correspond to the direction of travel, $\vec{\Omega}$. The spherical harmonic functions form an orthonormal set,

$$\int_0^{2\pi}\int_0^{\pi} Y_{l,m}(\theta,\phi)Y^*_{l',m'}(\theta,\phi)d\theta d\phi = \delta_{l,l'}\delta_{m,m'}, \tag{2.20}$$

where $Y^*_{l,m}$ is the complex conjugate of $Y_{l,m}$, and $\delta_{i,j}$ is the Kronecker delta function. The approximated intensity can be written as [103, 145]

$$I(\vec{x},\vec{\Omega},t) \approx \sum_{l=0}^{N}\sum_{m=-l}^{l} Y_{l,m}(\vec{\Omega})\left[\int_{4\pi} Y^*_{l,m}(\vec{\Omega}')I(\vec{x},\vec{\Omega}',t)d\Omega'\right], \tag{2.21}$$

where the expression is only approximate because a finite number (N) of moments are used, but converges to the exact value when $N \to \infty$. The closure approximation to Equation (2.21) is such that the moments are zero for any $l \geq N$ [42, 145], i.e.,

$$\int_{4\pi} Y^*_{l,m}(\vec{\Omega}')I(\vec{x},\vec{\Omega}',t)d\Omega' = 0 \qquad l \geq N. \tag{2.22}$$

The spherical harmonic method treats the radiation as waves and suffers from *wave effects* [42, 103, 144, 145], which are analogous to the *ray effects* from the discrete ordinate method's treatment of the radiation as rays. In the spherical harmonic method, particles travel as waves with finite speeds throughout the system. Each moment has a corresponding eigenvalue and wave speed; the more moments used, the more unique speeds (and corresponding wave patterns) are available for wave propagation, increasing the accuracy. The wave nature of the solutions may lead to negative solutions in the free-streaming limit [42, 145] or in the presence of steep gradients [42, 103, 144, 145]. Negative values for the energy density can lead to negative material temperatures, resulting in a discrepancy between the fundamental physics and the simulation. Possible solutions to this problem consider using a nonlinear closure equation [103, 145].

The P_1 approximation is closely related to the diffusion approximation, as both use the zeroth and first angular moments. A detailed explanation of this is given in the next section.

2.3.3.3 Diffusion

Governing equation. The diffusion equation, arguably the most important approximation to the RTE [49], is much easier to solve than higher-order methods. Although it is a crude, low-order approximation in both angle and frequency, it is widely used because it yields a simple form for the radiative flux [175]. The diffusion approximation is achieved by taking the zeroth and first angular moments of the RTE, similar to the P_1 method. We define the following variables for each moment of the radiation intensity:

$$E = \frac{1}{c}\int_{4\pi} I d\Omega \tag{2.23a}$$

$$\vec{F} = \int_{4\pi} \vec{\Omega} I d\Omega \tag{2.23b}$$

$$\mathbf{P} = \frac{1}{c}\int_{4\pi} \vec{\Omega}\vec{\Omega} I d\Omega, \tag{2.23c}$$

where E is the scalar intensity or energy density, $\vec{F}$ is the radiative flux, and $\mathbf{P}$ is the radiative pressure tensor. The second moment is included since it is needed for closure.

By taking the zeroth angular moment of the RTE,

$$\frac{1}{c}\frac{\partial I}{\partial t} + \vec{\Omega}\cdot\vec{\nabla} I = -(\sigma_a + \sigma_s)I + \frac{\sigma_s}{4\pi}cE + \frac{\sigma_a acT^4}{4\pi} + \frac{Q_r}{4\pi}, \tag{2.24}$$

we obtain

$$\frac{\partial E}{\partial t} + \vec{\nabla}\cdot\vec{F} = -\sigma_a cE + \sigma_a acT^4 + Q_r, \tag{2.25}$$

where the scattering terms (σ_s) have canceled. The units of each term of this equation are of power density (energy per unit time per unit volume). Before taking the first moment the RTE, note that the first moment of a constant (isotropic term) is zero,

$$\int_{4\pi} \vec{\Omega} d\Omega = \int_0^{2\pi}\int_0^{\pi} \vec{\Omega}\sin(\theta) d\theta d\phi \tag{2.26a}$$

$$= \hat{z}\int_0^{2\pi}\int_0^{\pi} \cos(\theta)\sin(\theta) d\theta d\phi \tag{2.26b}$$

$$= 2\pi\hat{z}\int_{-1}^{1} \mu d\mu \tag{2.26c}$$

$$= 0\hat{z}, \tag{2.26d}$$

where for simplicity the propagation direction is aligned with the z-axis [65], although any direction would yield the same result. The first moment of Equation (2.24) is

$$\frac{1}{c}\frac{\partial \vec{F}}{\partial t} + c\vec{\nabla}\cdot\mathbf{P} = -(\sigma_a + \sigma_s)\vec{F}, \tag{2.27}$$

where the last three terms of Equation (2.24) (scattering, emission, and source) vanish because they are isotropic. However, the isotropy of these terms is an assumption: scattering does not have to isotropic, and blackbody emission term is only isotropic in a frame co-moving with the fluid.

Note how both the zeroth and first moment equations, Equations (2.25) and (2.27), have a dependence on the first and second moment term, $\vec{F}$ and $\mathbf{P}$, respectively. This coupling with the next highest moment introduces a *closure problem* [146]. Issues from the closure problem occur in Equation (2.27), where we do not have an equation for the pressure tensor ($\mathbf{P}$). An equation for $\mathbf{P}$ could be obtained by taking another moment of the RTE, but then an equation for the third moment would be needed. This process needs to be closed (given an additional relation) in order to avoid the inconvenience of taking infinite moments.

Closing the angular moments. We now describe the closure for the P_1 method, from which the diffusion approximation can be obtained. Both of these methods assume the radiation intensity is a linear function of angle [42, 103]. This assumption is the spherical harmonic expansion of Equation (2.21) of lowest anisotropic order ($N = 1$), giving an isotropic term and a single angular term,

$$I \approx \frac{c}{4\pi} E + \frac{3}{4\pi} \vec{\Omega} \cdot \vec{F}. \tag{2.28}$$

Using the approximation in Equation (2.28), we then take the second moment of the radiation intensity,

$$\begin{aligned}
\mathbf{P} &= \frac{1}{c} \int_{4\pi} \vec{\Omega}\vec{\Omega} I d\Omega && \text{(2.29a)}\\
&\approx \int_{4\pi} \vec{\Omega}\vec{\Omega} \left[\frac{1}{4\pi} E + \frac{1}{c} \frac{3}{4\pi} \vec{\Omega} \cdot \vec{F} \right] d\Omega && \text{(2.29b)}\\
&\approx \frac{E}{4\pi} \int_{4\pi} \vec{\Omega}\vec{\Omega} d\Omega + 0 && \text{(2.29c)}\\
&\approx \frac{E}{3} \mathbf{I}, && \text{(2.29d)}
\end{aligned}$$

where $\mathbf{I}$ is the identity matrix and the flux term drops by integration [42, 65, 103].

Evaluating the following integral elucidates the last step,

$$\begin{aligned}\int_{4\pi} \vec{\Omega}\vec{\Omega} d\Omega &= \int_0^{2\pi}\int_0^{\pi} \vec{\Omega}\vec{\Omega}\sin(\theta) d\theta d\phi \\ &= 2\pi \int_0^{\pi} \cos^2(\theta)\sin(\theta) d\theta \\ &= 2\pi \int_{-1}^{1} \mu^2 d\mu \\ &= \frac{4\pi}{3},\end{aligned}$$

where we have assumed that $\vec{\Omega}$ is aligned to the z-axis [65]. Using the fact that the flux can pass through the angular integral, the same process for the $\vec{\Omega}\cdot\vec{F}$ term leads to an integral of μ^3, which is zero.

Using Equation (2.29d) with the first moment of the RTE, Equation (2.27), we obtain

$$\frac{1}{c}\frac{\partial \vec{F}}{\partial t} + \frac{c}{3}\vec{\nabla} E + (\sigma_a + \sigma_s)\vec{F} = 0, \tag{2.30}$$

where $\vec{\nabla}\cdot(\mathbf{I}E) = \vec{\nabla}E$. Equations (2.25) and (2.30) provides an expression for $\vec{F}$ and E without introducing another term, giving two equations and two unknowns. This completes the closure equation for the P_1 approximation.

The diffusion approximation proceeds from Equation (2.30) with one additional assumption: the time derivative term can be neglected. This assumption is called the 'quasi-static' approximation [42, 67, 103],

$$\frac{1}{c^2}\frac{\partial \vec{F}}{\partial t} \ll \frac{1}{3}\vec{\nabla} E, \tag{2.31}$$

allowing the time derivative term to be dropped from Equation (2.30). Mihalas and Mihalas [146] justify dropping the time derivative in the limit of optically thick systems because the time needed for photons to random walk a significant distance is long, resulting in a small change of flux with time, i.e., $[\frac{1}{c}\frac{\partial \vec{F}}{\partial t}]/[(\sigma_a+\sigma_s)\vec{F}] \approx 10^{-18}$ for typical values. This essentially is a comparison between the mean free path and the characteristic length of the flux [87].

Upon dropping the time derivative of the flux from Equation (2.30), a simple relation between the flux and the energy density is achieved,

$$\frac{c}{3}\vec{\nabla} E + (\sigma_a + \sigma_s)\vec{F} = 0. \tag{2.32}$$

Solving for the flux and introducing a total opacity, $\sigma_t = \sigma_a + \sigma_s$, we have

$$\vec{F} = -\frac{c}{3\sigma_t}\vec{\nabla}E, \tag{2.33}$$

which matches the form of Fick's first law, Equation (2.1a). Combining this result with the zeroth moment of the RTE, Equation (2.25), we arrive at the diffusion equation for radiative transfer,

$$\frac{\partial E}{\partial t} + \vec{\nabla} \cdot \vec{F} = \sigma_a c(aT^4 - E) + Q_r, \tag{2.34}$$

where $\vec{F} = -cD\vec{\nabla}E$ and $D = (3\sigma_t)^{-1}$. Note that Equation (2.34) is simply Fick's second law, Equation (2.1b), with $\sigma_a(acT^4 - E) + Q_r$ as a source term. Equation (2.14), the material energy equation, is rewritten using the scalar energy density as

$$\rho c_v \frac{\partial T}{\partial t} = \sigma_a c(E - aT^4) + Q_m. \tag{2.35}$$

Equation (2.34) and Equation (2.35) form the system of equations for the diffusion approximation of radiative transfer. Ignoring potential source effects (Q_m and Q_r), the energy gained/lost by the radiation field is equal to the energy lost/gained from the material via emission/absorption.

Validity and applicability. The simple form of the flux given by Equation (2.33) is one of the most appealing features of the diffusion approximation, as it reduces many of the complications in finding a solution. However, the approximation (Equation (2.31)) of dropping the time dependent term changes the class of the differential equation from hyperbolic to parabolic [42]. A hyperbolic system, such as the transport equation, restricts particles or energy to travel at *finite* speeds. However, a parabolic system, such as the diffusion equation, allows particles or energy to propagate at *infinite* speeds. A change in one part of the domain instantly affects the entire domain, though this effect is often so small that it is computationally negligible. The infinite propagation speed becomes a problem when the diffusion approximation is applied outside of its applicability range.

The diffusion approximation is only valid in an optically thick system. Three properties characterize a system as being optically thick or optically thin: diffusivity, opacity, and mean free path (MFP). These terms are related by

$$D = \frac{1}{3\sigma_t} = \lambda_{MFP}. \tag{2.36}$$

The MFP is the average distance a photon travels before an interaction event (i.e., scattering or emission-absorption) with matter. Systems with a small MFP and large opacities are called *optically thick*, corresponding to a system with many scattering and absorption/emission events compared to a characteristic length. Such a system is accurately modeled by a random walk and, thus, diffusion. A system with a large MFP and small opacity is *optically thin*, in which a photon travels relatively large distances between scattering and absorption events. Diffusion is a poor model for such systems because it allows energy to propagate faster than light speed.

The Cauchy-Schwarz inequality is applied to the radiation flux to determine an equation for when the diffusion approximation is valid physically,

$$\begin{aligned}
\|\vec{F}\| &= \left\| \int_{4\pi} \vec{\Omega} I d\Omega \right\| && \text{(2.37a)} \\
&\leq \int_{4\pi} \|\vec{\Omega}\| \|I\| d\Omega && \text{(2.37b)} \\
&\leq \int_{4\pi} I d\Omega && \text{(2.37c)} \\
&\leq cE, && \text{(2.37d)}
\end{aligned}$$

since $\vec{\Omega}$ in a unit vector and the radiation intensity is non-negative [103]. This limit is physically stating that the energy transported by the radiation flux cannot exceed the total amount of energy present. Inserting the diffusion flux approximation, Equation (2.33), into Equation (2.37d) gives

$$\begin{aligned}
\|\vec{F}\| &\leq cE \\
\|-cD\vec{\nabla}E\| &\leq cE \\
D &\leq \frac{E}{\|\vec{\nabla}E\|}, && \text{(2.38)}
\end{aligned}$$

where both D and E are always positive. In an optically thick region, $D \to 0$, easily satisfying Equation (2.38) unless there are large jumps in the energy density (i.e., $\|\vec{\nabla}E\|$ is large). In an optically thin region, however, $D \to \infty$, which means the RHS has to be even larger, which is difficult to satisfy. Therefore, the diffusion approximation is valid in optically thick regions with relatively smooth variations in energy density.

This completes the description of the diffusion approximation, which is accurate when Equation (2.38) is satisfied. However, this inequality restricts to the systems to which diffusion can be applied. The next section looks at expanding the applicability range of diffusion to optically thin systems by altering the flux to enforce the inequality of Equation (2.38).

2.3.3.4 Flux-Limited Diffusion

The major shortfall of the diffusion approximation, namely the possibility of energy propagation faster than light speed, is due to discarding the $\partial \vec{F}/\partial t$ term from Equation (2.30). Flux-limited diffusion (FLD) is an *ad hoc* method to artificially restrict the flux. The diffusivity, which comes from the assumption of Equation (2.31), is given as

$$D = \frac{1}{3\sigma_t}. \tag{2.39}$$

The flux and limit are summarized below:

$$\vec{F} = -cD\vec{\nabla}E \tag{2.40a}$$

$$D \leq \frac{E}{\|\vec{\nabla}E\|}. \tag{2.40b}$$

In the limit of the optically thick case, Equation (2.39) is appropriate. However, in the optically thin case, Equation (2.40b) must be enforced to prevent unphysical behavior, such as transporting energy faster than the speed of light.

Many different flux limiters are used in practice. The equations for the flux limiters may be specified in terms of the following dimensionless parameter,

$$R = \frac{1}{\sigma_t}\frac{\|\vec{\nabla}E\|}{E}, \tag{2.41}$$

which is known as the radiation Knudsen number [49, 127, 205], which is a ratio of the MFP (recall opacity is the inverse of MFP) and a characteristic length of the system (in this case, $E/\|\vec{\nabla}E\|$). Some exemplar flux limiters are

$$D(R) \ = \ \begin{cases} (3+R)^{-1} & \text{Sum [49, 146, 161]} \\ (\max(3,R))^{-1} & \text{Max [49, 161]} \\ (3^n + R^n)^{-\frac{1}{n}} & \text{Larsen [161]} \\ R^{-1}\left(\coth(R) - R^{-1}\right) & \text{Levermore-Pomraning [42, 49, 126, 161]} \end{cases} \tag{2.42}$$

where the first two are chosen to have the correct limits for small and large R, and have been shown to be zeroth-order accurate, while the modification by Larsen increases accuracy to first-order [161]. The Levermore-Pomraning limiter comes from solving the Chapman-Enskog problem [126], but works well beyond the original application [161].

FLD has the disadvantage of making D nonlinear in E [49, 161], and the accuracy of FLD must be examined for individual applications. Olson *et al.* [161] show that Larsen's $n = 2$ limiter is the best choice for FLD, but they determine that the $P_{\frac{1}{3}}$ method (a combination of the P_1 spherical harmonics method and diffusion) is superior to any FLD theories in geneal [161]. Despite its limitations, FLD yields more qualitatively physical results in a wider range of problems than diffusion and, hence, is widely used for radiative transfer.

This concludes the review of radiation transport. We now shift the focus from background and introductory material to the phase-interface diffusion problem.

2.4 Highly Discontinuous Material Properties

This section focuses on the primary problem of this dissertation: discontinuous material interfaces. Discontinuous interfaces commonly occur in multiphase diffusion (e.g., heat conduction between two phase) when two immiscible phases or fluids border each other with diffusivities that vary by several orders of magnitude. The discontinuous change in phase coefficients is not modeled well by many schemes that expect constant or smoothly varying values. While some work considers diffusivity ratios around two or three as a discontinuous phase, we are more interested in ratios of many orders of magnitude. When the diffusivity ratios are on orders of magnitude, the problems are referred to as *highly discontinuous* or *strongly discontinuous*. The ratio of the diffusivity for each phase is also referred to as the *jump strength*. In these multiphase cases, the transition regions between phases have mixed cells, which are grid cells containing more than one phase coefficient. Mixed cells are a large source of inaccuracy in such systems. This thesis explores the treatment of mixed cells in highly discontinuous cases.

We first describe the motivating problem in Section 2.4.1: energy transfer in a radiative, multiphase system. We describe our primary problem of interest in Section 2.4.2, where we show how the radiation transfer can be computationally made to look like a heat conduction problem. We end with a detailed review of previous work related to discontinuous material interfaces and mixed cells in Section 2.4.3.

Although most of the cited work only considers pure cells, it is still relevant to the mixed-cell problem. Note that the terms *fluid*, *material*, and *phase* may be used interchangeably; the method being developed can be applied to either fluids or solids with discontinuities in the diffusivity.

2.4.1 Motivating Problem

Radiative transfer problems in multiphase systems are frequently modeled by diffusion. Numerical issues occur when an opaque (optically thick) phase is bordered by a vitreous or translucent (optically thin) phase. These systems have highly discontinuous diffusivities. The ratio between the opaque and pellucid diffusivities can be extremely high, on the order of 10^{10} [181]. Such a scenario makes the computations very stiff, which can lead to sluggish or failed convergence [103, 181, 218].

A multi-fluid problem can be miscible (fluids mix) or immiscible (fluids segregate). We focus on the immiscible case, where the fluids maintain a definite boundary. The immiscible treatment of hydrodynamics can be considered a sharp-interface model. A Volume of Fluid (VoF) method only maintains the percentage of each fluid in a cell, discarding the exact interface position, smearing the interface across the entire cell. Such a method can still be considered as having a sharp interface if the interfaces are reconstructed rather than homogenizing (mixing or averaging) the cell.

At an immiscible interface between phases, both the energy and flux must be continuous. Consider 1D heat conduction where T is the temperature, $\vec{F}$ is the flux, and an interface located at $x = a$. The continuity equations for this 1D case are

$$\lfloor |T| \rfloor = T(a+) - T(a-) = 0 \tag{2.43a}$$

$$\lfloor |F| \rfloor = F(a+) - F(a-) = 0, \tag{2.43b}$$

where the notation $a-$ means x approaches a from smaller values, and $a+$ means x approaches a from larger values. The notation $\lfloor |\cdot| \rfloor$ means 'jump', which is the difference of the quantity in the brackets on either side of an interface, as in References [129, 139]. In 2D, we use similar notation where $\vec{a}-$ and $\vec{a}+$ means 'approaching interface from' below or above the interface. The continuity of flux is expressed in terms of $F_\perp$, where

$$F_\perp = \vec{F} \cdot \hat{n} \tag{2.44}$$

and $\hat{n}$ is the normal to the interface, pointing from above to below. Introducing the point of interest on an interface as $\vec{a} = (a_x, a_y)$, the 2D continuity equations are then

$$\lfloor |T| \rfloor = T(\vec{a}+) - T(\vec{a}-) = 0 \tag{2.45a}$$

$$\lfloor |F| \rfloor = F_{\perp}(\vec{a}+) - F_{\perp}(\vec{a}-) = 0. \tag{2.45b}$$

Note that there is no constraint on the tangential flux ($F_{\|}$). This is because flux tangent or parallel to an interface is just an internal flux; the heat is only moving within one phase with no energy exchanged between phases.

In the Section 2.3.3.3, we found the definition of flux, the diffusion approximation of the RTE, and the material energy equation, which we re-state below

$$\vec{F} = -D\vec{\nabla}E = -\frac{1}{3(\sigma_a + \sigma_s)}\vec{\nabla}E \tag{2.46a}$$

$$\frac{1}{c}\frac{\partial E}{\partial t} + \vec{\nabla} \cdot \vec{F} = \sigma_a(acT^4 - E) + Q_r \tag{2.46b}$$

$$\rho c_v \frac{\partial T}{\partial t} = \sigma_a(E - acT^4) + Q_m. \tag{2.46c}$$

FLD could easily be reintroduced to the model, but we are assuming that pure diffusion is adequate for the problems we are considering. Equation (2.46) is the set of equations that must be solved, where Equation (2.46b) is the equation for the radiation field, and Equation (2.46c) is the equation for the material. The heat capacity (c_v) and density (ρ) depend on the specific material and, in real materials, are temperature dependent functions ($c_v = c_v(T)$, $\rho = \rho(T)$), and may need to be inside the time derivative ($\frac{\partial}{\partial t}(\rho(T)c_v(T)T)$). Several assumptions are already inherent to Equation (2.46): the radiation intensity is linear in angle, the emission and absorption are isotropic, the material is in local thermodynamic equilibrium (LTE), and the flux varies much more slowly in time than the energy changes in space. Moreover, we are no longer solving for the radiation intensity, I, but its zeroth moment, the scalar intensity, E.

Note that by LTE, we mean that the material is in thermal equilibrium with itself, making a Planckian emission term physically appropriate. If the matter and radiation were in LTE with each other, then the emission would exactly cancel absorption, making the RHS of both Equation (2.46b) and Equation (2.46c) identically zero.

The CRASH program motivates this work. Approximately half of the effort in the CRASH program is experimental. In the quintessential experiment of this program [201, 209], a thin beryllium wafer is accelerated by a strong laser impulse in order to drive a shock down a xenon gas-filled tube. The process has enough energy to enter the radiative regime, where the shock is radiating energy upstream. Among many diagnostics, radiographs (an image produced from x-rays) are used to image the shock. Beryllium is x-ray transparent, which partially explains its use for the drive disk. This experiment involves hydrodynamics of up to five material species (phases), coupled with thermal and radiation transport.

The other half of the CRASH program is the development of a code to simulate the experiment. The CRASH code solves for either a gray radiation intensity or a set of multi-group intensities. Scattering is ignored. Three characteristic temperatures exist: ion temperature, electron temperature, and effective radiation temperature[1]. This follows from the LTE assumption: ions are in LTE with themselves; electrons are in LTE with themselves; and radiation is not in LTE with the ions, electrons, or, in general, with itself. Opacity is usually obtained by a look-up table, but the option is present for an opacity solver. The Rosseland mean opacity is used for the radiation diffusion coefficient, while the Planck mean opacity is used for emission and absorption [209]. The software solves the problem in three steps using operator-splitting, a computational technique to treat different types of physics processes separately (at least one in an implicit fashion) [49]. The first two steps are an explicit shock-capturing hydrodynamics solver and a linear advection of the radiation. The third step implicitly solves the stiff problem of radiation diffusion, heat conduction, and energy exchange. It is this third step on which we focus: energy exchanges between materials via diffusion/heat conduction. Note that CRASH solves radiation-hydrodynamics, where the fluid equations are coupled to the radiation transport. What we have presented in Equation (2.46) is a stationary version of this system of equations, where all the velocity-dependent terms are neglected.

2.4.2 A Simplified Problem

This section reduces some of the intricacies of the motivating problem in order to focus on improving the treatment of mixed cells. The three primary simplifications are to treat the phases as stationary; assume constant opacity; and ignore scattering, absorption, and emission. This enables us to examine the treatment of interfaces while eliminating many complications. The results of this work on the simplified problem are still directly applicable to more complex cases.

The first simplification we make is the assumption that phase boundaries are fixed, effectively freezing any hydrodynamics. The CRASH code solves the radia-

[1] The energy distribution of the radiation field may be approximated by an effective radiation temperature.

tion diffusion, heat conduction, and energy exchange problem separately from the hydrodynamics and advection problems [209]. Within a single time-step, the fluid boundaries can be considered fixed with initial temperatures for each phase. The phases then exchange energy based on the surface area of the interface, the length of the time-step, the temperature difference, each phase's thermal diffusivity and radiative opacity, and of course any photon emission and absorption. Since our goal is to examine treatment of mixed cells in diffusion problems, we can make the same simplification of treating the energy exchange problem separately. We can also ignore all hydrodynamic effects by assuming the phase boundaries are fixed. (The CRASH code does not make this simplification; it keeps velocity terms up to some order.) While this stationary assumption loses much of the dynamics of the system, these dynamics are unrelated to the energy exchange problem, at least for a single time step.

The second simplification we make is to assume a constant opacity for each phase, meaning the opacity for a phase does not vary with space, time, or temperature. The diffusivity of a phase depends on the opacity, and opacity of a real material can be an acutely complicated term. Let us consider the system in Equation (2.46), the material energy and diffusion approximation of the RTE. Opacity (σ_t) is inversely proportional to the MFP and the diffusivity ($D = (3\sigma_t)^{-1}$). From Equation (2.46), it is clear that emission is proportional to T^4. The gray absorption opacity is frequently modeled as $\sigma \propto T^{-3}$ [103, 153, 181, 182, 184, 217]. This relationship is highly nonlinear and shows why radiative transfer problems are so stiff. There are two kinds of opacities in the system in Equation (2.46): absorption opacity (σ_a) and scattering opacity (σ_s). These effects can be combined to form a total opacity, $\sigma_t = \sigma_a + \sigma_s$, although as noted by Brunner [42], this combination is only true for energy dependent equations, and may be untrue once averaged or discretized. In some systems scattering is a small effect and can be neglected [103], so $\sigma_t \approx \sigma_a$ (CRASH makes this assumption). For realistic materials, opacity depends on several factors, such as the density, temperature, and/or EoS. Also, opacity is a frequency-dependent variable. A lucid example of frequency dependence is that a person is opaque in visible light but transparent for x-rays, while bones are opaque for both frequency regimes. The computation of accurate opacities is challenging due to their highly nonlinear nature, their dependence on several material characteristics, and that they combine several possible interactions (scatter; absorb; bound-bound, bound-free, and free-free transitions). Calculations often use a mean opacity, one that averages all of the possible effects to give a single value. In practice, opacities can be obtained via a look-up table — often the SESAME opacity library from Los Alamos National Laboratories. The two most common types of averaged opacities used are the *Planck mean opacity* (σ_P) and the *Rosseland mean opacity* (σ_R), defined as

$$\sigma_P \equiv \frac{1}{B(T)} \int_0^\infty \sigma_\nu B_\nu d\nu \tag{2.47a}$$

$$\sigma_R \equiv \frac{\int_0^\infty \frac{\partial B_\nu}{\partial T} d\nu}{\int_0^\infty \frac{1}{\sigma_\nu} \frac{\partial B_\nu}{\partial T} d\nu}, \tag{2.47b}$$

where $B(T)$ and $B_\nu(T)$ is the frequency averaged and frequency dependent Planck function [65]. It is not our goal to improve the computation of opacities or to explore when to use σ_R instead of σ_P. Rather, our goal is examine the treatment of opacities on an interface. A mixed cell contains more than one phase, and our interest is to determine how best to model this as multiple pure phases and not some intermediate mixture. Therefore, it is logical to use a constant opacity for each material, avoiding a convolution of the complicating effects of a variable (and more physical) opacity.

The third and most drastic simplification we make is to ignore absorption, emission, and scattering, changing our problem to one of pure diffusion. By ignoring scattering, we follow the CRASH code. Comparing Fick's second law, Equation (2.1b), with the diffusion equation for the radiation field, Equation (2.46b), we see that the only difference is presence of the absorption and emission terms ($\sigma_a[acT^4 - E]$) in Equation (2.46b) that are not present in Fick's second law. If these terms where to cancel, the remaining equation would be

$$\frac{1}{c}\frac{\partial E}{\partial t} + \vec{\nabla} \cdot \vec{F} = Q.$$

Physically, the assumption that $\sigma_a[acT^4 - E] = 0$ means that the radiation and material are in LTE, and that the temperature of the radiation is equal to the temperature of the material. As mentioned in Section 2.1.1, Fick's laws of diffusion mirror Fourier's laws for heat conduction. Therefore, with our assumptions, we can choose our system to be one of heat conduction,

$$\rho c_v \frac{\partial T}{\partial t} = -\vec{\nabla} \cdot \vec{F} + Q, \tag{2.48}$$

where

$$\vec{F} = -D\vec{\nabla} T. \tag{2.49}$$

Thus, the problem we are solving is heat conduction for a discontinuous media. For this problem, we assume perfect thermal contact, which is akin to lacking thermal contact resistance. While this is true for fluids, this is an idealization for solid materials, which have voids or surface roughness. These voids (or roughness) create thermal contact resistance, which results in a temperature jump at an interface [107, 192]. However, inclusion of thermal contact resistance ($\lfloor |T| \rfloor \neq 0$) would not change the techniques other than adding a constant to the continuity equation.

Although some physics in the motivating problem has been lost, we gain the ability to garner deeper insight into the interface problem. Our assumption of constant opacity allows the examination of mixed cells to be straightforward: a mixed cell contains diffusivities $D_1 = 10^n$ and $D_2 = 10^{-n}$ (at various ratios), which is much more tractable for analysis than diffusivities based on physical opacity values that are dynamic functions of the system. We also eliminate hydrodynamics from our problem, which is justified because the hydrodynamics are frequently treated separately from radiation transport in practice, such as with the CRASH program. The problem we solve is no longer that of radiation diffusion, but rather heat conduction. This is a non-issue: since we wish to examine the treatment of mixed cells on a sharp interface, the results of the simplified problem will be directly transferrable to the more complicated radiative transfer case (as well as any mixed cell occurrence). Additionally, since heat conduction is, mathematically, still diffusion, the simplified problem uses the same numerical schemes as the motivating one. Consequently, rather than just simplifying the physics, we are transitioning the problem to a higher level of abstraction, increasing the breadth and scope of this work.

2.4.3 Previous Work

Much of the previous work on discontinuous coefficients does not involve mixed cells, but rather, concentrates on methods that handle the stiff, nonlinear problems of adjacent pure cells, with large jumps in coefficients. Although this thesis is focused on treating mixed cells, discontinuous pure cells are apropos to mixed-cell problems because the techniques involved are often similar.

Section 2.4.3.1: Solving the nonlinearities. The first techniques reviewed solve the nonlinearities of the stiff system. By treating nonlinearities, highly discontinuous coefficients do not lead to errors or instabilities because the method specifically converges nonlinear effects. This is related to the mixed-cell problem because cases where mixed-cell error is a dominant effect are typically also highly nonlinear, such as in radiative transfer.

Section 2.4.3.2: Fictitious points and expanding the stencil. Handling strong phase jumps is often done by expanding a 2D stencil from five to nine points while introducing fictitious unknowns on the faces of the cells. A nine-point stencil captures effects that are not aligned with the grid better than a five-point stencil. A fictitious point is an additional unknown that is added to the stencil. However, instead of solving for this point computationally, the value can be determined by algebra, physics, or some other means. The value of this fictitious point is then substituted into the equations. Since the unknown is determined before the calculations, it serves as an intermediate step, which is why it is called a fictitious unknown rather than simply an additional unknown. Most mixed-cell techniques also use an expanded stencil and fictitious unknowns.

Section 2.4.3.3: Mimetic finite difference methods. Another growing approach is that of mimetic finite difference methods (MFD), where discretizations are approached in a nonstandard way which yields superior results to standard finite difference methods (FDM). A mimetic method typically gives more accurate results, including in interfacial regions. Another benefit of a mimetic method is that the computational framework easily handles strongly discontinuous diffusivities. This section goes into detail of the origin and development of MFD because one of the main mixed-cell approaches used in this thesis is based on a mimetic method (see Chapter III).

Section 2.4.3.4: Previous mixed cell work. Finally, there are some techniques for the highly discontinuous problem that address mixed cells directly. This section is related to mixed-cell work because it discusses competing or alternative techniques. We do not discuss the most common solution to the mixed-cell problem (to avoid them by using body-fitted grids), as this is akin to the trivial solution of a differential equation. We mention only a few of the many cell-cutting methods (where the mixed cell is partitioned into smaller pure cells), which are the next most common approach and very related to body-fitted grids. A few mimetic approaches are mentioned as well as a family of methods based on emending the Taylor expansion from which FDM are derived. The largest family of the Taylor expansion techniques is related to the immersed interface methods.

2.4.3.1 Solving the Nonlinearities

When using the diffusion approximation of radiation transport and making the assumption that the matter and radiation are in LTE, then the radiation energy density assumes a blackbody distribution ($E \propto T^4$), giving a diffusion equation

$$\frac{\partial T}{\partial t} = \vec{\nabla} \cdot (D(T)\vec{\nabla}T^4), \tag{2.50}$$

where the standard assumption is that opacity depends inversely on T^3 ($D \propto \sigma^{-1} \propto T^3$). Equation (2.50) is a highly nonlinear and stiff system. The diffusivity at phase boundaries in such a system can have ratios as large as 10^{10} [181]. Although the problem is highly nonlinear, the standard solution methods integrate the governing equations with a linearized differential equation, with no attempt to converge the nonlinearities [29, 182, 184]. Such an approach may lead to instabilities. Furthermore, flux limiters are usually used in a linearized fashion (using explicit or time-lagged values to evaluate the limiter) [29, 184, 202].

Olson, Rider, and Knoll (ORK) have several related works that address multi-phase radiation transport for problems with highly discontinuous diffusivities using FLD [112, 113, 114, 181, 182, 184]. ORK solve the nonlinear equations using a highly efficient multigrid preconditioned Newton-Krylov method. They do not consider mixed cells, but rather consider the problem of highly discontinuous phases between pure cells. ORK consider both the equilibrium [184] and non-equilibrium [114] case between the material temperature and the radiation energy. They solve Equation (2.50) by implementing exceptionally efficient solvers.

A Newton-Krylov method contains an outer loop to solve the nonlinearities (via Newton's method) and an inner loop that solves the linear equations (via a Krylov method). The algorithm works as follows: The nonlinear term is estimated, and the Krylov system is solved using that value. The result is fed back into the Newton solver to better estimate the nonlinear term, at which point the new nonlinear estimate is used to solve the Krylov system again. This process repeats until a criterion is met that says the nonlinear value is converged.

ORK use the highly efficient Newton-Krylov method to solve this nonlinear heat conduction problem, ensuring that the nonlinearities are converged. Given a good initial guess, Newton's method deals well with nonlinearities. Krylov methods are exceptionally efficient at solving linear algebra equations. The particular Krylov method used is GMRES, which, although highly effective, has a disadvantage of increased storage cost of vectors and work requirements as the number of iterations grows. The combination of these two methods, Newton and Krylov, can yield accurate solutions with exceptional efficiency. However, these methods can require too many iterations or fail to converge without a good initial guess or effective preconditioner.

For this reason, ORK use a multigrid method for a preconditioner based on a Picard-type linearization of the governing equations.

Multigrid, first conceptualized in the 1960s [80, 81], was proposed and systemized by Brandt in the 1970s [31, 32]. Typically, non-multigrid schemes quickly converge high frequency errors but are slow to converge low frequency errors, where the frequency is related to the grid spacing. A multigrid method accelerates convergence by solving the problem on multiple levels, converging high frequency errors on fine grids and low frequency errors on coarse grids. The multigrid method can solve different types of equations and is considered among the most efficient iterative methods [93].

ORK utilize multigrid as a preconditioner because its efficiency stays relatively constant as the problem size grows, and it is more efficient per iteration than other typical preconditioners [181]. An advantage of Newton-Krylov with Picard-type multigrid preconditioning is that the Jacobian matrix is never formed — it shares this characteristic with some methods, but not others [57, 58, 184]. ORK also apply their techniques to other systems, such as the Navier-Stokes equations [112]. Applications of this work include astrophysical phenomena, inertially confined fusion, combustion, and hypersonic flow [184].

In the stiff, nonlinear, highly discontinuous diffusion problem, converging the nonlinearities improves accuracy. A method that does not converge the nonlinearities is forced to either use very small time steps or to give deficient results. While converging the nonlinearities as done by ORK is very efficient, it does not offer much insight in how to deal with mixed cells. However, any model that does treat mixed cells for radiative transfer (or other nonlinear) problems should consider ORK's numerical scheme in conjunction with the mixed-cell model.

2.4.3.2 Fictitious Points & Expanding the Stencil

A common theme in 2D work that deals with discontinuities in phase properties is the transition from the standard five-point stencil (center cell plus the cells west, east, north, and south) to the nine-point stencil (standard stencil plus the cells northwest, northeast, southeast, and southwest) [4, 199, 56, 75] as well as using a full tensor for the diffusivity. Alcouffe *et al.* [4] were one of the earliest to examine strongly discontinuous coefficients, originally motivated by neutron diffusion. They noted that existing methods could handle jumps with ratios under ten, while jumps greater than an order of magnitude had poor convergence. In their approach, the phase boundary lies on grid lines, leaving only pure cells. Using multigrid, Alcouffe *et al.* found that expanding to a nine-point operator from a five-point operator gave better results.

Shubin *et al.* [199] consider the problem for displacement fluid flow for oil recovery processes, calling the problem *adverse mobility ratio.* Since their application is fluid displacement, the phase coefficient is mobility rather than diffusivity, and adverse mobility ratio is another way to say *highly discontinuous coefficients.* They present an example where two finite-difference grids at 45° angles yield drastically different results, when they should be identical. Referring to this phenomenon as *grid orientation effects*, Shubin *et al.* note that grid refinement actually exacerbates this problem. They reduce the grid orientation effect by employing a nine-point stencil, using a full tensor diffusion coefficient, and introducing fictitious corner and face-centered values by interpolation. The physical diffusion term is replaced with a rotationally invariant numerical diffusion term to reduce spurious oscillations, with a similar term for artificial dispersion. Mixed cells can be considered a type of grid orientation effect because mixed cells occur when interfaces not aligned with a grid. Consequently, the work of Shubin *et al.* is related to mixed cells.

Crumpton *et al.* [56] consider the strongly-varying-coefficients problem with fictitious face-centered unknowns and interpolated vertex values. The method uses finite volume methods (FVM) with a full tensor diffusivity and a multigrid solver for solutions of elliptic diffusion equations. The fictitious points enable Crumpton *et al.* to partition each element into eight triangles, with each triangle defined by a cell-center, a face-center, and a vertex value (see Figure 2.1(a)). The face-centered values are solved using flux continuity, then substituted back into the discretization. The vertex points are found through interpolation. The method reduces to the standard discretization with harmonic means when the permeabilities are a diagonal tensor. They note that this method can be applied to non-uniform structured grids with more advanced interpolation. The main idea of the approach of Crumpton *et al.* could be used to partition a mixed cell into several pure subcells without actually cutting the cell and adding any unknowns, which is similar to the S3 method presented in Chapter III.

Edwards [73] examines flux continuity for mismatched grids resulting from adaptive mesh refinement (AMR) in cell-centered FVM by introducing fictitious unknowns on the face-centers of fine grids adjacent to course grids. This work was motivated by reservoir simulations while noting applications to fluid dynamic systems such as incompressible Euler and Navier-Stokes equations. In this work, an interface refers to a boundary where a coarse cell meets a fine cell rather than a division between phases. Implementing continuity of flux from cell-centered quantities across an interface causes severe errors because the line joining the centers is not orthogonal to the

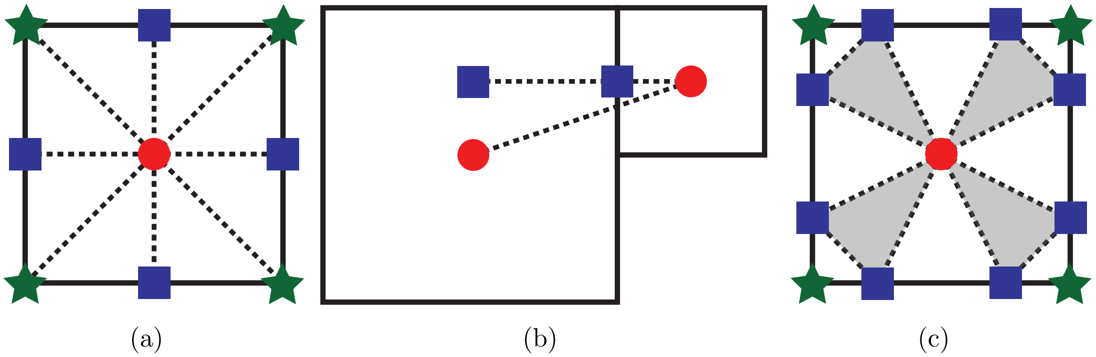


Figure 2.1: Sketches of cells to illustrate some concepts in the previous work section. Important cell-centers (red circles), vertices (green stars), and fictitious values (blue squares) are indicated. Subfigure (a) shows how Crumpton *et al.* [56] partition each cell into eight triangles. Subfigure (b) shows an example of a hanging node, where the face of one cell is larger than the face of its neighbor (commonly found in AMR grids). This figure elucidates how the center-to-center line of cell centers on a hanging node is not orthogonal to the cell face. Subfigure (c) shows how Edwards *et al.* [75] form four triangles in each cell.

interface (see Figure 2.1(b)). While mismatched cells are not identical to the discontinuous coefficient problem, Edwards' method is germane. In order to compute a flux normal to the interface, Edwards introduces fictitious values on the cell face. These face-centered values can be solved algebraically by flux continuity. The resulting flux at a face depends on the harmonic mean of each cell's permeability, a common occurrence in reservoir simulations [14, 73]. Interpolation is used between the cell-centered and face-centered values to define yet another new value. These final fictitious values are located in coarse cells such that the line joining these values and the cell-centered value of the fine cells is orthogonal to the interface (see Figure 2.1(b)). This process results in a fully conservative system with a symmetric positive-definite (SPD) coefficient matrix that can be solved by a preconditioned conjugate gradient method. All interfaces considered in this work are aligned to the grid since every cell is a rectangle, however the concept of solving for fictitious face-centered values in order to compute flux normal to an interface is applicable to mixed-cell problems where interfaces are not aligned to the grid.

Edwards and Rogers [75] consider the discontinuous coefficient problem, providing a conservative, flux continuous, cell-centered quadrilateral, FVM for full tensors. Full tensor equations arise whenever the computational grid is not parallel to the principal axes of the local tensor field. This occurs in the motivating problem of a heterogeneous reservoir simulation, where there are strong discontinuities and rapid variation in permeabilities. Other full tensor examples include any anisotropic medium not aligned with the computational grid, non-orthogonal or unstructured grids, or cross bedding. Full tensor schemes can be achieved in 2D using a nine-point stencil with some mesh options: a vertex based method [71, 74], a cell-face method [110], or a mixed method [7, 178]. A common mixed method is the cell-centered technique of introducing and then solving for face-centered (fictitious) values, which leads to effective permeabilities on the interface that are the harmonic mean of neighboring values. A cell-centered mixed method is possible with a five-point stencil only if the phase coefficient is a scalar or diagonal tensor.

Edwards and Rogers' method [75] of solving a full tensor, cell-centered problem first requires expanding the stencil to a nine-point scheme and then introducing two fictitious values on each cell face. The flux normal to each face of a quadrilateral is continuous. Each cell has a constant permeability (pure cells), but permeability may be discontinuous amongst neighbors. Each quadrilateral and its permeability tensor is transformed to a square domain by an isoparametric mapping. The vertex of each cell is then surrounded by four fictitious cell-face values, such that there are two new unknowns on each cell face. The term face-centered is not applicable because the values are not at the mid-point of a face. Four triangles per cell are introduced, defined by the center value of that cell and two face-centered values nearest to each vertex (see Figure 2.1(c)). A quadrature is introduced to enforce piecewise linear variation over each triangle. This leads to two expressions of flux between a cell-center value and one of its faces, with one flux equation for each cell-face unknown. An interior cell face then has four associated flux equations (two from each cell), and these four equations are related by continuity of flux. This defines a system of equations that can eliminate the cell-face unknowns by expressing them entirely in terms of cell-centered values. Substituting the cell-face values into the flux equations, an SPD coefficient matrix is formed that expresses the flux normal to each quadrilateral using only cell centers. Edwards' methods [73, 75] have applications to mixed-cell methods that cut cells into smaller fragments and mixed-cell methods involving AMR. His methods are also applicable as a mixed-cell method that fictitiously cuts a cell, similar to Crumpton *et al.* [56] as well as the new mixed-cell method that we propose in Chapter III.

2.4.3.3 Mimetic Finite Difference Methods

Mimetic finite difference methods (MFD) are a relatively recent discretization technique, originating in the mid 1990's [90]. Mimetic methods are a subset of finite difference methods (FDM). The pivotal difference is that standard FDM discretize a differential equation, while MFD discretize an operator (typically the gradient, divergence, and curl). The operators must take a coordinate invariant form, depending only on quantities such as volume, area, and angle [100]. Once the operators have a discrete form, any differential equation written in terms of these operators can be discretized by matrix manipulations, with different coordinate systems defined by changing the geometric quantities of the operators. By discretizing these vector-calculus operators in terms of the physics of the problem, symmetry and conservation properties survive the conversion from continuum to discrete. It can be shown that the discrete analog of the continuum equation shares or mimics many important properties; this is why schemes of this type are called mimetic [100]. Relevant vector-calculus identities are also preserved, notably the Divergence Theorem, which relates the volume integral of the divergence of a function to the outward-normal integral of the surface of said volume. MFD have a significant advantage over standard FDM: results are more physically accurate without a more complex discretization [90]. Mimetic methods have comparable accuracy to other FDM on 'nice' problems (simple grids, smoothly varying coefficients, etc.), but they are superior to FDM and mixed FEM on more difficult problems [100, 198]. All cited MDM papers considering interfaces make the assumption that the interface is aligned with the cells (no mixed cells) unless otherwise noted. This is a reasonable assumption for most MFD work because MFD typically use non-orthogonal grids, where the cells are body-fitted.

Mimetic methods are often derived by the Support-Operators Method (SOM) [100]. (For a detailed discussion of SOM; see Chapter III.) SOM decomposes to standard FDM when the grids are orthogonal (Cartesian). SOM requires a prime and derived operator, with an integral identity relating them. The derived operator is supported or referenced by the prime operator. The diffusion equation consists of the divergence of a gradient, called the diffusion operator,

$$\mathsf{L_D} = \vec{\nabla} \cdot (\mathbf{D}\vec{\nabla}). \tag{2.51}$$

SOM uses a mimetic description of the divergence operator ($\mathsf{D} = \vec{\nabla}\cdot$) as the prime operator. The derived operator is the modified gradient ($\mathsf{G} = -\mathbf{D}\vec{\nabla}$) or flux operator. The gradient operator is then defined in terms of the primary (divergence) such that they are adjoint ($\mathsf{G} = \mathsf{D}^*$) with respect to an inner product space defined by an integral relation. This scheme is conservative and contains many other advantageous properties that are discussed in Chapter III. The use of the flux operator instead of only the gradient is one of the key choices that makes SOM so applicable and advantageous for heterogeneous diffusion problems.

Samarskii *et al.* [190, 191] laid the foundation for mimetic methods in the early 1980's. They define two restrictions that should be imposed on any difference scheme: (1) the approximation and stability of the solution must converge to the exact value when the mesh intervals become small enough, and (2) the discrete analog must retain important properties of the original PDE (such as conservation, vector-calculus properties, etc.). Many schemes satisfy the first property, but few focus on satisfying the second. Samarskii *et al.* [190, 191] first approximate finite difference analogs of tensor calculus operators. Samarskii *et al.* use the term 'reference operator', which defines one operator in terms of another. This is the predecessor to SOM.

About a decade later, Shashkov built from Samarskii's foundation and established MFD and SOM. He and Liska [137] gave the underlying mathematical details while also providing a technique to use symbolic algorithms to construct FDM with SOM for a non-orthogonal logically rectangular grid. Use of symbolic algorithms is useful for 3D, where the equations become involved. Shashkov and Steinberg [196, 197, 198] provide a cornerstone for all subsequent MFD and SOM work. Shashkov's book [196] (edited by Steinberg), in particular, gives explicit mathematical details and descriptions of many of these operators (applicable to a general grid), as well as Fortran source code for implementation. In References [197, 198], they solve the diffusion equation with rough coefficients in rough grids using SOM. Shashkov has noted that the term Support-Operators Method is a flawed translation from the original Russian name, suggesting that *basic* or *reference* operator would be a truer translation [198].

Morel *et al.* [150] created a cell-centered Lagrangian method for general quadrilaterals not using SOM, but ended up with a matrix that was not symmetric positive-definite (SPD). Morel *et al.* [152] then use SOM to make a very similar method that yields approximately the same results and error values. However, this new method could be solved faster and more efficiently because the matrix was SPD. This simultaneously shows the advantage of solving an SPD matrix as well as using SOM. Later, Morel *et al.* [151] expanded their method to 3D with general six-sided cells (hexahedral meshes). While Shashkov's method [198] forms a dense matrix due to an inversion step, Morel's method [151, 152] is sparse, reducing the complexity of the matrix-vector multiplication used in the Krylov solvers. The sparseness of the matrix is because Morel *et al.* use a local SOM; all of the nonzero values in a row of the coefficient matrix correspond to the nearest neighbors of the cell associated with that row. The cost of making the matrix sparse is that additional unknowns on each face-center must be added. However, the gain in solving efficiency more than offsets the cost of extra unknowns.

Hyman and Shashkov (and often Steinberg and Morel), published work of a much more mathematical nature, formally proving the properties of not only SOM, but the mimetic discretizations in general of the divergence, gradient, and curl operators [95, 96, 97, 98, 99, 100]. They specifically explore the behavior of strongly heterogeneous problems [95, 99], while also looking at boundary conditions [97] and convergence properties [100]. Konstantin Lipnikov is another mathematically focused contributor to the development of SOM, often in collaboration with Shashkov. Lipnikov *et al.* explore diffusion for non-orthogonal, nonconformal meshes for general quadrilaterals [132] and general polyhedral meshes [135]. Lipnikov *et al.* [133] consider two-phase flow in porous media in a heterogeneous reservoir.

The following is a brief review of some of the many other works that explore or advance mimetic methods. Examples of other mathematical papers, either proving properties or examining convergence behaviors of mimetic methods include References [22, 24, 26, 35, 37, 186]. The relation of SOM to mixed FEM is mentioned in Morel *et al.* [152] and explored in References [22, 91]. SOM can be expanded from second-order to fourth- or sixth-order accuracy with grids that are uniform or general, in one or two dimensions [47, 48, 90, 149]. Many different grid types are explored, such as staggered 1D and general quadrilaterals. Some of the more complex grids developed include hexahedral [151], unstructured polygonal and polyhedral [37, 135, 136], and polyhedral with curved (nonplanar) faces [35, 36]. A polyhedral mesh, with each element having many faces, is more likely to have a face perpendicular to the flow [34], which is advantageous since only the flux normal to the flow is continuous. Works focusing on applications of the tensor capabilities (such as for an artificial viscosity) of mimetic methods include References [24, 34, 36, 44, 45, 134].

Domain decomposition is another method that can treat interface problems. Domain decomposition is a technique where the full domain is partitioned into smaller segments (such as by phase boundaries). FEM are typically used when domain decomposition splits the problem by phase into homogeneous regions. Each partition may have its own grid and resolution, which allows each domain to be solved in parallel. Domain decomposition may be overlapping, where interfacial regions are contained in multiple domains, or non-overlapping, where iterations are needed to constrain the interface values to match each other. Solution matching at mismatched grids can be tackled by introducing Lagrange multipliers at the boundaries; this space is called a mortar finite element space [23]. An example of a standard approach (not mimetic) is exampled by Braess *et al.* [30]: using a multigrid technique that is fast for homogeneous regions, they apply mortar elements with domain decomposition to handle strong phase jumps. Berndt *et al.* use mimetic mortar FDM for nonmatching grids from domain decomposition [23].

2.4.3.4 Previous Mixed-Cell Work

All work cited to this point has consisted entirely of pure cells, in which phase boundaries are aligned with cell faces. While body-fitted grids avoid mixed cells altogether, in many applications it is either not possible or not practical to align the mesh with interfaces, leading to the need for mixed cells. This section discusses some of the mixed-cell approaches in the literature, such as cutting the cell, mimetic techniques, and immersed interface methods.

Cutting the cell. The most logical approach to treat a mixed cell is to simply cut the cell into pure cells. This leads to irregularly shaped cells and body-fitted grids. There are examples of cut-cell approaches, many of which also involve merging (amalgamating small, irregular cells into a larger cell). Three examples follow, the latter two being mimetic methods.

Ye *et al.* [218] cut mixed cells and merge the smaller fragment with neighbors of the same phase. They examine large ratios of the phase property (strongly discontinuous coefficients) in the context of liquid to vapor phase changes for bubble dynamics. While body-fitted grids are a good choice for discontinuous phases, fixed grids are more suitable in some circumstances, such as when there is a phase transition with sizable volumetric changes. The approach of Ye *et al.* to this problem is to use an Arbitrary Lagrangian Eulerian (ALE) method, where marker cells track the interface in a Lagrangian method while the field calculations are performed on an Eulerian mesh. Rather than lose the ability to map one-to-one back to the Cartesian grid by creating more total cells after cutting, the smaller portion of a cut cell is merged with a neighbor of the same phase. This leads to a constant number of cells, with uniform cells far from the interface and irregular cells near the interface. FVM are used to solve the coupled fluid equations (mass, momentum, energy transfer) to account for both the Cartesian and trapezoid cells. Fluxes are estimated normal to any of the cell surfaces based on interpolation. Results are presented in terms of bubble dynamics, such as accuracy of drag coefficients or growth rates, making it difficult to compare the effectiveness of this method to other mixed-cell techniques. (See Ye *et al.* [218] for details on bubble dynamics.) The downsides of this method include the creation of irregular cells near the interface, as well as the standard downsides of using an ALE method, such as the cost associated with marker points and the cost and physical errors introduced by re-meshing.

There is a scant number of mimetic approaches to mixed-cell problems, primarily because mimetic methods are often employed with non-orthogonal grids in order to align with interfaces. While a mimetic method can use body-fitted grids, it can also approach a problem using a uniform, Cartesian grid. We know of only three examples of mixed-cell approaches using mimetic techniques [86, 118, 216]. Two of the approaches [86, 118] simply cut the mixed cell into pure cells, while the third approach is in 1D only [216]. *Two unique, 2D Cartesian grid, MFD mixed-cell models that do not cut the cells are presented as new work in Chapter III.*

Garimella and Lipnikov [86] cut mixed cells for a 3D mesh, solving a multiphase diffusion equation. Interfaces are reconstructed inside mixed cells using volume fractions, and then the mixed cells are cut into pure cells using these interfaces. This new mesh, having irregular or polyhedral elements, is then discretized and solved with MFD. Cells are not simply cut in two, but rather into several parts such that the mesh is conforming. "Conforming," in this sense, means that faces on either side of a interface have the same length, avoiding hanging nodes (see Figure 2.1(b)). Cells are amalgamated when possible to help reduce the cost of creating more elements and unknowns. Garimella and Lipnikov show their method is second-order accurate and has lower error than averaging methods (harmonic and arithmetic), which they show to be first order. They generate a linear and polynomial test solution using the method of manufactured solutions (MMS) (discussed in Section V) with a diffusivity ratio of 100. They demonstrate that a mixed-cell model can be benchmarked by the MMS, and its performance compared to the averaging techniques (harmonic and arithmetic mean) that are employed in standard practice. The mixed-cell models presented in Chapters III-IV also use the harmonic mean as a reference point in Chapters VII-VIII.

Kuznetsov *et al.* consider multiphase diffusion problems in References [91, 118, 119]. They first approach the problem using MFD on polygonal meshes, but they assume no mixed cells [119]. They achieve second-order convergence on locally refined, nonmatching meshes. Kuznetsov approaches the discontinuous problem again, this time with mixed cells, in a related paper [118] using mixed FEM (which has been shown to have a deep connection to MFD [22, 91, 152]). His approach is to cut the cells into several triangular pure cells. Kuznetsov works with Gvozdev and Shashkov in Reference [91] to perform a similar technique where mixed cells in polygonal meshes are triangulated into many pure cells. This method works in both 2D and 3D. Their results have a jump in error at the interface that is approximately an order of magnitude lower than the jump in error at the interface of a more typical method.

Noncutting MFD. Winters and Shashkov [216] consider the mixed-cell problem in diffusion using MFD (and SOM specifically) with two approaches, homogenization and fictitious unknowns. To the author's knowledge, Winters *et al.* is the only example of a mimetic technique to address mixed cells without cutting. The method is limited to 1D and two phases only, but they show it is easily extended to multiple phases. In the homogenization method (volume-weighted averaging of the diffusivities) they compare results using three different means (arithmetic, geometric, harmonic). The condensation method [118, 119] adds a fictitious unknown on the interface then algebraically solves for these unknowns using continuity. The resulting equation for the flux is a coupled 2×2 system that reduces to the pure-cell flux expression if the volume fraction is zero or unity. This system that determines flux for each phase is rather intricate, especially for a 1D problem, and will quickly grow in complexity in higher-dimensions. Test problems are created using the method of manufactured solutions (discussed in Section V). Test results demonstrate that mixed cells drop the convergence from second-order to first-order and that the harmonic mean performs slightly better than the arithmetic or geometric mean.

Winters and Shashkov [216] also provide an estimate of each phase's individual temperature in a mixed cell using a limiter reconstruction technique. The limiter reconstruction of the derivative is similar to the method used in Berger and LeVeque [21]. The success of this technique in finding each phase's temperature depends on the approximation of the spatial derivative of temperature, which employs a limiter. Winters and *et al.* compare the *Barth-Jespersen* and *minmod* limiters. Their results do not show a clear consensus as to which is better nor do they obtain both temperatures to equal accuracy. This indefinite conclusion is attributed to imprecise interface knowledge. This work is related to our Chapter IV, where we present a new method that gives the temperature of each phase in a mixed cell. Also of note, two errors are reported in the ∞-norm: global error and pure-cell error. The pure-cell error shows how well the computation performs away from the interface, which is useful because the interfacial region (mixed cells) dominate the error. While Winters *et al.* make an original approach to the mixed-cell problem using SOM, the problem is greatly simplified in that it is only in 1D, in which the volume fraction exactly locates the interface and all fluxes are automatically normal to the interface because there is only one possible axis.

Predictor-corrector method. Pember *et al.* [170] take a predictor-corrector approach to the mixed-cell problem. Their method uses a nine-point stencil in 2D and a twenty-seven-point stencil in 3D. It works on a uniform grid or coupled with AMR for both Cartesian and cylindrical coordinates. In the first (predictor) step, fluxes are calculated as in the standard method, with no regard to the interfaces. A reference state is then calculated based on these fluxes, and all pure cells not neighboring a mixed cell are updated to this reference state. In the second (corrector) step, two corrections are calculated: one correction is conservative but (potentially) unstable and the other correction is stable but nonconservative. These two corrections are used to re-establish conservation; fluxes are redistributed by the amount that conservation was violated in a mass-weighted manner to mixed cells and their neighbors. The geometric information needed for this scheme includes the volume fractions, the surface fractions, the interface normal, and the interfacial area. These can be found in a variety of ways, including assumption of a piecewise linear interface (with volume fractions provided). Pember *et al.* report the scheme to be globally second-order while first-order near interfaces. Two 1-norm errors are reported: one considering the entire domain and another considering only interfacial regions (mixed cells).

Improved Taylor expansions. There is a group of methods that addresses the discontinuous coefficient problem by taking a Taylor series on each side of an interface [21, 123, 124, 139, 142, 143, 215]. One of the assumptions for a Taylor series, the foundation for finite differences, is that the function is smoothly varying (infinitely differentiable). This assumption is violated near a discontinuous interface, implying that the standard FD discretization is invalid in such regions. This can be remedied by either correcting the Taylor expansion or by taking separate expansions on each side of the interface, where the phase parameters are constant or smooth.

An early example of the Taylor series approach is by MacKinnon and Carey [139]. They note that FEM with body-fitted grids naturally have flux continuity normal to the interface, but suboptimal convergence is seen if the element is not aligned with the grid. In FDM, an interface not aligned with the grid is typically accommodated using an intermediate value (homogenization), such as the arithmetic, geometric, or harmonic mean. Using a Taylor series on each side of the interface, MacKinnon and Carey calculate some correction terms to the standard FD discretization. Comparing results with and without these correction terms, as well as with results using different mean values (arithmetic, geometric, harmonic), they determine the following: (i) the harmonic mean is second order but does better for smaller diffusivity ratios; (ii) the arithmetic and geometric averages are only first order; (iii) the correction terms only improve the solution on coarse grids; and (iv) the harmonic mean is superior to their correction terms on a slightly refined grid.

Mayo [141] introduces a new approach to the discontinuous coefficient problem in terms of an integral equation. Mayo's integral method involves embedding a complex region inside a large, simple rectangular region [141, 142, 143]. Extending the domain to a rectangular region avoids common issues with integral techniques, and the nonsymmetric system can be solved with existing fast solvers. This method is applied to multiple problems and equations, but only the diffusion application is summarized here. The problem solves $\vec{\nabla} \cdot K\vec{\nabla}u = f$ in some complex shaped region D, which is embedded in a simple rectangular region R. This leads to three zones: cells where all neighbors are inside the boundary (interior), cells where all neighbors are outside the boundary (exterior), and cells with neighbors on either side (interfacial). The interior regions solve $\vec{\nabla} \cdot K\vec{\nabla}u = f$, while the exterior regions solve $\vec{\nabla} \cdot K\vec{\nabla}u = 0$. The interfacial values are treated with a new method. The value K is large inside D while small outside of it, leading to a discontinuous jump. Since the diffusivity K is not smooth in the transitional regions, the function u must have a kink. A Taylor expansion of a nonsmooth function is not valid. The Taylor expansion may be corrected if one knows the discontinuities in u and its derivatives in the normal direction. Mayo's method modifies the Taylor expansion by taking derivatives of the jump conditions in the normal and tangent direction ($\frac{\partial}{\partial \xi} \left\lfloor \left| u \right| \right\rfloor$ and $\frac{\partial}{\partial \xi} \left\lfloor \left| \vec{F} \right| \right\rfloor$ where ξ is the normal or tangent direction). Combining this with derivatives of the jump conditions for the differential equation ($\frac{\partial}{\partial \xi} \left\lfloor \left| \vec{\nabla} \cdot K\vec{\nabla}u \right| \right\rfloor$), terms in the Taylor series can be determined up to arbitrary order. The final result is a matrix equation $\mathbf{A}\vec{u} = \vec{b}$ where $\mathbf{A}$ is a standard n^{th}-order accurate discretization on the LHS and $\vec{u}$ is a vector of the unknowns. The RHS is $\vec{b}$, which is simply f inside region D with no boundary neighbors, $f + m$ inside region D with boundary neighbors, m outside region D with boundary neighbors, and 0 outside region D with no boundary neighbors. Since the correction term, m, only appears on the RHS, the matrix $\mathbf{A}$ is unaltered. Not altering the matrix allows rapid solutions with existing solvers, giving p^{th}-order solutions in irregular regions, where p is the order that is determined by the minimum of solver's accuracy as well as the accuracy of the Taylor expansion's correction.

Motivated by transient flows with strong shocks around complex boundaries, Berger and LeVeque [21] introduce a scheme with a rotated reference frame for mixed cells in Cartesian grids in conjunction with AMR. They desire to preserve the advantages of using uniform grid methods (i.e., no irregular or body-fitted cells), such as simple data structures and higher solution accuracy. In order to have continuity of flux normal to an interface, they create a rotated reference frame inside each mixed cell, calculating flux normal and tangential to the interface. This rotated difference scheme builds from related work [20]. Additional unknowns are needed to compute fluxes in this rotated frame. These fictitious values are created by taking a first order Taylor series on the interface, noting that a globally second order scheme can be first order near an interface [21, 123, 212]. Berger and LeVeque [21], citing work by Wendroff and White [212] state that, 'the local error can be one order lower than normally required in irregular grid cells without the error accumulating to destroy the global accuracy.' Fluxes are computed at these fictitious unknowns using a slope limiting linear reconstruction of the solution with the Taylor series expansion. For small cells, which emerge due to AMR, this method increases the stencil size. This method results in second-order convergence everywhere except the interfacial region, where it is first-order. This approach is very similar to the S2 method introduced in Chapter III, where a tensor rotates the diffusivity to calculate flux parallel and normal to the interface without introducing additional unknowns.

LeVeque and Li specify the immersed interface method (IIM) in References [123, 124]. This method is an offshoot of the immersed boundary method from Peskin [171], motivated by blood flowing through a beating heart. The immersed boundary method can only handle a delta function forcing term, not discontinuous coefficients. The IIM, however, is designed to handle discontinuous terms, such as a scalar diffusivity. Since cells near the interface form a lower-dimensional set, second-order global accuracy can still be maintained with only first-order truncation error near interfaces [21, 123, 212]. The main idea behind the IIM is to take Taylor expansions near the interface because the expansions from the standard discretization are not valid near a nonsmooth function.

We describe the 2D IIM in detail because it explicitly deals with mixed cells and has spawned several cognate methods, while References [21, 139] can be seen as predecessors. Since the diffusion coefficient is a scalar, a full nine-point (2D) scheme is not needed. For the IIM, the standard five-point stencil is modified by including one additional point for cells near boundaries. This sixth point is chosen to be one of the four additional values used in a nine-point stencil (NW, NE, SW, SE). A uniform Cartesian grid is assumed with mixed cells near interfaces. The calculation for a given mixed cell begins by selecting a point on the interface nearest to the cell-center; much of the subsequent calculations depend on this interface point. A coordinate transform is applied such that the local system has axes normal and tangent to the interface, with respect to the interface point selected. Note that the grid does not change. Rather, a second coordinate system is introduced, which is defined with respect to the interface at a chosen point. The jump conditions are then written in terms of values at the interface point. From the four corner cells (NW, NE, SW, SE), the one with the minimum distance from its cell-center to the interface point is selected as the sixth point in the stencil. A Taylor expansion is performed around this interface point for all six cells, with some being 'outside' the interface and some being 'inside'. The result for each term is an expression in the form of

$$\begin{aligned} u(\xi,\eta) = u^{\pm} &+ (\xi-\xi^*)u^{\pm}_{\xi} + \tfrac{1}{2}(\eta-\eta^*)u^{\pm}_{\eta} + (\xi-\xi^*)^2 u^{\pm}_{\xi\xi} \\ &+ \tfrac{1}{2}(\eta-\eta^*)^2 u^{\pm}_{\eta\eta} + (\xi-\xi^*)(\eta-\eta^*)u^{\pm}_{\xi\eta} + O(h^3), \end{aligned}$$

where h is the grid spacing, the $+$ and $-$ superscripts specify a side of the interface, ξ and η are the local coordinates, and (ξ^*, η^*) is the interface point (typically chosen to be the zero of the coordinate transform). The local truncation error for the differential equation can be expressed as a linear combination of the various terms in the six Taylor expansions (the standard five cells plus one additional cell). The truncation error, $T_{i,j}$, can be written as

$$\begin{aligned} T_{i,j} = a_1 u^- &+ a_2 u^-_{\xi} + a_3 u^-_{\eta} + a_4 u^-_{\xi\xi} + a_5 u^-_{\eta\eta} + a_6 u^-_{\xi\eta} \\ &+ a_7 u^+ + a_8 u^+_{\xi} + a_9 u^+_{\eta} + a_{10} u^+_{\xi\xi} + a_{11} u^+_{\eta\eta} + a_{12} u^+_{\xi\eta} + O(h), \end{aligned}$$

where the coefficients a_i come from the six Taylor series. A relation between each side of the interface is determined by the jump conditions, and these can be substituted into the local truncation error equation. By requiring only first order accuracy, the coefficients a_i must all be zero, which yields a system of six equations with six unknowns. For the case that the diffusion coefficient is the same on both sides of the interface, the coefficients for the sixth stencil point are zero, and the problem reduces to the standard five-point discretization. The values of the coefficients depend on the position of the interface relative to the stencil and the ratio of diffusivities. The discontinuous diffusion equation is then solved in terms of cell-centered values only, with mixed cells having an extra stencil point to specify the flux with respect to the interface.

The IIM method is powerful and accurate, showing second-order accuracy in the ∞-norm. The IIM also allows for discontinuities for the solution or the flux across the interface, which is equivalent to a double-layer or single-layer source at the interface, respectively, in potential theory [123, 129], or thermal contact resistance for heat conduction. However, the IIM comes with a few notable disadvantages. First, this method requires detailed knowledge of the interface, which is not always available. Sometimes the interface is only known discretely, so an appropriate interpolation must be used. Other times, such as with a VoF method, the exact interface position is not known, only the fraction of each fluid is given in a cell. Second, the diffusion coefficient cannot be a tensor due to the stencil (six points instead of nine), which limits the applicability of this method. Third, the discontinuities must be 'mild'; this approach fails for strong jumps in the discontinuity. Fourth and most important, the addition of a sixth stencil point breaks the symmetry of the matrix equation. While a five-point scheme forms an SPD coefficient matrix, the IIM adds one term to the matrix. The matrix is still diagonally dominant because the sixth stencil coefficient is smaller than the other five. However, the matrix loses symmetry, which is an important property because an SPD matrix can be inverted with some of the more efficient solvers (see Section 2.5.3.1).

Li [129] addresses the main deficiency of the IIM, where he restores symmetry, leading to an SPD matrix. Li preconditions the diffusion equation before applying the IIM. The critical result of this approach is that the RHS of the matrix equation is modified, leaving the matrix itself untouched and SPD. Li notes that most other methods for treating this problem (such as harmonic averaging, smoothing, and FEM) may have second-order convergence of E_1 or E_2, but are rarely second-order with E_∞ since they smooth out the solution near the interface. Second-order infinite norm attempts include Mayo's method [141, 142, 143] as well as the IIM [123, 124]. Li notes that some problems with large jumps in the coefficient cause the IIM to fail or converge very slowly. Li's approach results in two separate second-order problems to be solved: Poisson's equation in the pure regions and a Neumann boundary condition in the interface region. While the total system can be solved as a block matrix problem, it is more efficient to solve the two problems separately. The pure regions are solved via a Schur complement system with GMRES, while the interface region is solved via weighted least squares. Hence, the price of restoring symmetry to the coefficient matrix of the IIM is preconditioning the problem before applying the IIM and then solving a second, albeit smaller, system.

Wiegmann and Bube [214, 215] continue the development of the IIM with the

Explicit Jump Immersed Interface Method (EJIIM). In this approach, they combine LeVeque's IIM [123, 124] with some of Li's modifications [129], as well as some of Mayo's techniques [141, 142, 143]. Wiegmann uses Mayo's idea of embedding a complex geometry inside of a simple rectangular region, treating the boundary as an interface. They also use Mayo's modification of the Taylor expansion by taking derivatives of the phase jumps. First implementing in 1D to model traffic flow [214], Wiegmann and Bube [215] later extend their method to 2D. They report second-order convergence for their 2D method for discontinuous coefficient problems with diffusivity ratios of 5,000 and 1/5,000. Similar to Li's method [129], the system of equations to be solved is either a block matrix system or two separate systems of equations, consisting of the bulk values and interface values. While Wiegmann and Bube do not retain Li's fix to keep the coefficient matrix SPD, their approach can correct FDM of arbitrary order. The only restriction, and this is critical to both Mayo's method and the IIM family, is that the interface and all coefficient jumps must be known with precision. Knowing the phase coefficient jumps is simple if they are constant or linear, but there are many cases where they are nonlinear (such as in radiation transport where the diffusivity can depend on the third power of temperature). Of greater importance, however, is the fact that this method depends on detailed knowledge of the interface. The Taylor expansions are taken with respect to points on the interface closest to the cell-center of each mixed cell. While some numerical methods do explicitly track the interface, others must reconstruct it, often with a linear approximation. Basing the second-order (and potentially higher) Taylor expansion on a (frequently linear) reconstructed interface defeats the purpose of the higher-order Taylor expansion. The IIM and its consanguineous methods are a useful and accurate approximation, but they are best applied when the interface knowledge (both the formula for the interface and the phase properties on either side of it) are known with high fidelity.

Recapitulation. In summary, multi-fluid problems in an Eulerian frame can lead to mixed cells. AMR can be used to alleviate the effect of mixed cells, but the same inaccuracies still exist on a smaller scale (not to mention the many complications AMR introduces). A common way to treat these is to form a body-fitted grid or to cut (and possible merge) cells into pure cells. These techniques are effective, but they lead to irregularly shaped cells that encounter numerical issues and require interpolation. If mixed cells are not simply avoided or cut, they can be treated by homogenization, an averaging technique in which a (weighted) mean of each phase diffusivity is computed, and the cell is treated as a pure cell of this averaged value. Another common set of methods introduces fictitious unknowns on the cell-faces and solves for these using continuity. This is often accompanied by expanding the stencil from the standard five-point to a nine-point scheme. These approaches typically achieve second-order accuracy away from interface while being first-order accurate near the interface. However, in many cases, the global accuracy will eventually reduce to the accuracy of the lowest-order region of the domain.

Some unique approaches include a rotated-difference scheme [21], where the coordinates are transformed to a set parallel and perpendicular to the interface, and a predictor-corrector scheme [170], where the fluxes are modified in a conservative and stable way to account for the interfaces. Most mixed-cell schemes attain second-order accuracy globally but only first-order accuracy near the interface. However, there is a family of schemes that is second-order everywhere, which is accomplished by correcting the Taylor expansion for the finite difference derivatives. While these schemes are second-order on the interfaces, they require detailed knowledge of the interface and the coefficients near the interface, which are not always available. They perform best with only moderate jumps (jump strength is ultimately limited by order of the Taylor correction).

A mixed-cell method must be at least first-order accurate at the interfaces and second-order accurate globally to be equivalent to the majority of mixed-cell methods. Second-order accuracy at interfaces with strong jumps would be an unequivocally successful method, competing with the immersed interface family of methods as the best option for treating mixed cells without generating irregular (nonrectangular in 2D) cells. Use of E_∞ (defined in Section 5.2) as a metric is highly beneficial because it finds the least accurate area (which is near/on the interface), giving a measure of the interfacial accuracy. Any method that performs at second-order accuracy with E_∞ for systems with highly discontinuous coefficients could be a step forward to the field of mixed cells. Unfortunately, one would have to weigh the cost of such a method in terms of additional unknowns introduced and the required level of accuracy of interface knowledge, against those of competing methods. The new models proposed in this thesis require only volume fractions and a linear approximation of the interface. The new models do introduce additional unknowns, but they create a symmetric positive-definite (SPD) coefficient matrix, which can be solved efficiently. The next section discusses the benefits of an SPD matrix and explicates the conjugate gradient method as a means to solve an SPD system.

2.5 Efficient Iterative Methods

This section discusses the motivation for solving systems of equations iteratively and discusses an efficient class of solvers. Iterative methods are advocated due to the nonlinear and sparse nature of the system of equations. The class of efficient iterative solvers considered is the set of Krylov methods, focusing on the conjugate gradient technique.

2.5.1 Iterative vs. Direct Solvers

Any discretization method of a partial differential equation (PDE) leads to a system of algebraic equations connected to the grid points [93]. This system is often large, especially in industrial applications and 3D problems. The coefficient matrix is square (same number of rows as columns) and the entries are typically real. The coefficient matrix is nonsingular, which implies a unique solution exists. This system of equations can be written in the form of the matrix equation,

$$\mathbf{A}\vec{x} = \vec{b}, \tag{2.52}$$

where the coefficient matrix ($\mathbf{A}$) and the RHS ($\vec{b}$) are known, and $\vec{x}$ represents the unknowns. The sizes of these terms are $\vec{x} \in \Re^{n\times 1}$, $\vec{b} \in \Re^{n\times 1}$, $\mathbf{A} \in \Re^{n\times n}$, where n is some integer.

A system of equations, such as that described above, can be solved in two ways: directly and iteratively. If floating-point operations had no round-off error, a direct method would compute the exact solution in a finite number of steps. An iterative method, in contrast, makes successive approximations to the solution, with each iteration of the method (ideally) getting more accurate. Some methods do not have strictly monotonic convergence but do converge overall. Theoretically, an iterative solution converges to the exact solution after an infinite number of steps (iterations). In practice, exact solutions cannot be achieved because of the finite precision of floating point numbers. Iterative methods must have a way of judging how good a solution is at any iteration so that the iterative method will stop once sufficient quality has been achieved (stopping criterion). Therefore, solution approximations are often quantified with a residual,

$$\vec{r} = \vec{b} - \mathbf{A}\vec{x}, \tag{2.53}$$

where $\vec{x}$ is the current estimate of the solution and $\vec{r}$ is the residual. A common stopping criterion is

$$\|\vec{r}\| \leq \epsilon,$$

where ϵ is a chosen tolerance.

The number of operations to directly solve a general or dense n×n matrix is $\mathcal{O}(n^3)$, although some direct solvers require less than the third power of n, e.g., Strassen's algorithm of $\mathcal{O}(n^{2.81})$ and the Coppersmith-Winograd algorithm of $\mathcal{O}(n^{2.34})$ [207]. Direct methods are a good option for small or linear systems. However, as the size grows, the amount of time, the amount of memory, and the number floating point operations needed to solve the system may become more costly than an iterative method. Iterative methods require about $\mathcal{O}(n^2)$ operations to solve a dense system, but many require as few as $\mathcal{O}(n)$ to solve a matrix primarily composed of zeros (sparse) [207]. A direct method may or may not be able to take advantage of a sparse matrix. The round-off errors can compound for very large and dense systems in direct methods, preventing them from finding the correct solution. Nonlinear systems need an iteration scheme to converge the nonlinearities, making it economical to use an overall iterative scheme rather than iterating a direct method [93]. Therefore, nonlinear schemes are usually solved with iterative methods.

Iterative methods are simply trying to solve Equation (2.52) by inverting the matrix

$$\vec{x} = \mathbf{A}^{-1}\vec{b}. \tag{2.54}$$

Matrix inversion is typically the most computationally demanding step in solving a PDE and occupies most of the runtime. Considerable effort has been placed on creating efficient and cost effective methods of inverting matrices. Different discretization schemes and PDEs lead to different types of coefficient matrices, which require specialized inversion techniques. A coefficient matrix from a discretized PDE will typically be sparse and often SPD.

2.5.2 Sparse and SPD Matrices

A sparse matrix has mostly zeros for elements, with only a few bands of nonzero values. A band is a diagonal line through the coefficient matrix with nonzero values, always measured from the upper left (lower row and column) to the lower right (higher row and column). The sparsity from a PDE is due to the system being local. Local means that a given mesh element is only influenced by other elements within some radius or count (e.g., the two closest neighbors on each side). This is in contrast to a global method, such as the spectral method, where a mesh element is influenced by all other elements. Note that an AMR algorithm will invariably move nearest neighbors in the coefficient matrix; sparsity will be maintained, but neighbors may move far from their band.

As the name would suggest, a PDE involves derivatives, which are computed based on linear combinations of a cell with some of the cells around it. Each cell in the mesh has one corresponding row and column in the coefficient matrix. A given row in the coefficient matrix corresponds to the equation for that element. The main diagonal, when the row number is the same as the column number, has the largest magnitude in each row (may be most negative or most positive number). An off-diagonal element in the matrix defines the strength of the relation between the element corresponding to the row with the element corresponding to the column. The number of nonzero values in a row corresponds to the number of elements used to compute the derivative, which is also related to the number of bands in the coefficient matrix. Sparse matrices are ideal to work with because they require much less storage in memory as well as less computation. The multiplication of a vector by a matrix is much quicker if most of the matrix values are zero and may be skipped.

A symmetric matrix is one that is equal to its transpose,

$$\mathbf{A} = \mathbf{A}^T, \tag{2.55}$$

which means for an element (i, j), $A_{i,j} = A_{j,i}$. Symmetric matrices are advantageous because only the main-diagonal and the elements above (or below) it must be stored, for a savings of about half. The product of a matrix and its transpose is always symmetric, which can be seen by

$$\begin{aligned}(\mathbf{A}\mathbf{A}^T)^T &= (\mathbf{A}^T)^T(\mathbf{A})^T \\ &= \mathbf{A}\mathbf{A}^T,\end{aligned}$$

noting the reverse order of terms when distributing the transpose operator. A Hermetian matrix is one that is equal to its complex transpose, $\mathbf{A} = \mathbf{A}^*$. Since we are concerned with real matrices only, the transpose and complex transpose are identical.

A positive-definite matrix is the matrix analog to a positive number. For any arbitrary, nonzero vector of length n ($\forall\vec{x} \in \Re^{n\times 1}$), an $n\times n$ matrix ($\mathbf{A} \in \Re^{n\times n}$) is positive-definite if

$$\vec{x}^T\mathbf{A}\vec{x} > 0. \tag{2.56}$$

A semi-definite matrix is one that may yield zero for this product.

An SPD matrix is a matrix that has both symmetric and positive-definite properties. These matrices, especially when sparse, are an ideal matrix to work with for solving iterative systems. Such matrices often occur in practice; the standard, five-point discretization of the diffusion equation is SPD. An SPD matrix can be inverted with some of the most efficient solvers.

2.5.3 Krylov Methods

Krylov methods are some of the most powerful and efficient iterative solvers available today. When using them, it is not necessary to form the matrix explicitly, and they avoid any matrix-matrix multiplication, which is a costly function. Instead, a Krylov method's most costly operation is matrix-vector products. Another reason for their efficiency is that these algorithms work with vectors as opposed to an element-by-element method, such as Gauss-Seidel, making them efficient on modern pipelined CPU architectures.

A Krylov subspace is achieved by a power series: repeated multiplications of a matrix by a vector. For a typical Krylov method, the matrix is the coefficient matrix ($\mathbf{A}$) and the vector is the RHS of Equation (2.52) ($\vec{b}$). The i^{th} Krylov subspace is defined as

$$\mathcal{K}_i(\mathbf{A},\vec{b}) = \text{span}\left\{\vec{b}, \mathbf{A}\vec{b}, \mathbf{A}^2\vec{b}, ..., \mathbf{A}^{i-1}\vec{b}\right\}. \tag{2.57}$$

A Krylov based numerical solver adds one more dimension to its vector-space with each iteration. The first iteration has a very small vector space, $\mathcal{K}_1 = \{\vec{b}\}$, and the second has just two vectors to span the subspace, and so on. The matrix $\mathbf{A}$ need never be multiplied with itself, since the multiplications build with each iteration (i.e., $\vec{v}_0 = \vec{b}$, $\vec{v}_1 = \mathbf{A}\vec{v}_0$, $\vec{v}_2 = \mathbf{A}\vec{v}_1$, $\ldots$). Although the vector-space of a Krylov method grows with each power of iteration, the vectors are not orthogonal, sometimes having a large projection onto each other. Therefore, Krylov methods implement an orthogonalization algorithm, such as Gram-Schmidt, to form an orthonormal basis for the vector-space. Let the set $\{\vec{p}_0, \vec{p}_1, \ldots, \vec{p}_{i-1}\}$ be an orthonormal basis to the Krylov subspace $\mathcal{K}_i$. This basis spans the Krylov subspace as well,

$$\begin{aligned}\mathcal{K}_i(\mathbf{A},\vec{b}) &= \text{span}\left\{\vec{b}, \mathbf{A}\vec{b}, \mathbf{A}^2\vec{b}, ..., \mathbf{A}^{i-1}\vec{b}\right\} \\ &= \text{span}\{\vec{p}_0, \vec{p}_1, \ldots, \vec{p}_{i-1}\},\end{aligned}$$

with the property that $\vec{p}_j \cdot \vec{p}_k = \delta_{j,k}$ where $\delta_{j,k}$ is the Kronecker delta function.

A Krylov method for an $n \times n$ system theoretically converges in at most n steps, while in practice convergence can occur in many fewer steps. As with most iterative methods, convergence can be accelerated with preconditioning or a good guess of the solution. Krylov methods can be used to solve the matrix equation ($\mathbf{A}\vec{x} = \vec{b}$) or the eigenvalue equation ($\mathbf{A}\vec{x} = \lambda\vec{x}$). Common Krylov methods for eigenvalue equations are Lanczos for Hermitian matrices and Arnoldi for non-Hermitian matrices [207]. Krylov methods for solving the matrix equation for non-Hermitian matrices include GMRES, while a Hermitian matrix can be solved with the conjugate gradient method [207]. Some Krylov methods, like GMRES [188], require the storage of an additional vector for each iteration, which makes the solver become more costly with increasing accuracy. The conjugate gradient method, however, has a constant storage footprint, regardless of the number of iterations.

2.5.3.1 Conjugate Gradient

Introduced by Hestenes and Stiefel in 1952 [92], conjugate gradient (CG) is a Krylov subspace method for SPD matrices. The SPD requirement is restrictive and prevents CG from being applied more generally. However, CG is an excellent choice whenever applicable [49]. Properties of CG include rapid convergence, numerical stability, dependence of each step on the original data (i.e., the matrix $\mathbf{A}$), and that at any step, one can restart the algorithm using the previous solution as an initial guess [92]. This section elucidates the essence of how CG works.

Mathematical definitions. Before we proceed further, we introduce some mathematical notation. While a basis does not have to be orthogonal, no basis vector can be zero or parallel to another basis vector. Let the set $\{\vec{p_i}\}_{i=1}^{n}$ form a basis for a vector space of n vectors, with each vector having n elements. Any arbitrary vector $\vec{u}$ can be written as a linear combination of the basis vectors, using a set of scalars, $\{\alpha_i\}_{i=1}^{n}$, as shown,

$$\vec{u} = \alpha_1\vec{p}_1 + \alpha_2\vec{p}_2 + \cdots + \alpha_n\vec{p}_n. \tag{2.58}$$

The magnitude of a vector is defined as

$$\|\vec{u}\| = \sqrt{u_1^2 + u_2^2 + u_3^2 + \cdots + u_n^2}. \tag{2.59}$$

Let us define an inner product between two vectors of length n as

$$\langle\vec{u}, \vec{v}\rangle = \vec{u} \cdot \vec{v} = \vec{u}^T\vec{v} = u_1v_1 + u_2v_2 + \cdots u_nv_n. \tag{2.60}$$

Two nonzero vectors are orthogonal or normal if their inner product is zero, $\langle \vec{u}, \vec{v} \rangle = 0$. The inner product of a vector with itself gives its magnitude squared,

$$\langle \vec{u}, \vec{u} \rangle = \|\vec{u}\|^2 . \tag{2.61}$$

Equation (2.60) is the standard inner product, but an inner product can be defined with respect to other spaces. For the CG method, we introduce the A-norm, the inner product space for the real matrix $\mathbf{A}$. Let us define this as

$$\langle \vec{u}, \vec{v} \rangle_{\mathbf{A}} = \langle \vec{u}, \mathbf{A}\vec{v} \rangle = \langle \mathbf{A}^T \vec{u}, \vec{v} \rangle = \langle \mathbf{A}\vec{u}, \vec{v} \rangle, \tag{2.62}$$

where a transpose has already been defined, and the last equality is true because $\mathbf{A}$ is symmetric (SPD). Two nonzero vectors that have a zero A-norm ($\langle \vec{u}, \vec{v} \rangle_{\mathbf{A}} = 0$) are called A-orthogonal or conjugate. Indeed, this is what ‘conjugate’ refers to in CG.

Algorithm and properties. With these definitions in place, we now present the CG algorithm. The following algorithm solves Equation (2.52), $\mathbf{A}\vec{x} = \vec{b}$, where $\vec{p}_i$ are basis vectors or search directions, $\vec{r}_i$ are residual vectors, and α_i and β_i are a set of scalars. A residual vector is defined as $\vec{r}_i = \vec{b} - \mathbf{A}\vec{x}_i$, where $\vec{x}_i$ is the solution estimate for the current iteration.

```
CG Algorithm
    Initialize
        $\vec{x}_0 = 0$ ($x_0$ can be defined on input, default is zero)
        $\vec{p}_0 = \vec{r}_0 = \vec{b} - \mathbf{A}\vec{x}_0$
    Loop
        $\alpha_i = \dfrac{\langle \vec{p}_i, \vec{r}_i \rangle}{\langle \vec{p}_i, \vec{p}_i \rangle_{\mathbf{A}}}$
        $\vec{x}_{i+1} = \vec{x}_i + \alpha_i \vec{p}_i$
        $\vec{r}_{i+1} = \vec{r}_i - \alpha_i \mathbf{A}\vec{p}_i$
        $\beta_i = -\dfrac{\langle \vec{r}_{i+1}, \vec{p}_i \rangle_{\mathbf{A}}}{\langle \vec{p}_i, \vec{p}_i \rangle_{\mathbf{A}}}$
        $\vec{p}_{i+1} = \vec{r}_{i+1} + \beta_i \vec{p}_i$
```

The initial guess can be random, zero, or a user input. An initial guess of a zero vector is assumed in this section for simplicity. Each iteration of a loop involves one matrix-vector multiplication ($\mathbf{A}\vec{p}_i$), which is the most computationally intensive part of the algorithm. As is written above, each vector ($\vec{x}_i$, $\vec{p}_i$, and $\vec{r}_i$) as well as each scalar (α_i and β_i) must be stored. However, in practice, one needs to store one vector each for the solution, residual, and search-direction, as well as a single value for the scalar α and a single value for β (i.e., not one for each i). The only additional stored values needed are the previous and current residual magnitudes. For the highest performance, the matrix $\mathbf{A}$ does not need to be formed, only a procedure to store the result of multiplying the sparse matrix $\mathbf{A}$ with the current step's search direction, $\vec{p}_i$, is needed.

A large number of properties and relations can be written down for this algorithm, none more important than the following two. CG produces a set of orthogonal residual vectors and a set of mutually conjugate basis vectors [207], shown as follows:

$$\langle \vec{r}_i, \vec{r}_j \rangle = 0 \qquad j \neq i \tag{2.63a}$$

$$\langle \vec{p}_i, \mathbf{A}\vec{p}_j \rangle = 0 \qquad j \neq i. \tag{2.63b}$$

The implication is that each iteration searches for the solution in a new or orthogonal direction in the Krylov space. In other words, the search process chooses the most efficient direction to search. Having already found the closest solution for the previous Krylov subspace ($\mathcal{K}_{i-1}$), the algorithm looks in the new direction afforded by the current Krylov subspace ($\mathcal{K}_i$), which is conjugate to the previous subspace. There are two important forms for both α_i and β_i:

$$\alpha_i = \frac{\|r_i\|^2}{\langle \vec{p}_i, \mathbf{A}\vec{p}_i \rangle} = \frac{\langle \vec{p}_i, \vec{r}_i \rangle}{\langle \vec{p}_i, \mathbf{A}\vec{p}_i \rangle} \tag{2.64a}$$

$$\beta_i = \frac{\|r_{i+1}\|^2}{\|r_i\|^2} = -\frac{\langle \vec{r}_{i+1}, \mathbf{A}\vec{p}_i \rangle}{\langle \vec{p}_i, \mathbf{A}\vec{p}_i \rangle}. \tag{2.64b}$$

Initial studies by Hestenes and Stiefel [92] show that the expressions furthest to the right obtain better results with respect to round-off behavior, especially in poorly conditioned systems. Equation (2.64b) comes from substituting in $\vec{r}_{i+1} = \vec{p}_{i+1} - \beta_i \vec{p}_i$ with both Equation (2.63) and Equation (2.64a), and Equation (2.64a) is found in the proof of Equation (2.63a).

While the 'conjugate' part of 'conjugate gradient' has already been explained, it is not clear what 'gradient' is referring to. This comes from viewing the problem as one of optimization. The function being optimized, sometimes called the natural energy function [203], is defined as

$$\begin{aligned} \phi(\vec{x}) &= \tfrac{1}{2}\langle \vec{x}, \vec{x}\rangle_{\mathbf{A}} - \langle \vec{x}, \vec{b}\rangle \\ &= \tfrac{1}{2}\langle \vec{x}, \mathbf{A}\vec{x}\rangle - \langle \vec{x}, \vec{b}\rangle \\ &= \tfrac{1}{2}\vec{x}^T\mathbf{A}\vec{x} - \vec{x}^T\vec{b}. \end{aligned} \tag{2.65}$$

Taking the gradient of this function is as straightforward as if we assume we had a scalar function, $\frac{d}{dx}(\frac{1}{2}Ax^2 - bx) = Ax - b$ [203]. Hence, the gradient of the natural energy function is defined as $\mathbf{A}\vec{x} - \vec{b}$. Fundamental calculus states that the maximum or minimum of a function occurs where the derivative is zero. Because $\mathbf{A}$ is SPD (positive-definite), the energy function must be a minimum, not a maximum [206]. Therefore, the minimum of the energy function is when $\vec{x} = \vec{u}$, where we define $\vec{u}$ as the exact solution. The function $\phi(\vec{x})$ is called the natural energy because minimizing the energy equation solves $\mathbf{A}\vec{x} = \vec{b}$. Finding $\vec{x} = \vec{u}$ solves the matrix equation as well as optimizes the natural energy function. For this reason, the CG method can be viewed as a linear solver and/or as an optimizer. It can be shown that the choice for α_i ensures that the optimal step length is chosen along each search direction, and when the function $\phi(\vec{x})$ is minimized over $\vec{x}$, it is minimized over the entire vector space [207].

Expounding the algorithm. All necessary formulas, expressions, and definitions needed for the CG method have been introduced. In the following, we restate the loop part of the algorithm and then explicate what each step is doing.

$$\alpha_i = \frac{\langle \vec{p}_i, \vec{r}_i\rangle}{\langle \vec{p}_i, \mathbf{A}\vec{p}_i\rangle} \tag{2.66a}$$

$$\vec{x}_{i+1} = \vec{x}_i + \alpha_i\vec{p}_i \tag{2.66b}$$

$$\vec{r}_{i+1} = \vec{r}_i - \alpha_i\mathbf{A}\vec{p}_i \tag{2.66c}$$

$$\beta_i = -\frac{\langle \vec{r}_{i+1}, \mathbf{A}\vec{p}_i\rangle}{\langle \vec{p}_i, \mathbf{A}\vec{p}_i\rangle} \tag{2.66d}$$

$$\vec{p}_{i+1} = \vec{r}_{i+1} + \beta_i\vec{p}_i \tag{2.66e}$$

In Equation (2.66a), α_i is a scalar that weights how far along the $\vec{p}_i$ direction that $\vec{x}_{i+1}$ must be adjusted for optimal stability and convergence. It is the ratio of the projection of $\vec{p}_i$ onto $\vec{r}_i$ with the A-norm of $\vec{p}_i$.

The approximate solution, Equation (2.66b), is updated by adding the next weighted basis vector, $\vec{p}_i$, which was calculated in the previous iteration. This step is straightforward since $\vec{x}$ is expressed as a weighted sum of the basis for the current Krylov subspace, as in Equation (2.58). Another way to view Equation (2.66b) is that the search-direction $\vec{p}_i$ is a scalar multiple of the gradient of $\vec{x}_i$ ($\alpha_i \vec{p}_i = \vec{x}_{i+1} - \vec{x}_i$). This means that $\vec{p}_i$ goes in the direction of the gradient of the evolution of approximate solution, $\Delta \vec{x}_i = \vec{x}_{i+1} - \vec{x}_i$.

The residual, Equation (2.66c), is updated by subtracting the new, weighted basis vector after multiplication by $\mathbf{A}$. Since the solution update is $\alpha_i \vec{p}_i$ and the residual is defined as $\vec{r}_i = \vec{b} - \mathbf{A}\vec{x}_i$, the residual update is simply $-\alpha_i \mathbf{A}\vec{p}_i$. This can also be looked at as the step that enforces the A-orthogonality because this is where the contribution from the $\mathbf{A}\vec{p}_i$ term is removed, and the subsequent search-directions are equated to the residual (Equation (2.66e)). A-orthogonality ensures that each new residual is conjugate to the current Krylov subspace, while Equation (2.63a) ensures that each new residual is orthogonal to all previous residuals.

Equation (2.66d) is a scalar weight chosen to help select the next search-direction. The specific choice of β_i allows for the optimal solution of the energy function for each step, although different choices could be made for different CG-like methods.

The last step of the algorithm, Equation (2.66e), is the most subtle, however certain substitutions help illuminate what is happening. Substituting $\vec{p}_0 = \vec{r}_0$ and Equation (2.64b) into Equation (2.66e), the following can be obtained,

$$\begin{aligned}
\vec{p}_1 &= \vec{r}_1 + \frac{\|\vec{r}_1\|^2}{\|\vec{r}_0\|^2}\vec{r}_0 \\
\vec{p}_2 &= \vec{r}_2 + \frac{\|\vec{r}_2\|^2}{\|\vec{r}_1\|^2}\left(\vec{r}_1 + \frac{\|\vec{r}_1\|^2}{\|\vec{r}_0\|^2}\vec{r}_0\right) = \vec{r}_2 + \frac{\|\vec{r}_2\|^2}{\|\vec{r}_1\|^2}\vec{r}_1 + \frac{\|\vec{r}_2\|^2}{\|\vec{r}_0\|^2}\vec{r}_0 \\
\vec{p}_3 &= \|\vec{r}_3\|^2\left[\frac{\vec{r}_3}{\|\vec{r}_3\|^2} + \frac{\vec{r}_2}{\|\vec{r}_2\|^2} + \frac{\vec{r}_1}{\|\vec{r}_1\|^2} + \frac{\vec{r}_0}{\|\vec{r}_0\|^2}\right] \\
&\vdots \\
\vec{p}_i &= \|\vec{r}_i\|^2 \sum_{j=0}^{i} \frac{\vec{r}_j}{\|\vec{r}_j\|^2} = \|\vec{r}_i\|^2 \sum_{j=0}^{i} \hat{r}_j,
\end{aligned}$$

where $\hat{r}_j$ is the normalized residual. Since the residuals are mutually orthogonal, Equation (2.63a), the i^{th} search direction ($\vec{p}_i$) is formed from an orthonormal basis. The residuals themselves account for the A-orthogonality, Equation (2.66c). Therefore, each new residual $\vec{r}_{i+1}$ is conjugate to the previous Krylov space, $\mathcal{K}_i = \text{span}\{\vec{r}_0, \vec{r}_1, ..., \vec{r}_i\}$. Consequently, Equation (2.66e) creates a vector that is mutually conjugate with all previous search directions as well as A-orthogonal to the previous

Krylov space, making it a basis vector as well.

Each step of the CG algorithm has been explained. Starting from any initial guess, the algorithm finds the optimal search direction in a manner similar to the method of steepest decent, but in the A-norm. Each residual vector is orthogonal to all previous residuals, and each search direction is conjugate to the previous Krylov subspace. Choosing a stopping criterion for CG is not straightforward. The most obvious criterion, requiring that the Euclidean norm of the residual ($\|\vec{r}\|$) is smaller than some value, is often sufficient. However, the residual is not the best measure for accuracy because the residual is not the function being minimized [92]. Effective choices of stopping criterion for the CG are an active area of research [9, 10, 11]. While a good initial guess will quickly converge on the solution, the CG method can be accelerated with a preconditioner. Multigrid preconditioners are often used with CG in practice [1, 27, 114, 174, 206]. The problems considered in this dissertation converge in a small enough number of iterations that we do not find preconditioners to be necessary. However, many simplifications are made (e.g., no source terms, no interface dynamics, small domains), and preconditioners would be necessary in the more complicated cases.

There are many CG variants and related methods, such as bi-conjugate gradient, but the heart of the method remains as described in this section. If the coefficient matrix is not SPD, one could make the system SPD by multiplying by the transpose, giving a matrix equation of

$$\mathbf{A}^T\mathbf{A}\vec{x} = \mathbf{A}^T\vec{b}.$$

However, squaring the matrix in this manner also squares the conditions number [207], which makes a poorly conditioned system worse. Additionally, the extra steps of multiplying by the transpose (a matrix-matrix multiplication) may prove too costly, in which case a matrix that is not SPD can be solved by GMRES.

2.6 Summary

This marks the end of the Background and History chapter. We have introduced diffusion and fluid dynamics, as well as given examples of these phenomena in applied problems. We then discussed radiation transport, including the RTE and some common approximations of the RTE, including the diffusion approximation. We then discussed our motivation for highly discontinuous diffusivity problems, notably the diffusion approximation of radiation transport for the CRASH project. We were able to mathematically relate the diffusive radiation transport with heat conduction, enabling us to focus on a simpler case, while retaining applications to radiation transport and multiphase flow. We gave a detailed literature review of previous work involving either highly discontinuous coefficients or mixed cells. We concluded with a discussion of iterative methods for solving differential equations, with a particular emphasis on the conjugate gradient method.

A single temperature and a multi-temperature mixed-cell model for highly discontinuous multiphase, diffusive systems, solved with the CG method, are presented in Chapters III-IV. Numerical results for these models are presented in Chapters V-VIII.

CHAPTER III

Single-Temperature Methods

This chapter presents three single-temperature models to treat mixed cells for a 2D heat diffusion problem, as well as a detailed derivation of the Support-Operators Method (SOM) [95, 99, 132, 151, 152, 185, 198] applied to the problem. We note that it is customary in SOM to use 3D terminology even in 2D, so a volume has units of length squared, an area has units of length, etc. Section 3.1 specifies the single-temperature mixed-cell models. The first is a standard method (S1), while the second (S2) and third (S3) represent new work. All three models are implemented using SOM, which is detailed in Section 3.2. The chapter concludes with a summary in Section 3.3.

The S1 method homogenizes the diffusivities in a mixed cell by taking the harmonic mean of the diffusivity of each phase, weighted by volume fraction. S2 is based on a rotated tensor diffusivity, where a subcell interface is modeled using the tensor properties of the diffusivity. S3 is a 'split-zone' method, where the shape matrix allows each corner of a cell to have a unique diffusivity and volume fraction.

Although the S1 and S2 models may be solved with a variety of computational methods, we chose SOM for their numerical implementation. SOM is a fitting choice because it remains second-order accurate even with strong jumps in the scalar or tensor diffusivities (for pure cells). Consequently, the code based on SOM could implement all three models, giving a consistent framework to compare the methods.

SOM is general enough to treat non-orthogonal meshes, including those having more than four faces per cell in 2D (and more than six faces in 3D). However, our implementation of SOM is restricted to orthogonal grids. While this is a more complicated method than is necessary for the S1 model, its ability to treat the S2 and S3 models justify its use. Moreover, SOM is well suited for problems with discontinuous diffusivities. SOM can treat hanging nodes generated from adaptive mesh refinement (AMR) (see Figure 2.1(b) for an example) in 2D or 3D for x-y or r-z coordinates, and it is a natural choice to couple with a Lagrangian hydrodynamics code, making it a viable method for coupled diffusion-hydrodynamics problems. The primary disadvantage of using SOM compared to standard finite difference methods (FDM) is the larger number of unknowns, which may increase computational cost; SOM inputs and outputs cell-centered values, but computations are performed at the more numerous cell-faces. However, this is offset by increases in accuracy and computational efficiency per degree of freedom.

3.1 Three Single-Temperature Models

3.1.1 S1 - Harmonic-Mean Diffusivity

The harmonic-mean method (S1) is included as a reference model in the comparison, along with the new models presented in this dissertation. The harmonic-mean model has been similarly considered as a reference for mixed-cell analysis in the past [86, 216]. Additionally, it is a sensible comparison because mixed cells are treated in this manner in most methods that do not explicitly treat subcell interfaces. Justification for the use of the harmonic mean for a mean diffusivity is described by Pantankar [166], and is now common practice, although there is evidence that the harmonic mean is not always the best option in nonlinear heat conduction [105]. Furthermore, when two adjacent pure cells on an orthogonal grid have different diffusivities, SOM results are identical to the harmonic-mean model, and therefore the effective diffusivity set by the harmonic mean can be considered a reasonable approximation for diffusion in and near mixed cells.

3.1.2 S2 - Rotated Tensor Diffusivity

While the S1 model converges to SOM when there are no mixed cells, when mixed cells are present, it loses the anisotropy in the effective diffusivity that should be present. The rotated tensor diffusivity method (S2) retains the anisotropy of the diffusivity by exploiting the tensor form of the diffusion coefficient. In this model, diffusion near an interface between two different homogeneous phases, each with scalar (isotropic) diffusivity, is modeled with an anisotropic diffusion tensor, with flux tangent and normal to the interface given by the arithmetic and harmonic means, respectively. We define the arithmetic mean (D_A) and harmonic mean (D_H) of two diffusivities as

$$D_A = D_{\parallel} = f_1 D_1 + f_2 D_2 \tag{3.1a}$$

$$D_H = D_{\perp} = \left[\frac{f_1}{D_1} + \frac{f_2}{D_2}\right]^{-1}, \tag{3.1b}$$

where f_i is the volume fraction of phase i ($f_1 + f_2 = 1$), and D_i is the diffusivity of phase i. The arithmetic mean of a large and a small number yields a value similar to the large number; on the other hand, the harmonic mean yields a value similar to the small number.

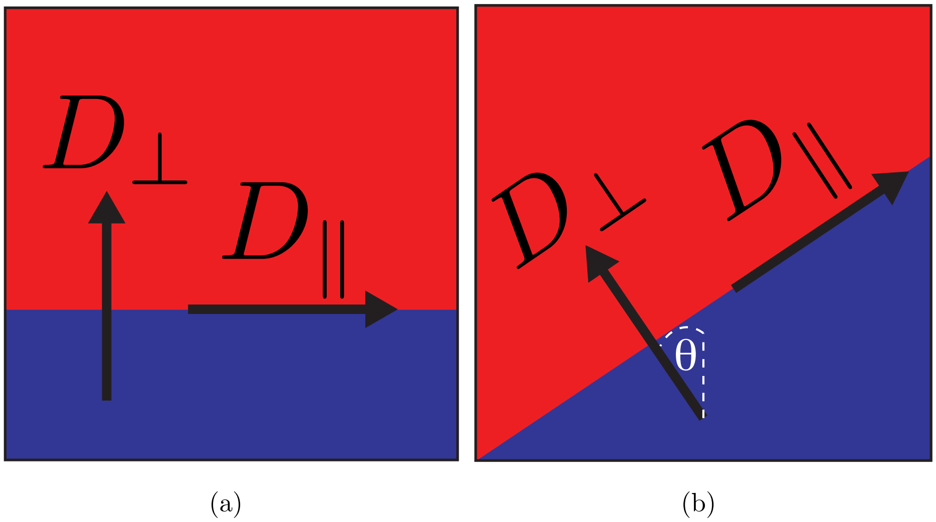


Figure 3.1: S2 model (rotated tensor diffusivity) for mixed cells in the single-temperature method. This model specifies the diffusivity tensor in terms normal and tangent to the interface for a mixed cell (33% volume fraction shown), with phase 1 shown in blue and phase 2 shown in red. The angle θ is defined as the angle between the interface normal and the y-axis. The examples shown have (a) θ=0° and (b) θ=33.7°. Parallel and perpendicular are defined with respect to the interface.

Figure 3.1 shows two examples of mixed cells configurations to help explain the S2 model, with phase 1 shown in blue and phase 2 shown in red. Figure 3.1(a) shows a mixed cell where the interface is aligned with the x-axis. Flux in the x-direction (parallel to the interface) is dominated by the larger diffusivity, in which case the rule of mixture for diffusivity gives the arithmetic mean. Flux in the y-direction (perpendicular to the interface) is limited by the smaller diffusivity since the flux of species must pass through both phases, in which case the rule of mixture gives the harmonic mean. Thus, for the mixed cell in Figure 3.1(a), the diffusivity tensor is specified by

$$\mathbf{D} = \begin{bmatrix} D_{\|} & 0 \\ 0 & D_{\perp} \end{bmatrix}. \tag{3.2}$$

The key concept of the S2 model is evident when considering Figure 3.1(b). In this case, applying the 2D rotation matrix specifies the diffusion tensor,

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{3.3}$$

to the diffusion tensor in Equation (3.2), which gives

$$\mathbf{D}_{S2} = \mathbf{R}\mathbf{D}\mathbf{R}' \tag{3.4a}$$

$$= \begin{bmatrix} D_{\|}\cos^2\theta + D_{\perp}\sin^2\theta & \frac{1}{2}(D_{\|} - D_{\perp})\sin 2\theta \\ \frac{1}{2}(D_{\|} - D_{\perp})\sin 2\theta & D_{\|}\sin^2\theta + D_{\perp}\cos^2\theta \end{bmatrix}, \tag{3.4b}$$

where θ is the angle between the interface normal and the y-axis. The geometric information required for this model is the volume fraction of each phase as well as the vector normal to the interface. In a Volume of Fluid (VoF) method, the volume fractions are given naturally, but the interface must be reconstructed (typically with a linear approximation) in order to approximate the interface normal. In level set methods (LSM), the interface normal is given naturally as the direction of the gradient of the level-set function, and the volume fractions can be calculated. Although the author is not aware of any other application of the rotated diffusivity tensor to a mixed-cell model, the rotated diffusivity has been used for test problems [56, 99, 197]. Berger and LeVeque [20, 21] present a similar idea, where the coordinate system is rotated via an isoparametric coordinate transform in order to describe the flux parallel and perpendicular to the interface.

3.1.3 S3 - Split-Zone Diffusivity

The split-zone method (S3) effectively gives one 'free' mesh refinement in a mixed cell with little additional computational cost when using SOM. However, SOM itself is more costly than other methods, and thus the 'free' refinement is in comparison to SOM without split zones, not to the standard FDM discretization of the diffusion equation. Every element or cell in SOM has a diffusivity-dependent shape matrix

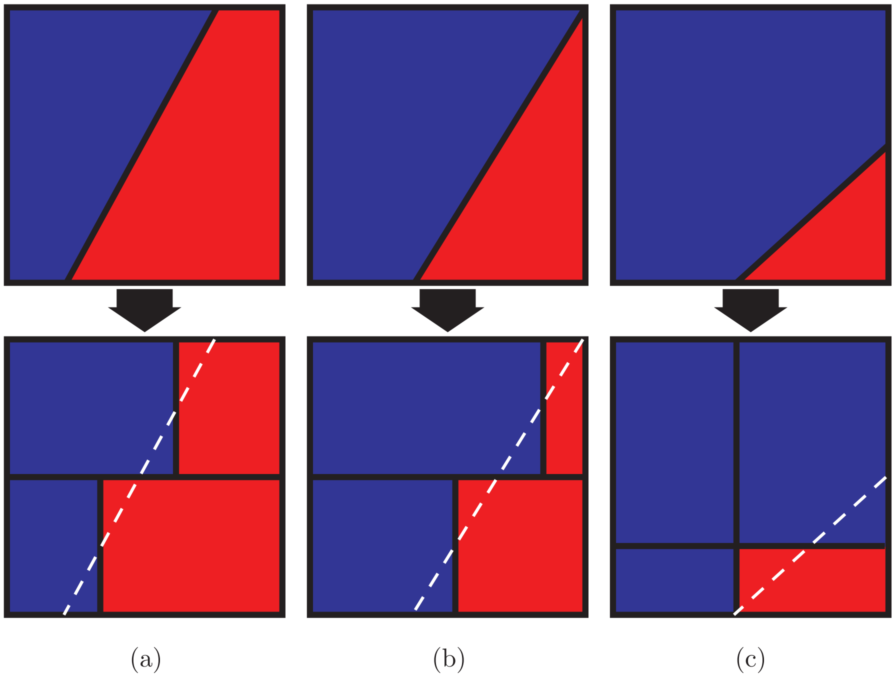


Figure 3.2: Three of the four classes of configurations that arise in the S3 (split-zone) model for mixed cells in the single-temperature method. The S3 model divides a mixed cell into four virtual subcells associated with each corner of the cell such that the volume of each phase is conserved. Note that only the phase and volume of each virtual subcell determine the diffusivity assigned at the associated cell corner in the S3 model; the orientation is for visualization purposes only. The top row shows the mixed cell and the bottom row shows the corresponding virtual subcells. The classes are distinguished by how the interface crosses the cell: the interface (a) crosses opposite faces (case (i)), (b) crosses one corner (case (ii)), and (c) crosses adjacent faces (case (iii)). The majority phase, phase 1, is rendered in blue, while the minority phase, phase 2, is rendered in red.

that contains information specific to that cell only (diffusivities, volume, angles of corners). This shape matrix considers the region associated with each corner of a cell separately, a feature that can be exploited to specify a different diffusivity and volume in the virtual subcell associated with a corner. The sum of the total volume and phase volumes of all virtual subcells within a cell must equal those of the cell. For a pure cell, the diffusivities associated with all four corners are equal to the pure cell value, and the corresponding subcell volumes are simply $\Delta x \Delta y/4$. The split-zone method takes advantage of an intrinsic capability of SOM, while not altering any of the properties or procedures of SOM. Although Morel, Hall, and Shashkov [151] mention that each corner can have a different diffusivity and that choice of volume

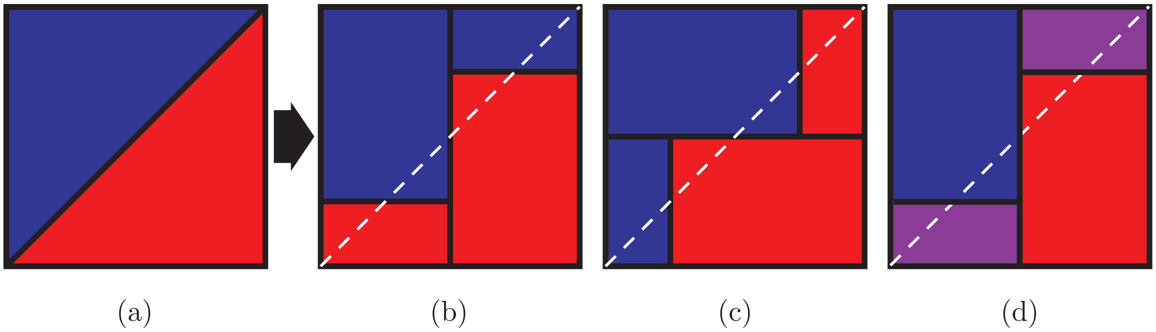


Figure 3.3: The fourth class of configurations for the S3 (split-zone) model for mixed cells in the single-temperature method (case (iv)). This case is when the interface crosses two corners, as shown in (a). Note that only the phase and volume of each virtual subcell determine the diffusivity assigned at the associated cell corner in the S3 model; the orientation is for visualization purposes only. Figures (b) and (c) show two ways to split this cell while preserving phase volumes and minimizing the ambiguous regions. Both options result in a large and small rectangle (3/8 and 1/8 of the cell volume) of each phase. The NW and SE virtual subcells clearly belong to one of the two phases, defining the diffusivities at these corners, but the SW and NE corners have two possible assignments of the phases. Case (d) resolves the ambiguity of phase assignments in the NE and SW virtual subcells via a harmonic-averaged diffusivity (shown as purple).

weights is flexible, to this author's knowledge, this capability of SOM has not been previously applied to a mixed-cell model.

Figures 3.2 and 3.3 show how the split-zone method approximates a mixed cell containing a linear interface as four virtual subcells. The four classes of interfacial configurations are: (i) interface crosses opposite sides of cell, (ii) interface crosses one corner, (iii) interface crosses adjacent sides of cell, and (iv) interface crosses two corners. While we will only discuss the orientations shown in these figures, each of these cases has between two and eight degenerate versions, found through rotation and/or reflection. Note that case (iv) results in two mixed and two pure virtual subcells, whereas all other cases result in four pure virtual subcells.

The areas of each cell face are not affected by the split-zone method; the surface area of each phase is found by interface reconstruction. The contribution of the virtual subcells is defined by the shape matrix, which is derived in Appendix B. Surface terms do not appear with the shape matrix, only the volume, phase, and corner angles. The purpose of the shape matrix is to account for effects from non-orthogonal corners, thus an orthogonal grid would give equal terms in each subcell. As mentioned earlier, the

S1 and S2 models are also solved using SOM; these cases use identical diffusivities and volumes ($\Delta x \Delta y/4$) in all four subcells for the shape matrix in each cell. For mixed cells only, the S3 model defines a different diffusivity and volume in each subcell for the shape matrix (pure cells use the same values in the four subcells).

The particular implementation of specifying the volume of each virtual subcell is given in Appendix A. Our implementation is based on the VoF method to track volume fractions, which is then used to reconstruct interfaces using a linear approximation, but other methods such as LSM can be applied. There are a variety of means to calculate the subcell volumes, and one method may prove more accurate than another. We have attempted to reduce the mixed virtual cell volumes as discussed in Appendix A. Finding an optimal, most accurate approach represents an important next step to the development of the S3 model, whereas this thesis provides a 'proof of concept' for the S3 model.

We summarize the split-zone formulae for the orientations shown in Figures 3.2 and 3.3, which are developed in Appendix A. Alternate orientations have analogous formulae. Some of the cases (when interface crosses opposite or adjacent faces) are specified in terms of surface fractions, the percentage of a face occupied by a particular phase. The surface area of the top face, for example, is specified in terms of the surface fraction as $A_T = a_T \Delta x$. Note that the following equations are written in terms of the majority phase, i.e., $f \geq 1/2$, and in all cases, $V_{LT} + V_{LB} + V_{RT} + V_{RB} = \Delta x \Delta y$.

Crosses opposite sides. The diffusivities are $D_{LT} = D_{LB} = D_1$ and $D_{RT} = D_{RB} = D_2$, and the subcell volumes are

$$V_{LT} = \frac{1}{4}\frac{3a_T + a_B}{a_T + a_B} f \Delta x \Delta y \tag{3.5a}$$

$$V_{LB} = \frac{1}{4}\frac{a_T + 3a_B}{a_T + a_B} f \Delta x \Delta y \tag{3.5b}$$

$$V_{RT} = \frac{1}{4}\frac{3(1-a_T) + (1-a_B)}{(1-a_T) + (1-a_B)}(1-f) \Delta x \Delta y \tag{3.5c}$$

$$V_{RB} = \frac{1}{4}\frac{(1-a_T) + 3(1-a_B)}{(1-a_T) + (1-a_B)}(1-f) \Delta x \Delta y. \tag{3.5d}$$

Crosses one corner. The diffusivities are $D_{LT} = D_{LB} = D_1$ and $D_{RT} = D_{RB} = D_2$, and the subcell volumes are

$$V_{RT} = \frac{1}{4}(1-f)\Delta x \Delta y \tag{3.6a}$$

$$V_{RB} = \frac{3}{4}(1-f)\Delta x \Delta y \tag{3.6b}$$

$$V_{LT} = \frac{1}{4}(1+f)\Delta x \Delta y \tag{3.6c}$$

$$V_{LB} = \frac{1}{4}(3f-1)\Delta x \Delta y. \tag{3.6d}$$

Crosses adjacent sides. The diffusivities are $D_{LT} = D_{LB} = D_{RT} = D_1$ and $D_{RB} = D_2$, and the subcell volumes are

$$V_{RB} = (1-f)\Delta x \Delta y \tag{3.7a}$$

$$V_{LB} = p(1-q)\Delta x \Delta y \tag{3.7b}$$

$$V_{LT} = (1-p)q\Delta x \Delta y \tag{3.7c}$$

$$V_{LB} = (1-p)(1-q)\Delta x \Delta y, \tag{3.7d}$$

where

$$p = \sqrt{\frac{(1-a_R)(1-f)}{(1-a_B)}} \tag{3.8a}$$

$$q = \sqrt{\frac{(1-a_B)(1-f)}{(1-a_R)}}. \tag{3.8b}$$

Crosses two corners. The diffusivities are $D_{LT} = D_1$, $D_{RB} = D_2$, and $D_{LB} = D_{RT} = \frac{2D_1D_2}{D_1+D_2}$. The subcell volumes are

$$V_{RB} = \frac{3}{8}\Delta x \Delta y \tag{3.9a}$$

$$V_{LB} = \frac{1}{8}\Delta x \Delta y \tag{3.9b}$$

$$V_{LT} = \frac{3}{8}\Delta x \Delta y \tag{3.9c}$$

$$V_{LB} = \frac{1}{8}\Delta x \Delta y. \tag{3.9d}$$

3.2 Support-Operators Method

SOM is not a specific computation method but is a technique to build a discretization of partial differential equation (PDE) using mimetic finite difference methods (MFD). Note that 'mimetic' is not equivalent to 'memetic,' which is an unrelated algorithm. MFD retain the vector-calculus mathematical properties of the continuum equation better than FDM. In standard FDM, the governing PDE is discretized and solved. MFD, however, first discretize vector-calculus operators while maintaining an integral identity, then build the differential equation using these operators, and subsequently solve the system. This extra step of creating invariant operators has numerous advantages, such as preserving certain continuum properties [151, 198] which will be discussed shortly.

Although we are technically applying a local SOM to discretize the diffusion equation in order to create a mimetic algorithm, we shall refer to the algorithm as simply "SOM," even though SOM is more general. The implementation of SOM used in this thesis is to solve the time-dependent, linear diffusion equation on a uniform, rectangular mesh. SOM could also model time-independent or nonlinear diffusion [151]. Additionally, SOM can be applied to many PDEs, such as Maxwell's equations involving divergences and curls of the electric and magnetic fields [198]. Furthermore, one of most the useful features of SOM is the flexibility for grid choice, such as handling 2D non-uniform or unstructured grids with quadrilateral, triangle, or polygonal cells (and hexahedral or polyhedral grids in 3D).

In this section, we present the details of the specific SOM implemented in our work. Detailed references to SOM and the more general MFD can be found in Section 2.4.3.3. However, fundamental references for SOM and those that were particularly valuable for this dissertation include References [95, 99, 132, 151, 152, 185, 198].

This SOM section is outlined as follows. We start in Section 3.2.1 by listing some important properties of this method. Section 3.2.2 shows the derivation of the operators in the continuum space. This section is useful when developing the mathematical comprehension and justification of the internal mechanisms of the method. Section 3.2.2.1 defines two specific inner-product spaces (Hilbert spaces) as well as an integral identity. This step leads to the two main operators of the method, divergence and gradient operators, which are adjoint to each other in the specified inner-product spaces. Section 3.2.2.2 shows how to construct the PDE in terms of these operators, expressing the system of equations in a single operator equation, $\mathsf{A}_C U = B$, where A_C is an SPD operator.

All subsequent steps occur in Section 3.2.3, which details the discrete derivation of the method. This section is most informative to those who are generating actual code. The discrete derivation begins in Section 3.2.3.1 with the discretization of the integral identity, which creates discrete analogs of the inner-product spaces. The key for this step is to maintain important properties of the continuum inner-product spaces. This step involves creating a discrete version of the divergence operator, which is perhaps the most complicated step of SOM. We do not discretize the modified gradient operator, but instead use the adjoint property to define it in terms of the divergence operator. As a result, the divergence operator is the *prime* operator and the modified gradient is the *derived* operator. The prime operator in SOM is the operator discretized directly from the integral identity; the derived operator is only specified in terms of the prime operator, due to the adjoint relationship of these two operators. The namesake ‘support operators’ refers to this prime and derived operator relationship.

The PDE is then discretized for a single cell in Section 3.2.3.2, which creates unique face-centered unknowns on all four faces of every cell, yielding five unknowns in each cell (one cell-centered and four face-centered). The case with both an orthogonal mesh and scalar diffusivity would allow elimination of the face-centered unknowns, resulting in one unknown per cell. However, since we use a full-tensor diffusivity, the equations for the face-centered values are coupled such that they cannot be eliminated. Instead, the cell-centered values are eliminated in Section 3.2.3.4, resulting in four unknowns per cell.

The final step, detailed in Section 3.2.3.5, is to form a global system. The global system connects adjacent mesh cells by enforcing continuity of temperature and flux at faces, which is equivalent to maintaining the adjoint relationship in the global system. This leads to a system of equations with a symmetric positive-definite (SPD) coefficient matrix. The system solves for a vector of face-centered unknowns, which are used to calculate the cell-centered value for each cell. In 2D, the number of face-centered unknowns is approximately double the number of unknowns for a cell-centered system. The only unknowns needed from the previous time-step are the cell-centered values, meaning the face-centered values are used only as an intermediate step and do not need to be stored.

3.2.1 Properties of SOM

There are many choices and options when constructing SOM, such as the coordinate system, dimension, and mesh type (orthogonal, quadrilaterals, polygons, unstructured, etc.). There are also two main technique options: the global SOM, where the system consists only of cell-centered unknowns, and the local SOM, where the system has both cell-centered and face-centered unknowns. A local stencil results in a sparse, banded matrix because cells are coupled only with neighbors that are spatially close rather than a global stencil where each cell is coupled to all others.

The following discussions apply to the specific formulation of SOM employed in our work. However, the consideration of non-orthogonal grids is retained so that it can be applied to a general quadrilateral mesh, even though this present work uses only orthogonal grids. Since SOM is a mimetic method, certain properties of the continuum problem (e.g., the gradient operator gives zero only when applied to a constant field) are maintained in its discrete form.

The most important properties associated with the specific SOM in the absence of mixed cells are [99, 152, 198]:

1. The finite difference scheme is conservative.
2. The finite difference scheme is second-order accurate for both smooth and non-smooth meshes, with or without phase discontinuities.
3. The system has both cell-centered and face-centered temperature unknowns.
4. The discrete diffusion operator, A, is SPD for Dirichlet, Neumann, and Robin boundary conditions.
5. A is sparse due to a local stencil at the cost of requiring face-centered unknowns.
6. A is the composition of the discrete divergence operator, D, and the discrete flux operator, G.
7. The discrete flux operator is adjoint to the discrete divergence operator, i.e., $\mathsf{G} = \mathsf{D}^*$.
8. D of a constant vector is zero for any grid.
9. The null space of G consists only of constant functions.
10. G is exact for linear functions if $\mathbf{D}$ is piecewise constant.
11. The normal component of flux, $F_\perp = \vec{F} \cdot \hat{n}$, is continuous across phase discontinuities.
12. The tangential component of flux, $F_\parallel$, may or may not be continuous along phase discontinuities.
13. Discontinuous isotropic (scalar) diffusivities (D) reduce to the harmonic average of the two adjacent cells for flux across the face between them for rectangular grids, as in the standard FDM.

14. All discrete operators reduce to a standard finite differencing for rectangular grids.

15. All discrete operators are linear.

Note that the diffusivity tensor, $\mathbf{D}$, the scalar diffusivity, D, and the divergence operator, D, are separate entities.

As mentioned, these properties assume that phase interfaces occur at the cell boundaries. Each cell is assumed to be homogeneous (i.e., there are no mixed cells), and phase properties can change abruptly only across cell boundaries. When mixed cells are present, some of these properties, such as second-order accuracy, change. These effects are explored in the numerical results (Chapters V-VIII), and, for now, we present SOM as intended, with only pure cells.

The operators are coordinate invariant (defined with quantities like volume, area, and angle), meaning the operators can be applied to any coordinate system merely by changing the formulas for these invariant quantities [95, 99, 198]. Accordingly, SOM can be applied to any coordinate system with much less effort than is typically required to change coordinate systems.

SOM creates a flux operator that is the adjoint to the divergence operator by defining an inner-product space weighted by the inverse of the diffusivity. Mixed FEM [7, 12, 33, 43, 178, 213] also define an inner-product space weighted by the inverse diffusivity. The inclusion of $\mathbf{D}^{-1}$ in the inner-product space is what allows SOM to handle discontinuous coefficients accurately [99, 198]. Physically, the diffusion tensor must be SPD, guaranteeing the existence of its inverse.

Morel *et al.* [152] discusses how the relationship between SOM and FEM is not just the similarly weighted inner product. For example, mixed FEM preserves the volume integral of the diffusion equation over each spatial cell, just like SOM. Mixed FEM are a class of FEM with a variety of specialized properties, such as being cell-centered and maintaining flux continuity with anisotropic, discontinuous diffusivity tensors [33, 178, 213]. The temperature unknowns at the cell centers in these FEM lead to an indefinite matrix that is difficult to solve [152]. This is similar to the global variant of SOM that has only cell-centered unknowns and yields a dense matrix. If FEM enforce continuity of temperature and the normal flux at the face-centers (rather than vertices), one can eliminate the continuity requirement from the trial-space and impose it with Lagrange multipliers. This results in an SPD system with both cell-centered and cell-edge temperature unknowns, very similar to SOM. In this case, the Lagrange multipliers can be shown to be equivalent to the face-centered unknowns in SOM [152]. However, SOM does not use basis functions in the formalism, and therefore it is fundamentally a FVM, despite some similarities to FEM discussed above [151, 152].

The primary disadvantage to SOM is the need for face-centered values. This leads to about twice as many unknowns as an equivalent cell-centered FDM. Additionally, each cell has a small (4×4) system of equations that must be inverted in order to form the global matrix. This additional computational overhead, as well as the increase in the number of unknowns, leads to SOM being outperformed by simpler methods (such as FDM) for problems with smooth meshes and continuous coefficients [185, 198]. On that account, there is little advantage to using SOM for problems with smooth (or rectangular) grids or smooth phase coefficients, as it offers comparable accuracy, but with additional cost. However, SOM excels for problems with highly non-uniform grids and/or coefficients that change rapidly in space, performing significantly better than traditional FVM and FDM [99, 198]. For our purposes, SOM is apt because the systems we consider have highly discontinuous coefficients, although we use a simple rectangular mesh.

The cost is offset in three ways. First, the small system of equations that must be inverted for each cell are independent from all other cells, enabling this computation to be performed in parallel with high efficiency. Second, the SPD nature of the coefficient matrix allows it to be solved with powerful and efficient linear matrix solvers (e.g., multigrid-preconditioned conjugate gradient), with only half of the coefficients needing to be stored [99, 198]. Third, because the method is second order, a smaller grid size is needed to give the same level of accuracy as common first-order methods, adding to the offset of the cost associated with larger number of unknowns.

3.2.2 Continuum Derivation of SOM

We here summarize the derivation of MFD applied to the diffusion equation by the means of local SOM by starting with the continuum problem. We mainly follow the steps described in [198]. We use a subscript C to distinguish continuum operators from discrete operators that appear in Section 3.2.3. The system of equations that describe diffusion in the volume V, bounded by surface ∂V, are

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot \vec{F} = Q, \qquad \text{on } V \tag{3.10a}$$

$$\vec{F} = -\mathbf{D}\vec{\nabla}U, \qquad \text{on } V \tag{3.10b}$$

$$\alpha U + \beta \mathbf{D}\vec{\nabla}U \cdot \hat{n} = \psi, \qquad \text{on } \partial V, \tag{3.10c}$$

where the third equation is the Robin boundary condition, $\hat{n}$ is the outward unit normal to the boundary surface, and α, β, and ψ are functions on the boundary. Additionally, U is a scalar temperature field, $\vec{F}$ is a vector flux, Q is a source or driving function, $\mathbf{D}$ is the diffusivity. Introducing the standard inner-product of two vectors as $\vec{J} \cdot \vec{F} = (\vec{J}, \vec{F})$, the boundary conditions, after substituting the expression for the flux, becomes

$$\alpha U - \beta(\vec{F}, \hat{n}) = \psi \qquad \text{on } \partial V. \tag{3.11}$$

Note the equivalent expressions for the inner product of two vectors,

$$\vec{J} \cdot \vec{F} = (\vec{J}, \vec{F}) = (\vec{J})^T \vec{F}, \tag{3.12}$$

where the superscript T indicates the transpose.

The derivation of SOM depends on the following two properties of inner-product spaces and adjoint operators [152, 198]: (i) the adjoint of an operator depends on the specific definition of the associated inner-product spaces (Hilbert space) of that operator; (ii) the adjoint is unique for a fixed inner product. Consequently, the divergence and modified gradient are adjoint to one another in a particular inner-product space. Moreover, the product of an operator and its adjoint is a Hermitian (self-adjoint) positive-definite operator. The adjoint of an operator is the conjugate transpose of the operator, which is just the transpose for a real operator.

Equations (3.10a), (3.10b), and (3.11) comprise the system to be solved via SOM. To solve this system, we must first construct the vector-calculus operators, and then build the PDE from these operators. The two most important operators for this flavor of SOM are the divergence operator, D_C, and the (modified) gradient operator, G_C, which is also the flux operator.

Constructing these operators involves defining an integral identity and appropriate inner-product spaces to express this identity, which is shown in Section 3.2.2.1. Once the integral identity is defined in terms of a primary and derived operator, Section 3.2.2.2 shows how to construct the PDE in terms of these operators, expressing the system of equations in a single operator equation, $\mathsf{A}_C U = B$, where A_C is an SPD operator.

3.2.2.1 Operators & Inner-Product Spaces

In this section we first derive a vector-calculus identity relating a vector and a scalar. An inner product space is then introduced to express this identity. A divergence and flux operator are then defined which express the integral identity in the inner-product space.

The integral identity originates from evaluating the divergence of the product between a vector and a scalar. Let W be a sufficiently smooth scalar and $\vec{J}$ be a sufficiently smooth vector, each defined in the volume V and on the boundary ∂V. The scalar-tensor product rules gives,

$$\vec{\nabla} \cdot (W\vec{J}) = \vec{J} \cdot \vec{\nabla} W + W(\vec{\nabla} \cdot \vec{J}). \tag{3.13}$$

Integrating this over volume yields,

$$\int_V \vec{\nabla} \cdot (W\vec{J}) dV = \int_V \vec{J} \cdot \vec{\nabla} W dV + \int_V W(\vec{\nabla} \cdot \vec{J}) dV \tag{3.14a}$$

$$\oint_{\partial V} (W\vec{J}) \cdot \hat{n} dS = \int_V \vec{J} \cdot \vec{\nabla} W dV + \int_V W(\vec{\nabla} \cdot \vec{J}) dV, \tag{3.14b}$$

where the Divergence theorem was applied to transform the LHS to a surface integral. Finally, this identity is written using inner-product notation, giving

$$\oint_{\partial V} W(\vec{J}, \hat{n}) dS = \int_V (\vec{J}, \vec{\nabla} W) dV + \int_V W(\vec{\nabla} \cdot \vec{J}) dV. \tag{3.15}$$

Equation (3.15) is a vector-calculus identity that is true for any sufficiently smooth values of $\vec{J}$ and W in the volume V. We refer back to this equation often because this is the general form of the integral identity that the SOM uses to relate the two main operators.

Now Equation (3.15) is applied to our specific system by substituting $W = U$. Keeping $\vec{J}$ as a general vector and moving some of the terms gives

$$\begin{aligned} \int_V U(\vec{\nabla} \cdot \vec{J}) dV - \oint_{\partial V} U(\vec{J}, \hat{n}) dS &= -\int_V (\vec{J}, \vec{\nabla} U) dV \\ &= -\int_V (\vec{J}, \mathbf{D}^{-1}\mathbf{D}\vec{\nabla} U) dV, \end{aligned} \tag{3.16}$$

where $\mathbf{D}^{-1}\mathbf{D} = \mathbf{I}$, and $\mathbf{I}$ is the identity matrix. This is the integral identity we wish to preserve using SOM. It embeds the relationship between the flux (vector) and temperature (scalar), including the phase terms.

In order to express Equation (3.16), two inner-product spaces must be introduced: one for vectors ($\mathbf{H}$) and another for scalars (H). Let V be the domain that is bounded by ∂V. Let X and W be arbitrary, smooth scalars in the scalar inner-product space: $X \in H, W \in H$. Let $\vec{J}$ and $\vec{Y}$ be arbitrary, smooth vectors in the vector inner-product space: $\vec{J} \in \mathbf{H}$, $\vec{Y} \in \mathbf{H}$. (See Reference [198] for more details on these inner-product spaces.) X and $\vec{Y}$ are arbitrary and will be eliminated. These SOM inner products are

$$(X, W)_H = \int_V XWdV + \oint_{\partial V} XWdS \tag{3.17a}$$

$$(\vec{J}, \vec{Y})_{\mathbf{H}} = \int_V \vec{J} \cdot (\mathbf{D}^{-1}\vec{Y})dV = \int_V (\vec{J}, \mathbf{D}^{-1}\vec{Y})dV. \tag{3.17b}$$

SOM makes an important choice by weighting the vector inner-product space by the inverse diffusivity in this definition, a key for accurately solving problems with discontinuous diffusivities [99, 198]. Note that the two SOM inner products will have a H or $\mathbf{H}$ subscript, while the standard inner product will have no subscript.

In order to write Equation (3.16) in terms of the inner-product spaces, Equation (3.17), two operators must be defined,

$$\mathsf{D}_C\vec{F} = \begin{cases} \vec{\nabla} \cdot \vec{F} & (x, y) \in V \\ -\beta(\vec{F}, \hat{n}) & (x, y) \in \partial V \end{cases} \tag{3.18a}$$

$$\mathsf{G}_C U = -\mathbf{D}\vec{\nabla}U \qquad (x, y) \in V. \tag{3.18b}$$

The operators D_C and G_C are the operators at the heart of SOM, and the discrete forms of these are used to construct a mimetic algorithm to solve the diffusion equation. The choice of modifying the gradient operator to a flux operator is an important step to allow SOM to treat strongly discontinuous coefficients. The flux operator does not need any boundary conditions because the divergence operator accounts for them.

Using the inner-product spaces specified in Equation (3.17) and the operators defined in Equation (3.18), the integral identity of Equation (3.16) can be expressed as

$$(\mathsf{D}_C\vec{J}, U)_H = (\vec{J}, \mathsf{G}_C U)_{\mathbf{H}}, \tag{3.19}$$

where $\mathsf{D}_C\vec{J}$ is a scalar, and $\mathsf{G}_C U$ is a vector. The implication of Equation (3.19) is that D_C and G_C are adjoint to each other,

$$\mathsf{D}_C = \mathsf{G}_C^*, \tag{3.20}$$

where the superscript $*$ is the adjoint (conjugate transpose). However, since all terms involved are real, this is equivalent to simply the transpose, such that $\mathsf{G}_C^* = \mathsf{G}_C^T$. Since D_C and G_C are real and nonzero, the products $\mathsf{D}_C\mathsf{G}_C$ and $\mathsf{G}_C^*\mathsf{D}_C^*$ form SPD operators.

3.2.2.2 Diffusion Equation in Operator Form

This section expresses the diffusion equation using the divergence (D_C) and modified gradient (G_C) operators. This step in the derivation involves discretization in time. Note that the continuum operators specifically refer to continuous in space.

The diffusion equation and its boundary terms,

$$\frac{\partial U}{\partial t} + \vec{\nabla} \cdot \vec{F} = Q \tag{3.21a}$$

$$\vec{F} = -\mathbf{D}\vec{\nabla} U \tag{3.21b}$$

$$\alpha U - \beta(\vec{F}, \hat{n}) = \psi, \tag{3.21c}$$

must be expressed using the operators of Equation (3.18). Let U be the current value of the temperature and U^o be the old value (at a time Δt in the past). Thus, $\frac{\partial U}{\partial t} = \lim_{\Delta t \to 0}(U - U^o)/\Delta t$. The boundary value and the current portion of the time-derivative term can be accounted for by introducing another operator,

$$\Omega_C U = \begin{cases} \frac{1}{\Delta t} U & (x,y) \in V \\ \alpha U & (x,y) \in \partial V \end{cases} . \tag{3.22}$$

The system is fully specified by accounting for the source term (Q), boundary values (ψ), and the old portion of the time derivative, in the term,

$$B = \begin{cases} Q + \frac{1}{\Delta t} U^o & (x,y) \in V \\ \psi & (x,y) \in \partial V \end{cases} , \tag{3.23}$$

where B is a scalar.

The system of equations, Equation (3.21), can now be expressed as

$$\Omega_C U + \mathsf{D}_C \vec{F} = B, \tag{3.24}$$

where the boundary conditions are embedded in the operators. Expressing the problem using the flux operator yields

$$(\mathsf{\Omega}_C + \mathsf{D}_C\mathsf{G}_C)U = B. \tag{3.25}$$

Since all the operators are acting on one value (U), they can be combined into one operator, A_C, reducing the problem to

$$\mathsf{A}_C U = B, \tag{3.26}$$

where

$$\mathsf{A}_C = \mathsf{\Omega}_C + \mathsf{D}_C\mathsf{G}_C. \tag{3.27}$$

To show A_C is SPD, we can specify $\alpha \geq 0$ with appropriate definitions of β and ψ (i.e., if $\alpha < 0$, multiply the boundary condition by negative one). Since $\Delta t > 0$ as well, $\mathsf{\Omega}_C = \mathsf{\Omega}_C^* \geq 0$ [99, 198]. Substituting Equation (3.20), which states $\mathsf{D}_C = \mathsf{G}_C^*$, we have

$$\mathsf{A}_C = \mathsf{\Omega}_C + \mathsf{G}_C^*\mathsf{G}_C = \mathsf{\Omega}_C + \mathsf{D}_C\mathsf{D}_C^*. \tag{3.28}$$

It logically follows that $\mathsf{A}_C = \mathsf{A}_C^*$ [198]. Thereby, A_C is a Hermetian or self-adjoint operator, which leads to the property that

$$(U, \mathsf{A}_C V)_H = (\mathsf{A}_C U, V)_H. \tag{3.29}$$

3.2.3 Discrete Derivation of SOM

This section derives the discrete form of SOM. Note that the discrete forms of the operators are matrices. However, for clarity of presentation, we shall continue to use the operator symbol with the explicit understanding that operators in this section are discrete analogs of the continuum operators and represent real matrices.

There are three primary steps. First, the integral identity is discretized over a single cell (Section 3.2.3.1). Second, the diffusion equation is discretized over a single cell using the integral identity Section 3.2.3.2. Third, a global system is formed by combining cells such that continuity of temperature and flux are enforced (Section 3.2.3.5).

3.2.3.1 Discretization of the Integral Identity

Starting from Equation (3.15) with the substitutions that $W = U$ and $\vec{\nabla} W = \mathbf{D}^{-1}\mathbf{D}\vec{\nabla} U = -\mathbf{D}^{-1}\vec{F}$, we have

$$\int_V \vec{J} \cdot (\mathbf{D}^{-1}\vec{F})dV = \int_V U(\vec{\nabla} \cdot \vec{J})dV - \oint_{\partial V} (U\vec{J}) \cdot \hat{n}dS, \tag{3.30}$$

where $\vec{J}$ is an arbitrary vector. Equation (3.30) becomes the integral identity central to SOM, which is discretized below. The integral discretization choices are based on References [151, 152, 185]. Note that this integral embeds the relationship between the flux and the gradient of the temperature without containing the gradient operator. The RHS is simpler, and thus we being with these terms.

We assume general quadrilateral cells where the volume integrals are taken over a single cell. Since our work is in 2D, the volume is in fact an area. Local values are defined on the left, right, top, and bottom sides of a quadrilateral (as well as the center for scalars) for each cell. A scalar has the following definitions,

$$U_C \quad \text{cell center} \tag{3.31a}$$

$$U_L \quad \text{left face center} \tag{3.31b}$$

$$U_R \quad \text{right face center} \tag{3.31c}$$

$$U_T \quad \text{top face center} \tag{3.31d}$$

$$U_B \quad \text{bottom face center.} \tag{3.31e}$$

Similarly, vector components are defined as

$$J_L = \vec{J} \cdot \hat{n}_L \quad \text{left face center} \tag{3.32a}$$

$$J_R = \vec{J} \cdot \hat{n}_R \quad \text{right face center} \tag{3.32b}$$

$$J_T = \vec{J} \cdot \hat{n}_T \quad \text{top face center} \tag{3.32c}$$

$$J_B = \vec{J} \cdot \hat{n}_B \quad \text{bottom face center,} \tag{3.32d}$$

where $\hat{n}_s$ is the outward unit normal from side s. In addition, for rectangular cells, $\hat{n}_L = -\hat{x}$, $\hat{n}_R = \hat{x}$, $\hat{n}_T = \hat{y}$, and $\hat{n}_B = -\hat{y}$, where the hat symbol indicates a unit vector.

Considering the first term on the RHS of Equation (3.30), we have

$$\int_V U(\vec{\nabla} \cdot \vec{J})dV \approx U_C \left[A_L J_L + A_R J_R + A_T J_T + A_B J_B\right], \tag{3.33}$$

where A_s is the areas of side s. For rectangular grids, this reduces to

$$\int_V U(\vec{\nabla} \cdot \vec{J})dV \approx U_C \left[\Delta y(J_L + J_R) + \Delta x(J_T + J_B)\right]. \tag{3.34}$$

Note that all components of $\vec{J}$ are added (rather than some being subtracted) because the elements are defined with respect to the outward unit normal (see Equation (3.32)).

The second term on the RHS of Equation (3.30) is a surface integral, so we have

$$\oint_{\partial V} (U\vec{J}) \cdot \hat{n} dS \approx A_L U_L J_L + A_R U_R J_R + A_T U_T J_T + A_B U_B J_B, \tag{3.35}$$

where we must use the surface components of the scalar temperature. For a rectangular cell, this reduces to

$$\oint_{\partial V} (U\vec{J}) \cdot \hat{n} dS \approx \Delta y(U_L J_L + U_R J_R) + \Delta x(U_T J_T + U_B J_B). \tag{3.36}$$

It should be evident how to generalize both Equation (3.33) and Equation (3.35) to cells with greater or fewer sides, since these integral approximations both reduce to summations of surface components.

The remaining term of the integral identity, LHS of Equation (3.30), introduces a diffusivity-modified shape matrix, which we will simply refer to as the shape matrix. The derivation of the shape matrix requires a protracted explanation, and the full details of the discretization of this term are given in Appendix B. As we alluded to when discussing the S3 model, this integral is vertex based. The vertex dependence is evident in the resulting discretization,

$$\begin{aligned}\int_V \vec{J} \cdot (\mathbf{D}^{-1}\vec{F})dV \approx &(\vec{J}_{LT} \cdot \mathbf{S}_{LT}\vec{F}_{LT})V_{LT} + (\vec{J}_{RT} \cdot \mathbf{S}_{RT}\vec{F}_{RT})V_{RT} \\ &+(\vec{J}_{LB} \cdot \mathbf{S}_{LB}\vec{F}_{LB})V_{LB} + (\vec{J}_{RB} \cdot \mathbf{S}_{RB}\vec{F}_{RB})V_{RB},\end{aligned} \tag{3.37}$$

where $\mathbf{S}_{ss'}$ is a 2×2 corner shape matrix for terms at the vertex at the intersection of side s(=R or L) and side s'(=T or B). Similarly, $\vec{J}_{ss'}$ and $\vec{F}_{ss'}$ are two-element vectors pointing in the $\hat{n}_s$ and $\hat{n}_{s'}$ directions. $V_{ss'}$ represents the volumetric weight of the ss' corner. For a rectangle, $V_{ss'} = \frac{1}{4}\Delta x \Delta y$ for all four corners. The shape matrix for the ss' corner is specified as

$$\mathbf{S}_{ss'} = \begin{bmatrix} S_{ss'}^{s,s} & S_{ss'}^{s,s'} \\ S_{ss'}^{s',s} & S_{ss'}^{s',s'} \end{bmatrix}. \tag{3.38}$$

The components are

$$S^{s,s}_{ss'} = \frac{1}{\sin^2\theta_{ss'}}[K^{xx}_{ss'}\sin^2\theta_{xs'} + K^{yy}_{ss'}\cos^2\theta_{xs'} - K^{xy}_{ss'}\sin(2\theta_{xs'})] \tag{3.39a}$$

$$S^{s',s'}_{ss'} = \frac{1}{\sin^2\theta_{ss'}}[K^{xx}_{ss'}\sin^2\theta_{xs} + K^{yy}_{ss'}\cos^2\theta_{xs} - K^{xy}_{ss'}\sin(2\theta_{xs})] \tag{3.39b}$$

$$\begin{aligned} S^{s,s'}_{ss'} = S^{s',s}_{ss'} = \frac{1}{\sin^2\theta_{ss'}}[&K^{xy}_{ss'}\sin(\theta_{xs}+\theta_{xs'}) \\ &- (K^{xx}_{ss'}\sin\theta_{xs}\sin\theta_{xs'} + K^{yy}_{ss'}\cos\theta_{xs}\cos\theta_{xs'})] \end{aligned} \tag{3.39c}$$

where $\mathbf{K} = \mathbf{D}^{-1}$. We have $S^{s',s}_{ss'} = S^{s,s'}_{ss'}$ because $D^{xy} = D^{yx}$, since $\mathbf{D}$ is physically required to be SPD [16, 111, 159, 163, 164]. The shape matrices from the other three corners are defined analogously. From Equation (3.39) it is clear that $(\mathbf{S}_{ss'})^{-1}$ acts as an effective diffusivity tensor that converts the x-y diffusivity and angular dependence with respect to the x-y grid into the face-normal system.

Equations (3.33), (3.35), and (3.37) complete the discretization of Equation (3.30). While the specific choices for discretization are not unique [152], they are straightforward. The integral identity,

$$\int_V \vec{J}\cdot(\mathbf{D}^{-1}\vec{F})dV = \int_V U(\vec{\nabla}\cdot\vec{J})dV - \oint_{\partial V}(U\vec{J})\cdot\hat{n}dS, \tag{3.40}$$

is discretized as

$$\begin{aligned} (\vec{J}_{LB}\cdot\mathbf{S}_{LB}\vec{F}_{LB})V_{LB} + &(\vec{J}_{RB}\cdot\mathbf{S}_{RB}\vec{F}_{RB})V_{RB} + (\vec{J}_{LT}\cdot\mathbf{S}_{LT}\vec{F}_{LT})V_{LT} \\ +(\vec{J}_{RT}\cdot\mathbf{S}_{RT}\vec{F}_{RT})V_{RT} = &U_C\left[A_LJ_L + A_RJ_R + A_TJ_T + A_BJ_B\right] \\ &- \left[A_LU_LJ_L + A_RU_RJ_R + A_TU_TJ_T + A_BU_BJ_B\right] \end{aligned} \tag{3.41}$$

Terms on the RHS can be grouped, giving a slightly more compact form,

$$\begin{aligned} (\vec{J}_{LB}\cdot\mathbf{S}_{LB}\vec{F}_{LB})V_{LB} + (\vec{J}_{RB}\cdot\mathbf{S}_{RB}\vec{F}_{RB})V_{RB} \\ +(\vec{J}_{LT}\cdot\mathbf{S}_{LT}\vec{F}_{LT})V_{LT} + (\vec{J}_{RT}\cdot\mathbf{S}_{RT}\vec{F}_{RT})V_{RT} = \\ A_LJ_L(U_C - U_L) + A_RJ_R(U_C - U_R) + A_TJ_T(U_C - U_T) + A_BJ_B(U_C - U_B). \end{aligned} \tag{3.42}$$

3.2.3.2 Local System of Equations

The diffusion equation, combined with Equation (3.42), can be expressed in the form of the matrix equation, $\mathbf{A}\vec{x} = \vec{b}$, where $\mathbf{A} \in \Re^{5\times1}$ and $\vec{x}, \vec{b} \in \Re^{5\times1}$. First the discrete analog of the integral identity must be written in matrix form.

The LHS of Equation (3.42) can be expressed as the following matrix products,

$$\begin{bmatrix} J_L & J_R & J_T & J_B \end{bmatrix} \mathbf{S} \begin{bmatrix} F_L \\ F_R \\ F_T \\ F_B \end{bmatrix} = \begin{bmatrix} A_L J_L (U_C - U_L) \\ A_R J_R (U_C - U_R) \\ A_T J_T (U_C - U_T) \\ A_B J_B (U_C - U_B) \end{bmatrix}, \tag{3.43}$$

where $\mathbf{S} \in \Re^{4\times 4}$ is the shape matrix. $\mathbf{S}$ is defined as

$$\mathbf{S} = \begin{bmatrix} (V_{LT}S_{LT}^{L,L} + V_{LB}S_{LB}^{L,L}) & 0 & V_{LT}S_{LT}^{L,T} & V_{LB}S_{LB}^{L,B} \\ 0 & (V_{RT}S_{RT}^{R,R} + V_{RB}S_{RB}^{R,R}) & V_{RT}S_{RT}^{R,T} & V_{RB}S_{RB}^{R,B} \\ V_{LT}S_{LT}^{T,L} & V_{RT}S_{RT}^{T,R} & (V_{RT}S_{RT}^{T,T} + V_{LT}S_{LT}^{T,T}) & 0 \\ V_{LB}S_{LB}^{B,L} & V_{RB}S_{RB}^{B,R} & 0 & (V_{LB}S_{LB}^{B,B} + V_{RB}S_{RB}^{B,B}) \end{bmatrix}, \tag{3.44}$$

where the components are defined in Equation (3.39). Note that $\mathbf{S}$ is SPD. For an orthogonal grid with equal diffusivities in each corner, the shape matrix is

$$\mathbf{S} = \frac{\Delta x \Delta y}{4} \begin{bmatrix} 2K^{xx} & 0 & -K^{xy} & K^{xy} \\ 0 & 2K^{xx} & K^{xy} & -K^{xy} \\ -K^{xy} & K^{xy} & 2K^{yy} & 0 \\ K^{xy} & -K^{xy} & 0 & 2K^{yy} \end{bmatrix}. \tag{3.45}$$

Up to this point, the vector $\vec{J}$ has been arbitrary; these equations hold true for *any* $\vec{J}$. Therefore, one can define a *particular* $\vec{J}$ such that a simple system of equations can be generated. SOM selects the standard basis ([1,0,0,0], [0,1,0,0], [0,0,1,0], and [0,0,0,1]) as $\vec{J}$, which correspond to the cell faces $\{L, R, T, B\}$ [151, 152, 185]. Note that we have chosen a particular order for the vector components; this choice is free and arbitrary, but once made, it must remain consistent.

Substituting $\vec{J}$ in this manner reduces the system of equations to,

$$\mathbf{S}\vec{F} = \mathbf{A}(U_C\vec{1} - \vec{U}), \tag{3.46}$$

where the following matrix and three vectors are defined as

$$\mathbf{A} = \begin{bmatrix} A_L & 0 & 0 & 0 \\ 0 & A_R & 0 & 0 \\ 0 & 0 & A_T & 0 \\ 0 & 0 & 0 & A_B \end{bmatrix} \tag{3.47a}$$

$$\vec{F}^T = \begin{bmatrix} F_L & F_R & F_T & F_B \end{bmatrix} \tag{3.47b}$$

$$\vec{1}^T = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix} \tag{3.47c}$$

$$\vec{U}^T = \begin{bmatrix} U_L & U_R & U_T & U_B \end{bmatrix}. \tag{3.47d}$$

This can be expressed succinctly by adding an additional column to the area matrix and an additional row to the temperature matrix,

$$\mathbf{S}\vec{F} = -\mathbf{A}'\vec{U}', \tag{3.48}$$

where

$$\mathbf{A}' = \begin{bmatrix} A_L & 0 & 0 & 0 & -A_L \\ 0 & A_R & 0 & 0 & -A_R \\ 0 & 0 & A_T & 0 & -A_T \\ 0 & 0 & 0 & A_B & -A_B \end{bmatrix} \tag{3.49a}$$

$$(\vec{U}')^T = \begin{bmatrix} U_L & U_R & U_T & U_B & U_C \end{bmatrix}. \tag{3.49b}$$

The expression for flux becomes

$$\vec{F} = -\mathbf{S}^{-1}\mathbf{A}'\vec{U}'. \tag{3.50}$$

Comparing this with $\vec{F} = -\mathbf{D}\vec{\nabla}U$, we can identify the discrete form of the modified gradient operator, $\mathbf{S}^{-1}\mathbf{A}'$. Equation (3.50) gives an expression for the flux solely in terms of known values (diffusivities, face areas, corner angles, and cell volumes) multiplied by the temperature vectors, which consist of both cell- and face-centered unknowns.

The inverted shape matrix is required to write the expression for flux. Every cell then has an associated 4×4 matrix that must be inverted, which can be done iteratively or directly. We henceforth no longer write the exact expressions for the matrix products and restrict ourselves to forms such as Equation (3.50) rather than

Equation (3.49).

Discretize the diffusion equation. The integral identity, Equation (3.30), as well as the divergence and flux operator have been discretized, culminating in Equation (3.50). However, the diffusion equation has not yet been discretized. The diffusion equation integrated over the volume of one cell is

$$\int_V \frac{\partial U}{\partial t} dV = \int_V \vec{\nabla} \cdot \vec{F} dV = \int_V q dV. \tag{3.51}$$

Let the cell-centered value of the temperature be defined as

$$\int_V U dV = V U_C, \tag{3.52}$$

and let the time derivative be discretized using backwards-Euler time differencing,

$$\frac{\partial U_C}{\partial t} \approx \frac{U_C - U_C^o}{\Delta t}, \tag{3.53}$$

where U_C^o is the value at the previous time-step. The source term is defined as

$$Q = \int_V q dV + U_C^o / \Delta t. \tag{3.54}$$

Applying the divergence theorem to the flux term leads to [185]

$$\int_V \vec{\nabla} \cdot \vec{F} dV \approx \sum_{i=1}^{4} A_i F_i, \tag{3.55}$$

where this term is discretized similarly to Equation (3.35). Thus, the discrete diffusion equation for a single cell is

$$\frac{V}{\Delta t} U_C + \sum_{i=1}^{4} A_i F_i = Q. \tag{3.56}$$

Equation (3.50) is an expression for the flux, which, when multiplied by $\mathbf{A}$, gives

$$\mathbf{A}\vec{F} = -\mathbf{A}\mathbf{S}^{-1}\mathbf{A}'\vec{U}'. \tag{3.57}$$

Straightforward matrix multiplication finds that

$$
\begin{aligned}
A_iF_i = -A_i[P_{i1}(A_LU_L - A_iU_C) + P_{i2}(A_RU_R - A_iU_C) \\
+P_{i3}(A_TU_T - A_iU_C) + P_{i4}(A_BU_B - A_iU_C)],
\end{aligned}
\tag{3.58}
$$

where P_{ij} is the i^{th} row and j^{th} column of $\mathbf{P} = \mathbf{S}^{-1}$ and A_i is the value on the diagonal of the i^{th} row of $\mathbf{A}$. Let $\mathbf{M} = -\mathbf{A}\mathbf{S}^{-1}\mathbf{A}'$; thus Equation (3.57) becomes

$$\mathbf{A}\vec{F} = \mathbf{M}\vec{U}', \tag{3.59}$$

where $\mathbf{M} \in \Re^{4\times5}$. Then the area-flux face product (A_sF_s) for a side s can be expressed as the sum of a column of the matrix-vector product $\mathbf{M}\vec{U}'$,

$$A_sF_s = \sum_{j=1}^{5} M_{sj}U'_j, \tag{3.60}$$

where M_{sj} is the s^{th} row and j^{th} column of $\mathbf{M}$ and s is one of $[L,\ R,\ T,\ B]$.

By substituting Equation (3.60) into Equation (3.56), the discrete diffusion equation becomes

$$\frac{V}{\Delta t}U_C + \sum_{i=1}^{4}\left(\sum_{j=1}^{5} M_{ij}U'_j\right) = Q, \tag{3.61}$$

where U'_j is the j^{th} element of the five element temperature vector that contains the cell-center and face-center values. Rewriting Equation (3.61) with the cell-centered values separated gives

$$\sum_{i=1}^{4}\sum_{j=1}^{4} M_{ij}U_j + \left(\frac{V}{\Delta t} + \sum_{i=1}^{4} M_{i5}\right)U_C = Q, \tag{3.62}$$

which makes the following steps clearer.

A single system of equations containing Equation (3.62) and $\mathbf{A}\vec{F} = \mathbf{M}\vec{U}'$ can be formed by creating a 5×5 zonal matrix $\mathbf{Z}$, of which the first four rows are equal to $\mathbf{M}$. The first four elements of the fifth row is equal to the transpose of the fifth column of $\mathbf{M}$. The last element (located in fifth row, fifth column of $\mathbf{Z}$) is defined by the coefficient of U_C in Equation (3.62). Thus the following system,

$$\mathbf{Z}\vec{U}' = \vec{B}, \tag{3.63}$$

is defined by

$$Z_{ij} = \begin{cases} M_{ij} & i \in \{1,\ldots,4\}, j \in \{1,\ldots,5\} \\ M_{j5} & i = 5, j \in \{1,\ldots,4\} \\ -\dfrac{V}{\Delta t} - \displaystyle\sum_{i=1}^{4} M_{i5} & i = j = 5 \end{cases} \tag{3.64a}$$

$$B_i = \begin{cases} -A_i F_i & i \neq 5 \\ -Q & i = 5 \end{cases}, \tag{3.64b}$$

where B_i are elements of $\vec{B}$, Z_{ij} are elements of $\mathbf{Z}$, and $\mathbf{M} = -\mathbf{A}\mathbf{S}^{-1}\mathbf{A}$. The sizes of these matrices are $\mathbf{M} \in \Re^{4\times5}$, $\mathbf{Z} \in \Re^{5\times5}$, $\vec{B} \in \Re^{5\times1}$, $\vec{U}' \in \Re^{5\times1}$. Taking the negative of the diffusion equation in the fifth row of $\mathbf{Z}$ makes the system of equations SPD, as opposed to symmetric negative-definite.

The diffusion equation and integral identity for a single cell are now expressed as a 5×5 matrix equation, $\mathbf{Z}\vec{U}' = \vec{B}$. In order to obtain the equations for the entire mesh, a global matrix equation must be assembled from the single-cell matrix equations. This global matrix equation is what is given as input to a matrix solver (e.g., a conjugate gradient solver). However, two additional steps must occur before assembling the global system. First, the boundary conditions must be implemented, as this alters both $\mathbf{Z}$ and $\vec{B}$ from the matrix equation. Second, the cell-centered unknown must be eliminated from the system, resulting in a 4×4 matrix equation. This reduced system is then assembled into a global matrix equation, where the unknown vector consists entirely of face-centered unknowns. The following three sections are thus presented in the order in which they must logically occur: enforcing boundary conditions (Section 3.2.3.3), eliminating cell-centered unknowns (Section 3.2.3.4), and assembling the system of equations (Section 3.2.3.5).

3.2.3.3 Boundary Conditions

Equation (3.63) is the 5×5 equation for each cell. Boundary conditions are enforced on this system prior to forming the global system. Note that while most cells do not have any boundaries, some cells (four in 2D) have multiple boundaries. Multiple boundary conditions are applied by considering one boundary first, making the appropriate zonal matrix and solution vector modifications, and then repeating for each additional boundary.

The general boundary condition is

$$\alpha U - \beta(\vec{F}, \hat{n}) = \psi \qquad (\text{on } \partial V), \tag{3.65}$$

where $\alpha \neq 0$ and $\beta = 0$ yields a Dirichlet boundary condition, $\alpha = 0$ and $\beta \neq 0$ corresponds to a Neumann boundary condition, and $\alpha\beta \neq 0$ is a mixed or Robin boundary type. Note that both α and β cannot be simultaneously zero. In SOM, boundary conditions are enforced on cell faces, as opposed to the cell centers of a ghost zone. As a result, for a particular boundary face k, the boundary condition is

$$\alpha U_k - \beta F_k = \psi_k. \tag{3.66}$$

Dirichlet boundary condition. A Dirichlet boundary condition is

$$U_k = \frac{\psi_k}{\alpha}. \tag{3.67}$$

This is enforced in the equation $\mathbf{Z}\vec{U}' = \vec{B}$ by setting the k^{th} row and column of $\mathbf{Z}$ to zeros everywhere except for on the diagonal, i.e., $Z_{ij} = \delta_{i,k}\delta_{j,k}$, and $B_k = \psi_k/\alpha$. However, in order for the system to remain conservative, the contributions to the flux from the k^{th} face must be subtracted from $\vec{B}$. This is possible because the exact value of U_k is known. The matrices now become

$$B_i^{\text{DBC}} = \begin{cases} B_i - Z_{ik}\dfrac{\psi_k}{\alpha} & i \neq k \\ \dfrac{\psi_k}{\alpha} & i = k \end{cases} \tag{3.68}$$

$$Z_{ij}^{\text{DBC}} = \begin{cases} Z_{ij} & (i \neq k) \text{ and } (j \neq k) \\ 0 & ((i = k) \text{ or } (j = k)) \text{ and } (i \neq j) \\ 1 & i = j = k \end{cases}, \tag{3.69}$$

where DBC indicates ‘Dirichlet boundary condition.’ The result is that the matrix multiplication for this row of Equation (3.63) is simplified to

$$\begin{bmatrix} 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} U_L \\ U_R \\ U_T \\ U_B \\ U_C \end{bmatrix} = \frac{\psi_k}{\alpha}, \tag{3.70}$$

where we have shown the case $k = 3$, which clearly gives $\alpha U_T = \psi_k$.

Neumann boundary conditions. The Neumann boundary conditions are written in terms of gradient, which makes it natural to impose these conditions on the flux-based SOM. For a particular boundary face k, a Neumann boundary condition is

$$F_k = -\frac{\psi_k}{\beta}. \tag{3.71}$$

Recall that the system of equation $\mathbf{Z}\vec{U}' = \vec{B}$ represents $A_i F_i$ with the first four rows (the fifth enforces the diffusion equation). Hence, a Neumann boundary condition update is enforced as

$$B_i^{\text{NBC}} = \begin{cases} B_i & i \neq k \\ B_i + A_i \dfrac{\psi_k}{\beta} & i = k \end{cases}, \tag{3.72}$$

with no modification of the zonal matrix, $\mathbf{Z}$, and where NBC indicates Neumann boundary condition.

Robin boundary conditions. The Robin boundary condition is more complicated since both U_k and F_k must be specified. To proceed, Equation (3.66) is rewritten as

$$A_k F_k - A_k \frac{\alpha}{\beta} U_k = -A_k \frac{\psi_k}{\beta}. \tag{3.73}$$

From $\mathbf{Z}\vec{U}' = \vec{B}$ we have that

$$A_k F_k = \sum_{j=1}^{5} Z_{kj} U_j', \tag{3.74}$$

the boundary condition can be expressed again in terms of the matrix $\mathbf{Z}$ as,

$$\sum_{j=1}^{5} Z_{kj} U'_j - A_k \frac{\alpha}{\beta} U_k = -A_k \frac{\psi_k}{\beta}. \tag{3.75}$$

This can be enforced by modifying $\vec{B}$ and $\mathbf{Z}$ as

$$B_i^{\text{RBC}} = \begin{cases} B_i & i \neq k \\ B_i - A_k \dfrac{\psi_k}{\beta} & i = k \end{cases} \tag{3.76}$$

$$Z_{ij}^{\text{RBC}} = \begin{cases} Z_{ij} & i \neq k \\ Z_{ij} - A_k \dfrac{\alpha}{\beta} & i = k \end{cases}, \tag{3.77}$$

where RBC indicates Robin boundary condition.

3.2.3.4 Eliminating the Cell-Centered Unknown

Once the boundary conditions have been enforced, there is enough information to completely eliminate the cell-centered unknown from the system. Hence, the 5×5 system $\mathbf{Z}\vec{U}' = \vec{B}$ can be reduced to a 4×4 system, and the temperature vector reverts back to simply $\vec{U}$ from $\vec{U}'$.

The last row of Equation (3.63) is

$$\sum_{j=1}^{4} Z_{5j} U_j + Z_{55} U_C = B_5, \tag{3.78}$$

which can be solved for U_C, giving

$$U_C = \frac{1}{Z_{55}} \left(B_5 - \sum_{j=1}^{4} Z_{5j} U_j \right). \tag{3.79}$$

This expresses the cell-centered temperature only in terms of known matrix elements and the face-centered temperatures. Note that the values of B_5 and Z_{5j} that appear in Equation (3.79) *must be stored* so that the cell-centered value can be recovered at the end of the time step.

Next, Equation (3.79) is substituted for row $i \in \{1, \ldots, 4\}$ of Equation (3.63) (the matrix equation):

$$\sum_{j=1}^{4} Z_{ij}U_j + Z_{i5}U_C = B_i \tag{3.80a}$$

$$\sum_{j=1}^{4} Z_{ij}U_j + \frac{Z_{i5}}{Z_{55}}\left(B_5 - \sum_{j=1}^{4} Z_{5j}U_j\right) = B_i \tag{3.80b}$$

$$\sum_{j=1}^{4} Z_{ij}U_j - \frac{Z_{i5}}{Z_{55}}\sum_{j=1}^{4} Z_{5j}U_j = B_i - \frac{Z_{i5}}{Z_{55}}B_5 \tag{3.80c}$$

$$\sum_{j=1}^{4}\left(Z_{ij} - \frac{Z_{i5}}{Z_{55}}Z_{5j}\right)U_j = B_i - \frac{Z_{i5}}{Z_{55}}B_5. \tag{3.80d}$$

This eliminates the cell-centered value from Equation (3.63). The new system of equations is

$$\tilde{\mathbf{Z}}\vec{U} = \tilde{B}, \tag{3.81}$$

where

$$\tilde{B}_i = B_i - \frac{Z_{i5}}{Z_{55}}B_5 \qquad i \in \{1,\dots,4\} \tag{3.82a}$$

$$\tilde{Z}_{ij} = Z_{ij} - \frac{Z_{i5}}{Z_{55}}Z_{5j} \qquad i,j \in \{1,\dots,4\}, \tag{3.82b}$$

and $\tilde{\mathbf{Z}} \in \Re^{4\times 4}$ and $\tilde{B} \in \Re^{4\times 1}$. The boundary conditions must be implemented prior to defining $\tilde{\mathbf{Z}}$ and $\tilde{B}$.

3.2.3.5 Assembling the Global System

Up until this point, the discrete derivation has only considered a single cell in the mesh. The face-centered coordinates have been local and completely independent. The global matrix equation is constructed by assembling the single-cell matrix equations. In order to maintain the discrete analog of the integral identity, Equation (3.30), continuity of temperature and flux must be satisfied. This converts the surface integral of Equation (3.30) from the surface of an individual cell to the external boundaries where the boundary conditions are imposed [152].

Let us introduce a global notation for temperature and flux. A specific cell in the mesh is referred to by the i^{th} row and j^{th} (e.g., $U_{i,j}$), while the four faces of a cell are enumerated and referenced by s, giving

$$U^s_{i,j} \tag{3.83a}$$

$$F^s_{i,j}. \tag{3.83b}$$

Note that i and j correspond to a position within the spatial grid, not an element of the coefficient matrix as with the local coordinate system. Also note that the comma in terms like $U_{i,j}$ do not imply a derivative.

Continuity of temperature requires that cells sharing a face must have the same temperature on that face, e.g.,

$$U^R_{i,j} = U^L_{i+1,j} \tag{3.84a}$$

$$U^T_{i,j} = U^B_{i,j+1}. \tag{3.84b}$$

This leads to a single-temperature value on each face, allowing it to be uniquely referred to by the face-centered index,

$$U_{i+\frac{1}{2},j} = U^R_{i,j} = U^L_{i+1,j} \tag{3.85a}$$

$$U_{i,j+\frac{1}{2}} = U^T_{i,j} = U^B_{i,j+1}. \tag{3.85b}$$

Therefore, we no longer need to identify a temperature as being cell-centered. (A cell-centered temperature has been denoted by integer indices; a face-centered temperature now has a half-integer index.)

Continuity of flux demands that the flux leaving one face of a cell is equal to the amount of flux entering the cell sharing that face,

$$F^R_{i,j} = -F^L_{i+1,j} \tag{3.86a}$$

$$F^T_{i,j} = -F^B_{i,j+1}, \tag{3.86b}$$

where the negative sign is due to flux components all being defined as outward fluxes. In order to construct a global matrix that is SPD [152, 185], the flux continuity expression must be negative, e.g.,

$$-A^R_{i,j}F^R_{i,j} - A^L_{i,j}F^L_{i+1,j} = 0 \tag{3.87a}$$

$$-A^T_{i,j}F^T_{i,j} - A^B_{i,j}F^B_{i,j+1} = 0. \tag{3.87b}$$

This means that the values of A_iF_i in Equation (3.64b) (and all subsequent expressions for $\vec{B}$ and $\tilde{B}$) are not actually needed unless the face is on an external boundary or has a surface source. Accordingly, one can set A_iF_i to zero for the faces where the fluxes are continuous before assembling the global system, and then manipulate the matrices as described.

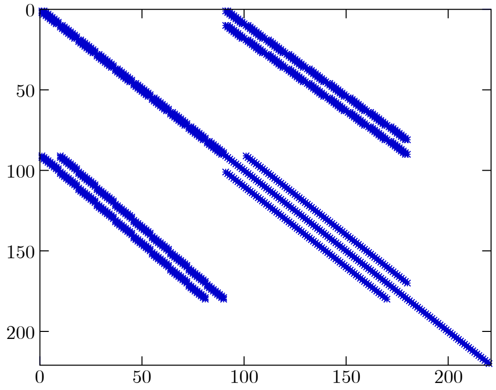


Figure 3.4: Sparse matrix structure for a 10×10 grid from SOM. Nonzero values are marked with blue asterisks, and the seven bands are visible. With 1,188 nonzero values and 220^2 elements, this matrix is 2.45% nonzero.

The global system is then given by

$$\mathcal{Z}\vec{\mathcal{U}} = \vec{\mathcal{B}}, \tag{3.88}$$

where $\vec{\mathcal{U}}$ is the vector consisting of each face-centered temperature, and $\vec{\mathcal{B}}$ is assembled from $\tilde{B}$ of each cell. The global matrix, $\mathcal{Z}$, is an SPD matrix that can be inverted with efficient linear solvers such as the conjugate gradient solver. Figure 3.4 shows an example of the sparse matrix $\mathcal{Z}$ for a 10×10 system, which corresponds to 100 cell-centered unknowns and 220 face-centered unknowns. The vector $\vec{\mathcal{U}}$ contains only face-centered temperatures, while $\vec{\mathcal{B}}$ contains source terms, boundary values, and terms that were used to eliminate cell-centered temperatures in Equation (3.82a). Once the matrix is inverted and $\vec{\mathcal{U}}$ is determined, the cell-centered values of the temperature can be recovered via Equation (3.79) and the fifth row values from the single cell equation. Note that script variables (e.g., $\vec{\mathcal{U}}$, $\mathcal{F}$) correspond to global variables.

The global mapping process is similar to that of finite element methods (FEM), and further details can be obtained from literature of that method (e.g., Zienkiewicz and Taylor [222]). An equation of flux for each face in the mesh is collected into a system of equations expressed as $\mathcal{Z}\vec{\mathcal{U}} = \vec{\mathcal{B}}$. For an interior face, flux (multiplied with the area) from two cells sharing a face are combined to sum to zero (in the absence of source terms). (Three cells would be used for a flux expression for a hanging node.) The equation for a boundary face depends on a single cell; the k^{th} row of $\tilde{Z}$ and $\tilde{B}$ are mapped to $\mathcal{Z}$ and $\vec{\mathcal{B}}$, respectively, from the boundary face (k) of the boundary cell.

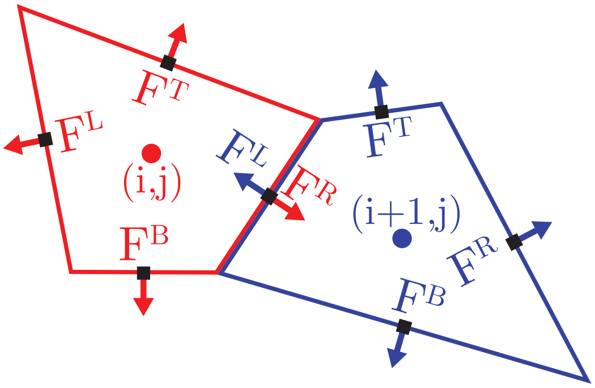


Figure 3.5: Flux between a face shared between two cells. The flux for each face of a cell is indicated, with each cell having a single color (red for cell (i, j) and blue for cell $(i + 1, j)$). The expressions for flux in the single cell system couples all four face-centered temperatures in each cell. Consequently, the equation for flux of one face in the global system (which depends on two cells) leads to an equation with seven unknowns. This is evident in the picture by the seven different face-centered temperatures (marked with black boxes).

We consider one specific example of assembling the global equation of the flux for one particular face shared between nonboundary cells. Let us consider the flux between the right face of cell (i, j) and the left face of cell $(i+1, j)$ for two nonboundary cells, as shown in Figure 3.5. Let the global element number for the $(i + \frac{1}{2}, j)$ face be n (i.e., $F_{i+\frac{1}{2},j} = \mathcal{F}_n$). Using the local coordinate system, where the subscript (i, j) is the cell coordinate and the superscript is the side of that cell, the flux equation for the n^{th} face is $-A^R_{i,j}F^R_{i,j} - A^L_{i+1,j}F^L_{i+1,j} = 0$. In the most general case, where the diffusivity is a tensor and the cells are not rectangles, all four faces of a cell are coupled. Thus, the local equation for flux to the right in cell (i, j) depends on $U^L_{i,j}$, $U^R_{i,j}$, $U^T_{i,j}$, and $U^B_{i,j}$; while the local flux equation to the left for cell $(i + 1, j)$ depends on $U^L_{i+1,j}$, $U^R_{i+1,j}$, $U^T_{i+1,j}$, and $U^B_{i+1,j}$. Since $U^R_{i,j} = U^L_{i+1,j}$, the global equation for flux in the mesh at face $\mathcal{F}_n$ depends on seven unknowns (four from each cell, with one shared value). All the elements of $\mathcal{Z}$ in row n are zero except for the seven columns corresponding to these seven unknowns. These nonzero values correspond to the four values from the second row (i.e., right face) of $\tilde{Z}$ for cell (i, j) and the four values from the first row (i.e., left face) of $\tilde{Z}$ in cell $(i+1, j)$. Similarly, the n^{th} row of $\vec{\mathcal{B}}$ is the combination (negative sum) of the second element (i.e., right face) of $\tilde{B}$ of cell (i, j) and first element (i.e., left face) $\tilde{B}$ of cell $(i+1, j)$.

3.3 Summary

We have discussed three single-temperature mixed-cell models. The first of which, the S1 model, is a standard approach where the harmonic mean of each phase's diffusivity is used as an effective diffusivity for a mixed-cell. This S1 model is used as a baseline for comparison for the other two single-temperature models, as well as for the three multi-temperature models presented in the next chapter. The S2 model is a rotated tensor diffusivity, where the effective diffusivity of a mixed cell is approximated in the form of a tensor in attempt to capture the effect of the phase interface. The S3 model effectively splits a mixed cell into four subcells, and we show a means to approximate these subcells as four pure cells whenever possible while maintaining the correct volumes of each phase.

The S1 and S2 model can be implemented with a variety of numerical schemes. The S3 model, however, is more restrictive in its implementation. We chose to execute all three of these models using SOM, a mimetic technique that yields excellent computational and mathematical properties, particularly with discontinuous diffusivities. SOM is a pertinent method for solving problems with highly discontinuous diffusivities, such as those explored in this thesis. Moreover, SOM is unheralded due to its short history and abstract description. This chapter serves as a useful introduction and resource to broaden the application of SOM in apposite computational problems.

CHAPTER IV

Multi-Temperature Methods

In this chapter, we present three multi-temperature models to treat mixed cells for a 2D diffusion problem. To accomplish this, we derive the diffusion equation in a phase-based form, where there is an equation for each phase and an exchange between phases. Section 4.1 presents the derivation and discretization of this method. After deriving the method, we specify all three mixed-cell models in Section 4.2, which are based on progressively more accurate approximations. The first model is a rough approximation whose main advantage is simplicity, which is achieved by avoiding interface reconstruction. The second and third models both reconstruct the interface in order to compute the centroids for each phase and compute face areas, centroid distances, and (ideally) temperatures more accurately.

The multi-temperature method is discretized via finite difference methods (FDM). However, the equations must be integrated over the volume of a single cell in order to obtain average values for each phase. (Note, as with SOM, we use 3D terminology for 2D terms.) This step results in the generation of volume fractions, which are used to compute the temperature of phases separately. Furthermore, since the method enforces flux continuity between faces, this is technically a finite volume method (FVM), albeit with an orthogonal grid. While the presentation of this method is restricted to only two phases (and two temperatures), in principle, the number of phases (and associated temperatures) can be extended to n phases.

The original motivation for the work presented in this chapter was to create a multiphase, multi-temperature code that increases mixed-cell accuracy without requiring interface reconstruction. It is desirable to avoid interface reconstruction when extrapolating from 2D to 3D, where interface reconstruction is much more difficult. Interface reconstruction is used, however, in the M2 and M3 methods in order to determine the location of the centroids, as well as to approximate areas of faces and interface normals. Should these two models achieve significantly higher accuracy than the M1 model, it may be an indication that the knowledge of the interfacial location is essential for accurately capturing subcell effects when using this approach. The centroids are calculated in the M2 and M3 models only for two phases while assuming a linear interface. However, there are a variety of ways to determine the centroids, and an astute next step would be to employ a more robust method, such as the Moment of Fluid (MoF) algorithm [69, 70], which tracks volume fractions and centroids of multiple phases and which has been extended to 3D [2].

4.1 Multi-Temperature Derivation

Two-phase and multi-phase flow are well-established [66, 88, 89, 157, 158, 162, 168]. The motivation for volume averaging is frequently associated with porous media flows, where one averages the fine structure over a volume to obtain average values at lower resolution (this is often referred to as coarse graining). However, the assumptions of our model problem (see Section 2.4.2) greatly simplify the typical exposition by their lack of a velocity dependence (which reduces the equations, from a fluids perspective, to just energy, eliminating mass and momentum transfer). Note that by having adjacent phases with different temperatures, particularly within a single cell, the phases are not in LTE with each other.

The forthcoming derivation applies to all three models (M1, M2, and M3) unless specifically noted.

4.1.1 Notation

Directions and cells are labeled with respect to the center cell (C). The four faces (and neighbors) on the left, right, top, and bottom of C are represented with labels L, R, T, and B, with the variable ζ used to indicate one of the labels (e.g., $\zeta = L$). There is a fifth possible type of face that occurs only in mixed cells, denoted as I, that corresponds to a interface. Two phases exist: phase 1 and phase 2. The arbitrary designation of these phases allows for the presentation to apply to either phase, thus most equations will be written only for phase 1. However, the corresponding expressions can be obtained for phase 2 in a similar manner. There are five quantities of interest: flux ($\vec{F}$), face area (A), temperature (U), distance (l), and diffusivity (D). We introduce the following notation, which applies to all five of these quantities,

$$F^1_{C\zeta}, \tag{4.1}$$

where the superscript indicates the phase, the first subscript indicates the position, and the second subscript (when present) indicates the face or direction with respect to the position. Flux, area, and distance require the double subscript to indicate what face of the cell is being measured, and the order of the subscripts can represent the normal direction (e.g., l^1_{TC} points from the top cell to the center cell). Temperature and diffusivity are scalars who only require a single subscript to identify the cell where the variable is located. For example, the phase 1 flux component from the left cell to the center cell is F^1_{LC}, which is equal and opposite to the phase 1 flux from the center cell to the left cell, F^1_{CL}. The phase 2 area that the right cell shares with the center cell is A^2_{RC} , and D^1_B is the diffusivity of phase 1 of the bottom cell.

In most cases, the faces shared between two cells are of the same phase and size and thus,

$$A^1_{C\zeta} = A^1_{\zeta C} \tag{4.2a}$$

$$A^2_{C\zeta} = A^2_{\zeta C}, \tag{4.2b}$$

where the ζ face of the center cell (C) equals the face that cell ζ shares with cell C. Figure 4.1 is an example of a mixed cell with the majority phase (phase 1) rendered as blue and the minority phase (phase 2) rendered as red. Note that when a face has only one phase, such as the left and top faces in Figure 4.1, the face area of the other (nonpresent) phase is defined as zero.

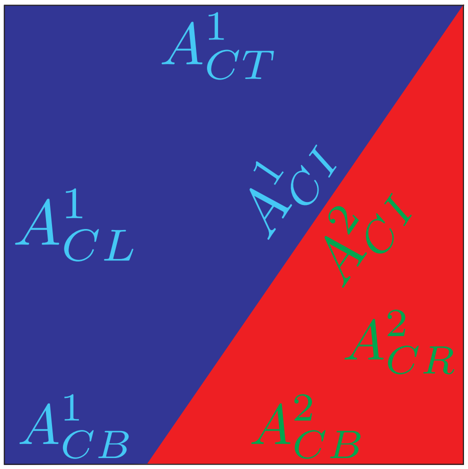


Figure 4.1: Multi-temperature area labels for faces in a typical mixed cell. The superscript indicates the phase, where the blue region is phase 1 and the red region is phase 2.

Certain boundaries violate Equation (4.2), such as when the interface has structure below the resolution of a cell; under the present work, we ignore such cases, i.e., the interfacial structures are sufficiently resolved by the meshes used. However, Equation (4.2) can also be violated if the interface lies along a cell face. This leads to adjacent pure cells of different phase, in which case the areas of different phases are equal,

$$A^1_{C\zeta} = A^2_{\zeta C}. \tag{4.3}$$

This case is quite common and must be accounted for, which leads to two types of flux possible: within the same phase (*intraphase*) and between different phases (*interphase*). Note that the flux across A^1_{CI} is always *interphase*, while flux across a face $A^1_{C\zeta}$ is usually *intraphase*. Similarly, flux between phases within a cell is *intracell* flux, and flux between two cells is *intercell* flux.

There are five directions in which the flux could be directed: the four cardinal directions and normal to the interface. Five directions and two phases yield ten possible fluxes in any given cell,

$$\begin{matrix} F^1_{CL} & F^1_{CR} & F^1_{CT} & F^1_{CB} & F^1_{CI} \\ F^2_{CL} & F^2_{CR} & F^2_{CT} & F^2_{CB} & F^2_{CI} \end{matrix}.$$

Only four to eight of these are nonzero at any given time. Any of these ten fluxes can be interphase, but only F^1_{CI} and F^2_{CI} cannot be intraphase.

4.1.2 Determining the Fluxes

Consider the general system shown in Figure 4.2, which can be considered flux between two phases in different cells, flux between two phases in the same cell, or flux between the same phase in different cells (where the colors would not signify phase). Accordingly, α and β can be the same phase ($\alpha = \beta = 1$ or $\alpha = \beta = 2$) or different phases ($\alpha \neq \beta$). Similarly, A and B can represent adjacent cells or, in the case of a mixed cell, the same cell. The interface ζ between A and B may represent any face (e.g., $\zeta = L$), including the interface in a mixed cell. The flux from the centroid of cell A to the surface ζ between cells A and B is

$$F^\alpha_{A\zeta} = -D^\alpha_A \vec{\nabla} U^\alpha_A \cdot \hat{n}_\perp, \tag{4.4}$$

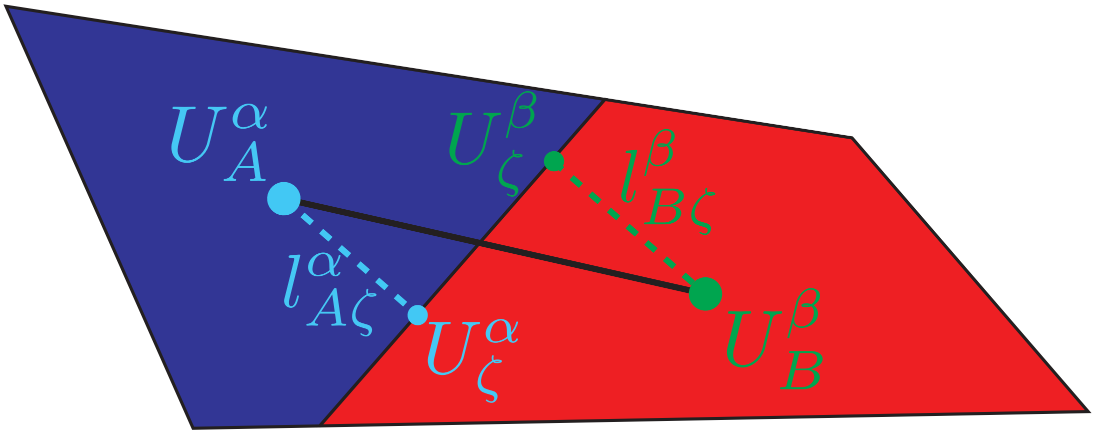


Figure 4.2: Generic example of flux between two quadrilaterals. U_ζ^α and U_ζ^β, fictitious unknowns on the surface ζ, are required to calculate the flux. The superscripts α and β may be equal or distinct phases, and A and B may be the same or neighboring cells. For each phase, the centroid and its normal distance to surface ζ is indicated.

where $\hat{n}_\perp$ is the outward unit normal, which is perpendicular to face ζ. We can discretize the flux as

$$F_{A\zeta}^\alpha \approx -D_A^\alpha \frac{U_\zeta^\alpha - U_A^\alpha}{l_{A\zeta}^\alpha}, \tag{4.5}$$

where the interface temperature U_ζ^α is defined at the minimum distance (normal distance) between U_A^α and the interface ζ. (See Figure 4.3 for a definition of normal distance.) Similarly, the flux from the centroid of cell B to the surface ζ is

$$F_{B\zeta}^\beta \approx -D_B^\beta \frac{U_\zeta^\beta - U_B^\beta}{l_{B\zeta}^\beta}. \tag{4.6}$$

The flux leaving the face of one cell must equal the flux entering the adjacent cell through the same face,

$$F_{A\zeta}^\alpha + F_{B\zeta}^\beta = 0. \tag{4.7}$$

Note that, as was the case in Support-Operators Method (SOM), the fluxes are defined via the outward normal. Therefore, the sums combine to zero rather than the differences.

In order to solve this under-determined system, we make the approximation that the temperatures on a shared surface are equal:

$$U_\zeta^\alpha = U_\zeta^\beta. \tag{4.8}$$

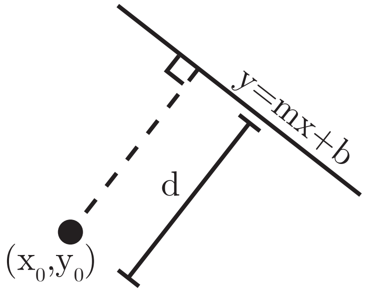


Figure 4.3: Normal (minimum) distance from a point to a line. The minimum distance from point (x_0, y_0) to line $y = mx + b$ is $d = \sqrt{\dfrac{(mx_0 + b - y_0)^2}{1 + m^2}}$.

For ease of notation, let us refer to the interfacial temperature as U_ζ, and let us assume the expressions for flux are exact (so we can use full equal signs). Substituting Equations (4.5) and (4.6) into (4.7) and solving for U_ζ gives

$$U_\zeta = \frac{\dfrac{D^\alpha_A}{l^\alpha_{A\zeta}} U^\alpha_A + \dfrac{D^\beta_B}{l^\beta_{B\zeta}} U^\beta_B}{\dfrac{D^\alpha_A}{l^\alpha_{A\zeta}} + \dfrac{D^\beta_B}{l^\beta_{B\zeta}}}. \tag{4.9}$$

Substituting this expression for the interfacial temperature back into Equation (4.4), we find

$$\begin{aligned} F^\alpha_{A\zeta} &= -\frac{D^\alpha_A}{l^\alpha_{A\zeta}} (U_\zeta - U^\alpha_A) \\ &= -\frac{D^\alpha_A}{l^\alpha_{A\zeta}} \left(\frac{\dfrac{D^\alpha_A}{l^\alpha_{A\zeta}} U^\alpha_A + \dfrac{D^\beta_B}{l^\beta_{B\zeta}} U^\beta_B}{\dfrac{D^\alpha_A}{l^\alpha_{A\zeta}} + \dfrac{D^\beta_B}{l^\beta_{B\zeta}}} - U^\alpha_A \right) \\ &= -\frac{D^\alpha_A}{l^\alpha_{A\zeta}} \frac{D^\beta_B}{l^\beta_{B\zeta}} \frac{U^\beta_B - U^\alpha_A}{\dfrac{D^\alpha_A}{l^\alpha_{A\zeta}} + \dfrac{D^\beta_B}{l^\beta_{B\zeta}}} \\ &= -\frac{U^\beta_B - U^\alpha_A}{\dfrac{l^\alpha_{A\zeta}}{D^\alpha_A} + \dfrac{l^\beta_{B\zeta}}{D^\beta_B}}. \end{aligned} \tag{4.10}$$

The flux from the other side follows the same derivation,

$$F^{\beta}_{B\zeta} = -\frac{U^{\alpha}_{A} - U^{\beta}_{B}}{\dfrac{l^{\alpha}_{A\zeta}}{D^{\alpha}_{A}} + \dfrac{l^{\beta}_{B\zeta}}{D^{\beta}_{B}}}, \tag{4.11}$$

where flux expressions are clearly equal and opposite, $F^{\alpha}_{A\zeta} + F^{\beta}_{B\zeta} = 0$. We note that in the case where the distances are equal and sum to Δx ($l^{\alpha}_{A\zeta} = l^{\beta}_{B\zeta} = \Delta x/2$), the flux $F^{\alpha}_{A\zeta}$ is

$$F^{\alpha}_{A\zeta} = -\frac{2D^{\alpha}_{A}D^{\beta}_{B}}{D^{\alpha}_{A} + D^{\beta}_{B}}\frac{U^{\beta}_{B} - U^{\alpha}_{A}}{\Delta x}, \tag{4.12}$$

which gives an effective diffusivity as the equally weighted harmonic mean of the two diffusivities. This expression gives insight as to why the harmonic mean is used and how it arises.

The general flux expressions of Equations 4.10 and (4.11) can be applied to the three specific cases of flux:

$$F^{1}_{C\zeta} = -\frac{D^{1}(U^{1}_{\zeta} - U^{1}_{C})}{l^{1}_{C\zeta} + l^{1}_{\zeta C}} \qquad \text{Intraphase, intercell flux} \tag{4.13a}$$

$$F^{1}_{C\zeta} = -\frac{U^{2}_{\zeta} - U^{1}_{C}}{\dfrac{l^{1}_{C\zeta}}{D^{1}_{C}} + \dfrac{l^{2}_{\zeta C}}{D^{2}_{\zeta}}} \qquad \text{Interphase, intercell flux} \tag{4.13b}$$

$$F^{1}_{CI} = -\frac{U^{2}_{C} - U^{1}_{C}}{\dfrac{l^{1}_{CI}}{D^{1}_{C}} + \dfrac{l^{2}_{CI}}{D^{2}_{C}}} \qquad \text{Interphase, intracell flux,} \tag{4.13c}$$

with equivalent expressions for the phase 2 fluxes.

4.1.3 Volume-Averaged Diffusion in a Single Cell

The volume average of a cell in the mesh is also known as a control volume, as well as a representative elementary volume. We desire to take the volume average of the diffusion equation,

$$\rho c\frac{\partial u}{\partial t} + \vec{\nabla}\cdot\vec{F} = q, \tag{4.14}$$

where the temperature (u), diffusivity (D), flux ($\vec{F}$), source term (q), and the heat capacity and density (ρc) are multi-phase values. We restrict the number of phases to two for this work, with the knowledge that this can be generalized to n phases. We further restrict the diffusivity, density, and heat capacity to be constant for each phase.

Let the temperature be denoted as u_C^1 and u_C^2, where the superscripts refer to phase 1 and phase 2, respectively. For explicit clarity, any powers of terms in this chapter would be written outside of parenthesis, e.g., $(u^2)^2$. The subscript, which indicates position, is C for all terms because we are working in a single cell. The volume of phase 1 is V_C^1, the volume of phase 2 is V_C^2, and the total volume is $\Delta V = V_C^1 + V_C^2$. Note that all cells have the same total volume, ΔV. We define the volume fraction as

$$f_C^1 = \frac{V_C^1}{\Delta V}. \tag{4.15}$$

Region V_C^2 is non-overlapping with V_C^1, and phase 1 variables are zero in V_C^2 and vice versa. Thus, we have

$$\int_{\Delta V} u_C^1 dV = \int_{V_C^1} u_C^1 dV + \int_{V_C^2} u_C^1 dV = \int_{V_C^1} u_C^1 dV. \tag{4.16}$$

Then, the volume average of u_C^1 in a cell of volume ΔV is

$$\frac{1}{\Delta V}\int_{\Delta V} u_C^1 dV = f_C^1 \frac{1}{V_C^1}\int_{V_C^1} u_C^1 dV. \tag{4.17}$$

Note that phase 1 is equivalent to phase 2 (i.e., we assume they have the same physics in each phase), so any expression can exchange 1 and 2. Consequently, most equations are only written for phase 1.

The volume average of the phase 1 of the diffusion equation is found as follows,

$$\begin{aligned}
\frac{1}{\Delta V}\int_{\Delta V} (\rho c)_C^1 \frac{\partial u_C^1}{\partial t} dV + \frac{1}{\Delta V}\int_{\Delta V} \vec{\nabla}\cdot \vec{F}_C^1 dV =& \frac{1}{\Delta V}\int_{\Delta V} q_C^1 dV \\
(\rho c)_C^1 f_C^1 \frac{\partial}{\partial t}\frac{1}{V_C^1}\int_{V_C^1} u_C^1 dV + \frac{1}{\Delta V}\int_{\delta V_C^1} \vec{F}_C^1 \cdot \hat{n}_{\text{out}} dS =& \frac{f_C^1}{V_C^1}\int_{V_C^1} q_C^1 dV \\
(\rho c)_C^1 f_C^1 \frac{\partial U_C^1}{\partial t} + \frac{1}{\Delta V}\sum_{\zeta=1}^{s} A_{C\zeta}^1 F_{C\zeta}^1 =& f_C^1 Q_C^1,
\end{aligned} \tag{4.18}$$

where s is the number of sides or faces, $\hat{n}_{\text{out}}$ is the outward surface normal, and δV_C^1 is the surface bounding the volume V_C^1. We introduce phase average values U_C^1 and

Q_C^1,

$$U_C^1 = \frac{1}{V_C^1}\int_{V_C^1} u_C^1 dV \tag{4.19a}$$

$$Q_C^1 = \frac{1}{V_C^1}\int_{V_C^1} q_C^1 dV. \tag{4.19b}$$

Note that ζ, while still representing the sides of a cell, has been enumerated (e.g., $\zeta = 1$ corresponds to the face). The most general case of the flux term in Equation (4.18) is

$$\frac{1}{\Delta V}\sum_{\zeta=1}^{s} A_{C\zeta}^1 F_{C\zeta}^1 = \frac{1}{\Delta V}[A_{CL}^1 F_{CL}^1 + A_{CR}^1 F_{CR}^1 + A_{CT}^1 F_{CT}^1 + A_{CB}^1 F_{CB}^1 + A_{CI}^1 F_{CI}^1], \tag{4.20}$$

where the interface term, $A_{CI}^1 F_{CI}^1$, is the intracell, interphase flux between phase 1 and phase 2.

4.1.4 Discretized Volume-Averaged Diffusion Equation

With an expression for flux, Equations (4.10) and (4.11), and the discrete diffusion equation, Equation (4.18), we can now create our system of equations. Using a backward-Euler time discretization, we have

$$(\rho c)_C^1 f_C^1 \frac{U_C^1 - \tilde{U}_C^1}{\Delta t} - \frac{1}{\Delta V}\sum_{\zeta=1}^{s} A_{C\zeta}^1 \frac{U_\zeta - U_C^1}{\frac{l_{C\zeta}^1}{D_C^1} + \frac{l_{\zeta C}}{D_\zeta}} = f_C^1 Q_C^1, \tag{4.21}$$

where $\tilde{U}_C^1$ is the value of U_C^1 from the previous time step, and ζ represents the sides of the cell or the neighbor adjacent to that side of the cell. Terms in cell ζ do not have a phase indicated because either phase is possible. We group the temperature terms for the center cell to obtain,

$$\left(\frac{(\rho c)_C^1 f_C^1}{\Delta t} + \frac{1}{\Delta V}\sum_{\zeta=1}^{s} \frac{A_{C\zeta}^1}{\frac{l_{C\zeta}^1}{D_C^1} + \frac{l_{\zeta C}}{D_\zeta}}\right) U_C^1 = f_C^1 Q_C^1 + \frac{(\rho c)_C^1 f_C^1 \tilde{U}_C^1}{\Delta t} + \frac{1}{\Delta V}\sum_{\zeta=1}^{s} \frac{A_{C\zeta}^1 U_\zeta}{\frac{l_{C\zeta}^1}{D_C^1} + \frac{l_{\zeta C}}{D_\zeta}}. \tag{4.22}$$

This equation can be expressed in a more polished form as

$$\begin{aligned}\xi_C^1 U_C^1 = B_C^1 + \xi_C^{12} U_C^2 &+ (\xi_L^{11} U_L^1 + \xi_R^{11} U_R^1 + \xi_T^{11} U_T^1 + \xi_B^{11} U_B^1) \\ &+ (\xi_L^{12} U_L^2 + \xi_R^{12} U_R^2 + \xi_T^{12} U_T^2 + \xi_B^{12} U_B^2),\end{aligned} \tag{4.23}$$

where we define

$$\xi_\zeta^{12} = \frac{A_{C\zeta}^1}{\Delta V}\left[\frac{l_{C\zeta}^1}{D_C^1} + \frac{l_{\zeta C}^2}{D_\zeta^2}\right]^{-1} \tag{4.24a}$$

$$\xi_\zeta^{11} = \frac{A_{C\zeta}^1 D^1}{\Delta V(l_{C\zeta}^1 + l_{\zeta C}^2)} \tag{4.24b}$$

$$\tilde{\xi}_C^1 = \frac{1}{\Delta t} f_C^1 (\rho c)_C^1 \tag{4.24c}$$

$$B_C^1 = f_C^1 Q_C^1 + \tilde{\xi}_C^1 \tilde{U}_C^1 \tag{4.24d}$$

$$\xi_C^1 = \tilde{\xi}_C^1 + \xi_C^{12} + (\xi_L^{11} + \xi_R^{11} + \xi_T^{11} + \xi_B^{11}) + (\xi_L^{12} + \xi_R^{12} + \xi_T^{12} + \xi_B^{12}), \tag{4.24e}$$

and ζ can represent any cell face or the interface inside a mixed cell, and faces with zero areas make the corresponding ξ terms zero. The equation for phase 2 is found by swapping 1 and 2 in all terms.

There are two options for discretizing an $N \times M$ mesh with two phases: every cell can have both phase temperatures, or cells will only have phase temperatures for the phases that exist in the cell. The former method is easier to implement but is more costly. We use the latter option, where we only place a phase temperature in a cell if that cell contains the associated phase. The result is a smaller system, but the sizes of variables for each phase is not known at runtime. The sizes can only be determined after the volume fractions are determined; thus, we use allocatable variables in our implementation.

Boundary conditions are implemented via ghost cells, a layer of dummy cells surrounding the mesh whose values are specified in terms of the Robin boundary conditions,

$$b_1 U + b_2(\vec{\nabla} U, \hat{n}) = b_3. \tag{4.25}$$

Using U_G for the boundary (ghost) temperature, U_I for the interior temperature, and

d for the distance between them, we find that

$$b_1 U_G + b_2 \frac{U_G - U_I}{d} = b_3 \tag{4.26a}$$

$$\left(b_1 + \frac{b_2}{d}\right) U_G = b_3 + \frac{b_2 U_I}{d} \tag{4.26b}$$

$$U_G = \frac{b_3 d + b_2 U_I}{b_1 d + b_2}. \tag{4.26c}$$

We form a system of equations,

$$\mathbf{A}\vec{U} = \vec{B}, \tag{4.27}$$

where the elements of $\vec{B}$ are defined as in Equation (4.24d). The global system, $\mathbf{A}$, is an SPD matrix. We solve this system using the conjugate gradient method. The output is the cell-centered temperature for pure cells and the centroid-centered temperatures for mixed cells. The ordering of the elements in the temperature vector is a free choice, however, the ordering must remain consistent once a choice has been made. We choose to define all the phase 1 temperature values first, followed by the phase 2 temperature values. Each row in a matrix contains at most ten nonzero elements, as shown in Equation (4.23). A cell in the mesh uses a row/column (i, j) notation to refer to each cell uniquely.

The error on a test problem can be computed in two ways. The phase-separate computes a centroid-centered error for each phase. The combined-phase error computes a cell-centered error by taking the volume weighted arithmetic mean,

$$U_C^{\text{average}} = f_C^1 U_C^1 + f_C^2 U_C^2. \tag{4.28}$$

4.2 Three Multi-Temperature Models

We introduce three mixed-cell models using the framework of the phase-averaged diffusion equation outlined in the previous section. The models, referred to as the M1, M2, and M3 models, account for progressively more accurate approximations of the subgrid dynamics. The distinguishing feature between the models is the way in which distance is defined, which is used for flux calculations.

The flux between phase 1 of cell C and generic phase α of cell ζ is defined as

$$F^1_{C\zeta} = -\frac{U^\alpha_\zeta - U^1_C}{\dfrac{l^1_{C\zeta}}{D^1_C} + \dfrac{l^\alpha_{\zeta C}}{D^\alpha_\zeta}}. \tag{4.29}$$

Intraphase flux ($\alpha = 1$) only occurs for intercell flux. Interphase flux ($\alpha = 2$) occurs both inside mixed cells (intracell) and when there are different phases on either side of a cell face (intercell).

Equation (4.29) requires two distances to compute the flux, the normal distance from each centroid to the face. However, the M1 and M2 models only approximate the total distance,

$$l_{C\zeta} = l^1_{C\zeta} + l^\alpha_{\zeta C}. \tag{4.30}$$

The individual distances, $l^1_{C\zeta}$ and $l^\alpha_{\zeta C}$, must be approximated for the M1 and M2 models in terms of the volume fractions (f^1_C and f^α_ζ) and total distance ($l_{C\zeta}$). The approximation we use for the M1 and M2 models is

$$l^1_{C\zeta} = \frac{f^1_C}{f^1_C + f^\alpha_\zeta} l_{C\zeta} \tag{4.31a}$$

$$l^\alpha_{\zeta C} = \frac{f^\alpha_\zeta}{f^1_C + f^\alpha_\zeta} l_{C\zeta}, \tag{4.31b}$$

where ζ can be a neighbor or cell C in the case of mixed cells.

We can plug these distances into Equation (4.29), which gives us

$$F^1_{C\zeta} = -\left(\frac{f^1_C + f^\alpha_\zeta}{\dfrac{f^1_C}{D^1_C} + \dfrac{f^\alpha_\zeta}{D^\alpha_\zeta}} \right) \frac{U^\alpha_\zeta - U^1_C}{l_{C\zeta}}, \tag{4.32}$$

where the only distance term is the combined distance, and the term in parenthesis is the weighted harmonic mean. For intraphase flux, $\alpha = 1$, the assumption of constant diffusivities for each phase allows the diffusivities to be factored out, and the flux expression reduces to

$$F^1_{C\zeta} = -D^1 \frac{U^1_\zeta - U^1_C}{l_{C\zeta}}, \tag{4.33}$$

which is the standard flux approximation.

We now specify how each model approximates the distances. This is the main difference between the models. The only other difference is how the surface areas are specified, where the M1 model is different from the M2 and M3 models. Figure 4.4 shows the difference between distance definitions. The M1 model defines the total distance as the cell center spacing (black lines). The M2 model defines the total distance as the centroid spacing (dashed lines). The M3 model uses the normal distance between the centroid and the interface.

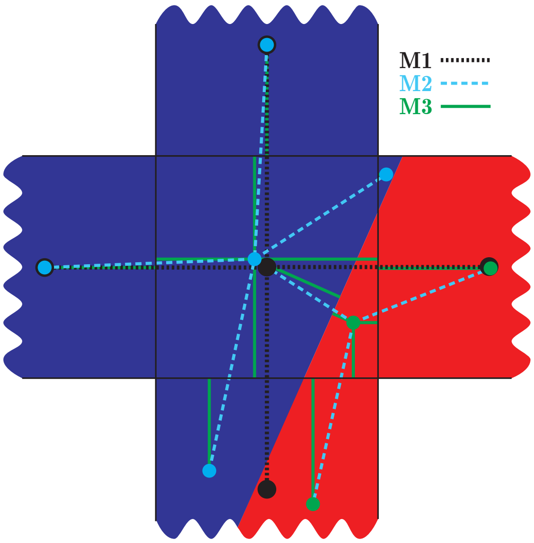


Figure 4.4: Distance definitions for multi-temperature models. The M1 model uses cell-centered spacing (black dotted lines), the M2 model uses centroid spacing (cyan dashed lines), and the M3 model uses the normal distance between each centroid and face (green solid lines). Cell centers are indicated with black circles, phase 1 centroids are indicated with cyan circles, and phase 2 centroids are indicated with green circles. Note that parts of the neighboring cells are truncated.

4.2.1 M1 Model

Using the local volume fractions (a cell and its nearest neighbors) and an assumption of a linear interface, the surface fractions and interface normal can be approximated as follows. The surface fractions, $a^1_{C\zeta}$, in the M1 model, are approximated as $a^1_{C\zeta} = \frac{1}{2}(f^1_C + f^1_\zeta)$, where f^1_ζ is the volume fraction of phase 1 in cell ζ and f^1_C is the volume fraction of phase 1 in cell C. However, if either cell is a pure cell (i.e., $f^1 = 1$), the surface fraction is unity. The surface fraction for phase 2 complements that of phase 1 (i.e., $a^1_{CL} + a^2_{CL} = 1$). The surface area for a particular phase is the surface fraction multiplied by the area of the cell face (e.g., for a left face, the area is $A^1_{CL} = a^1_{CL}\Delta y$). The surface area of the interface within a mixed cell can be approximated from the surrounding face areas, $A_{CI} = \sqrt{(A^1_{CL} - A^1_{CR})^2 + (A^1_{CT} - A^1_{CB})^2}$. The angle of the phase interface, θ, is readily approximated from the face areas.

The M1 model makes a simple assumption that the centroid of each phase is located at the cell center for calculating the distance for flux between cells. Thus, the distance between any two adjacent cells is either Δx or Δy. However, for flux within a mixed cell, this same assumption results in a distance of zero between two phases. This is undesirable because these distances occur in the denominator of the flux term, and infinite flux within mixed cells would not allow unique phase temperatures. Consequently, a second assumption of the M1 model is required to give the distances between the phases in a mixed cell a nonzero value. A distance of $\Delta x/2$ is reasonable for a vertical interfaces, as is a distance of $\Delta y/2$ for a horizontal interface. Thus, the distance between phases in a mixed cell is approximated for any interfacial orientation as $\frac{1}{2}\sqrt{\Delta x^2 \cos^2\theta + \Delta y^2 \sin^2\theta}$. This expression for distance works in the two limiting cases of horizontal and vertical interfaces, but may over or underestimate for intermediate angles, depending on the volume fractions. In summary, the approximated distance of the M1 model are given as

$$l^1_{CL} = l^1_{CR} = l^2_{CL} = l^2_{CR} = \tfrac{1}{2}\Delta x \tag{4.34a}$$

$$l^1_{CT} = l^1_{CB} = l^2_{CT} = l^2_{CB} = \tfrac{1}{2}\Delta y \tag{4.34b}$$

and

$$l^1_{CI} + l^2_{CI} = \tfrac{1}{2}\sqrt{\Delta x^2 \cos^2\theta + \Delta y^2 \sin^2\theta}. \tag{4.35}$$

The advantage of the M1 model is its simplicity, especially for implementation. The face areas and interface normal are estimated from only the volume fractions, and they are sufficient in specifying the distances. The diffusivity for a flux between phases within a mixed cell uses the harmonic mean, weighted by volume fractions, Equation (4.32).

4.2.2 M2 Model

The M2 and M3 models approximate the centroid of each phase in every cell. For pure cells, the centroid coincides with the cell center; more mixed cells, the centroids are calculated for each phase. Using only the volume fractions and an assumption of a linear interface, the centroids (as well as the surface fractions and face normals) can be computed for a 2D mesh. The volume fractions in a 3×3 stencil are used to distinguish the different cases of how the interface divides the cell and the orientations of the interface. (See Figure 3.2 for examples.) By doing so, each subcell region occupied by a phase becomes a polygon with three, four, or five vertices. We use these vertices to calculate the face areas and centroids of these polygons. Note that this procedure is only accurate for a linear interface. There are more advanced interface reconstruction algorithms (see Dyadechko and Shashkov [68] for a review); however, the current implementation is sufficient for the work presented in this dissertation.

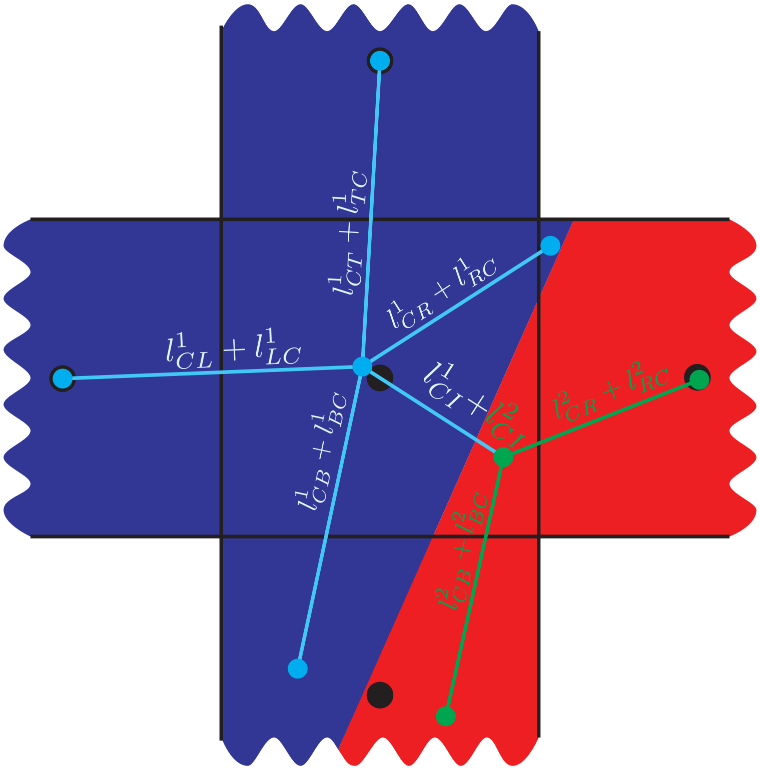


Figure 4.5: The M2 model for mixed cells in the multi-temperature method. Distances are computed from centroid to centroid in this model. Phases are indicated by background colors (phase 1 is blue, phase 2 is red), and all distances needed for this center cell are labeled for phase 1 (white) and phase 2 (green). Note that parts of the neighboring cells are truncated.

The M2 model defines distances between two values as the centroid-to-centroid distance, shown in Figure 4.5. This is significantly more accurate than the M1 model, albeit at the cost of a more complicated algorithm (such as distinguishing the cases in Figure 3.2 and their proper orientation). Only the total centroid-to-centroid distances are found in this model, therefore, only the distance sums are shown in Figure 4.5. The flux is calculated using Equation (4.32). However, since only flux normal to an interface is continuous physically, this model is flawed, e.g., the fluxes to the right in Figure 4.5 where it is clear these are oblique to the faces.

4.2.3 M3 Model

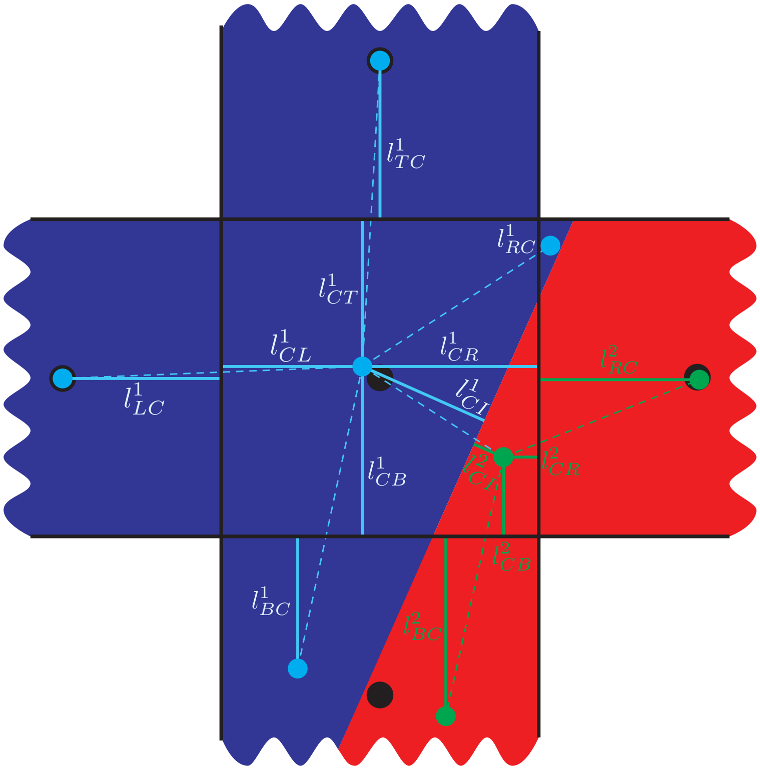


Figure 4.6: M3 model for mixed cells in the multi-temperature method. Distances are specified as the normal distance from each centroid to the appropriate face, which makes the flux normal to a face continuous. Phases are indicated by background colors (phase 1 is blue, phase 2 is red), and all distances needed for this center cell are labeled for phase 1 (white) and phase 2 (green). Solid lines represent these distances while dashed lines represent the centroid-to-centroid distances of the M2 model. Note that parts of the neighboring cells are truncated.

The M3 model, which also locates the centroids, accounts for the oblique intersections that are neglected in the M2 model. Physically, only the flux normal to an interface is continuous; therefore the M3 method defines the distances as the normal (minimum) distance from the centroid to a face, as shown in Figure 4.6. (See Figure 4.3 for a precise definition of normal distance.) Accordingly, the individual distances are defined in this model, unlike the M1 and M2 models. Thus, Equation (4.29) is used for flux, which is more accurate than Equation (4.32), which is used for both the M1 and M2 models.

4.3 Discussion

The main advantage of the multi-temperature method is that it allows each phase in a mixed cell to be treated separately. Physically, adjacent phases not in equilibrium may have unique temperature with a heat flux from the warmer to the cooler material. In standard methods, a mixed cell is treated as a single entity with a set of effective material properties obtained from a combination of the properties of each phase. In the multi-temperature model, the fluxes in a mixed cell are computed separately for each phase. Thus, homogenization is avoided while maintaining an orthogonal grid. Consequently, the multi-temperature model maintains the simplicity of uniform meshes while achieving a higher degree of physical accuracy.

However, although the overall mesh structure is uniform, the differencing in the mixed cells effectively introduces a cut cell, which gives a non-uniform mesh size near the interface. Because the change in cell width is discontinuous (as opposed to smoothly varying) and because the subcell zones are on the order of the original cell (i.e., $f\Delta x$ is $\mathcal{O}(\Delta x)$), the multi-temperature model introduces a first-order error [93]. The upshot of this method is that the physics is more accurate for the mixed cells. Thus, we expect the multi-temperature models to have lower error than a single-temperature model for rough grids while having a lower convergence rate. Consequently, the multi-temperature models will begin with a lower error, but the single-temperature models will become more accurate with sufficient refinement. We will look for when the multi-temperature models attains second-order convergence, but the better measure of the success of these models occurs by considering the error on coarse grids compared to a single-temperature model's error at the same refinement.

The models are *implemented* with the assumption of a linear interface. However, the models themselves do not *require* an assumption of a linear interface. Therefore, the multi-temperature models could be implemented with more advanced interface tracking or interface reconstruction in order to increase the accuracy of the simulations.

Since the M2 and M3 models require the centroid to be computed for each phase in each cell, a next step in the development of this method would be to integrate with a scheme that naturally tracks centroids, such as the Moment of Fluid (MoF) method [2, 3, 6, 68, 69, 70]. The MoF method is the Volume of Fluid (VoF) method with one additional complexity. The VoF method tracks the zeroth volumetric moment of position (volume), while the MoF method tracks both the volume and the first volumetric moment (centroid). In addition, when multiple phases are present (greater than two), the MoF naturally handles the problem of ordering the phases, which is an issue with multi-interface reconstruction. Interface reconstruction methods for greater than two phases yield different results depending on the order they reconstruct the interface for each phase. The MoF method compares the results from each ordering possibility and determines which is best, thus leading to accurate interface reconstruction in multiphase systems.

The multi-temperature methods could be coupled with a fluid dynamics scheme that provides the volume fractions for an orthogonal grid. An example of a such a method is Arbitrary Lagrangian Eulerian (ALE) method. The fluid advection is performed using a Lagrangian technique, where the mesh moves with the fluid. After each time step, the irregular mesh is projected onto an orthogonal mesh (which creates mixed cells) and the fluid state updated (e.g., chemical diffusion or heat conduction). The multi-temperature model could be implemented during the Eulerian cycle of an ALE method, calculating the properties of each phase separately. The results for each phase in each cell could be used in the subsequent Lagrangian cycle in order to advect the fluid, all the while avoiding any homogenization.

Although the work presented involving the multi-temperature methods is restricted to a linear interface and two phases, the models have the potential to be applied to n phases with nonlinear interfaces. Therefore, the multi-temperature models are not limited to the cases presented, but rather, these are a proof-of-concept of this mixed-cell approach.

4.4 Summary

We have developed a multi-temperature method to calculate temperature in heat conduction problems with two phases, where the method may be generalized to n phases. This method allows each material in a mixed cell to have a unique temperature, which is more physically accurate. By integrating over the volume of a cell and solving the problem via surface fluxes, this FDM is technically a FVM on an orthogonal grid.

Specification of certain parameters that compose the flux, notably distances required to approximate the gradient, culminated in three different multi-temperature models. The M1 model simply assumes that the spacing between neighboring cells is equal to the grid spacing, while the distance for the exchange flux within a mixed cell is half the grid spacing modified by the interfacial orientation. The M2 model, while more accurate, requires the location of the centroids for each phase existing in every cell. The centroid separation is used to define the distances for in the flux calculation. The M3 model makes improvements from the M2 model by computing only flux normal to a surface or interface, calculating the normal distance from each centroid.

These three models may be implemented using a variety of numerical techniques with potential applications to two- and multi-phase flow. We use simple finite differences to implement these models, although a modified SOM could be used in the future. These models are evaluated in Chapters VI-VIII using numerical tests, with comparisons made between the standard mixed-cell model (the harmonic-mean model (S1) described in the previous chapter) to determine if these methods are advantageous.

CHAPTER V

Code Verification

This chapter summarizes the verification of the single- and multi-temperature codes using the Method of Exact Solutions (MES) and the Method of Manufactured Solutions (MMS). Technically, a code can never be fully verified, but rather evidence can be collected that suggests a code has the correct behavior. Consequently, when we use the word verify, we mean to state that a certain set of tests has given evidence that particular aspects of a code are functioning as expected, and we assume these aspects are trustworthy. The purpose of this chapter is to demonstrate the reliability of the codes which implement the models for a standard, pure-cell problem prior to their applications to mixed-cell problems.

In Section 5.1, we discuss our definition of code verification in more detail. In Section 5.2, we define p-norms, error, and convergence rates. We present the results of four numerical tests employed to verify various aspects of the single-temperature and multi-temperature code for pure-cell cases in Section 5.3. We conclude with a discussion in Section 5.4.

5.1 Definition of Code Verification

Verification (of a code that solves a set of equations) checks that the governing equations are solved consistently, often examining the order-of-accuracy against the accuracy of the discretization method [189]. Code verification does not ensure that the equations are solved with the most efficient numerical methods, nor that the computed solutions describe the system being modeled [189]. Model *validation* examines whether the governing equations selected to be solved sufficiently model the physical system [189]. We consider only *verification* and not validation here.

There are many ways to verify a code. Much of this section is based on Salari and Knupp [189], which gives a thorough discussion and review of code verification techniques. Some of the less rigorous methods are trend testing and symmetry testing. In trend testing, one varies input parameters in order to qualitatively check the behavior of the code, i.e., observe trends. Symmetry testing may involve setting up a problem with a known symmetric solution to verify the results are symmetric, or checking that translations and rotations of an asymmetric solution give expected results. Both trend and symmetry testing give only qualitative results and require expert judgment — review by a person with experience and knowledge of the correct solution behavior. A method that is slightly more quantitative is comparison testing, where a code is compared with a verified code that solves the same problem. Issues with the comparison method include the fact that different codes rarely solve identical systems, and that comparable codes that are verified may not exist or may not be accessible.

One of the most widely used and quantitatively sound methods to verify codes is the Method of Exact Solutions (MES) [189]. In MES, computational results are compared with an analytic solution, a closed-form mathematical expression for the solution at every position and time in the domain. Such solutions can be found in literature or derived via solution methods such as separation of variables (SoV) or integral transforms. One issue with this method is that the analytic solution is often less general than the code's capabilities (e.g., being of lower dimension, requiring homogeneous coefficients, or specifying a simple domain), meaning that certain aspects of the code will not be tested. This lack of comprehensiveness in testing a code with a single analytic solution is often addressed by testing against a suite of analytic solutions [189]. Some important issues with MES are that a solution often (i) does not exist for the system, (ii) is difficult to numerically implement, or (iii) does not test critical aspects of the code. See Chapter VIII for an example of a MES that is difficult to implement due to the solution containing special functions and infinite sums.

The Method of Manufactured Solutions (MMS) is a more flexible and often simpler alternative to the MES. A closed form solution U is created (manufactured) by the user. The solution U can be anything, but it is most useful if it is written in terms of easily computed functions (e.g., sine, logarithms, exponentials) as well as polynomials of the time or spatial variables. A solution can be created that is general enough to exercise many (if not all) of the code's capabilities, including those that cannot be verified quantitatively by MES or other means. The solution U is then operated on by the differential operator (L), resulting in a source term (Q). For the diffusion equation, this process can be mathematically described as

$$Q = \mathsf{L}[U], \tag{5.1}$$

where

$$\mathsf{L}[U] = \frac{\partial U}{\partial t} - \vec{\nabla} \cdot \mathbf{D} \vec{\nabla} U. \tag{5.2}$$

This source term (Q) is implemented into the code, effectively driving the code to the chosen solution U. The source term may be complicated, however algebraic programs such as *Maple* and *Mathematica* can compute the source term and output the results in proper FORTRAN or C syntax. The best choices for domain size/shape and boundary conditions would yield a solution that tests the most aspects of a code, or instead, that tests aspects that are not otherwise tested. Despite this flexibility in choosing the domain and boundary conditions, all MMS problems presented in this thesis are computed on a square domain and use Dirichlet boundary conditions that are set by the appropriate analytical solution. The discretization error of a code can then be computed by comparing the code's calculated solution (u) to the known exact solution (U).

5.2 Discretization Error & Convergence Rate

5.2.1 Discretization Error & Its Quantification

Discretization is the process of converting a continuous domain into a discrete domain, subdividing the domain into finite-sized cells. The difference between the continuum and discrete solutions is called the discretization error. Higher resolution (smaller cells) should always yield lower error, which is termed the 'convergence' of the numerical solution. A time dependent equation, such as the diffusion equation, is discretized in both space and time. Therefore, it has spatial as well as temporal discretization errors.

We choose to measure errors in terms of the p-norm, defined as

$$\|x\|_p = \left(|x_1|^p + |x_2|^p + |x_3|^p + \cdots + |x_n|^p\right)^{\frac{1}{p}}, \tag{5.3}$$

where x is a discrete n-element vector, $p \geq 1$, and $|x_i|$ is the absolute value of the i^{th} element. For a 1D solution with N cells in a unit length domain, the error between an exact solution (U) and an approximate solution (u) is defined as

$$\|U-u\|_p = \left[\sum_{i=1}^{N} |U_i - u_i|^p \frac{1}{N}\right]^{\frac{1}{p}}. \tag{5.4}$$

For a 2D solution on $N \times M$ cells, the error measured in the p-norm is defined as

$$\|U-u\|_p = \left[\sum_{i=1}^{N}\sum_{j=1}^{M} |U_{i,j} - u_{i,j}|^p V_{i,j}\right]^{\frac{1}{p}}, \tag{5.5}$$

where $V_{i,j} = \frac{1}{NM}$ for a uniform mesh on the unit square.

We use three p-norms as metrics for the error analysis in this thesis: $p \in \{1, 2, \infty\}$. The p-norms for these three values are

$$\|U-u\|_1 = \sum_{i=1}^{N}\sum_{j=1}^{M} |U_{i,j} - u_{i,j}| V_{i,j} \tag{5.6a}$$

$$\|U-u\|_2 = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{M} (U_{i,j} - u_{i,j})^2 V_{i,j}} \tag{5.6b}$$

$$\|U-u\|_\infty = \max_{i,j} |U_{i,j} - u_{i,j}|. \tag{5.6c}$$

However, we use the relative error to specify results, which is the usual p-norm of the difference $(U - u)$ divided by the p-norm of the exact solution (U) [129, 151, 152]. Accordingly, our three error metrics are

$$E_1 = \frac{\sum_{i,j} |U_{i,j} - u_{i,j}|}{\sum_{i,j} |U_{i,j}|} \tag{5.7a}$$

$$E_2 = \sqrt{\frac{\sum_{i,j} (U_{i,j} - u_{i,j})^2}{\sum_{i,j} (U_{i,j})^2}} \tag{5.7b}$$

$$E_\infty = \frac{\max_{i,j} |U_{i,j} - u_{i,j}|}{\max_{i,j} |U_{i,j}|}, \tag{5.7c}$$

where the summations have been combined and the summation limits suppressed for brevity. The volume factor cancels since we are using a uniform grid and $V_{i,j} = \Delta x \Delta y$, which can be factored from the sum. For the multi-temperature methods, the dividing through by the volume factor leads to subcells with different volumes and thus to the appearance of volume fractions in the error expressions. Thus, the three relative error metrics for phase 1 are

$$E_1^1 = \frac{\sum_{i,j} |U_{i,j}^1 - u_{i,j}^1| f_{i,j}^1}{\sum_{i,j} |U_{i,j}^1| f_{i,j}^1} \tag{5.8a}$$

$$E_2^1 = \sqrt{\frac{\sum_{i,j} (U_{i,j}^1 - u_{i,j}^1)^2 f_{i,j}^1}{\sum_{i,j} (U_{i,j}^1)^2 f_{i,j}^1}} \tag{5.8b}$$

$$E_\infty^1 = \frac{\max_{i,j} |U_{i,j}^1 - u_{i,j}^1| f_{i,j}^1}{\max_{i,j} |U_{i,j}^1| f_{i,j}^1}, \tag{5.8c}$$

where both the calculated and analytical solutions are computed at the centroid of each cell i, j. Note that the volume fractions do not cancel out, although many are zero or one, and the expressions for the phase 2 error are found by substituting 2 in for the phase superscript.

5.2.2 Error Convergence & Its Rate

The rate at which the error reduces as the resolution increases is known as the convergence rate, q. The standard assumption is that the error resulting from spatial discretization scales directly with the cell width, h, as

$$\|E(h)\| \approx Ch^q + \mathcal{O}(h^{q+1}), \tag{5.9}$$

where C is the convergence rate constant [56, 99, 151, 152, 189, 197, 198]. For a 2D solution with a grid $N \times M$, the grid spacing (h) dependence of the error is dominated by the larger direction for isotropic systems,

$$h = \max\left(\frac{1}{N}, \frac{1}{M}\right). \tag{5.10}$$

Dividing error values at two grid sizes (h_1 and h_2) gives

$$\frac{\|E(h_1)\|}{\|E(h_2)\|} \approx \frac{Ch_1^q + \mathcal{O}(h_1^{q+1})}{Ch_2^q + \mathcal{O}(h_2^{q+1})}. \tag{5.11}$$

Solving for q, after dropping the terms with powers greater than q, gives

$$q \approx \frac{\log \frac{\|E(h_1)\|}{\|E(h_2)\|}}{\log \frac{h_1}{h_2}} = \frac{\log \|E(h_1)\| - \log \|E(h_2)\|}{\log h_1 - \log h_2}. \tag{5.12}$$

A convergence rate is always compared between two spatial resolutions. However, the convergence rate is not necessarily the same between different sets grid resolutions. Therefore, in the tests presented in this thesis, we compare the convergence for a series of spatial resolutions, with the grid spacings listed as $h_1 \rightarrow h_2$ for each pair of resolutions.

For convergence studies, the grid sizes are often chosen to increase by factors of two in each refinement, such that $\frac{h_1}{h_2} = 2$, which gives (via the logarithm change of base) the simpler form of the convergence rate,

$$q \approx \log_2 \frac{\|E(h)\|}{\|E(h/2)\|}. \tag{5.13}$$

However, we use Equation (5.12) to maintain the flexibility in the manner resolution refinements are made. Note that there is a temporal convergence rate as well, where Δt takes the place for h, but we focus on the spatial convergence.

Since the error at the interface is the largest error in the domain and since we are focused on treating this interfacial error, the error metric that emphasizes the largest discrepancy is the best metric for mixed-cell analysis. The error characterized by the ∞-norm is a better measure of the mixed-cell error than any other norm because the it calculates only the least accurate cell (which is a mixed cell for the problems tested). Thus, the E_∞ is the most stringent measure of mixed-cell accuracy. E_1 and E_2 are included for completeness and additional information; however the figures and discussion will include E_∞ alone. Appendix C contains any E_1 and E_2 information not shown in the main text.

The phrase 'convergence rate' in this work will always refer to the convergence of the relative error, and phrase 'order of accuracy' is synonymous with 'order of convergence.' Thus, the accuracy of a method is measured using the convergence rate.

5.3 Pure-Cell Test Problems

We present four test problems. The first two are time dependent with no source terms. The first test uses MES with a constant scalar diffusivity. The second test uses MMS with a scalar diffusivity that changes with position. The third and fourth tests are steady-state solutions created by MMS. The third test has a constant tensor diffusivity, which applies to the single-temperature code. The fourth test is 1D with a single interface. This interface test is performed for a pure-cell case here, while a mixed-cell version of this test is performed in Chapter VI.

5.3.1 Test 1: MES with Fundamental Solution

The first test employs what is known as the fundamental solution, an analytic solution found by using a Green's function on an infinite domain. This test demonstrates that the code correctly solves the time-dependent diffusion equation. The problem is heat conduction on an infinite plane of homogeneous diffusivity, D, initially with a unit heat source concentrated at the origin. In 2D, the solution is

$$U(x,y,t) = \frac{t_i}{t} \exp\left(-\frac{x^2+y^2}{4Dt}\right), \tag{5.14}$$

where D is the diffusivity (we use $D = 1$ for this test), and t_i is the initial time, which must be greater than zero when implemented in order to avoid a singularity (t_i=0.01 is used). The solution decays to zero as the distance from the origin approaches infinity. The computational domain, however, is finite, and zero Dirichlet boundary conditions are used. The simulation is terminated before the heat wave from the impulse is affected by the boundary conditions. For a domain of $x, y \in [-2, 2]$, a maximum time of $t = 0.11$ is sufficient to avoid boundary effects. This is determined by ensuring the total amount of energy remains constant (i.e., none is lost due to boundary effects). Unlike the single-temperature code, the multi-temperature code requires an interface defined. We define this interface outside of the computational domain, reducing the multi-temperature code to standard finite difference methods (FDM).

	Single-T			Multi-T		
grid size	q_1	q_2	q_∞	q_1	q_2	q_∞
20→40	2.33	2.38	2.33	2.34	2.38	2.41
40→60	2.19	2.22	2.21	2.19	2.22	2.23
60→80	2.16	2.19	2.19	2.16	2.19	2.20
80→100	2.11	2.14	2.14	2.11	2.14	2.15
mean	2.20	2.23	2.22	2.20	2.23	2.25

Table 5.1: Convergence rates for the first pure-cell verification problem. For both the single-temperature and multi-temperature codes, the convergence rate is calculated for three p-norms. The leftmost column gives the two grid sizes between which the convergence rate is being calculated.

Table 5.1 lists the time-averaged convergence rates for this test problem. The solution is obtained at ten equally spaced times in the interval from $t_{\min} = 0.01$ to $t_{\max} = 0.11$. For the time range of this test, the difference in error for a given grid resolution is insignificant (i.e., the convergence rates have little variance with time for pair of grids). Thus, we present the results after averaging the ten outputs produced over the duration of the simulation. Both the single-temperature and multi-temperature codes achieve better than second-order accuracy in all three error norms at every sample time.

The single-temperature and multi-temperature models are equivalent for this problem because there are no mixed cells. The only difference between the multi-temperature and single-temperature codes for this test is the numerical method for solving the differential equation: SOM for the single-temperature code and FDM for the multi-temperature code. SOM gives comparable accuracy to FDM for this problem, which is expected because the grid is rectangular and the diffusivities are constant. SOM is only advantageous compared to FDM for skewed grids and/or discontinuous coefficients.

This test demonstrates that both codes solve the time-dependent heat equation (diffusion equation with unit diffusivities) with second-order accuracy.

5.3.1.1 Multi-temperature Numerical Issue

We note some interesting solution behavior for the multi-temperature code when the interface lies within the computational domain while setting the diffusivities of each phase equal. Table 5.2 shows the results of this test problem with seven different interface locations, one outside the domain, three along an axis of symmetry, and three asymmetric interfaces. The domain is $x, y \in [-2, 2]$, and the solution has circular symmetry from the origin. Therefore, any linear interface that crosses through the origin will lie on an axis of symmetry. We find that the results are second-order accurate when the interface is outside the domain or on an axis of symmetry ($y = 0$, $y = x$, $y = 2x$). When the interface is elsewhere ($y = x+1$, $y = -2x+0.3$, $y = 2x+0.3$) the convergence rate reduces to as low as $q_1 = 1.2$. We demonstrate this behavior using the test problem of this section with the M1 model, but this behavior was observed in any problems when equal diffusivities were set on two phases. Therefore, this is an inherent artifact appearing in the code.

The multi-temperature code only recovers a second-order, single-temperature FDM solution in certain cases when introducing an interface between equivalent phases. While this behavior is worrisome, it may or may not represent a significant flaw. It is

Interface	q_1	q_2	q_∞
Outside	2.34	2.38	2.41
y=0	2.34	2.38	2.41
y=x	2.34	2.38	2.41
y=2x	2.34	2.37	2.41
y=x+1	1.85	1.84	1.80
y=-2x+0.3	1.19	1.18	1.36
y=2x+0.3	1.19	1.18	1.36

Table 5.2: Results for the multi-temperature code for the first pure-cell verification problem showcasing the interface-dependent convergence rates. The interfaces fall into three groups: those outside the computational domain, those along an axis of symmetry (line through the origin for this problem), and those not along an axis of symmetry. The convergence rate is calculated for three p-norms between square grids of size 10 and 20 with the M1 model.

possible that the multi-temperature model simply cannot recover the single temperature model due to the dividing and subsequent differencing of the mixed cell. The first-order error introduced by differentiating on a subgrid mesh may manifest with these asymmetric interfaces while canceling otherwise. However, this behavior may instead represent a problem in the discretization or implementation of the model. This problem would not manifest in 1D but may skew the 2D results. Further study is needed to fully explore this issue.

5.3.2 Test 2: MMS with Spatially Dependent Diffusivity

The second problem is a MMS test problem that showcases a spatially dependent diffusivity. It is necessary to demonstrate that the correct solution is obtained when the diffusivity has a spatial dependence because the numerical test presented in Chapter VIII uses a nonlinear interfacial geometry, which leads to a spatially dependent diffusivity near the interface. This test, adopted from Salari and Knupp [189], was designed to be a time-dependent problem with a zero source term. The solution and diffusivity are [189]

$$U(x,y,t) = e^x \cos(y) \exp\left(-\frac{t}{\rho c}\right) \tag{5.15a}$$

$$D(x,y) = e^x \sin(y) - x. \tag{5.15b}$$

Note that a physics-based assumption necessary for the code is that D must be positive. As a result, we shift the domain from the standard unit square to $x \in [-2,-1]$ and $y \in [0,1]$.

	Single-T			Multi-T		
grid size	q_1	q_2	q_∞	q_1	q_2	q_∞
20→40	1.98	1.98	1.93	2.05	2.15	2.20
40→60	2.00	2.00	1.97	2.04	2.09	2.12
60→80	2.00	2.00	1.98	2.02	2.05	2.08
80→100	2.00	2.00	1.98	2.03	2.06	2.08
100→200	2.00	2.00	1.99	2.00	2.02	2.03
mean	2.00	1.99	1.97	2.03	2.07	2.10

Table 5.3: Convergence rates for the second pure-cell verification problem. For both the single-temperature and multi-temperature codes, the convergence rate is calculated for three p-norms.

As in the first test, this is a time-dependent problem, and the solution error is measured at ten equally spaced intervals during the simulation time from $t = 0$ to $t = 0.1$. We find that the errors at a particular grid size remain effectively constant with time (variance below 10^{-12} with errors between 10^{-5} and 10^{-7}) for the range of time examined. Therefore, the convergence rates between two particular grid sizes are also effectively constant with time. The convergence rates for both codes, averaged over all ten times, are shown in Table 5.3. Both problems are second-order accurate in all three error norms, although the multi-temperature code has slightly higher accuracy in all cases (2-6% higher convergence rates).

This test demonstrates that both codes solve a variable diffusivity time-dependent scalar diffusion problem with second-order accuracy.

5.3.3 Test 3: MMS with Anisotropic Diffusivity

The third test, generated by MMS, verifies the tensor diffusion capabilities of the single temperature code. Verification of the tensor capabilities of the code is highly desirable because one of the single temperature mixed-cell models (S2) uses a tensor approach. The multi-temperature code is not evaluated with this test because it cannot handle tensor diffusivities. Since we have two time dependent tests already, this is designed to be a steady-state problem. Choosing a solution of

$$U(x,y) = x\sin(2\pi y) + e^y \ln(x+1), \tag{5.16}$$

with a generic constant diffusivity tensor,

$$\mathbf{D} = \begin{bmatrix} D^{xx} & D^{xy} \\ D^{xy} & D^{yy} \end{bmatrix}, \tag{5.17}$$

the source term is

$$Q(x,y,t) = \frac{D^{xx}e^{y}}{(x+1)^2} + D^{yy}\left(4\pi^2 x\sin(2\pi y) - e^{y}\ln(x+1)\right) - 2D^{xy}\left(2\pi\cos(2\pi y) + \frac{e^{y}}{x+1}\right). \tag{5.18}$$

Note that $\mathbf{D}$ must be symmetric physically ($D^{xy} = D^{yx}$) due to Onsager's principle [16, 111, 159, 163, 164].

grid size	q_1	q_2	q_∞
10→20	2.02	2.01	2.04
20→30	2.00	2.00	2.01
30→40	2.00	2.00	2.01
40→50	2.00	2.00	2.02
50→100	2.00	2.00	2.01
mean	2.00	2.00	2.02

Table 5.4: Convergence rates of the single-temperature code for the third pure-cell verification problem. The convergence rate is calculated for three p-norms.

This problem is run to steady state on the unit square. The diffusivity values are randomly chosen to be $D^{xx} = 629$, $D^{yy} = 85.3$, and $D^{xy} = 0.741$. The convergence rates, shown in Table 5.4, demonstrate that single-temperature code remains second-order accurate for tensor diffusivities.

5.3.4 Test 4: MMS with Two Phases

The fourth test is an MMS problem that displays each code's ability to handle a phase interface where a jump in the diffusivity occurs. This is a 1D test that is run to steady-state (time-independent) with a discontinuous diffusivity. This test is executed such that the discontinuity occurs at a cell face such that there are only pure cells.

The solution is constant in the y-direction and could be run on a grid with an arbitrary number of points in that dimension. However, for consistency with all other tests, this problem is run on square mesh with $\Delta x = \Delta y$. Thus, the domain of the problem is the unit square. The computed results are constant in the y-direction, effectively giving a 1D solution. This test has a quadratic solution, which was inspired by a test with a linear solution in Shashkov and Steinberg [198], and is designed to give a constant source term in the domain. The solution, diffusivity, and source are defined as follows,

$$U(x,y) = \begin{cases} D_2 x^2 + a^2(D_1 - D_2) & (x < a) \\ D_1 x^2 & (x > a) \end{cases} \tag{5.19a}$$

$$D = \begin{cases} D_1 & (x < a) \\ D_2 & (x > a) \end{cases} \tag{5.19b}$$

$$Q = -2D_1 D_2, \tag{5.19c}$$

where $x = a$ is the location of the interface. The interface location is chosen to lie along cell faces, avoiding any mixed cells. Note that temperature and flux are continuous at the interface.

This problem tests that the code can correctly account for the effect of an interface. The interface is orthogonal to the grid such that there are no mixed cells. It is crucial to show the accuracy of the codes for a pure-cell, two phase problem prior to exploring the behavior of a mixed-cell case. This test is a precursor to the more advanced 1D test problem presented in Chapter VII, which is time-dependent and has an analytic function to describe the evolution from initial condition to steady-state.

	Single-T			Multi-T		
grid size	q_1	q_2	q_∞	q_1	q_2	q_∞
10→20	2.00	1.99	2.04	1.85	1.87	1.91
20→30	2.00	2.00	2.03	1.74	1.86	1.95
30→40	2.00	2.00	2.02	2.19	2.05	1.97
40→50	2.00	2.00	2.01	1.52	1.78	1.97
mean	2.00	2.00	2.03	1.83	1.89	1.95

Table 5.5: Convergence rates for the fourth pure-cell verification problem. The jump ratio is 10^2 and the interface is near the middle of the domain. For both the single-temperature and multi-temperature codes, the convergence rate is calculated for three p-norms.

This test is implemented using diffusivities $D_1 = 10$ and $D_2 = 0.1$ (ratio of 10^2) on the unit square. The interface is placed at $a = 0.5$ for the single-temperature code, ensuring that there are no mixed-cells. The multi-temperature grid is shifted by half a cell with respect to the single-temperature grid because the boundaries are defined differently in the respective codes (face centered versus cell centered). Consequently, the interface for the multi-temperature code must be shifted with each grid size to ensure that there are no mixed cells. It is placed as near to $a = 0.5$ as possible without going over.

Table 5.5 presents the convergence rates for this test problem. The single-temperature code remains second-order accurate for a discontinuous diffusivity, as expected for SOM. The multi-temperature method is slightly below second-order accurate, with average convergence rates of about 1.89, compared to the average rates of 2.00 for SOM. Since the multi-temperature method with no mixed cells reduces to the standard FDM, these results also show that the FDM is not as accurate for a discontinuous coefficient.

To show that the lower performance of the multi-temperature code is not particular to the set value of the jump strength, another test is performed on the first three grids of the multi-temperature code where the jump strength changes (with no mixed cells). As Table 5.6 shows, the changes in the convergence rates with the jump strength r are negligible. Therefore, the sub-second-order convergence for the two-phase problem appears to be an inherent characteristic of the multi-temperature code..

ratio	grid size	q_1	q_2	q_∞
5	10→20	1.83	1.87	1.92
	20→30	1.78	1.87	1.93
10^2	10→20	1.85	1.87	1.91
	20→30	1.74	1.86	1.95
10^4	10→20	1.85	1.87	1.91
	20→30	1.74	1.86	1.95

Table 5.6: Convergence rates for various jump ratios for the multi-temperature code on the fourth pure-cell verification problem. The convergence rate is calculated for three p-norms. The interface lies on the cell boundaries such that no mixed cells are present.

This test indicates that both codes treat a discontinuous diffusivity problem at or near second-order accuracy for pure cells. The results of this test also show that SOM is more accurate than the multi-temperature code on such problems.

5.4 Summary

In this chapter we have defined verification, error metrics, and convergence rates. Convergence rates, using the relative error in three p-norms, are used in this and the following numerical results chapters in order to make an objective measure of the performance of the six mixed-cell models. The convergence rate of E_∞ is used as the primary metric in the analysis of the mixed-cell codes because it most accurately measures the mixed-cell error.

Four test problems are performed in this chapter, two of which are dynamic and two steady-state. These tests are used in an effort to build confidence that the single- and multi-temperature diffusion codes perform in the expected manner. The first test demonstrates that the codes solve a single-phase diffusion problem at second-order accuracy. The second test substantiates that the codes solve a diffusion problem with a position-dependent diffusion coefficient at second-order accuracy. The third test, which is only executed on the single-temperature code, indicates that the single-temperature code correctly solves a full tensor diffusion problem, which is necessary for the S2 mixed-cell model. The fourth test demonstrates that the single- and multi-temperature codes solve a two-phase problem at or near second-order accuracy for exclusively pure-cell problems, which is a typical assumption required for many multiphase codes (phrases such as 'the interface lies on cell boundaries' generally accompany such methods).

CHAPTER VI

Numerical Results: 1D & 2D Manufactured Solution

This chapter presents two test problems using the method of manufactured solutions (MMS) in order to explore the accuracy of the mixed-cell models. (MMS is detailed in Section 5.1.) The similarities and differences between the six mixed-cell models are summarized in Table 6.1. Both test problems are run to steady state, with the first problem in 1D and the second problem in 2D. The convergence rates for a set of grids will be given for each model in three variants of each test. These tests act as a segue to the more complicated method of exact solution (MES) mixed-cell tests presented in the subsequent two chapters.

We present the MMS test and convergence rates of each model for the 1D test in Section 6.1 and for the 2D test in Section 6.2. This chapter concludes with a discussion in Section 6.3.

Single-Temperature		
Similarities		Mixed cells have only one temperature, calculated at the cell-center. Implemented using SOM.
Differences		The distinguishing trait is how the diffusivity that represents both phases for a mixed cell is specified.
S1	Section 3.1.1	Effective scalar diffusivity from volume-fraction-weighted harmonic mean of diffusivities of each phase in a mixed cell.
S2	Section 3.1.2	Effective tensor diffusivity from volume-fraction-weighted averages of diffusivities of each phase in a mixed cell: arithmetic mean for direction parallel to interface, harmonic mean for direction perpendicular to interface normal.
S3	Section 3.1.3	The calculation of flux for the mixed cell is partitioned into four separate calculations in subcell regions, where each subcell, associated with one of the corners, has a diffusivity, volume, and corner angle.

Multi-Temperature		
Similarities		Mixed cells have a temperature for each phase, calculated at the phase centroid. Implemented using a flux-based FDM approach, where the flux across each face (including interfaces within a mixed cell) is computed.
Differences		The flux is computed differently between the three multi-temperature models. The flux requires the diffusivity of each phase, the area of the interface, and the distance from each centroid to the face.
M1	Section 4.2.1	Assumes all intercell distances are Δx or Δy, and intracell distances are about half a cell apart. Face areas are approximated using the volume fractions of neighboring cells.
M2 & M3		Centroids computed for each phase in all cells, so distances and face areas are accurately computed.
M2	Section 4.2.2	Centroid-to-centroid distance used for all flux calculations, regardless of orientation with respect to the interface.
M3	Section 4.2.3	Normal distance from each centroid to common interface is used for flux calculations.

Table 6.1: Description of the six mixed-cell models.

6.1 1D Mixed-Cell Test

This section uses the same 1D test problem as in Section 5.3.4 with one difference: the interface is positioned such that it passes through a cell, creating mixed cells. This problem allows for an arbitrary discontinuity in the diffusivity to be placed in the domain, creating a single mixed cell in the 1D solution.

The solution is constant in the y-direction and could be run on a grid with an arbitrary number of points in that dimension. However, for consistency with all other tests, this problem is run on square mesh with $\Delta x = \Delta y$. Thus, the domain of the problem is the unit square. The computed results are constant in the y-direction, effectively giving a 1D solution with a single mixed cell. This test has a quadratic solution, which was inspired by a test with a linear solution in Shashkov and Steinberg [198], and is designed to give a constant source term in the domain. The solution, diffusivity, and source are the same as Section 5.3.4:

$$U(x,y) = \begin{cases} D_2 x^2 + a^2(D_1 - D_2) & (x < a) \\ D_1 x^2 & (x > a) \end{cases} \tag{6.1a}$$

$$D = \begin{cases} D_1 & (x < a) \\ D_2 & (x > a) \end{cases} \tag{6.1b}$$

$$Q = -2D_1 D_2, \tag{6.1c}$$

where $x = a$ is the location of the interface, chosen to cut through cell faces, creating mixed cells. Note that temperature and flux are continuous at the interface.

6.1.1 1D MMS Results

This test is implemented using diffusivities $D_1 = 10$ and $D_2 = 0.1$ (ratio of 10^2), as in Section 5.3.4. The difference here is that we introduce mixed cells. Thus, we position the interface near $a = 0.5$ such that the volume fraction of the mixed cell is constant for each grid. We explore three different volume fractions for the mixed cell: 33%, 50%, and 85%. The error (in the particular, E_∞) are plotted in Figure 6.2, and the convergence rates obtained from these errors are summarized in Table 6.2 for the six models (three each for single-temperature and multi-temperature codes). Additional convergence information (such as the rates for each grid and other norms) is available in Table C.1. Any convergence rate equal to or above 1.9 shown as bold in every convergence results tables for mixed-cell problems.

Unlike the pure-cell results, the multi-temperature results differ when taking the numerical solution at a phase-centered position versus at a cell-centered position. For pure-cell cases, the phase-center for each cell is the cell center. However, for a mixed cell, the phase-centered temperature is located at the centroid of each phase. A cell-centered temperature can be computed by taking an arithmetic average of the temperature from each phase, weighted by the volume fraction. However, this averaging introduces a first-order error even though the results for the individual

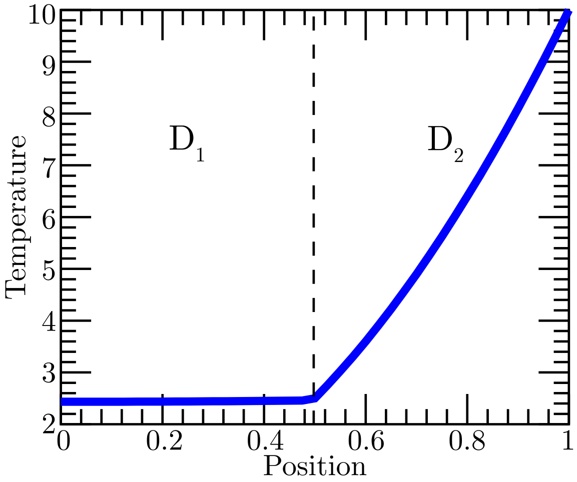


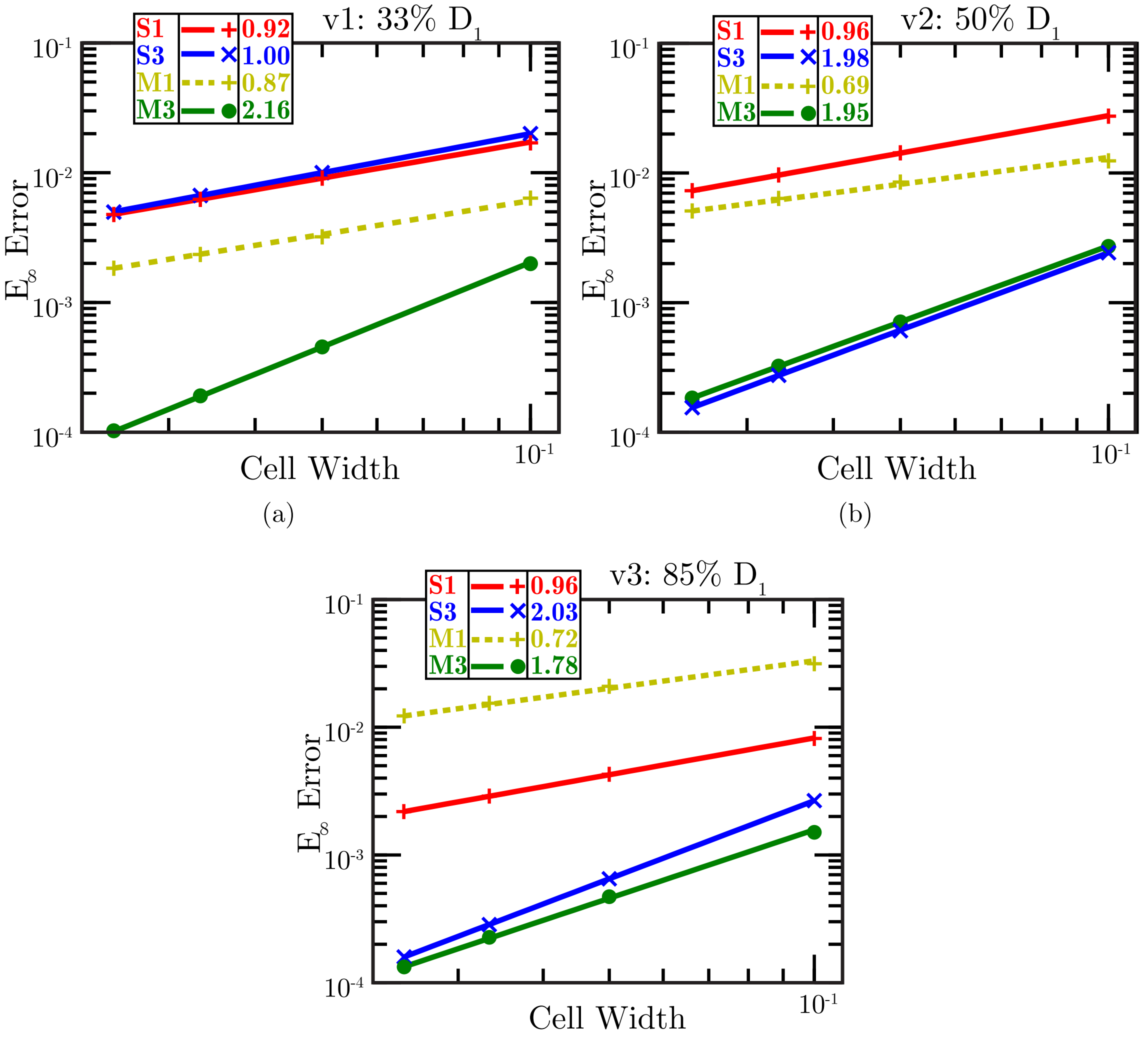
Figure 6.1: Analytical solution profile for the 1D MMS test. The interface is indicated with a dashed line, with D_1=10 on the LHS and D_2=0.1 on the RHS.

phases may be second-order accurate. In fact, the cell-centered results for the M3 model are approximately the same as the S1 model, meaning that the extra effort for the multi-temperature model offers no gain compared to the standard FDM using the harmonic mean. Thus, the advantage of the M3 model is only realized using phase-centered temperature values.

Furthermore, for the values used on this test, the phase 2 results are of greater interest because the solution has a much higher slope than in phase 1 (see Figure 6.1). In fact, the error analysis for the multi-temperature code in each mixed cell test problem uses the phase where the solution has a higher slope. Thus, for the multi-temperature models we use the phase 2 solution for the error analysis for this problem. We show the cell-centered results in many of the tables, where the three models are labeled as M1c, M2c, and M3c, where the ‘c’ indicates combined. The phase-centered results displayed will not indicate which phase is used (this choice will always be stated), and the three models are indicated as M1, M2, and M3.

Examination of Figure 6.2 shows that the phase 2 results of the M3 model have lower relative error values than the S1 and M1 models. The S3 model has equivalent error to the M3 model in v2 and v3, where both are second-order accurate. However, the S3 model actually has the highest error in v1.

S1 and S2 models. The S1 model is only first-order accurate in the ∞-norm, and it performs at approximately the same accuracy in each volume fraction (see Table 6.2). The S2 model, while having slightly lower q_1 and q_2 compared to the S1



(a)

(b)

(c)

Figure 6.2: Error versus cell width calculated with the ∞-norm for each volume fraction in the 1D MMS mixed cell test problem. The volume fractions shown are (a) v1, (b) v2, and (c) v3. Only phase 2 results are shown for the multi-temperature models, and the M2 model is not shown because it is identical to the M3 model. Similarly, the S2 model is not shown because it very similar to the S1 model.

	S1	S2	S3	M1c	M2c	M3c	M1	M2	M3
v1	0.92	0.68	1.00	0.74	0.97	0.97	0.87	**2.16**	**2.16**
v2	0.96	0.68	**1.98**	0.69	0.90	0.90	0.69	**1.95**	**1.95**
v3	0.96	0.86	**2.03**	0.72	1.14	1.14	0.72	1.78	1.78

Table 6.2: Average convergence rates of all six mixed-cell models for the 1D MMS mixed-cell test problem. The interface is near the center of the domain and makes the volume fractions indicated for the mixed cells. The jump strength is 10^2, and phase 2 is shown for the multi-temperature models. The convergence rate is measured using E_∞ and the rates are averaged over all pairs of grids. Table C.1 is an expanded version of this table. Bold indicates $q_i \geq 1.90$.

model, does especially poor in the ∞-norm (see Table C.1). This is surprising, as the S2 model should reduce to the S1 model in 1D because in the xx element of the diffusivity tensor in the S2 model uses the harmonic mean of the diffusivities of two phases. The differences between the S2 and S1 models are due to the anisotropy of the S2 diffusivity, which uses the arithmetic mean in the yy component of the tensor diffusivity, while the S1 model is isotropic and uses the harmonic mean.

S3 model. In v1 (33% volume fraction), the S3 model performs slightly better than the S1 model in all three norms, but it is still only first-order accurate in the ∞-norm; see Table 6.2. However, for v2 and v3, the S3 model is second-order accurate in all three norms (Table 6.2) and has the error much lower than the S1 and M1 models. Second-order performance is the desired result; however the fact that this is only achieved when the larger diffusivity is the majority fraction may limit the success of this model in 2D. This behavior is discussed further for the 1D MES test problem in Section 7.3.2.

M1, M2, and M3 models. The M1 model is first-order accurate or less in each error norm and each volume fraction. This indicates that the simplistic approximations made with this model are too rough to yield any gain in accuracy. The M2 and M3 models are equivalent in 1D since the distinguishing feature, off-axis interfaces, does not occur in 1D. Thus, in 1D, we will only discuss the M3 model.

The M3 model is second-order accurate in all three error normals for v1 and v2 (see Table C.1 in the appendix). However, the convergence rates decrease as the volume fraction increases (e.g., q_∞ are 2.16, 1.95, and 1.78 for v1, v2, and v3, respectively). Thus, when the smaller diffusivity is the majority phase, the M3 model converges more rapidly. This is opposite to the trend in the S3 model. However, while the smallest q_∞ for the M3 model is 1.78, which is much larger than the worst convergence rate of the S3 model, which is only 1.00. Therefore, the M3 model has the best overall accuracy for E_∞ for this test, as supported by the error values shown in Figure 6.2.

6.2 2D Mixed-Cell Test

This is a 2D MMS problem that creates a solution that has continuous temperature and flux across a linear interface, $y = mx + b$, where m is the slope and b is the y-intercept. The source term is equal for both phases, which is important for implementation purposes; it can introduce an ambiguity when two different sources must be assigned to a single cell, which may affect the rest of the test. The solution, diffusivity, and source are defined as follows,

$$T(x,y) = \begin{cases} D_2(y - mx - b)\sin(2\pi x)\sin(2\pi y) & (y < mx + b) \\ D_1(y - mx - b)\sin(2\pi x)\sin(2\pi y) & (y > mx + b) \end{cases} \tag{6.2}$$

$$D = \begin{cases} D_1 = 10 & (y < mx + b) \\ D_2 = 0.1 & (y > mx + b) \end{cases} \tag{6.3}$$

$$\begin{aligned} Q = -4\pi D_1 D_2 \big[&- 2\pi(y - mx - b)\sin(2\pi x)\sin(2\pi y) \\ &+ \sin(2\pi x)\cos(2\pi y) - m\cos(2\pi x)\sin(2\pi y)\big]. \end{aligned} \tag{6.4}$$

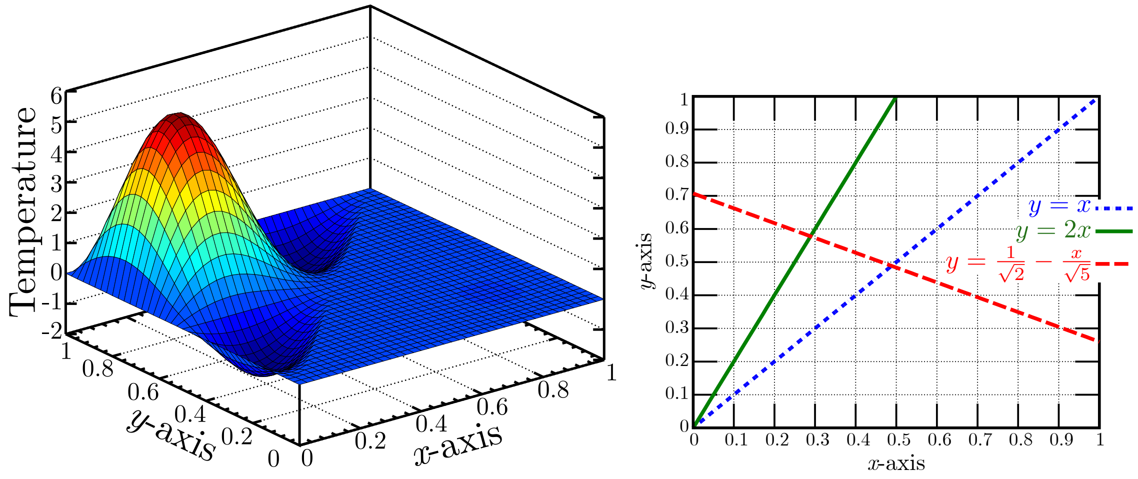
This is just one example of potential 2D MMS tests for mixed-cells. The test is implemented on the unit square, and the solution is zero for every external boundary as well as along the interface. The solution, shown in Figure 6.3(a), contains negative temperatures, which is not an issue for the codes tested. If, however, this is problematic, a constant ψ could be added to the solution, only changing the Dirichlet boundary conditions from zero to ψ.

A line only divides a rectangular cell when it crosses at two points. The points have four possible arrangements: two opposite corners, one corner and a face, opposite faces, or adjacent faces. In order to test each of these four cases, we run this 2D MMS test problem using the following three interfaces (shown in Figure 6.3(b)):

$$\text{L1:} \quad y = x \tag{6.5a}$$

$$\text{L2:} \quad y = 2x \tag{6.5b}$$

$$\text{L3:} \quad y = \tfrac{1}{\sqrt{2}} - \tfrac{x}{\sqrt{5}}. \tag{6.5c}$$



(a)

(b)

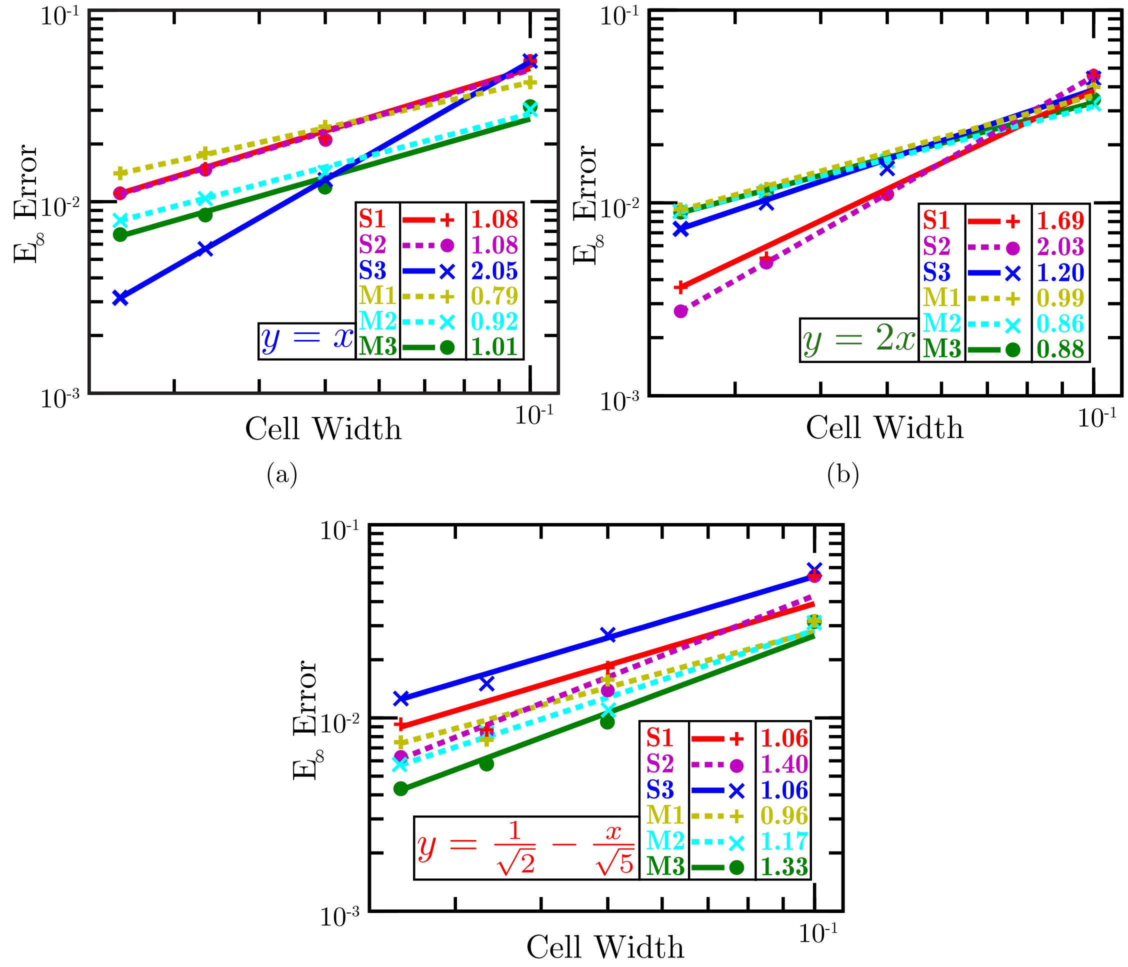
Figure 6.3: 2D MMS mixed cell test. (a) 2D analytical solution obtained via the MMS which is use for the mixed-cell test with D_1=10, D_2=0.1, and interface y=x. (b) The three interfaces used for this test.

Equation (6.5a) creates an interface that cuts two corners of every cell along the diagonal, making volume fractions of one half in every mixed cell. Equation (6.5b) crosses one corner of every mixed cell, making volume fractions of one quarter and three quarters. By using irrational numbers, Equation (6.5c) should not intersect any corners of cells in the grid, resulting in mixed cells with interfaces that cross opposite sides and adjacent sides. These mixed cells should have a spread of volume fractions. Note that these lines are given for the grid of the single-temperature code. The multi-temperature grid has a half grid shift in both the x- and y-directions, requiring Equation (6.5b) to be $y = 2x - \Delta x/2$, while the other two interfaces remain unchanged. No changes are required for L1 or L3 for the multi-temperature grid since Equation (6.5a) still cuts two corners along the diagonal and Equation (6.5c) still does not touch any corners.

As implemented, phase 1 is below the interface ($y < mx + b$). Thus, the phase with relatively constant temperature in Figure 6.3(a) is phase 1. Therefore, we will only report the results of error analysis for phase 2 for the multi-temperature models, as this phase has larger variation in the solution (temperature) than phase 1.

6.2.1 2D MMS Results

The average convergence rates for each interface and model are presented in Table 6.3, and additional convergence information (including convergence rates for other



(a)

(b)

(c)

Figure 6.4: Error versus cell width calculated with the ∞-norm for each interface of the 2D MMS mixed cell test problem. Only phase 2 results are shown for the multi-temperature models. The interfaces shown are (a) L1, (b) L2, and (c) L3. Bold indicates $q_i \geq 1.90$.

	S1	S2	S3	M1c	M2c	M3c	M1	M2	M3
L1	1.08	1.08	**2.05**	0.79	0.93	1.00	0.79	0.92	1.01
L2	1.69	**2.03**	1.20	1.08	1.47	1.62	0.99	0.86	0.88
L3	1.06	1.40	1.06	0.96	1.15	1.13	0.96	1.17	1.33

Table 6.3: Average convergence rates of all six mixed-cell models for the 2D MMS mixed-cell test problem. The convergence rate is measured using E_∞. Results are shown for three different linear interfaces, defined in Equation (6.5). The jump strength is 10^2, and phase 2 is shown for the multi-temperature models. Table C.2 is an expanded version of this table. Bold indicates $q_i \geq 1.90$.

norms as well as rates for each set of grids) can be found in Table C.2. Log-log plots of the error versus cell width are shown in Figure 6.4. The slope of each best-fit line, indicated with the legends, represents the convergence rates.

S1 and S2 models. As shown in the first row of Table 6.3, the S1 and S2 models give equivalent results in for the L1 interface, each with an average q_∞ of 1.08. The S1 model has its best performance using the L2 interface, starting out at second-order accuracy ($q_\infty = 2.04$) but dropping to first-order with increasing resolution ($q_\infty = 1.15$), as seen in Table C.2. This decreasing convergence rate indicates that the mixed-cell error is below the discretization error for lower resolutions, but that the discretization error converges more quickly than the mixed-cell error. This leads to mixed-cell dominating error for higher resolutions.

The S2 model performs at second-order accuracy for the L2 interface for each set of grids tested (as well as all three error norms); see Table C.2. This consistent performance in each grid indicates that the anisotropic tensor, using arithmetic and harmonic means, accurately computes the mixed-cell temperature for this case. It is not surprising that both the S1 and S2 models perform best with the same interface, as they are related methods (homogenization involving the harmonic mean).

Neither model performs well in the L3 interface, although for q_∞, the S2 model (1.40) is better than the S1 model (1.06), as shown in the third row of Table 6.3. Thus, when considering all three interfaces for this test case, the S2 model represents a slight improvement over the S1 model.

S3 model. Table 6.3 shows that the S3 model achieves second-order accuracy for the L1 interface, where $y = x$. Each mixed cell in this case involves two pure subcells and two subcells homogenized by the harmonic mean. This indicates that homogenization can yield second-order accuracy, at least when combined with the split-zone method. Second-order accuracy for the harmonic-mean approach also occurs for L2 with the S2 model, as discussed above.

The S3 model is only first-order accurate for the L2 and L3 interfaces. However, Table C.2 shows that q_1 and q_2 for the S3 model are between 1.88 and 2.00 on average for these interfaces. Figure 6.4(c) shows that the S3 model has the highest error of all models for the L3 interface. In fact, even in the L1 interface where S3 is second-order accurate, it has the largest error for the coarsest grid; see Figure 6.4(a).

The L3 interface crosses cells on adjacent sides as well as opposite sides. The cells crossed on opposite sides should more or less alternate which phase is the majority. As we learned from the 1D MMS test, the S3 model has a performance drop for mixed cells with interfaces that cross opposite sides when the majority phase is the smaller diffusivity. Since this occurs frequently in L3, the large error for this interface is related to the observations from the 1D test: the S3 model is only first-order accurate when the majority phase is the smaller diffusivity (for cells with an interface that crosses opposite sides).

M1, M2, and M3 models. The multi-temperature results are all first-order accurate or less, as shown in Table 6.3 and Table C.2. The M3 model has peak q_∞ (1.3) for L3, which, similar to the 1D MMS results, is also the interface where the S3 model performs worst. The fact that M3 tends to perform best in the instances where S3 performs worst (and vice versa) suggests that a model that combines the S3 and M3 models may yield the best overall performance.

Although M3 is only first-order accurate, examination of Figure 6.4 shows that it has the lowest error for first-order order results in all cases. In fact, it is evident from Figure 6.4 that the M3 model along with the M2 model have the lowest error for the coarsest grid in every case. Examination of Tables 6.3 and C.2 show that the M2 and M3 models yield similar convergence rates. The M1 model, while having the lowest convergence rate and highest error of the multi-temperature methods, has error lower than all three single-temperature models at the coarsest grid.

The strength of the multi-temperature models is most apparent in this test. Although they all have only first-order convergence rates, we find that the error is lower than the single-temperature models for the coarsest grids. This indicates that the multi-temperature models are approximating the physics better. The fact that the convergence is lower than second-order is a consequence of the subzonal model: the discretization of the mixed cells effectively yields a non-uniform grid. The non-uniformity is discontinuous (i.e., not smoothly varying) and the change in size is on the order of the grid spacing (distances from mixed-cell centroids to cell faces are on the order of $\Delta x/2$). Hirsch [93] shows that if either of these properties is true (discontinuity in grid size or spacing on the order of Δx), the resulting discretization is only first-order. The multi-temperature model could potentially achieve second-order accuracy by accounting for the non-uniformity of the grid spacing. This can be achieved by supplementing the discretization near the interface with higher-order terms [93].

6.2.2 Modified S3 (Split-Zone) Results

Through the results presented above, we realized that it was possible to modify the split-zone method to achieve better accuracy. The change applies to mixed cells where the interface crosses adjacent sides as well as where the interface crosses one corner. In each case, the minority phase is mostly associated with a single corner. Say that phase 2 is localized in the bottom-right corner with volume fraction $1 - f$. We then specify

$$\begin{aligned} V_{RB} &= (1-f)\Delta x \Delta y && (6.6)\\ V_{LB} &= \tfrac{1}{3} f \Delta x \Delta y && (6.7)\\ V_{LT} &= \tfrac{1}{3} f \Delta x \Delta y && (6.8)\\ V_{RT} &= \tfrac{1}{3} f \Delta x \Delta y. && (6.9) \end{aligned}$$

This is in contrast to Equation (3.7) for when the interface crosses adjacent cells and Equation (3.6) for when the interface crosses one corner.

		L2: $y = 2x$			L3: $y = \frac{1}{\sqrt{5}} - \frac{x}{\sqrt{2}}$		
S3	grid size	q_1	q_2	q_∞	q_1	q_2	q_∞
old	10→20	**1.95**	**1.96**	1.55	**1.97**	**1.93**	1.11
	20→30	1.89	1.87	1.01	**2.03**	**1.99**	1.44
	30→40	1.83	1.80	1.03	**1.98**	1.87	0.62
	mean	1.89	1.88	1.20	**2.00**	**1.93**	1.06
new	10→20	**1.98**	**2.03**	**2.04**	**1.93**	**2.00**	**2.05**
	20→30	**2.00**	**2.01**	**2.01**	**2.08**	**2.00**	1.83
	30→40	**2.00**	**2.00**	**2.01**	**1.93**	**1.92**	-0.31
	mean	**1.99**	**2.01**	**2.02**	**1.98**	**1.97**	1.19

Table 6.4: Comparison of convergence rates for 2D MMS mixed-cell test problem for the initial (old) and modified (new) S3 model implementations. The convergence rate is calculated for three p-norms. Results for each pair of grids are presented along with the average convergence rate. Updated results are shown for the L2 and L3 interfaces (there was no change for the L1 interface). The diffusivity ratio is 10^2. Bold indicates $q_i \geq 1.90$.

Table 6.4 compares the convergence rates using this new S3 model compared to the old S3 model. The L1 interface case, where S3 is second-order accurate, is identical in both variants of the S3 model because the modifications do not affect this case. We see that the S3 model now achieves second-order q_∞ for the L2 interface. Now both the S2 and S3 models are second-order accurate for this interface. We also see increased performance for the S3 model in the L3 interface case. While the S3 model starts out at second-order for the L3 interface, it drops to negative convergence for the highest resolution tested. This indicates that the mixed-cell error starts below the discretization error but eventually grows and peaks above the discretization error, leading to a negative convergence rate. The improved results at low resolution demonstrate that the error from the adjacent crossing cells is reduced, but the error from the opposite crossing cells still dominates the results at higher resolution.

We see for L1 in Table C.2 and for L2 and L3 in Table 6.4 that the modified S3 model is now second-order in all three error norms for all three interfaces except for q_∞ using the L3 interface. Therefore, the S3 model yields the highest accuracy and convergence rates for this 2D MMS mixed-cell test.

6.3 Summary

We presented MMS mixed-cell tests in 1D and 2D. The method of manufactured solutions creates a test problem by choosing an analytic solution and then calculating result of operating the governing differential equations on the analytic solution. This result is then used as a source term that, along with the boundary conditions, drives the system to the analytic solution.

The 1D test maintains a constant volume fraction in the mixed cells at each grid size. In 1D, we find that the S1 and M1 models are only first-order accurate. We also find that the S3 and M3 models give second-order accuracy for E_∞, however the S3 model only performs at second-order for two of the three volume fractions, while the M3 model performs well in all cases. The S3 model is second-order only when the majority phase ($f \geq 1/2$) is the larger diffusivity.

In 2D, we find that the S1, M1, M2, and M3 models are only first-order accurate. However, we also see that the error associated with the multi-temperature models is lowest for the coarsest grids, even though they may have a lower convergence. The S2 model performs at second-order accuracy for one interface ($y = 2x$), but only first-order for the other two interfaces.

The S3 model has similar performance to the S2 model, achieving second-order accuracy in only one of the three interfaces. However, we were able to modify the S3 model, which provides second-order accuracy in two of the three cases tested. The lower convergence rate for the S3 model in the L3 interface is likely related to the first-order performance in the 1D MMS case. Thus, when the interface crosses opposite sides and the majority phase is the smaller diffusivity, the S3 model is first-order accurate but second-order otherwise. Further research is needed to explore techniques to mitigate this issue.

CHAPTER VII

Numerical Results: 1D Exact Solution

This chapter summarizes the error analysis to quantify the fidelity of different mixed-cell models using a 1D composite rod solution. This solution serves as a benchmark test against which the models introduced in this thesis are compared. The analytical solution we present does not represent new work [61, 62, 147, 221]; however using this solution as a mixed-cell benchmark is unique. The chapter begins in Section 7.1 with the derivation of the analytic solution via separation of variables (SoV). The resulting eigenvalue solution is then discussed in Section 7.2, showing the dependence on the diffusivity ratios, the time variable, and the number of eigenvalues. This is followed with results for each mixed-cell technique in Section 7.3. The chapter concludes in Section 7.4 with a discussion.

The problem solved in this chapter is the 1D heat equation for an insulated composite rod, which can only conduct heat along the length of the rod. The 1D heat equation is

$$\rho c \frac{\partial T(x,t)}{\partial t} = \frac{\partial}{\partial x}\left(D \frac{\partial}{\partial x} T(x,t) \right), \tag{7.1}$$

where ρ is density, c is heat capacity, $T(x,t)$ is temperature, D is diffusivity, x is position, and t is time. The rod has length L with a domain of $0 \leq x \leq L$. An interface at $x = a$ separates the two phases, with D_1 from $0 \leq x \leq a$ and D_2 from $a \leq x \leq L$. Figure 7.1 shows the basic problem set up.

We assume continuity of temperature at the interface. However, physical situations may have thermal contact resistance at material interfaces due to surface roughness and voids. Thermal contact resistance leads to a jump in the temperature at the interface [107, 192]. However, this complication could be added to the test problems with relatively little difficulty in order to match the test with realistic materials.

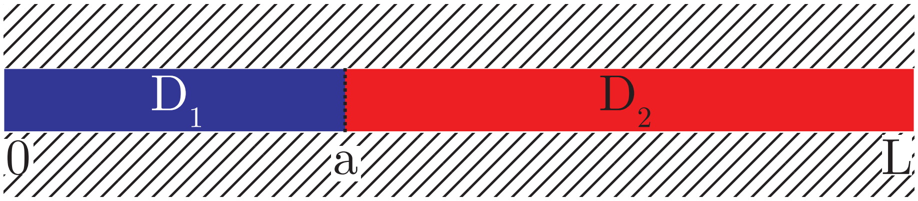


Figure 7.1: One dimensional heat conduction problem domain. Phase 1 is the blue region with diffusivity D_1, and phase 2 is the red region with diffusivity D_2. The rod is insulated along its length, which only allows for head conduction across the endpoints.

Initially, the rod is at a uniform temperature, $T(x, 0) = T_i$ for $0 \leq x \leq L$. For $t > 0$, the two ends of the rod are held at constant temperatures, $T(0, t) = T_0$ and $T(L, t) = T_L$. The results presented make the assumption that $T_0 < T_L < T_i$, but the solution is valid for any values of these three temperatures. Further constraints include continuity of temperature and flux across the interface. Mathematically, these constraints are given as

$$
\begin{aligned}
T(x,0) &= T_i && \text{Initial Condition} && (7.2a)\\
T(0,t) &= T_0 && \text{Left Boundary} && (7.2b)\\
T(L,t) &= T_L && \text{Right Boundary} && (7.2c)\\
T(a-,t) &= T(a+,t) && \text{Continuity of Temperature} && (7.2d)\\
D_1 \frac{\partial T(x,t)}{\partial x}\bigg|_{x=a-} &= D_2 \frac{\partial T(x,t)}{\partial x}\bigg|_{x=a+} && \text{Continuity of Flux.} && (7.2e)
\end{aligned}
$$

The notation $x = a-$ indicates x approaching a from below (smaller values), and $x = a+$ indicates x approaching a from above (larger values).

Using a constant initial condition is a good choice because it avoids numerical complications. Namely, the assignment for temperature in a mixed cell is less clear if there is a gradient or if each material has a different temperature. The test can become contaminated by the initial implementation of the temperature profile if the initial condition has a gradient or kink at the interface. A smooth initial condition avoids this problem entirely, allowing the analysis to focus on the fidelity of the mixed cells without the complication of introducing any potential inaccuracy at the start of the problem.

7.1 Derivation of the Analytical Solution

This derivation is a simpler case of an n-layered, composite media, analytical heat equation solution [61, 62, 147]. The approach presented in this chapter for a two-layered composite media problem follows that of Zauderer [221]. Although we can express the thermal conductivity (D) as a function of x in terms of the Heaviside step function,

$$D(x) = (D_2 - D_1)H(x - a) + D_1, \tag{7.3}$$

we choose instead to break the problem in two parts. By treating each phase separately, the diffusivity no longer depends on x, simplifying the problem. The solution for each phase is related by enforcing continuity of temperature and flux at the interface. There exists a set of solutions (eigenfunctions) that satisfies Equation (7.1) in each phase, with a constant for each eigenfunction determined by the initial condition.

This section first applies SoV and determines the time dependence of the solution (Section 7.1.1). This is followed by the steady-state solution (Section 7.1.2) and the transient solution (Section 7.1.3). The full solution is presented in Section 7.1.5, summarizing all terms and constants required to express the analytical solution.

7.1.1 Separation of Variables & Time Dependence

Beginning with the SoV assumption, we write the two-variable temperature as the product of a function of time, $\tau(t)$, with a function of position, $X(x)$, so that

$$T(x,t) = \tau(t)X(x). \tag{7.4}$$

Because the diffusivity is no longer a function of x, it can pass through the spatial derivative. As a result, the differential equation becomes

$$\begin{aligned} \rho c \tau_t(t)X(x) &= D\tau(t)X_{xx}(x) \\ \frac{\tau_t(t)}{\tau(t)} &= \frac{D}{\rho c}\frac{X_{xx}(x)}{X(x)} = -k, \end{aligned} \tag{7.5}$$

where k is some constant relating the time-dependent LHS to the position-dependent RHS, and the subscript notation here indicates a derivative of a single variable function (i.e., $X_x(x) = dX(x)/dx$, $\tau_t(t) = d\tau(t)/dt$). By treating the two equalities of Equation (7.5) separately, two ordinary differential equations, one each for $\tau(t)$ and $X(x)$, are obtained.

The solution for $\tau(t)$ in Equation (7.5) is given by

$$\tau(t) = b\exp(-kt) + b_0\exp(kt), \tag{7.6}$$

where b and b_0 are arbitrary constants. We choose k to be non-negative, which requires $b_0 = 0$ to avoid unphysical exponential growth with time. Therefore, the solution to the temporal function is

$$\tau(t) = b\exp(-kt), \tag{7.7}$$

where k must be either zero or positive. This restriction on k guides the process of finding the spatial function, $X(x)$.

7.1.2 Steady-State Solution ($k = 0$)

When $k = 0$, the time equation reduces to a constant, which we choose to be unity. Let $X_1(x)$ describe the function from $x = 0$ to $x = a$, with phase parameters $\rho_1 c_1$ and D_1 and boundary conditions $X_1(0) = T_0$ and $X_1(a) = T_a$. We can then solve the differential equation for $x < a$,

$$\frac{D_1}{\rho_1 c_1} X_1''(x) = 0, \tag{7.8}$$

as follows:

$$\begin{aligned}
X_1(x) &= c_0 x + c_1 && \text{General Solution} \\
X_1(0) &= T_0 = c_1, && \text{Left Boundary Condition} \\
X_1(a) &= T_a = c_0 a + T_0 && \text{Right Boundary Condition} \\
X_1(x) &= \frac{T_a - T_0}{a} x + T_0. && \qquad (7.9)
\end{aligned}$$

Following the same approach for $x \geq a$, with boundary conditions $X_2(a) = T_a$ and $X_2(L) = T_L$, we find

$$X_2(x) = \frac{T_a - T_L}{L - a}(L - x) + T_L \quad (x \geq a) \tag{7.10}$$

We combine Equations (7.9) and (7.10) to write our piecewise expression of the steady-state (ss) temperature,

$$T_{\text{ss}}(x) = \begin{cases} T_0 + \frac{x}{a}(T_a - T_0) & (x \leq a) \\ T_L + \frac{L-x}{L-a}(T_a - T_L) & (x \geq a), \end{cases} \tag{7.11}$$

where we have $T(x,t) = T(x) = X(x)$ because $k = 0$ and $\tau(t)$ is constant. Equation (7.11) satisfies the nonhomogeneous boundary conditions, so any other contributions from nonzero values of k will have a boundary condition of $T(0, t > 0) = T(L, t > 0) = 0$.

Enforcing continuity of temperature and flux completes the steady-state solution. Continuity of temperature was enforced by defining the value T_a, which is not yet determined. The additional equation necessary to solve for T_a is given from continuity of flux:

$$\begin{aligned} -D_1 \frac{dT(x)}{dx}\bigg|_{x=a-} &= -D_2 \frac{dT(x)}{dx}\bigg|_{x=a+} \qquad (7.12) \\ \frac{D_1}{a}(T_a - T_0) &= \frac{D_2}{L-a}(T_L - T_a) \\ T_a\left(\frac{D_2}{L-a} + \frac{D_1}{a}\right) &= \frac{D_2}{L-a}T_L + \frac{D_1}{a}T_0 \\ T_a &= \frac{\dfrac{D_2}{L-a}T_L + \dfrac{D_1}{a}T_0}{\dfrac{D_2}{L-a} + \dfrac{D_1}{a}}. \qquad (7.13) \end{aligned}$$

Substituting Equation (7.13) into Equation (7.11) and simplifying the resulting expression gives the steady state temperature,

$$T_{\text{ss}}(x) = \begin{cases} T_0 + \dfrac{x}{a}\dfrac{D_2}{L-a}\left(\dfrac{T_L - T_0}{\dfrac{D_2}{L-a} + \dfrac{D_1}{a}}\right) & (x < a) \\ \\ T_L + \dfrac{L-x}{L-a}\dfrac{D_1}{a}\left(\dfrac{T_0 - T_L}{\dfrac{D_2}{L-a} + \dfrac{D_1}{a}}\right) & (x > a). \end{cases} \tag{7.14}$$

Introducing combinations of constants, we rewrite this linear expression for reference later as

$$T_{\text{ss}}(x) = \begin{cases} T_0 + \sigma x & (x < a) \\ T_L + \beta(L - x) & (x > a). \end{cases} \tag{7.15}$$

This is the final form of the steady-state solution.

7.1.3 Transient Solution ($k > 0$)

For positive values of k, the temporal solution is exponential decay, with an argument that depends on the values of k. The temporal solution is given in Equation (7.7). For the spatial solution, we require that $X(0) = X(L) = 0$ as the non-homogeneous boundary conditions are accounted for by the steady-state portion of the solution. Starting with $x \leq a$ for the spatial solution of Equation (7.5) and introducing constants (b_i's) to be determined, we have

$$X_1(x) = b_1 \sin\left(\sqrt{\frac{k\rho_1 c_1}{D_1}}x\right) + b_2 \cos\left(\sqrt{\frac{k\rho_1 c_1}{D_1}}x\right). \tag{7.16}$$

The condition $X_1(0) = 0$ means that $b_2 = 0$. Similarly, we can write the spatial solution for $x \geq a$ as

$$X_2(x) = b_3 \sin\left(\sqrt{\frac{k\rho_2 c_2}{D_2}}(L - x)\right) + b_4 \cos\left(\sqrt{\frac{k\rho_2 c_2}{D_2}}(L - x)\right). \tag{7.17}$$

Again, we find the cosine term vanishes because $X_2(L) = 0 = b_4$.

Let us define four new constants for ease of exposition:

$$\lambda = \sqrt{k} \tag{7.18a}$$

$$\mu = \sqrt{\frac{\rho_1 c_1}{D_1}} \tag{7.18b}$$

$$\nu = \sqrt{\frac{\rho_2 c_2}{D_2}} \tag{7.18c}$$

$$\eta = L - a. \tag{7.18d}$$

We identify k as the eigenvalue for the differential equation to be solved. However, the presentation is more crisp to discuss the eigenvalues in terms of λ rather than of k. Consequently, we will refer to λ as the eigenvalues, with the implicit knowledge that the eigenvalues are technically the square of λ. There is no loss of information since k is non-negative.

Substituting Equation (7.18), the spatial solutions are then

$$X_1(x) = b_1 \sin(\lambda\mu x) \tag{7.19a}$$

$$X_2(x) = b_3 \sin(\lambda\nu(L - x)). \tag{7.19b}$$

Continuity of temperature, $X_1(a) = X_2(a)$, gives a relation between b_1 and b_3, allowing us to write the solutions as

$$X_1(x) = b\frac{\sin(\lambda\mu x)}{\sin(\lambda\mu a)} \qquad (x < a) \tag{7.20a}$$

$$X_2(x) = b\frac{\sin(\lambda\nu(L - x))}{\sin(\lambda\nu\eta)} \qquad (a < x). \tag{7.20b}$$

Continuity of flux, Equation (7.12), gives

$$\mu D_1 \cot(\lambda\mu a) + \nu D_2 \cot(\lambda\nu\eta) = 0. \tag{7.21}$$

Equation (7.21) is a transcendental equation and the eigenvalue equation for Equation (7.5). There exists an infinite set of values for $\lambda > 0$ that satisfy Equation (7.21). We enumerate this ordered set of eigenvalues with n, where $\lambda_n \in \{\lambda_1, \lambda_2, \lambda_3, ...\}$.

The set eigenfunctions, $v^{(n)}(x)$, is specified on a domain of $0 \leq x \leq L$ in two piecewise-continuous sections. The set of eigenfunctions, corresponding to the eigenvalues specified by Equation (7.21), are defined as

$$v^{(n)}(r) = \begin{cases} v_1^{(n)}(r) & (0 \leq x \leq a) \\ v_2^{(n)}(r) & (a \leq x \leq L) \end{cases} \tag{7.22}$$

where

$$v_1^{(n)}(x) = \frac{\sin(\lambda_n \mu x)}{\sin(\lambda_n \mu a)} \tag{7.23a}$$

$$v_2^{(n)}(x) = \frac{\sin(\lambda_n \nu(L - x))}{\sin(\lambda_n \nu\eta)}, \tag{7.23b}$$

where the scalar constant series b_n is absorbed into the temporal solution, and the subscripts 1 and 2 refer to the regions of the rod (see Figure 7.1). The transient solution to this 1D heat conduction problem for discontinuous media is

$$T_{\text{transient}}(x, t) = \sum_{n=1}^{\infty} b_n e^{-\lambda_n^2 t} v^{(n)}(x). \tag{7.24}$$

7.1.4 The Full Solution

Combining the steady-state, Equation (7.15), and the transient, Equation (7.24), solutions, we find a combined solution of

$$T(x,t) = \begin{cases} T_0 + \sigma x + \displaystyle\sum_{n=1}^{\infty} b_n e^{-\lambda_n^2 t} v_1^{(n)}(x) & (x < a) \\ T_L + \beta(L-x) + \displaystyle\sum_{n=1}^{\infty} b_n e^{-\lambda_n^2 t} v_2^{(n)}(x) & (x > a). \end{cases} \tag{7.25}$$

We must determine the scalar constants series b_n in order to fully specify the solution. This involves two steps. First, in Section 7.1.4.1, we must demonstrate that the eigenfunctions are orthogonal and determine their magnitude, creating an orthonormal basis. Second, in Section 7.1.4.2, we determine the constants b_n from the initial conditions.

7.1.4.1 Eigenfunctions: Orthonormal Basis

Each eigenfunction, Equation (7.23), is a solution to the governing equation, Equation (7.5). The eigenfunctions are orthogonal and form a basis of the eigenspace. In this section we first demonstrate the orthogonality of the eigenfunctions, and then determine the magnitude in order to normalize the basis. The orthogonalization discussion and inner product definition, which come from Sturm-Liouville problems, are adapted from Zauderer [221].

Let λ_i and λ_j be eigenvalues corresponding to eigenfunctions $v^{(i)}$ and $v^{(j)}$. Let L be the diffusion operator, written as $\mathsf{L}[v^{(n)}(x)] = -\vec{\nabla}\cdot(D\vec{\nabla}v^{(n)}(x)) = \lambda\rho c v^{(n)}(x)$. The 1D form is simply $\mathsf{L}[v^{(n)}(x)] = -Dv_{xx}^{(n)}(x) = \lambda\rho c v^{(n)}(x)$, where $v_{xx}^{(n)}(x) = \frac{\partial^2}{\partial x^2}v^{(n)}(x)$. Orthogonality can be shown with the integral

$$\int_0^L (v^{(i)}\mathsf{L}v^{(j)} - v^{(j)}\mathsf{L}v^{(i)})dx$$

and

$$\mathsf{L}v^{(i)} = -Dv_{xx}^{(i)} = \lambda_i \rho c v^{(i)}. \tag{7.26}$$

Beginning with the first definition of the operator L, we find

$$\begin{aligned}\int_0^L (v^{(i)}\mathsf{L}v^{(j)} - v^{(j)}\mathsf{L}v^{(i)})dx &= \int_0^L (v^{(j)}Dv_{xx}^{(i)} - v^{(i)}Dv_{xx}^{(j)})dx \\ &= \int_0^L \frac{d}{dx}\left(Dv^{(j)}v_x^{(i)} - Dv^{(i)}v_x^{(j)}\right)dx \qquad (7.27)\end{aligned}$$

Repeating the integral using the second definition of L, we have

$$\begin{aligned}\int_0^L (v^{(i)}\mathsf{L}v^{(j)} - v^{(j)}\mathsf{L}v^{(i)})dx &= \int_0^L (v^{(i)}\lambda_j\rho c v^{(j)} - v^{(j)}\lambda_i\rho c v^{(i)})dx \\ &= (\lambda_j - \lambda_i)\int_0^L \rho c v^{(i)}v^{(j)}dx \\ &= (\lambda_j - \lambda_i)\ \langle v^{(i)}, v^{(j)}\rangle, \qquad (7.28)\end{aligned}$$

where we define the inner product for this eigenvalue problem as [221]

$$\langle v^{(i)}(x), v^{(j)}(x)\rangle = \int_0^L \rho c v^{(i)}v^{(j)}dx = \mu^2 D_1\int_0^a v_1^{(i)}v_1^{(j)}dx + \nu^2 D_2\int_a^L v_2^{(i)}v_2^{(j)}dx, \quad (7.29)$$

where $\rho_1 c_1 = \mu^2 D_1$ and $\rho_2 c_2 = \nu^2 D_2$ by Equation (7.18)(b-c).

Combining Equation (7.27) and Equation (7.28), we find the expression

$$(\lambda_j - \lambda_i)\ \langle v^{(i)}, v^{(j)}\rangle = \int_0^L \frac{d}{dx}(Dv^{(j)}v_x^{(i)} - Dv^{(i)}v_x^{(j)})dx. \quad (7.30)$$

Therefore, demonstrating orthogonality of the eigenfunctions is now a matter of demonstrating that the RHS of Equation (7.30) is zero when $\lambda_j \neq \lambda_i$, since this would imply $\langle v^{(i)}, v^{(j)}\rangle = 0$.

We have shown the following three properties of the eigenfunctions in Section 7.1.3:

$$v^{(n)}(0) = v^{(n)}(L) = 0 \quad (7.31a)$$

$$v_1^{(n)}(a) = v_2^{(n)}(a) \quad (7.31b)$$

$$D_1\tfrac{d}{dx}[v_1^{(n)}(x)]_{x=a-} = D_2\tfrac{d}{dx}[v_2^{(n)}(x)]_{x=a+}, \quad (7.31c)$$

which correspond to (a) zero at the endpoints, (b) continuous across the interface, and (c) flux continuity across the interface. All three of these properties were enforced when constructing the eigenfunctions. Using Equation (7.31), we find that

$$\int_0^L (v^{(i)}Lv^{(j)} - v^{(j)}Lv^{(i)})dx = \int_0^a \frac{d}{dx}(D_1 v^{(j)} v_x^{(i)} - D_1 v^{(i)} v_x^{(j)})dx \\ + \int_a^L \frac{d}{dx}(D_2 v^{(j)} v_x^{(i)} - D_2 v^{(i)} v_x^{(j)})dx = 0. \qquad (7.32)$$

This integral is identically zero because the eigenfunctions vanish at the endpoints, and the functions $Dv^{(j)}v_x^{(i)}$ and $Dv^{(i)}v_x^{(j)}$ each occur with opposite signs due to continuity of temperature and flux at $x = a$. Since Equation (7.32) is identically zero, we see that Equation (7.30) must be true for $\lambda_j \neq \lambda_i$. Therefore, the eigenfunctions are orthogonal.

We now evaluate the inner product, Equation (7.29), between identical eigenfunctions to determine the magnitude,

$$\|v^{(n)}\|^2 = \langle v^{(n)}, v^{(n)} \rangle. \qquad (7.33)$$

The norm of the n^{th} eigenfunction is found by substituting Equation (7.23) into Equation (7.29) and evaluating the integral:

$$\begin{aligned}\|v^{(n)}(x)\|^2 &= \int_0^L (\rho c)\left(v^{(n)}(x)\right)^2 dx = \mu^2 D_1 \int_0^a \left(v_1^{(n)}\right)^2 dx + \nu^2 D_2 \int_a^L \left(v_2^{(n)}\right)^2 dx \\ &= \mu^2 D_1 \frac{\lambda_n \mu a - \cos(\lambda_n \mu a)\sin(\lambda_n \mu a)}{2\lambda_n \mu \sin^2(\lambda_n \mu a)} + \nu^2 D_2 \frac{\lambda_n \nu \eta - \cos(\lambda_n \nu \eta)\sin(\lambda_n \nu \eta)}{2\lambda_n \nu \sin^2(\lambda_n \nu \eta)} \\ &= \frac{\mu^2 D_1 a}{2\sin^2(\lambda_n \mu a)} + \frac{\nu^2 D_2 \eta}{2\sin^2(\lambda_n \nu \eta)} - \frac{1}{2\lambda_n}\left(\mu D_1 \cot(\lambda_n \mu a) + \nu D_2 \cot(\lambda_n \nu \eta)\right) \\ &= \frac{\mu^2 D_1 a}{2\sin^2(\lambda_n \mu a)} + \frac{\nu^2 D_2 \eta}{2\sin^2(\lambda_n \nu \eta)} \\ &= \frac{1}{2}\left[\mu^2 D_1 a \csc^2(\lambda_n \mu a) + \nu^2 D_2 \eta \csc^2(\lambda_n \nu \eta)\right], \end{aligned} \qquad (7.34)$$

where the cotangent terms vanish due to Equation (7.21).

We use Equation (7.34) as the magnitude of the eigenfunctions in order to create an orthonormal basis for constructing the initial condition, where in turn defines the constant set b_n.

7.1.4.2 Constructing the Initial Conditions

The full solution is now determined by specifying b_n, which is accomplished by constructing the initial condition in terms of the eigenfunctions, Equation (7.23).

Up to this point, the initial condition has not influenced the derivation. Therefore, this is the only step that needs to change for using different initial conditions for this problem. The scalars are specified using a Fourier series. The Fourier series was developed to solve the heat equation, transforming an arbitrary function between two limits into a summation of sines and cosines.

The Fourier series can be generalized [8, 221] in terms of inner products of basis vectors defined for an interval. We summarize the basic formulae needed for a general Fourier series. The Fourier series for a function $f(x)$ in an interval is written as

$$f(x) = \sum_{n=1}^{\infty} b_n v^{(n)}(x), \tag{7.35}$$

provided the basis vectors satisfy the orthogonality condition $(v^{(n)}(x)\frac{\partial}{\partial x}v^{(n)}(x) = 0)$ at the endpoints of the interval. The constants in the Fourier series, b_n, are defined [8] as

$$b_n = \frac{\langle v^{(n)}(x), f(x)\rangle}{\langle v^{(n)}(x), v^{(n)}(x)\rangle}. \tag{7.36}$$

The standard sine and cosine Fourier series uses an interval of 2π, with $v^{(n)}(x) = \sin(\alpha_n x)$ and/or $v^{(n)}(x) = \cos(\alpha_n x)$, inner products of $\langle v^{(i)}, v^{(j)}\rangle = \int_0^{2\pi} v^{(i)} v^{(j)} dx$, and magnitudes of $\|v(x)\| = \sqrt{\pi}$.

To determine our set of scalars b_n, we must solve Equation (7.36), which depends on $f(x)$. We must express our solution, Equation (7.25), at $t = 0$ in the form of Equation (7.35), in order to determine $f(x)$. Since $T(x, 0) = T_i$, we write Equation (7.25) as

$$(T_i - T_0) - \sigma x = \sum_{n=1}^{\infty} b_n v_1^{(n)}(x) \qquad (x < a) \tag{7.37}$$

$$(T_i - T_L) - \beta(L - x) = \sum_{n=1}^{\infty} b_n v_2^{(n)}(x) \qquad (x > a). \tag{7.38}$$

Comparison with Equation (7.35) allows us to define $f(x)$ as

$$f(x) = \begin{cases} (T_i - T_0) - \sigma x & (x < a) \\ (T_i - T_L) - \beta(L - x) & (x > a). \end{cases} \tag{7.39}$$

The Fourier constants are then solved as

$$
\begin{aligned}
b_n &= \frac{\langle f(x), v^{(n)}(x)\rangle}{\langle v^{(n)}(x), v^{(n)}(x)\rangle} \\
b_n &= \frac{1}{\|v^{(n)}(x)\|^2}\left[\mu D_1 \int_0^a f(x)v_1^{(n)}(x)dx + \nu D_2 \int_a^L f(x)v_2^{(n)}(x)dx\right]. \qquad (7.40)
\end{aligned}
$$

7.1.5 Summary of the Full 1D Solution

We can evaluate Equation (7.40) by straightforward integration and $\|v^{(n)}(x)\|^2$ in Equation (7.34). After some simplification, we have the Fourier coefficients as

$$
\begin{aligned}
b_n = \Big[&\mu D_1\big[(T_i - T_0)(\csc(\lambda_n\mu a) - \cot(\lambda_n\mu a)) + \frac{\sigma}{\lambda\mu}(\lambda\mu a\cot(\lambda_n\mu a) - 1)\big] \\
&+\nu D_2\big[(T_i - T_L)(\csc(\lambda_n\nu\eta) - \cot(\lambda_n\nu\eta)) + \frac{\beta}{\lambda\nu}(\lambda\nu\eta\cot(\lambda_n\nu\eta) - 1)\big]\Big] \Big/ \\
&\left[\frac{\lambda_n}{2}\left[\mu^2 D_1 a\csc^2(\lambda_n\mu a) + \nu^2 D_2\eta\csc^2(\lambda_n\nu\eta)\right]\right]. \qquad (7.41)
\end{aligned}
$$

The following is a summary of the one-dimensional solution of this discontinuous media heat conduction problem,

$$
T(x,t) = \begin{cases} T_0 + \sigma x + \displaystyle\sum_{n=1}^{\infty} b_n e^{-\lambda_n^2 t} v_1^{(n)}(x) & (x < a) \\ T_L + \beta(L - x) + \displaystyle\sum_{n=1}^{\infty} b_n e^{-\lambda_n^2 t} v_2^{(n)}(x) & (x > a), \end{cases} \qquad (7.42)
$$

where the constants needed are

$$
\begin{aligned}
\lambda &= \sqrt{k} && (7.43a) \\
\eta &= L - a && (7.43b) \\
\mu &= \sqrt{\frac{\rho_1 c_1}{D_1}} && (7.43c) \\
\nu &= \sqrt{\frac{\rho_2 c_2}{D_2}} && (7.43d) \\
\sigma &= \frac{D_2}{a\eta}\frac{T_L - T_0}{\frac{D_2}{\eta} + \frac{D_1}{a}} = \frac{D_2(T_L - T_0)}{aD_2 + \eta D_1} && (7.43e) \\
\beta &= \frac{D_1}{a\eta}\frac{T_0 - T_L}{\frac{D_2}{\eta} + \frac{D_1}{a}} = \frac{D_1(T_0 - T_L)}{aD_2 + \eta D_1}. && (7.43f)
\end{aligned}
$$

The eigenfunctions and eigenvalue equation are

$$v_1^{(n)}(x) = \frac{\sin(\lambda_n \mu x)}{\sin(\lambda_n \mu a)} \tag{7.44a}$$

$$v_2^{(n)}(x) = \frac{\sin(\lambda_n \nu (L - x))}{\sin(\lambda_n \nu \eta)} \tag{7.44b}$$

$$\mu D_1 \cot(\lambda_n \mu a) = -\nu D_2 \cot(\lambda_n \nu \eta). \tag{7.44c}$$

7.2 Dependence of Analytical Solution on Eigenvalues

In order to apply the analytic solution, Equation (7.42), one must be able to compute the eigenvalues. There is an infinite set of eigenvalues, but we must use a finite number when computing the solution. We must be able to determine how many eigenvalues we need for the truncated solution to retain sufficient accuracy. This section explores these two topics: finding eigenvalues (Section 7.2.1) and deciding how many to use (Section 7.2.2). This section concludes by choosing the diffusivity ratio and simulation time for the numerical testing of the mixed-cell models (Section 7.2.3).

The results discussed in the text and shown in plots in this and the remainder of this chapter use the following constants:

$$T_0 = 1 \tag{7.45a}$$

$$T_L = 10 \tag{7.45b}$$

$$T_i = 30 \tag{7.45c}$$

$$L = 1 \tag{7.45d}$$

$$a = 0.6 \tag{7.45e}$$

$$\rho_1 c_1 = 1 \tag{7.45f}$$

$$\rho_2 c_2 = 1. \tag{7.45g}$$

While various ratios of D_1 to D_2 are used, the figures and examples given, for consistency, use $D_1 = 10^{-3}$ and $D_2 = 10^3$ whenever possible.

7.2.1 Evaluating the Eigenvalues

Finding the eigenvalues is a key step in the implementation of this analytic solution. The most important eigenvalues in Equation (7.42) are those with small n, due to the negative exponential term and the fact that the eigenvalues are well ordered. While eigenvalues can be found graphically (plotting the functions and manually locating the intersections), an automated method is superior for speed, accuracy, and robustness. Transcendental equation, Equation (7.21), is in terms of cotangents, which have a known period (the period of $\cot ax$ is π/a). The transcendental equation can be written in terms of two functions,

$$f_1(\lambda) = \mu D_1 \cot \lambda\mu a \quad \text{and} \quad f_2(\lambda) = -\nu D_2 \cot \lambda\nu\eta, \tag{7.46}$$

where the eigenvalues exist at the intersections of these functions. We make a distinction between λ_n, the discrete set of eigenvalue, and λ, which is the continuous argument of the functions f_1 and f_2. Finding the eigenvalues is made easier because $\lambda = 0$ marks the beginning of both periods. Calculation of eigenvalues for these equations becomes an exercise in finding the minimum of $|f_1(\lambda) - f_2(\lambda)|$ for λ in the interval defined by the asymptotes of each function, as shown in Figure 7.2.

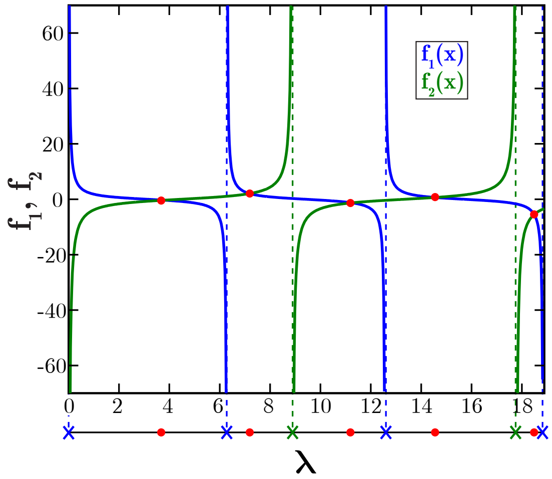


Figure 7.2: The transcendental functions, Equation (7.46), for the 1D eigenvalue solution for $D_1 = 1$ and $D_2 = 2$. A search interval is bounded by the asymptotes for each function, marked with as '×' on the axis below the figure. Each intersection (an eigenvalue) is marked with a red circle.

The algorithm for finding eigenvalues consists of four steps, with the goal of finding the first N eigenvalues. The first step determines the search intervals based on the periodicity of the transcendental functions, as shown in Figure 7.2. More than N intervals should be searched because some intervals do not contain an intersection. The second step finds the minimum of $|f_1 - f_2|$ for each interval; this corresponds to an early estimate of the eigenvalue. Step three searches in a range around each approximate eigenvalue to refine the solution of the transcendental equation as well as to ensure that the functions indeed intersect at the that point. This removes points where the functions do not intersect, as some search intervals contain no intersection (e.g., when one function goes to positive infinity while the other goes to negative infinity). The intersection condition is either that the absolute difference between the functions at a proposed intersection is below some tolerance (0.1 is used) or that the normalized difference is below some tolerance (1 is used),

$$\left|f_1(\Lambda) - f_2(\Lambda)\right| < \text{(absolute difference tolerance)} \tag{7.47a}$$

$$\frac{\left|f_1(\Lambda) - f_2(\Lambda)\right|}{\max(\left|f_1(\Lambda)\right|, \left|f_2(\Lambda)\right|)} < \text{(normalized difference tolerance),} \tag{7.47b}$$

where Λ is the proposed eigenvalue. Failure to normalize by the largest function value causes the algorithm to miss eigenvalues for cases with very large jumps (e.g., 10^6 and higher) where the absolute difference may be in the tens of thousands, but the normalized difference is less than one. Simply looking for a sign change is inefficient for stronger ratios due to the resolution required in λ-space in order to capture the intersection as well the fact that the asymptotes also rapidly change signs. Step four refines the value by iteratively searching in smaller ranges around each intersection point until the change falls to machine precision.

The eigenvalue-finding algorithm has been explicitly verified for ratios from 10^{-10} to 10^{10} up to the first 200 eigenvalues by graphing the functions and confirming intersections. Figure 7.2 illustrates a simple case with a diffusivity ratio of two, while Figure 7.3 shows a case with a much stronger jump ratio (10^{-6}). The latter plot is focused on the first instance where the function with the longer period crosses the λ-axis.

7.2.2 Accuracy of a Truncated Solution

There are three solutions we must distinguish. There is the exact analytical solution, which requires an infinite summation of eigenvalues. There is a truncated analytical solution, which uses a finite number of eigenvalues to approximate the solution. Lastly, there is the calculated solution, which is generated from codes, for which error is quantified using the analytical solution. It is pragmatically necessary to use a truncated analytical solution to compare with the calculated solution, but we must quantify the accuracy of the truncated solution, lest our analysis be in vain.

While it takes many thousands of eigenvalues to resolve a strongly discontinuous solution at $t = 0$, far fewer eigenvalues are needed at any other time because the contribution of an eigenvalue decreases exponentially in both time and magnitude

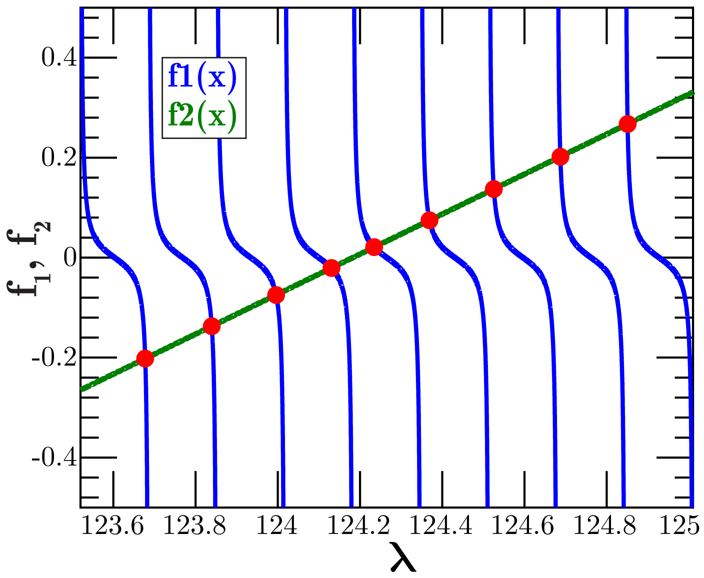


Figure 7.3: Transcendental functions, Equation (7.46), for the 1D eigenvalue solution for $D_1 = 10^{-3}$ and $D_2 = 10^3$. Eigenvalues are indicated with red circles.

$(\exp(-\lambda_n^2 t))$. (Recall that the eigenvalue is technically λ_n^2, but we are referring to λ_n as eigenvalue for brevity.) Thus, at each point in time there is only a finite number of eigenvalues that will change the accuracy of the truncated solution above the accuracy of the calculated solution. Consequently, computational cost can be reduced by using the minimum number of terms (corresponding to the number of eigenvalues) needed to have a truncation error below the discretization error of the codes being validated. There are three factors to consider when comparing the accuracy versus the number of eigenvalue terms: diffusivity ratio, simulation time, and error metric at which the comparisons are made.

Diffusivity ratios. We compare diffusivity ratio cases using $D_1 = 10^{\pm r/2}$ and $D_2 = 10^{\mp r/2}$ with even $s \in \{2, 4, \ldots, 10\}$. The definition of the power of the ratio is

$$s = \pm \log_{10} \frac{D_1}{D_2}. \tag{7.48}$$

Inclusion of ratios up to ten orders of magnitude is considered because some discontinuous coefficient problems in radiative transfer have such high ratios [181].

Simulation time. The choice of the time at which to sample solutions with different diffusivity ratios is more involved. Solutions with different diffusivity ratios take various amounts of time to reach steady-state. In order to compare solutions with various s at equivalent times, one may rescale time in terms of energy content (per unit area for this 1D problem). We define the energy at a given time as

$$\mathcal{E}(t) = \int_0^a \rho_1 c_1 T_1(x,t)dx + \int_a^L \rho_2 c_2 T_2(x,t)dx, \tag{7.49}$$

where ρ_i is the density, c_i is the heat capacity, and the subscript i indicates the phase. Using this energy definition, we introduce a function that characterizes the energy evolution with respect to the steady-state energy for problems with Dirichlet boundary conditions:

$$\xi(t) = \frac{\|\mathcal{E}(t) - \mathcal{E}_{\text{final}}\|}{\|\mathcal{E}_{\text{initial}} - \mathcal{E}_{\text{final}}\|}, \tag{7.50}$$

where $\mathcal{E}_{\text{initial}}$ is the initial energy, $\mathcal{E}_{\text{final}}$ is the steady-state energy, and we assume that $\mathcal{E}_{\text{initial}} \neq \mathcal{E}_{\text{final}}$. The metric $\xi(t)$ would not work with most non-Dirichlet boundary conditions, such as a no-flux or time dependent boundary condition. Equation (7.50) gives unity at $t = 0$ and approaches zero as $t \to \infty$.

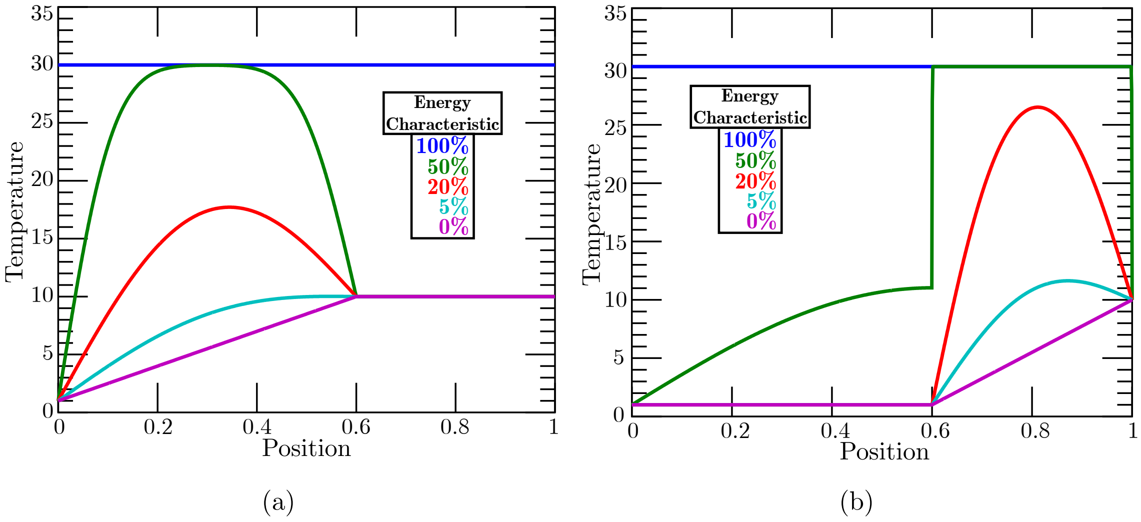


Figure 7.4: 1D MES solutions at different points in the energy evolution. The diffusivity ratio shown are (a) $s = -6$ and (b) $s = 6$. Other parameters are given in Equation (7.45). The energy characteristic is defined in Equation (7.50), where 100% indicates the system has all of its initial energy, while 0% means it has lost all excess energy and equilibrated to steady-state.

Using the energy characteristic, $\xi(t)$, as a means to measure the progress of the solution towards equilibrium, we define three points in time: early, when $\xi(t)$=0.50;

intermediate, when $\xi(t)$=0.20; and late, when $\xi(t)$=0.05. It would not be useful to use times after steady-state is reached because the eigenvalues make no contribution to the solution by that point. Table 7.1 presents the simulation time to reach the early-, intermediate, and late-stage points for various s values. Figure 7.4 shows plots of the analytic solution for $s = -6$ and $s = 6$, each plotted at five different energy states: initial condition, early, intermediate, late, and steady-state, $\xi(t) \in \{1.0, 0.5, 0.2, 0.05, 0\}$. Note that the sign of the ratios is important because of the arrangement of the phases with respect to the boundary conditions. The specific values of $|r|$, however, do not make a significant difference because the smaller diffusivity is already much smaller than the larger value, even for $|r| = 2$. Thus, solutions with different diffusivity ratios (of the same sign) compared at the same point of their energy evolution look almost identical.

Error metric. The truncation error, e_2, is measured with the 2-norm,

$$e_2 = \sqrt{\sum_{i=1}^{M} \left[s_i(t) - S_i(t)\right]^2 \frac{1}{M}}, \tag{7.51}$$

where M is the number of points in space at which the solution is sampled, $s(t)$ is the n-term truncated solution, $S(t)$ N-term truncated solution representing the exact solution, and t is the time the solutions are compared. The truncation error is computed for $s(t)$ for $n \in \{1, 2, \dots\}$ until the error falls to machine precision (2.2×10^{-16}). In the event that that $s(t)$ requires as nearly many eigenvalue terms as $S(t)$, i.e., n is near N (say $n > 0.85N$), N is increased and the error is recalculated. The result of this calculation is the truncated-solution error versus the number of eigenvalues terms used. Figure 7.5 plots the error versus number of eigenvalues for various diffusivity ratios at early-stage time, and Table 7.1 shows number of eigenvalues needed to reach machine precision truncation error for all the energy characteristic stages and each s value considered.

Note that the truncation error, Equation (7.51), is only weakly dependent on the spatial resolution (M). This is verified by comparing various M values for early-stage results between $N = 1000$ and $n = 16$ on a problem with $s = -6$. Spatial resolutions $M \in \{250, 500, 1000, 2000\}$ have corresponding errors $e_2(M) \in \{2.8929, 2.8981, 2.8998, 2.8997\} \times 10^{-13}$, which shows that a factor of two increase in the spatial resolution has negligible effect on the overall error.

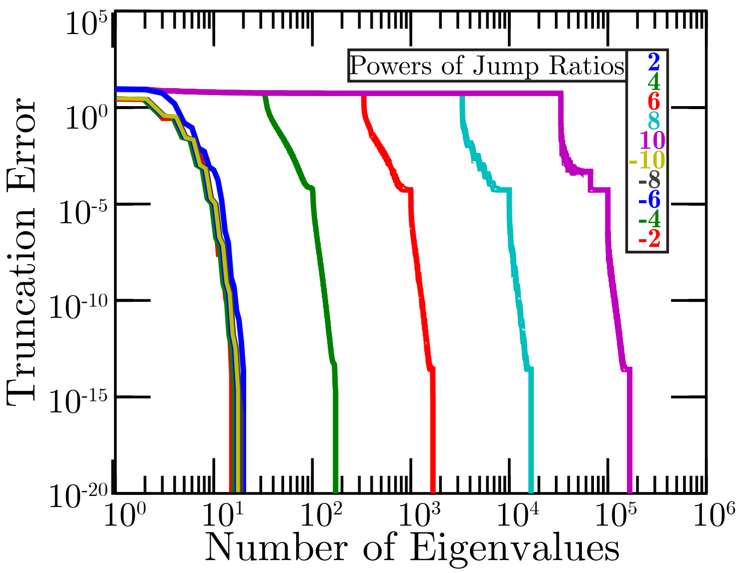


Figure 7.5: Solution accuracy versus number of eigenvalues for 1D analytic solution at early-stage. The logarithmic plot measures the solution at early-stage, defined by $\xi(t) = 0.50$.

Table 7.1 shows the time for solutions with various values of s to reach early-, intermediate-, and late-stage energy contents. The table also shows the number of eigenvalues that are needed to represent the solution to machine precision. It can be seen for all intermediate- and late-stages (and early-stage for $s < 0$) that the time to reach each stage increases by about a factor of ten for every factor of 100 increase in $|s|$. This trend is because the diffusion time scales inversely with diffusivity ($\Delta t \propto 1/D$) and because the time scale is controlled primarily by the smaller diffusivity, which decreases by a factor of ten when $|s|$ increase by a factor of 100. Only the early times for positive ratios break this trend, where their time decreases by about a factor of ten for every factor of one hundred increase in $|s|$. This trend is because most of the energy loss occurs in phase 1, which has a larger diffusivity and lower boundary condition (i.e., this trend would be reversed if $T_0 > T_L$). Thus, the time to reach the early-stage is governed by the faster diffusivity, which increases by a factor of ten.

Another trend in Table 7.1 is that the positive ratio cases evolves more rapidly than the equivalent negative ratio cases. This is because the boundary condition is lower for phase 1, thus when $D_1 > D_2$, energy can escape the system more rapidly than $D_1 < D_2$. Because the $s > 0$ cases evolve more rapidly, more eigenvalues are needed at all stages compared to the $s < 0$ cases, simply because the number of eigenvalue terms needed depends on the simulation time (both cases use the same eigenvalues). All

ratio	Time			Number of Eigenvalues		
s	Early	Int.	Late	Early	Int.	Late
2	1.564E-2	9.019E-2	0.3189	21	9	5
4	1.867E-3	0.6232	2.863	173	9	4
6	1.904E-4	6.207	28.60	1,676	9	4
8	1.904E-5	62.07	286.0	16,674	9	4
10	1.904E-6	620.7	2860	166,674	9	4
-2	4.593E-2	0.3655	0.8781	16	6	3
-4	0.3653	3.517	8.574	18	5	3
-6	3.644	35.16	85.72	18	5	3
-8	36.44	351.6	857.2	19	5	3
-10	364.3	3516	8,572	19	6	4

Table 7.1: 1D MES analytic solution trends depending on ratio of phase coefficients. Solution times to reach early-, intermediate-, and late-stage are listed. Also, the number of eigenvalues needed for the solution at these three times to have an error in the 2-norm near machine precision are listed.

intermediate- and late-stage solutions need less than ten eigenvalues, and most early-stage cases need twenty or fewer. However, $s > 0$ cases need thousands of eigenvalues for the early-stage. A comparison of the early-stage plot (green) in Figure 7.4(a) and Figure 7.4(b) shows that the positive-ratio case contains sharp temperature gradients (which require more eigenvalues to resolve than a smooth curve) because the system is at the earlier time even though half of the energy has been lost. If we instead compared the different ratio cases at the same time for a given set up (rather than at the same energy characteristic stage), we would find that they require an equivalent number of eigenvalues.

Figure 7.5 shows the 2-norm solution truncation error versus number of eigenvalues at early-stage time, where the negative ratio cases overlap and are difficult to distinguish. The trend in Table 7.1 that $s < 0$ diffusivity ratios require more eigenvalues is also clear in Figure 7.5.

Note that the behavior of a positive ratio versus a negative ratio depends on the choice of boundary conditions, where $T_0 < T_L$. If this were reversed ($T_0 > T_L$), the trends and numbers would all switch between positive and negative ratios. Using $T_0 = T_L$ may give different trends altogether, but the purpose of this chapter is to use this solution to examine mixed cells, not to fully explore the behavior of the analytic solution. Therefore, we use the parameters specified in Equation (7.45) for the results presented in the remainder of this chapter.

7.2.3 Selection of Numerical Testing Parameters

The purpose of this chapter is to create an analytic solution to test the mixed-cell models. Therefore, it is not important what s and simulation time are used for the comparison, so long as the mixed-cell is a critical point (i.e., the temperature profile has distinguishing features in the mixed cell) and the computational effort is minimized.

We can use Table 7.1 and Figure 7.4 to choose the values for the computational analysis. Examining Figure 7.4 shows a temperature kink at the interface for all plots except the $\xi(t) = 0.05$ value for the negative ratio case. This kink is important because it makes the temperature of the mixed cell more meaningful for a test of accuracy than if the solution were smooth at the interface.

We choose the early-stage time, $\xi(t) = 0.50$, for the negative ratio diffusivities, $s < 0$, for the mixed-cell test. As shown in Table 7.1, this particular set occurs at an early simulation time (minimizing the runtime of the simulations) and requires fewer than twenty eigenvalues (unlike the early-stage for the $s > 0$ case). Furthermore, examination of Figure 7.4(a) shows the temperature profile at this stage contains a prominent discontinuity in the gradient.

7.3 Results of Mixed-Cell Models for 1D Analytical Test

This section presents the performance of the mixed-cell models used in this study to this 1D MES test problem. The similarities and differences between the six models are summarized in Table 6.1. Three single temperature models are presented: the harmonic mean (S1), the rotated tensor-diffusivity method (S2), and the split-cell method (S3). (See Chapter III for the description of these models.) A multi-temperature model is also presented with three variants, based on the level of the approximation (M1, M2, M3). (See Chapter IV for the description of these models.) Five of these six models represent new work, with the harmonic mean (S1) used as a baseline. The accuracy of a model's results in comparison to the harmonic mean determines if the method is advantageous as compared to the standard treatment.

7.3.1 Implementation

The implementation of the 1D analytic solution is straightforward (in comparison with the 2D analytic solution of Chapter VIII). No-flux (reflecting) boundary conditions are implemented for the y-direction boundaries, while Dirichlet boundaries are implemented for the two x boundaries. The no-flux boundaries for the y-axis convert the problem from 2D to 1D, with the physical interpretation that the rod is insulated along its length, only able to conduct heat at the ends. The Dirichlet boundary conditions are $T(0,t) = T_0$ and $T(L,t) = T_L$. The problem is initialized by setting $T(x) = T_i$. The mixed-cell simulations are run for various grid refinements to characterize convergence trends. The temporal discretization is first-order accurate while the spatial discretization is second-order accurate. The time-step is scaled conservatively as $\Delta t \propto \Delta x^3$ in order to ensure the temporal error is below the spatial error. The analytic solution is not needed until the errors of the mixed-cell model are computed.

We first note that the time when the solutions are sampled does not affect the trends. The caveat to this observation is that all methods have the same initial conditions and converge to steady-state after sufficient time, so the time chosen to sample the solutions must be between these two points. In this interval, the plots of the errors of a method versus time do not cross; if one method has a lower error, the error is lower at *any* time in this interval. This is verified with a ratio of 10^{-6} on a 20×20 grid. E_1, E_2, or E_∞ plots versus time do not cross for neither the mixed nor pure case. Therefore, we can present the results by sampling the solutions at a single time, which we choose to be when the energy content reaches the early-stage. Second, we note that all the methods yield the same level of error when using pure-cells only.

Relative errors in three different p-norms are used: E_1, E_2, and E_∞. Each of these errors has an equivalent convergence rates: q_1, q_2, and q_∞. All cases are compared at a single time, defined by $\xi(t) = 0.50$. Each case is tested on six grid sizes, $N \in \{10, 20, 30, 40, 50, 100\}$, where N is the number of elements in the 1D solution. Three diffusivity ratios are used, $s = \log_{10} \frac{D1}{D2} \in \{-2, -6, -10\}$.

In this 1D problem, there is only one mixed cell, having a volume fraction of any value between zero and one. Three volume fraction cases are tested: $\frac{1}{3}, \frac{1}{2}, \frac{17}{20}$. Note that in order to have a set volume fractions on a uniform grid, the interface location must be flexible from one resolution to the next. Additionally, since the boundaries are defined differently in the single- and multi-temperature cases (face centered versus cell centered), each of these methods has a unique set of interface locations. Interface position, a, is constrained to be as near to $a = 0.6$ as possible in a range of $0.5 \leq a \leq 0.6$ for the single-temperature model and a range of $0.55 \leq a \leq 0.65$ for the multi-temperature model. The interface locations are presented in Table 7.2.

	Single Temperature			Multiple Temperature		
grid	v1 (33%)	v2 (50%)	v3 (85%)	v1 (33%)	v2 (50%)	v3 (85%)
10	8/15	11/20	117/200	7/12	3/5	127/200
20	17/30	23/40	237/400	71/120	3/5	247/400
30	26/45	7/12	119/200	107/180	3/5	367/600
40	7/12	47/80	477/800	143/240	3/5	487/800
50	44/75	59/100	597/1000	179/300	3/5	607/1000
100	89/150	119/200	1197/2000	359/600	3/5	1207/2000

Table 7.2: Interface location for single- and multi-temperature methods required for 1D MES test problem. The values listed in the table are the interface location, a, for a domain size of unity. Phase 1 occupies the LHS of the domain, and the volume fractions refer to this phase. The interface location depends on the grid size (denoted simply as 'grid'), volume fraction of the mixed cell ($\frac{1}{3}$, $\frac{1}{2}$, $\frac{17}{20}$), and model type (single- or multi-temperature). Overall, phase 1 accounts for about 60% of the volume fraction.

7.3.2 Results

This section presents the convergence rates of all six mixed-cell models for the 1D test case developed in this chapter. Nine different variations of the test are examined (three volume fractions and three diffusivity ratios) for six different mixed-cell models. The volume fractions, f, used are v1 (33%), v2 (50%), and v3 (85%). The phase 1 multi-temperature results are given as well as the phase-averaged results. The phase 2 results are not informative because, as shown in Figure 7.4(a), equilibrium is reached in this phase. Therefore, the temperature in this phase is (nearly) constant, and the error does not represent discretization error.

Table 7.3 presents the averaged q_∞, while additional convergence information (such as q_1 and q_2 and the convergence rate for each grid) can be found in Tables C.3, C.4, and C.5. The two most obvious conclusions from Table 7.3 are that the S1 and S2 models give similar results, as do the M2 and M3 models.

The degeneracy of the S1 and S2 models in 1D occurs because the tensor rotation model (S2) reduces to the harmonic mean along the x-direction, which is the approach of the S1 method. The results are not identical, which is only distinguishable in the $s = -10$ case. The S1 model uses a scalar diffusivity that is equal to the harmonic mean, weighted by volume fractions, of the diffusivities in the mixed cell. The S2 model uses a tensor diffusivity, which is, for this set up, the harmonic mean for the

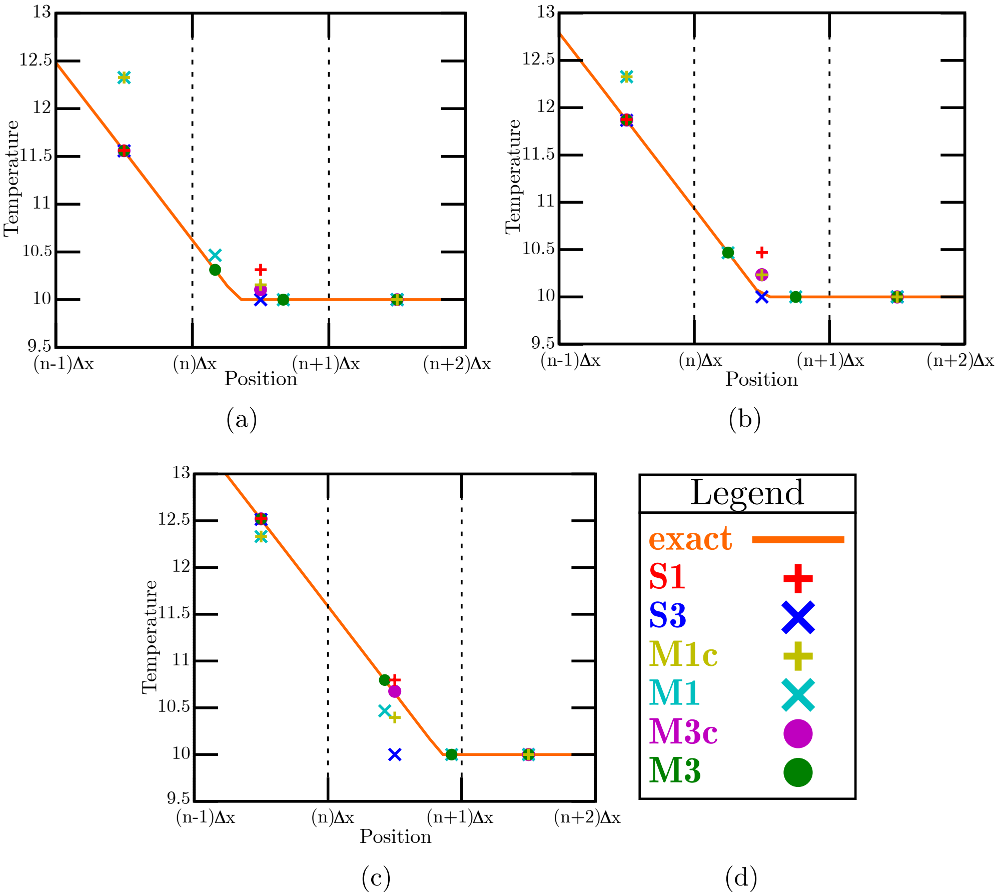
xx tensor component, the arithmetic mean for the yy tensor component, and zero for the xy tensor component. Although the y-direction is constant due to the no-flux boundary conditions, the anisotropic tensor causes slight variations between these models due to the matrix operations performed by SOM when calculating the flux.

The degeneracy of the M2 and M3 methods occurs because, in 1D, the centroid-to-centroid lines can only be orthogonal to a cell face or a interface. Thus, the distinguishing feature between the M2 and M3 models is not present. The centroid-to-centroid line can only be off-axis in 2D (see Figure 4.2), which is only accounted for by the M3 model. Because of these degeneracies, we limit our discussion to the S1, S3, M1, and M3 models.

f	s	S1	S2	S3	M1c	M2c	M3c	M1	M2	M3
33%	-2	1.21	1.21	**2.12**	0.55	0.99	0.99	0.55	1.78	1.78
	-6	1.17	1.17	**2.07**	0.58	1.02	1.02	0.58	1.83	1.83
	-10	1.18	1.15	1.53	0.58	1.03	1.03	0.58	1.82	1.82
50%	-2	1.17	1.17	**2.07**	0.63	1.00	1.00	0.51	1.67	1.67
	-6	1.17	1.17	**2.07**	0.63	1.00	1.00	0.53	1.67	1.67
	-10	1.17	1.15	1.51	0.63	1.00	1.00	0.53	1.66	1.66
85%	-2	1.41	1.41	0.99	0.74	**1.92**	**1.92**	0.74	**1.95**	**1.95**
	-6	1.45	1.45	0.99	0.83	**1.91**	**1.91**	0.83	**1.93**	**1.93**
	-10	1.45	1.43	0.99	0.82	**1.95**	**1.95**	0.82	**1.98**	**1.98**

Table 7.3: Average q_∞ for the 1D MES mixed cell test problem. Three mixed-cell volume fractions, f, are tested using three different diffusivity ratios. The convergence rates for phase 2 are shown for the multi-temperature models. The convergence rates are averaged for grids of size 10, 20, 30, 40, and 50. Additional convergence rates are presented in Tables C.3, C.4, and C.5. Bold indicates $q_i \geq 1.90$.

Figure 7.6 plots the temperature profile for all three volume fractions for the $s = -6$ ratio by aligning the interface location. Each case shows the results for the S1, S3, M1, and M3 models. The multi-temperature models show both the cell-centered value (indicated as M1c and M3c) as well as the phase-centered values (indicated as M1 and M3). Three cells are shown in each graph, from left to right: pure phase 1, mixed cell, pure phase 2. In all phase 2 pure cells, the six methods overlap. This is because the solution for this phase is nearly constant and all methods perform equally well in this case. This justifies the omission of the phase 2 convergence rates (which are zero) for the multi-temperature method from Table 7.3.



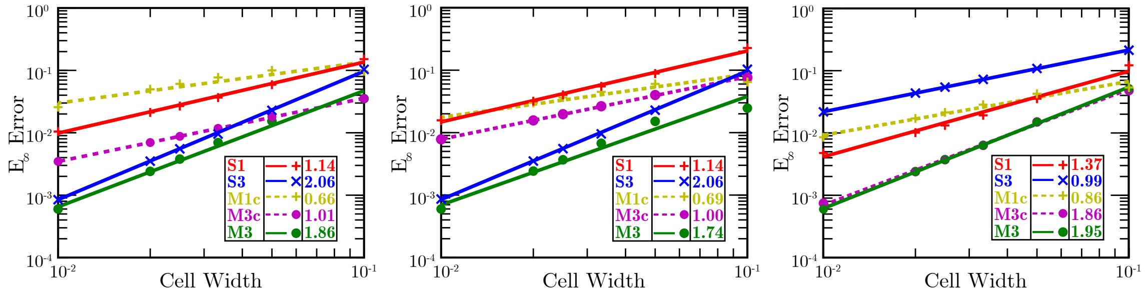
(a)

(b)

(c)

(d)

Figure 7.6: Results for 1D MES test problem for various mixed-cell models. The problem uses a phase ratio of $s = -6$ on a 100 element grid. Each figure shows three cells: a mixed cell flanked by two pure cells (i.e., the dotted lines and vertical plot boundaries represent cell faces). The coordinate system of the single- and multi-temperature methods, which differ due to face centered versus cell centered boundaries, have been shifted such that the interfaces align. Three volume fractions (in terms of the phase left of the interface) are shown: (a) v1 (33%), (b) v2 (50%), and (c) v3 (85%). The legend is shown in (d). The interface occurs at the kink of the exact solution. Both the phase-averaged (M1c, M3c) and phase-centered (M1, M3) solutions are shown for the multi-temperature method.



(a) (b) (c)

Figure 7.7: Error versus cell width calculated with the ∞-norm for 1D MES test problem. The $s = -6$ case is shown logarithmically for volume fractions (a) 1/3, (b) 1/2, and (c) 17/20. The lines are the least-squares fit to E_∞ data (circles). The slopes (convergence rates) are indicated in the legend of each plot. Four mixed-cell models are shown, with the M3 model showing both the phase-averaged (labeled as M3c) and phase 1 (labeled as M3) results.

Figure 7.7 plots E_∞ versus cell width on a log-log scale, where the slope of each least-squares fit line (indicated in the legend) corresponds to the convergence rate. The $s = -6$ ratio is shown for all three volume fractions. The relative error values of each model can be compared within each figure. We observe that the M3 model has the lowest error value in each case, and the S3 model has the second lowest error overall for v1 and v2, but it has the highest error in v3.

7.3.3 Discussion

As noted earlier, we discuss only the S1, S3, M1, and M3 models: in 1D, the results of the S2 model are equivalent to the S1 model, and the M2 model is indistinguishable from the M3 model.

S1. Table 7.3 shows that the S1 model, which is used as a reference point for all the other models, performs at consistent rates for each diffusivity ratio in a given volume fraction. The average convergence rate for E_∞ in all cases is 1.23. Thus, the S1 model does not have second-order accuracy in the ∞-norm for any case.

For the $s = -2$ and $s = -6$ ratios, the average convergence rates are 2.01 and 1.79 for the 1-norm and 2-norm, respectively; see Tables C.3 and C.4. However, for the $s = -10$ ratio, these convergence rates are drop to 1.54 and 1.58. This performance degradation suggests that such this large of a ratio may be beyond the capabilities of SOM. Figure 7.7 shows that, in terms of absolute error value, the S1 model is the among the least accurate methods. Examination of Figure 7.6 shows that the S1 model performs best in v3, which is where the S3 model performs worst.

S3. It is clear from Table 7.3 that the S3 model has the highest convergence rates for a cell-centered result (i.e., only considering the single-temperature and phase-averaged multi-temperature codes). It is also evident that the S3 model has a performance drop for v3 and $s = -10$ ratio cases. Interestingly, every other model (S1, M1, M3) achieves its best performance in v3 and maintains consistent performance in all three diffusivity ratios for a given volume fraction. Only the S3 model breaks these trends.

Ignoring the $s = -10$ ratio, we observe from Tables C.3 and C.4) that the S3 model is second-order accurate in the ∞-norm for when $f \leq \frac{1}{2}$, but it is only first-order accurate when $f > \frac{1}{2}$. This is consistent with the results found in the 1D MES test in Section 6.1. For v3 (f=85%), the S3 model has the least accuracy, as is evident in Figures 7.7(c) and 7.6(c).

If we consider only v1 and v2 while continuing to ignore the $s = -10$ ratio, the S3 model has the ideal performance, which is evident in plots (a) and (b) in Figures 7.7 and 7.6. Consulting Tables C.3 and C.4 for these cases shows the average convergence rates of the S3 model are 2.02, 2.10, and 2.07, for q_1, q_2, and q_∞, respectively. Therefore, the S3 model is the ideal mixed-cell model for these cases.

However, the performance of the S3 model is inconsistent since it has second-order convergence rates only in certain cases. Using Table 7.3, let us compare the convergence rates between the S3 and S1 models for the cases where S3 is not second-order accurate. The S3 model has higher convergence in the ∞-norm for the $s = -10$ ratio and volume fractions v1 and v2, but S1 has higher rates for the 1-norm and 2-norm. Since the ∞-norm is the best indicator for the accuracy of a mixed-cell model, we conclude the S3 model outperforms the S1 model for all ratios in the first two volume fractions. For v3, where phase 2 (the larger diffusivity) is the minority phase, the S3 model is outperformed by the S1 model in q_∞ at every diffusivity ratio. The fact that the S1 model performs well in this case is not surprising because the harmonic mean weights the smaller diffusivity. The drop in the performance of the S3 model for v3 suggests that, unlike the harmonic mean, the S3 model weights the larger diffusivity more. The fact that the S3 model predicts the same temperature, equal to the nearly constant phase 2 temperature, in all three volume fraction cases (see Figure 7.6) supports this observation.

The main trend for the S3 model is that it is second-order accurate in all three error norms when the majority phase is the larger diffusivity (and $|s| < 10$) but it is only first-order accurate in the ∞-norm when the smaller diffusivity is the majority phase. Modifications to the split-zone model were tested (including changing the subcell volumes and diffusivities), but none was found to be equally accurate or better than the implementation described in Appendix A (note that in 1D, interface can only cross opposite sides for a mixed cell). This limitation of the S3 model in 1D could potentially hinder the success of this model in 2D.

M1. As seen in Tables 7.3, the M1 model has the lowest convergence rate, less than first-order accurate in every case. This includes both the cell-centered and phase-centered results. Figure 7.6 shows that the M1 model is farthest away from the exact solution in phase 1, while Figure 7.7 shows that it consistently ranks amongst the two least accurate models. Hence, we can conclude that the M1 model, which makes a very simple assumption, is insufficient. The S1 model easily outperforms the M1 model, indicating one would be better off using a standard single-temperature discretization with a harmonic mean of diffusivities for mixed cells rather than the M1 model.

M3. Considering first the phase-averaged results, we observe that the M3 model is second-order accurate in v3, but is only first-order accurate in v1 and v2 (the inverse of the S3 model). The average convergence rate for the cell-centered M3 model is 1.31, which is slightly better than the average rate of the S1 model (1.26).

However, the main advantage of the M3 model is truly evident when the phase-centered results are considered. When we consider the phase 1 results, the M3 model is indisputably the most accurate mixed-cell model in this test. The M3 model maintains consistent performance for every diffusivity ratio tested, i.e., it does not suffer a performance drop in the $s = -10$ ratio as in the S3 model. The average convergence rates for the M3a model in Table 7.3 is 1.81, while the S3 model has an average of 1.59 when all volume fractions and ratios are considered. Examination of Figure 7.6 shows that values of the M3 models match the exact solution for both phases in all three volume fractions. It is evident from Figure 7.7 that the M3 model has the lowest error and highest or second-highest convergence rate in all three volume fractions. The low error the M3 model for the coarsest grids, most noticeable in Figure 7.7(b), makes the convergence rate lower for the $10 \rightarrow 20$ case than any other set of grids. We compute the convergence rates while ignoring the coarsest grid, as shown in Table 7.4, using Tables C.3, C.4, and C.5. When the coarsest grid results are neglected, we find that the M3 model is second-order accurate in all three volume fractions for all three ratios, as shown in Table 7.4. Thus, the phase-centered M3 model has the best overall performance for this test.

	v1			v2			v3		
s	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
-2	**1.99**	**2.01**	**2.01**	**1.94**	**1.96**	**1.98**	**1.94**	**1.95**	**1.98**
-6	**2.01**	**2.03**	**2.03**	**1.97**	**1.98**	**2.00**	**1.95**	**1.96**	**1.99**
-10	**2.01**	**2.03**	**2.04**	**1.96**	**1.97**	**1.99**	**1.97**	**1.96**	**1.99**

Table 7.4: Average convergence rates for the M3 model in 1D MES mixed cell test problem. The results come from averaging the convergence rates in Tables C.3, C.4, and C.5 for all sets of grids except 10 → 20. Bold indicates $q_i \geq 1.90$.

Potential New Models Results indicate that the S1 and S3 models could be combined to form a new, more accurate single-temperature model. The S3 model has excellent performance in the volume fractions where the majority phase has a larger diffusivity (v1 and v2) but does poorly in v3. The average convergence rate of the S1 model in v3 is 1.43, which is better than that of the S3 model (0.99). This suggests that a best single-temperature model may be one that dynamically chooses between the S1 and S3 models, depending on the diffusivity of the majority phase. Tables C.3, C.4, and C.5 indicate that such a model would have accuracy of 1.9 or better in all three error norms for all cases save one; E_∞ when the majority phase is the smaller diffusivity (S1 model) would only have a convergence rate of 1.4. (In the above discussion, we ignore the $s = -10$ ratio, where the performance of the single-temperature methods suffers.) Furthermore, a combination of the S3 model with the phase-centered M3 model would give a q_∞ greater than 1.9 in every case.

7.3.4 Conclusion

The M3 model (as well as the 1D degenerate M2 model) is the best method as measured by this 1D MES analytic test, while the M1 model is the worst model. These models perform consistently in all volume fractions and diffusivity ratios, including the largest ratio. The S3 method has the second best performance in this test, followed by the S1 and its 1D degenerate S2 model. The S3 model is only first-order accurate when the majority phase is the smaller diffusivity and when the $s = -10$; it is second-order in every other case.

However, the final verdict for best method depends on the application. The motivation for this thesis was to create a cell-centered diffusion scheme that could be coupled to a cell-centered hydrodynamics scheme. While the phase-centered M3 method has the highest convergence rate, the results are not cell centered, and the averaging process introduces additional errors (as seen in the M3c results in Table 7.3). Consequently, the M3 method would only be recommended if the code for the application employing the mixed-cell models (e.g., a hydrodynamics scheme) could input distinct temperatures for each phase in a cell. If the application requires a cell-centered solution, the S3 method would be the best choice.

Another key difference is that the M3 model requires interface reconstruction in order to accurately locate the centroids prior to computing the flux for each phase. The S3 model, alternatively, only uses volume fractions and makes a much less sophisticated estimate of the interfacial position. Therefore, if the code does not already have higher-order interface reconstruction, the S3 model is a superior choice. Furthermore, avoiding interface reconstruction is beneficial in 3D due to its computational demand and complexity.

7.4 Summary

We presented a derivation of a 1D, time dependent, two phase, analytical solution to a composite-media heat-conduction problem. The solution is an eigenvalue problem expressed as a Fourier series that accepts any ratio of diffusivities. Details for finding the eigenvalues and quantifying a truncated solution are discussed.

This solution is used to test the six mixed-cell models presented in Chapters III and IV. We were indeed able to find models that improved upon the convergence rate of the S1 model, which is a standard method of using the harmonic mean of the diffusivities in a mixed cell. We found that the M1 model is the least accurate, while the M2 and M3 models (which give identical results in 1D) provide the highest accuracy. However, the M2/3 models performance is best when resulting temperatures of two phases in a mixed cell are not averaged. Thus, the M2/3 models would be most beneficial if if the application requiring a mixed-cell model were able to track the temperatures for each phase separately. If the application can only take a single, cell-centered temperature for a mixed cell, the S3 model is the next best choice. We also proposed two new models that would improve upon the deficiencies of the S3 model.

CHAPTER VIII

Numerical Results: 2D Exact Solution

This chapter summarizes the error analysis to investigate the precision of different mixed-cell models using a discontinuous-media problem for a 2D composite disk. This problem is solved by a radial solution in cylindrical coordinates that we project onto a 2D grid to create mixed cells with various volume fractions and interface angles. The analytical solution to this problem serves as a 2D benchmark test against which the mixed-cell models introduced in this thesis are compared. The solution is a simplified version of that presented by Mikhailov, Özişik, and Vulchanov [147]. In Section 8.1, we first derive the analytic solution via separation of variables (SoV). This is followed by an analysis of the eigenvalue behavior in Section 8.2, considering how accuracy is affected by diffusivity ratios, solution time, and number of eigenvalues used to represent the solution. Results of each mixed-cell model are compared in Section 8.3, followed by a discussion in Section 8.4.

The heat equation in cylindrical coordinates is

$$\rho c \frac{\partial T(r,\phi,z,t)}{\partial t} = D\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \phi^2} + \frac{\partial^2}{\partial z^2}\right) T(r,\phi,z,t), \tag{8.1}$$

where ρ is density, c is heat capacity, $T(r,\phi,z)$ is temperature, D is diffusivity, r is the radial coordinate, ϕ is the polar angle, z is the height coordinate, and t is time. Our problem is a disk of radius R with phase 1 for $0 \leq r \leq a$ and phase 2 for $a \leq r \leq R$, as shown in Figure 8.1. In 2D, we assume T to be independent of z, thus $T = T(r,\phi,t)$.

We assume continuity of temperature at the interface. However, physical situations may have thermal contact resistance at phase or material interfaces due to surface roughness and voids. Thermal contact resistance leads to a jump in the temperature at the interface [107, 192]. However, this complication could be added to the test problems with relatively little difficulty in order to match the test with realistic materials.

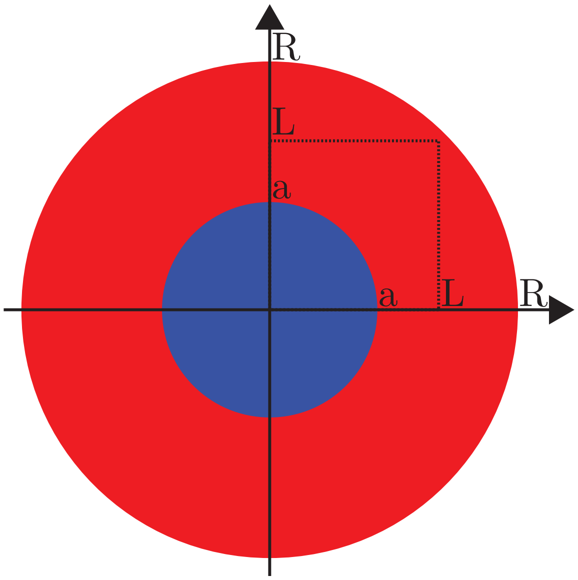


Figure 8.1: Two dimensional domain considered for the circular-interface heat-conduction problem. Phase 1 is the inner (blue) circle of radius a, phase 2 is the outer (red) region from radius a to R. The computational domain is a square of length L (dashed line), where $0 < a < L < R$.

As in the 1D case, we define the PDE on each domain occupied by a phase, and relate the two domains by continuity of temperature and flux. In this problem, a composite disk is initially at a uniform temperature T_i. After $t = 0$, the perimeter of the disk is held at constant temperature T_R. A physical example of this problem is a short, thick rod at uniform temperature T_i that is dipped into a thermal reservoir (at temperature T_R). The rod is insulated at the ends and only conducts heat along the radial direction (the opposite situation of the 1D rod in the previous chapter). The temperature profile is identical for any disk cross section perpendicular to the axis of the rod.

The set of boundary conditions and continuity constraints that specify the problem are given as

$$T(r,0) = T_i \qquad \text{Initial Condition} \tag{8.2a}$$

$$\left.\frac{\partial T(r,t)}{\partial r}\right|_{r=0} = 0 \qquad \text{Inner Boundary} \tag{8.2b}$$

$$T(R,t) = T_R \qquad \text{Outer Boundary} \tag{8.2c}$$

$$T(a-,t) = T(a+,t) \qquad \text{Continuity of Temperature} \tag{8.2d}$$

$$D_1 \left.\frac{\partial T(r,t)}{\partial r}\right|_{r=a-} = D_2 \left.\frac{\partial T(r,t)}{\partial r}\right|_{r=a+} \qquad \text{Continuity of Flux.} \tag{8.2e}$$

The notation $r = a-$ indicates r approaching a from the center, and $r = a+$ indicates r approaching a from the outside.

The justification for the choice of a constant initial condition is the same as in Chapter VII: it avoids numerical complications. It is less clear how to assign the temperature of a mixed cell when there is a gradient or if each material has a different temperature as compared to when the temperature is constant across the entire cell. The accuracy of the test becomes contingent on the initial implementation of the temperature profile if the initial condition has a gradient (or kink) at the interface. A constant initial condition avoids this problem entirely, allowing the analysis to focus on the fidelity of the mixed cells without the complication of introducing any potential inaccuracy at the start of the problem.

The solution is a 1D radial function in cylindrical coordinates because we assume symmetry in ϕ and z. This 1D solution is then projected onto a Cartesian mesh. The advantage of this projection is that a Cartesian mesh cannot resolve a circular interface at any resolution. Although normally this would be a disadvantage, this projection guarantees the presence of mixed cells at various ratios and orientations and, thus, provides a more challenging test for our mixed-cell models. Only some of the circular domain can be represented on the Cartesian computational domain. The problem domain is $0 \le r \le R$, while the computational domain is a square that fits inside the circle of radius R. While this rectangle could be anywhere inside the circle, including at an asymmetric location, we choose to use a square of length L in the positive quadrant, where $0 < a < L < R$ and $\sqrt{2}L < R$; see Figure 8.1. This (symmetrical) choice of computational domain reduces the computational demand when implementing the boundary conditions. (See Section 8.3.2.1 for alternative configurations for this test.)

In the 1D MES test problem (Chapter VII), the computational domain and boundary conditions matched those of the analytic solution; therefore the eigenvalues and analytic solution needed to be computed only as a post-processing step. For the 2D problem, both the computational domain and boundary conditions are different from those of the analytic problem. The computational domain is a subset of the analytic domain, and the boundaries of the computational domain must be specified as Dirichlet boundary conditions using the known analytic values at every time-step. Prescribing the boundary conditions in such a manner makes this a computationally expensive test, especially since the solution depends on numerous eigenvalues and is comprised of special (Bessel) functions. However, this is still a valuable mixed-cell test, especially since it tests two-dimensional effects.

8.1 Derivation of the Analytical Solution

This 2D solution is a simplified version of that presented by Mikhailov, Özişik, and Vulchanov [147]. The derivation of the 2D MES analytical solution closely mirrors the derivation for the 1D MES solution presented in Section 7.1. As in the 1D case, the problem is simplified by treating each phase separately, yielding a diffusivity that is not a function of position. The solution for each phase is related by enforcing continuity of temperature and flux at the interface. There exists a set of solutions (eigenfunctions) that satisfies the heat equation in each phase, with a constant for each eigenfunction determined by the initial condition.

We first apply SoV and determine the time dependence of the solution (Section 8.1.1). This is followed by the steady-state solution (Section 8.1.2) and the transient solution (Section 8.1.3). The full solution is presented in Section 8.1.4, summarizing all terms and constants required to express the analytical solution.

8.1.1 Separation of Variables and Time Dependence

We begin with the SoV assumption that the solution depends on the product of functions of a single variable, $T(r, \phi, t) = \tau(t)\Gamma(r)\Phi(\phi)$. The diffusion equation in polar coordinates leads to a separated equation in the following steps:

$$\begin{aligned}\left(\frac{\partial}{\partial t} - \frac{D}{\rho c}\left(\frac{\partial^2}{\partial r^2} + \frac{1}{r}\frac{\partial}{\partial r} + \frac{1}{r^2}\frac{\partial^2}{\partial \phi^2}\right)\right)\tau(t)\ \Gamma(r)\Phi(\phi) &= 0 \\ \tau_t\ \Gamma\Phi &= \frac{D}{\rho c}\left(\tau\ \Gamma_{rr}\Phi + \frac{1}{r}\tau\ \Gamma_r\Phi + \frac{1}{r^2}\tau\ \Gamma\Phi_{\phi\phi}\right) \\ \frac{\tau_t}{\tau} &= \frac{D}{\rho c}\left(\frac{\Gamma_{rr}}{\Gamma} + \frac{1}{r}\frac{\Gamma_r}{\Gamma} + \frac{1}{r^2}\frac{\Phi_{\phi\phi}}{\Phi}\right) = -k, \qquad (8.3)\end{aligned}$$

where we now have functions of time isolated from functions of space. The subscript notation here indicated a derivative of a single variable function (e.g., $\Phi_{\phi\phi}(\phi) = d^2\Phi(\phi)/d\phi^2$). As in the 1D case (see Equation (7.7)), we define k as non-negative. This physically implies that the solution for $\tau(t)$ must be

$$\tau(t) = e^{-kt}, \tag{8.4}$$

where $k \geq 0$, and we have set the function constant to unity. Next we assume no angular dependence, reducing our spatial equation to simply

$$r^2\Gamma_{rr} + r\Gamma_r + \frac{k\rho c}{D} r^2\Gamma = 0, \tag{8.5}$$

with the physical restriction that D, ρ and c are positive

8.1.2 Steady-State Solution ($k = 0$)

When $k = 0$, there is no time dependence, and Equation (8.5) reduces to simply

$$r\Gamma_{rr} + \Gamma_r = 0, \tag{8.6}$$

which is solved as

$$\Gamma(r) = c_1 \log r + c_2, \tag{8.7}$$

where c_1 and c_2 are arbitrary constants. Imposing boundary conditions, Equations (8.2b) and (8.2c), eliminates the r dependence, reducing the spatial solution to a constant,

$$\Gamma(r) = T_R. \tag{8.8}$$

This solution accounts for the nonhomogeneous boundary conditions and is added to the transient case, for which k is nonzero.

8.1.3 Transient Solution ($k > 0$)

Equation (8.5) for $k > 0$ is Bessel's equation of order zero, which has a general solution,

$$\Gamma(r) = c_1 J_0\left(\sqrt{\frac{\rho c k}{D}} r\right) + c_2 Y_0\left(\sqrt{\frac{\rho c k}{D}} r\right), \tag{8.9}$$

where J_0 is the Bessel function of the first kind and Y_0 is the Bessel function of the second kind, both of zeroth order. Since the $T(R, t) = T_R$ boundary condition has already been satisfied with the steady-state solution, the outer boundary condition for the transient solution becomes homogeneous,

$$T_{\text{transient}}(R,t) = 0, \tag{8.10}$$

while the inner boundary condition, Equation (8.2c), remains unchanged.

As in the 1D case, we define some constants for ease of notation:

$$\lambda = \sqrt{k} \tag{8.11a}$$

$$\mu = \sqrt{\frac{\rho_1 c_1}{D_1}} \tag{8.11b}$$

$$\nu = \sqrt{\frac{\rho_2 c_2}{D_2}}. \tag{8.11c}$$

We identify k as the eigenvalue for the differential equation to be solved. However, the presentation is more concise to discuss the eigenvalues in terms of λ rather than of k. Consequently, we will refer to λ as the eigenvalues, with the implicit knowledge that the eigenvalues are technically the square of λ. No information is lost by the substitution since k is non-negative.

Substituting Equation (8.11), the spatial solutions are then

$$\Gamma_1(r) = b_0 J_0\left(\lambda\mu r\right) + b_1 Y_0\left(\lambda\mu r\right) \qquad (r < a) \tag{8.12}$$

$$\Gamma_2(r) = b_2 J_0\left(\lambda\nu r\right) + b_3 Y_0\left(\lambda\nu r\right) \qquad (r > a), \tag{8.13}$$

where b_i are constants to be determined. The Bessel functions at zero are $J_0(0) = 1$ and $Y_0(0) \to -\infty$. Since the domain for phase 1 ($0 \leq r \leq a$) includes zero and $\Gamma_1(r)$ must be bounded at $r = 0$ (i.e., have a finite value), we have $b_1 = 0$. Thus, the phase 1 solution is

$$\Gamma_1(r) = b_0 J_0(\lambda\mu r). \tag{8.14}$$

We enforce Equation (8.10) to determine the phase 2 solution below,

$$\begin{aligned}\Gamma_2(R) &= b_2 J_0\left(\lambda\nu R\right) + b_3 Y_0\left(\lambda\nu R\right) = 0\\ \Gamma_2(r) &= b_2\left(J_0\left(\lambda\nu r\right) + \beta Y_0\left(\lambda\nu r\right)\right),\end{aligned} \tag{8.15}$$

where we define b_3 in terms of b_2 via the convenient value β,

$$\beta = -\frac{J_0\left(\lambda\nu R\right)}{Y_0\left(\lambda\nu R\right)}. \tag{8.16}$$

Imposing the continuity condition, Equation (8.2d), gives solutions

$$\Gamma_1(r) = b_n \frac{J_0(\lambda\mu r)}{J_0(\lambda\mu a)} \qquad (r < a) \tag{8.17a}$$

$$\Gamma_2(r) = b_n \frac{J_0(\lambda\nu r) + \beta Y_0(\lambda\nu r)}{J_0(\lambda\nu a) + \beta Y_0(\lambda\nu a)} \qquad (a < r). \tag{8.17b}$$

For simplicity of the remaining chapter, we assume that $J_0(\lambda\mu a) \neq 0$ and $J_0(\lambda\nu a) + \beta Y_0(\lambda\nu a) \neq 0$ and, although this could be treated as a special case, ensure a posteriori that this is satisfied. Also note that Equation (8.2b) is satisfied because $d/dr[J_0(r)]_{r=0} = -J_1(0) = 0$.

Enforcing continuity of flux across the interface, Equation (8.2e), gives the transcendental equation, which determines the set of eigenvalues for our solution. Flux continuity of Equation (8.17) is evaluated as follows:

$$\begin{aligned}
D_1 \frac{d\Gamma_1(r)}{dr}\bigg|_{r=a} &= D_2 \frac{d\Gamma_2(r)}{dr}\bigg|_{r=a} \\
c_n D_1 \frac{\partial}{\partial r}\left(\frac{J_0(\lambda\mu r)}{J_0(\lambda\mu a)}\right)\bigg|_{r=a} &= c_n D_2 \frac{\partial}{\partial r}\left(\frac{J_0(\lambda\nu r) + \beta Y_0(\lambda\nu r)}{J_0(\lambda\nu a) + \beta Y_0(\lambda\nu a)}\right)\bigg|_{r=a} \\
-b_n \lambda\mu D_1 \frac{J_1(\lambda\mu a)}{J_0(\lambda\mu a)} &= -b_n \lambda\nu D_2 \frac{J_1(\lambda\nu a) + \beta Y_1(\lambda\nu a)}{J_0(\lambda\nu a) + \beta Y_0(\lambda\nu a)} \\
\mu D_1 \frac{J_1(\lambda\mu a)}{J_0(\lambda\mu a)} &= \nu D_2 \frac{J_1(\lambda\nu a) + \beta Y_1(\lambda\nu a)}{J_0(\lambda\nu a) + \beta Y_0(\lambda\nu a)}, \qquad (8.18)
\end{aligned}$$

where we use the identity that

$$\frac{\partial}{\partial r} B_0(\alpha r) = -\alpha B_1(\alpha r), \tag{8.19}$$

where $B_0(\alpha r)$ can represent any $J_0(\alpha r)$ or $Y_0(\alpha r)$ (or any linear combinations). here exists an infinite set of values of $\lambda > 0$ which satisfy Equation (8.18). We order and enumerate this set with n, where the set of eigenvalues is $\lambda_n \in \{\lambda_1, \lambda_2, \lambda_3, \dots\}$.

We define the set of eigenfunctions, $v^{(n)}(r)$, on a domain of $0 \leq r \leq R$ in two piecewise-continuous sections, corresponding to the eigenvalues λ_n, as

$$v^{(n)}(r) = \begin{cases} v_1^{(n)}(r) & (0 \leq r \leq a) \\ v_2^{(n)}(r) & (a \leq r \leq R) \end{cases} \tag{8.20}$$

where

$$v_1^{(n)}(r) = \frac{J_0(\lambda_n \mu r)}{J_0(\lambda_n \mu a)} \tag{8.21a}$$

$$v_2^{(n)}(r) = \frac{J_0\left(\lambda_n \nu r\right) + \beta_n Y_0\left(\lambda_n \nu r\right)}{J_0\left(\lambda_n \nu a\right) + \beta_n Y_0\left(\lambda_n \nu a\right)}, \tag{8.21b}$$

where the scalar constants, b_n, are absorbed into the temporal solution. The transient solution to this 2D heat conduction problem for discontinuous media is

$$T_{\text{transient}}(x,t) = \sum_{n=1}^{\infty} b_n e^{-\lambda_n^2 t} v^{(n)}(r). \tag{8.22}$$

8.1.3.1 The Full Solution

Combining the transient solution, Equation (8.22), with the steady-state solution, $T_{\text{ss}}(r,0) = T_R$, yields a combined solution of

$$T(r,t) = \begin{cases} T_R + \displaystyle\sum_{n=1}^{\infty} b_n e^{-\lambda_n^2 t} v_1^{(n)}(r) & (r < a) \\ T_R + \displaystyle\sum_{n=1}^{\infty} b_n e^{-\lambda_n^2 t} v_2^{(n)}(r) & (r > a). \end{cases} \tag{8.23}$$

We must determine b_n in order to fully specify the solution. This involves two steps. First, in Section 8.1.3.2, we must demonstrate that the eigenfunctions are orthogonal and determine their magnitude, creating an orthonormal basis. Second, in Section 8.1.3.3, we determine b_n from the initial conditions.

8.1.3.2 Eigenfunctions: Orthonormal Basis

As shown in texts such as those by Jackson [101] and Arfken and Weber [8], Bessel's equation are self-adjoint, and the solutions are orthogonal when Sturm-Liouville boundary conditions are satisfied. Specifically, letting $\alpha_{\gamma m}$ be the m^{th} zero of the γ^{th} order Bessel function of the first kind ($J_\gamma(\alpha_{\gamma m}) = 0$), one can show for a finite interval $r \in [0, R]$,

$$\int_0^R J_\gamma\left(\alpha_{\gamma m}\frac{r}{R}\right) J_\gamma\left(\alpha_{\gamma n}\frac{r}{R}\right) r\ dr = \frac{R^2}{2}[J_{\gamma+1}(\alpha_{\gamma n})]^2 \delta_{m,n}, \tag{8.24}$$

where $\delta_{m,n}$ is the Kronecker delta function. This represents orthogonality with respect to the roots of the Bessel functions of fixed order γ. The necessary conditions for an eigenfunction to be orthogonal on a finite interval $r \in [0, R]$, is that $v(r)v'(r)$ is zero at the endpoints,

$$v(0)v'(0) = 0 \tag{8.25a}$$

$$v(R)v'(R) = 0. \tag{8.25b}$$

These conditions are satisfied for our 2D analytic solution by Equation (8.2b) and Equation (8.2c). The discontinuity in our eigenfunctions does not prevent orthogonality because, by construction, both the eigenfunctions and their flux are continuous across the interface which are other conditions that must be met to give orthogonality. Therefore, the eigenfunctions, Equation (8.21), are orthogonal.

We define the inner product for this problem as

$$\langle f(r), g(r) \rangle = \int_0^R \rho c \ fgrdr = \mu^2 D_1 \int_0^a fgrdr + \nu^2 D_2 \int_a^R fgrdr, \tag{8.26}$$

where $\rho_1 c_1 = \mu^2 D_1$ and $\rho_2 c_2 = \nu^2 D_2$ by Equation (8.11).

A necessary integral identity, confirmed by *Maple* [140], is

$$\int_b^c B_0(\lambda r)^2 \ r \ dr = \left[\frac{r^2}{2} \Big(B_0(\lambda r)^2 + B_1(\lambda r)^2 \Big) \right]_{r=b}^{r=c}, \tag{8.27}$$

where $B_0(\lambda r)$ can represent $J_0(\lambda r)$ or $Y_0(\lambda r)$ (or a linear combination of the two).

The inner product of the n^{th} eigenfunction is

$$\begin{aligned} ||v^{(n)}(r)||^2 &= \left\langle v^{(n)}, v^{(n)} \right\rangle \\ &= \mu^2 D_1 \int_0^a \left(v_1^{(n)}\right)^2 rdr + \nu^2 D_2 \int_a^R \left(v_2^{(n)}\right)^2 rdr \equiv P_1 + P_2, \end{aligned}$$

where we find P_1 and P_2 separately. Using the integral identity in Equation (8.27), the first term is solved as

$$\begin{aligned} P_1 = \mu^2 D_1 \int_0^a \left(v_1^{(n)}\right)^2 rdr &= \mu^2 D_1 \int_0^a \left(\frac{J_0(\lambda_n \mu r)}{J_0(\lambda_n \mu a)} \right)^2 rdr \\ &= \frac{D_1}{2} \left(\frac{a\mu}{J_0(\lambda_n \mu a)} \right)^2 \left[J_0(\lambda_n \mu a)^2 + J_1(\lambda_n \mu a)^2 \right] - 0 \\ &= \frac{a^2 \mu^2 D_1}{2} \left[1 + \left(\frac{J_1(\mu \lambda_n a)}{J_0(\mu \lambda_n a)} \right)^2 \right]. \end{aligned} \tag{8.28}$$

Defining γ_n as the inverse of the denominator for $v_2^{(n)}$ for intermediate steps,

$$\frac{1}{\gamma_n} = J_0\left(\lambda_n \nu a\right) + \beta_n Y_0\left(\lambda_n \nu a\right), \tag{8.29}$$

the second term is

$$\begin{aligned}
P_2 &= \nu^2 D_2 \int_a^R \left(v_2^{(n)}\right)^2 r dr = \nu^2 D_2 \int_a^R \gamma_n^2 \left[J_0\left(\lambda_n \nu r\right) + \beta_n Y_0\left(\lambda_n \nu r\right)\right]^2 r dr \\
&= \frac{D_2}{2}(\nu\gamma_n)^2 \Bigg[R^2\bigg(\big(J_0(\lambda_n \nu R) + \beta_n Y_0(\lambda_n \nu R)\big)^2 + \big(J_1(\lambda_n \nu R) + \beta_n Y_1(\lambda_n \nu R)\big)^2\bigg) \\
&- a^2\bigg(\big(J_0(\lambda_n \nu a) + \beta_n Y_0(\lambda_n \nu a)\big)^2 + \big(J_1(\lambda_n \nu a) + \beta_n Y_1(\lambda_n \nu a)\big)^2\bigg)\Bigg] \\
&= \frac{R^2\nu^2 D_2}{2}\left(\frac{J_1(\lambda_n \nu R) + \beta_n Y_1(\lambda_n \nu R)}{J_0(\lambda_n \nu a) + \beta_n Y_0(\lambda_n \nu a)}\right)^2 - \frac{a^2\nu^2 D_2}{2}\left[1 + \left(\frac{J_1(\lambda_n \nu a) + \beta_n Y_1(\lambda_n \nu a)}{J_0(\lambda_n \nu a) + \beta_n Y_0(\lambda_n \nu a)}\right)^2\right],
\end{aligned} \tag{8.30}$$

where we used the outer boundary condition, Equation (8.10), to eliminate one term.

Combing Equation (8.28) and Equation (8.30) gives us the square of the norm of the n^{th} eigenfunction,

$$\begin{aligned}
||v^{(n)}(r)||^2 &= P_1 + P_2 \\
&= \frac{a^2}{2}(\mu^2 D_1 - \nu^2 D_2) + \frac{R^2\nu^2 D_2}{2}\left(\frac{J_1(\lambda_n \nu R) + \beta_n Y_1(\lambda_n \nu R)}{J_0(\lambda_n \nu a) + \beta_n Y_0(\lambda_n \nu a)}\right)^2 \\
&+ \frac{a^2}{2}\left[\frac{1}{D_1}\left(\mu D_1 \frac{J_1(\mu\lambda_n a)}{J_0(\mu\lambda_n a)}\right)^2 - \frac{1}{D_2}\left(\nu D_2 \frac{J_1(\lambda_n \nu a) + \beta Y_1(\lambda_n \nu a)}{J_0(\lambda_n \nu a) + \beta Y_0(\lambda_n \nu a)}\right)^2\right] \\
&= \frac{a^2}{2}\left[(\mu^2 D_1 - \nu^2 D_2) + \left(\frac{1}{D_1} - \frac{1}{D_2}\right)\left(\mu D_1 \frac{J_1(\mu\lambda_n a)}{J_0(\mu\lambda_n a)}\right)^2\right] \\
&+ \frac{R^2\nu^2 D_2}{2}\left(\frac{J_1(\lambda_n \nu R) + \beta_n Y_1(\lambda_n \nu R)}{J_0(\lambda_n \nu a) + \beta_n Y_0(\lambda_n \nu a)}\right)^2,
\end{aligned} \tag{8.31}$$

where the last step combines two terms by using the square of Equation (8.18).

As in the 1D case, the magnitude of the eigenfunctions, Equation (8.31), and their orthogonality forms an orthonormal basis. The initial conditions are constructed as a Fourier-Bessel series, where b_n specifies the weight of each basis function.

8.1.3.3 Constructing the Initial Conditions

The full solution is now determined by specifying b_n, which is accomplished by constructing the initial condition in terms of the orthogonal eigenfunctions. Prior to this point, the derivation has been generic for any initial condition. Only the determination of the Fourier-Bessel constants, b_n, depend on the initial condition.

We use formulae for a general Fourier series introduced in Section 7.1.4.2, as they are applicable here. The set of constants for a Fourier-Bessel series is defined as [8]

$$\begin{aligned} b_n &= \frac{\langle v^{(n)}(r), f(r)\rangle}{\langle v^{(n)}(r), v^{(n)}(r)\rangle} && \text{(8.32a)}\\ &= \frac{1}{||v^{(n)}(r)||^2}\left[\mu^2 D_1 \int_0^a f(r) v_1^{(n)}(r) r dr + \nu^2 D_2 \int_a^R f(r) v_2^{(n)}(r) r dr\right], && \text{(8.32b)} \end{aligned}$$

where $f(r)$ can be any function in general. However, for our problem $f(r)$ is defined by the initial condition, Equation (8.2a), and the expression for the temperature, Equation (8.23). Consequently, the function $f(r)$ is defined as

$$f(r) = \sum_{n=1}^{\infty} b_n v^{(n)}(r) = T_i - T_R. \tag{8.33}$$

Note that if $T_i = T_R$, the constants would all be zero. This is the correct behavior because that initial condition would equal the steady-state condition, meaning there would be no transient term.

Since the function $f(r) = T_i - T_R$ is a constant, the set of Fourier-Bessel constants is given by

$$\begin{aligned} b_n &= \frac{T_i - T_R}{||v^{(n)}||^2}\left[\mu^2 D_1 \int_0^a v_1^{(n)}(r) r dr + \nu^2 D_2 \int_a^R v_2^{(n)}(r) r dr\right] && \text{(8.34a)}\\ &= (T_i - T_R)\frac{\langle 1, v^{(n)}\rangle}{||v^{(n)}||^2}. && \text{(8.34b)} \end{aligned}$$

The denominator is already determined in Equation (8.31), and we examine the inner product of 1 and $v^{(n)}$ into two parts as when we found the eigenfunction magnitude,

$$\begin{aligned} \langle 1, v^{(n)}\rangle &= \mu^2 D_1 \int_0^a v_1^{(n)}(r) r dr + \nu^2 D_2 \int_a^R v_2^{(n)}(r) r dr\\ &\equiv P_1 + P_2. \end{aligned}$$

The integral needed for this inner product is

$$\int_b^c B_0(\lambda r) r dr = \left[\frac{r}{\lambda} B_1(\lambda r) \right]_{r=b}^{r=c}, \tag{8.35}$$

where $B_0(\lambda r)$ represents $J_0(\lambda r)$ or $Y_0(\lambda r)$ (or a linear combination). The first term is then

$$\begin{aligned} P_1 = \mu^2 D_1 \int_0^a \left(v_1^{(n)}\right) r dr &= \mu^2 D_1 \int_0^a \frac{J_0(\lambda_n \mu r)}{J_0(\lambda_n \mu a)} r dr \\ &= \frac{\mu^2 D_1}{J_0(\lambda_n \mu a)} \left[\frac{a}{\lambda_n \mu} J_1(\lambda_n \mu a) - 0 \right] \\ &= \frac{a}{\lambda_n} \left(\mu D_1 \frac{J_1(\lambda_n \mu a)}{J_0(\lambda_n \mu a)} \right). \end{aligned} \tag{8.36}$$

The second term is evaluated below,

$$\begin{aligned} P_2 = \nu^2 D_2 \int_a^R \left(v_2^{(n)}\right) r dr &= \gamma_n \nu^2 D_2 \int_a^R \left(J_0\left(\lambda_n \nu r\right) + \beta_n Y_0\left(\lambda_n \nu r\right)\right) r dr \\ &= \frac{\gamma_n \nu^2 D_2}{\lambda_n \nu} \left[R(J_1\left(\lambda_n \nu R\right) + \beta_n Y_1\left(\lambda_n \nu R\right)) - a(J_1\left(\lambda_n \nu a\right) + \beta_n Y_1\left(\lambda_n \nu a\right)) \right] \\ &= \frac{R \nu D_2}{\lambda_n} \frac{J_1\left(\lambda_n \nu R\right) + \beta_n Y_1\left(\lambda_n \nu R\right)}{J_0(\lambda_n \nu a) + \beta_n Y_0(\lambda_n \nu a)} - \frac{a}{\lambda_n} \left[\nu D_2 \frac{J_1\left(\lambda_n \nu a\right) + \beta_n Y_1\left(\lambda_n \nu a\right)}{J_0(\lambda_n \nu a) + \beta_n Y_0(\lambda_n \nu a)} \right]. \end{aligned} \tag{8.37}$$

Combining Equation (8.36) and Equation (8.37) gives

$$\begin{aligned} \langle 1, v^{(n)} \rangle &= P_1 + P_2 \\ &= \frac{a}{\lambda_n} \left[\left(\mu D_1 \frac{J_1(\lambda_n \mu a)}{J_0(\lambda_n \mu a)} \right) - \left(\nu D_2 \frac{J_1\left(\lambda_n \nu a\right) + \beta_n Y_1\left(\lambda_n \nu a\right)}{J_0(\lambda_n \nu a) + \beta Y_0(\lambda_n \nu a)} \right) \right] \\ &+ \frac{R \nu D_2}{\lambda_n} \frac{J_1\left(\lambda_n \nu R\right) + \beta_n Y_1\left(\lambda_n \nu R\right)}{J_0(\lambda_n \nu a) + \beta Y_0(\lambda_n \nu a)} \\ &= \frac{R \nu D_2}{\lambda_n} \frac{J_1\left(\lambda_n \nu R\right) + \beta_n Y_1\left(\lambda_n \nu R\right)}{J_0(\lambda_n \nu a) + \beta Y_0(\lambda_n \nu a)}, \end{aligned} \tag{8.38}$$

where the term in brackets is eliminated using the eigenvalue equation, Equation (8.18).

From Equation (8.34b) and Equation (8.38), the set of constants is defined as

$$b_n = \frac{(T_i - T_R) R \nu D_2}{||v^{(n)}||^2 \lambda_n} \frac{J_1\left(\lambda_n \nu R\right) + \beta_n Y_1\left(\lambda_n \nu R\right)}{J_0(\lambda_n \nu a) + \beta Y_0(\lambda_n \nu a)}, \tag{8.39}$$

where $||v^{(n)}||^2$ is defined in Equation (8.31).

8.1.4 Summary of the Full 2D Solution

The two-dimensional solution to the discontinuous media circular heat conduction problem is

$$T(r,t) = \begin{cases} T_R + \displaystyle\sum_{n=1}^{\infty} c_n e^{-\lambda_n^2 t} v_1^{(n)}(r) \quad (r < a) \\ \\ T_R + \displaystyle\sum_{n=1}^{\infty} c_n e^{-\lambda_n^2 t} v_2^{(n)}(r) \quad (a < r). \end{cases} \tag{8.40}$$

The following definitions are needed for the eigenfunctions:

$$\mu = \sqrt{\frac{\rho_1 c_1}{D_1}} \tag{8.41a}$$

$$\nu = \sqrt{\frac{\rho_2 c_2}{D_2}} \tag{8.41b}$$

$$\beta_n = -\frac{J_0\left(\lambda_n \nu R\right)}{Y_0\left(\lambda_n \nu R\right)} \tag{8.41c}$$

$$v_1^{(n)}(r) = \frac{J_0(\lambda_n \mu r)}{J_0(\lambda_n \mu a)} \tag{8.41d}$$

$$v_2^{(n)}(r) = \frac{J_0\left(\lambda_n \nu r\right) + \beta_n Y_0\left(\lambda_n \nu r\right)}{J_0\left(\lambda_n \nu a\right) + \beta_n Y_0\left(\lambda_n \nu a\right)}. \tag{8.41e}$$

The scalar constants are defined with the following two equations,

$$\begin{aligned} ||v^{(n)}|| =& \frac{a^2}{2}\left[(\mu^2 D_1 - \nu^2 D_2) + \left(\frac{1}{D_1} - \frac{1}{D_2}\right)\left(\mu D_1 \frac{J_1(\mu\lambda_n a)}{J_0(\mu\lambda_n a)}\right)^2\right] \\ &+ \frac{R^2\nu^2 D_2}{2}\left(\frac{J_1(\lambda_n \nu R) + \beta_n Y_1(\lambda_n \nu R)}{J_0(\lambda_n \nu a) + \beta_n Y_0(\lambda_n \nu a)}\right)^2 \end{aligned} \tag{8.42a}$$

$$b_n = \frac{(T_i - T_R) R\nu D_2}{||v^{(n)}||^2 \lambda_n} \frac{J_1\left(\lambda_n \nu R\right) + \beta_n Y_1\left(\lambda_n \nu R\right)}{J_0(\lambda_n \nu a) + \beta Y_0(\lambda_n \nu a)}, \tag{8.42b}$$

where λ_n is the ordered set of eigenvalues which satisfy

$$\mu D_1 \frac{J_1(\lambda_n \mu a)}{J_0(\lambda_n \mu a)} - \nu D_2 \frac{J_1\left(\lambda_n \nu a\right) + \beta_n Y_1\left(\lambda_n \nu a\right)}{J_0\left(\lambda_n \nu a\right) + \beta_n Y_0\left(\lambda_n \nu a\right)} = 0. \tag{8.43}$$

8.2 Dependence of the Analytical Solution on Eigenvalues

Finding the eigenvalues is a crucial step for implementing this analytic solution. While there exists an infinite number of eigenvalues, we can only use a finite number of terms when computing the analytic solution. We must use enough eigenvalues such that the truncated solution accuracy is sufficient to measure the error of the computed solution. Use of the 2D analytic solution to validate the accuracy of a mixed-cell model is computationally expensive because the analytic solution must be evaluated on the computational boundary at every time step. This requires that all the necessary eigenvalues be computed as a pre-processing step. We must determine how many eigenvalue terms are needed to compute the truncated solution at every time the analytic solution is required. This section explores these two topics: finding eigenvalues (Section 8.2.1) and deciding how many terms of eigenfunctions to keep (Section 8.2.1.1). This section concludes by choosing the diffusivity ratio and simulation time for the numerical testing of the mixed-cell models (Section 8.2.1.2).

The parameters used in the results for this problem are

$$T_R = 10 \tag{8.44a}$$

$$T_i = 30 \tag{8.44b}$$

$$R = 1.5 \tag{8.44c}$$

$$a = 0.6 \tag{8.44d}$$

$$L = 1 \tag{8.44e}$$

$$\rho_1 c_1 = 1 \tag{8.44f}$$

$$\rho_2 c_2 = 1, \tag{8.44g}$$

which only differ from the 1D case in that L is the computational domain and the problem domain goes to $R = 1.5$; see Figure 8.1 for the configuration.

8.2.1 Evaluating the Eigenvalues

Finding the eigenvalues for the 2D case is more complicated than in the 1D case. The primary reason for this difference is that the two functions are trigonometric in the 1D case, having constant spacing between zeros, whereas the two functions in the 2D case are combinations of Bessel functions, which do not have constant spacing between zeros. This makes it more difficult to determine the search intervals, such as those shown in Figure 7.2 for the 1D case.

The transcendental equation, Equation (8.18), can be written in terms of the following two functions,

$$f_1(\lambda) = \mu D_1 \frac{J_1(\lambda\mu a)}{J_0(\lambda\mu a)} \quad \text{and} \quad f_2(\lambda) = \nu D_2 \frac{J_1(\lambda\nu a) - \dfrac{J_0(\lambda\nu R)}{Y_0(\lambda\nu R)} Y_1(\lambda\nu a)}{J_0(\lambda\nu a) - \dfrac{J_0(\lambda\nu R)}{Y_0(\lambda\nu R)} Y_0(\lambda\nu a)}, \tag{8.45}$$

where the eigenvalues occur at the intersections of these functions. A sample plot of these functions is shown in Figure 8.2. Note that the plotted diffusivity values (D_1 and D_2) are chosen to be close in value so that the figure can show multiple cycles of each function. More typical values, where the diffusivities have ratios from 10^2 to 10^{10}, are more difficult to plot because the value of the transcendental functions are larger and change greatly from on point to the next. We distinguish the continuous variable λ, used as arguments of f_1 and f_2 to explore λ-space, from the discrete set of eigenvalues λ_n, which occur only when f_1 and f_2 are equal.

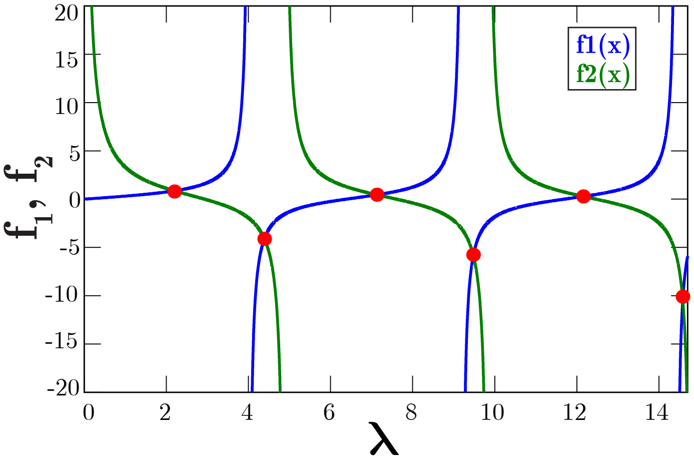


Figure 8.2: Eigenvalues for 2D analytic solution from transcendental equations, Equation (8.45). A small diffusivity ratio ($D_1 = 1$ and $D_2 = 2$) is shown for clarity. Each intersection (an eigenvalue) is marked with a red circle.

The algorithm for finding eigenvalues consists of six steps. The first step is to determine search intervals. This is accomplished by estimating and then refining the average period of each function. Half integer increments of these periods are used as search intervals because the periods are not exactly constant. Moreover, since eigenvalues occur mostly at the endpoints (asymptotes) of each period, searching at half-integer periods helps prevent missing an eigenvalue. The second step finds the minimum for each search interval for both the difference and normalized difference, defined in Equation (7.47) for the 1D eigenvalue-finding algorithm. The normalized difference is only used if the functions are larger than unity in the interval, $\max(|f_1|, |f_2|) > 1$. This prevents normalizing when the function values are near zero. If the differences are less than a tolerance (1 is used), they are stored for the next step, with a potential of two values for each search range (the standard-difference and the normalized-difference). The third step searches around each potential intersection, finding the minimum standard or normalized difference for this small range. If the difference is small enough (0.1 is used for the standard difference and 1 is used for the normalized difference), these potential intersection points are passed to the fourth stage. Step four refines the intersections, as in step three, but now uses a tighter criterion to determine whether or not to keep an intersection (10^{-3} for standard difference and 10^{-1} for normalized difference). For the normalized difference, often one function will be small (less than one) and the other will be large (greater than one thousand); in cases when both functions are smaller than a tolerance (for ratios less than 10^7, 0.5 is used, and 15 is used for stiffer ratios), that point is also kept as a potential intersection. Step five finds the average spacing of potential eigenvalues and examines points that are deemed too close together (less than 20% of average), keeping only the value with the smaller difference (normalized-difference used if values are large). Step six searches in smaller and smaller ranges around each eigenvalue until the change falls to machine precision. This step simply increases the accuracy of the eigenvalues found, which is crucial for very stiff problems. This algorithm has been graphically verified to find the first 200 eigenvalues for the following ten ratios $10^s : s \in \pm\{2, 4, 6, 8, 10\}$.

8.2.1.1 Accuracy of a Truncated Solution

We must distinguish three solutions: (i) the analytic solution, resulting from an infinite summation of eigenvalue terms, (ii) the truncated solution, resulting from a finite summation of eigenvalue terms, and (iii) calculated solution, generated from the numerical approximation of the equations. For the test to be valid, it is necessary that the error for the truncated solution is less than the discretization error of the calculated solution.

We quantify the behavior of this problem by comparing ten diffusivity ratios, measuring the truncation error using Equation (7.51), at three sampling times based on the energy evolution versus stead state, Equation (7.50). (Refer to Section 7.2.2 for details on these choices.) The diffusivity ratio is expressed in terms of s, defined as

$$s = \log_{10} \frac{D1}{D2}. \tag{8.46}$$

Note that the specific units of time, diffusivity, etc., are arbitrary. Furthermore, recall that we refer to λ_n as the eigenvalues, but technically the eigenvalues are λ_n^2.

ratio	Time			Number of Eigenvalues		
s	Early	Mid	Late	Early	Mid	Late
2	2.3894	5.7748	10.8881	4	3	2
4	23.873	57.71	108.818	4	3	2
6	238.73	577.1	1,088.15	4	3	2
8	2,387.3	5,771	10,881.75	4	3	2
10	23,873	57,710	108,817.5	4	3	2
-2	0.035115	0.46355	1.34735	22	6	3
-4	3.6486E-3	4.1085	12.7926	189	5	3
-6	3.6283E-4	41.033	127.859	1,804	6	3
-8	3.626E-5	410.315	1,278.58	17,931	5	3
-10	4.2211E-6	4,530.47	13,197.85	169,751	5	3

Table 8.1: 2D MES analytic solution trends depending on ratio of phase coefficients. Solution times to reach early-, intermediate-, and late-stage are listed. Also list is the number of eigenvalues needed to construct the solution at those times with an error near machine precision.

Table 8.1 shows the time for solutions with various diffusivity ratios to reach early-, intermediate-, and late-stage, which corresponds to $\xi(t) = 0.50$, $\xi(t) = 0.20$, and $\xi(t) = 0.05$, respectively. Equation (7.50) defines $\xi(t)$, which a function that characterizes the energy evolution with respect to the steady-state energy; a value of $\xi(t) = 0.90$ still has 90% of the energy at the initial condition, and $\xi(t) = 0$ indicates the solution is at equilibrium (i.e. the system has lost all energy above steady-state). Also shown is the number of eigenvalues needed to represent the solution to machine precision at these three times. The trends relating the time and diffusivity ratios in Table 8.1 are opposite to those Table 7.1 (i.e., the same trends for opposite signs of s). The time to reach intermediate- and late-stage (and early-stage for $s > 0$) increases by a factor of ten for every factor of 100 increase in s. This trend is because the diffusion time scales inversely with diffusivity ($\Delta t \propto 1/D$) and because the time scale is controlled primarily by the smaller diffusivity, which decreases by a factor of ten when $|s|$ increases by a factor of 100. The exception to this trend is the early time behavior for $s < 0$, where the time decreases by a factor of ten for increasing $|s|$ by 100. This is due to heat flux at the $r = R$ boundary being faster for this ratio because the phase 2 diffusivity is large. In other words, when D_2 is large, energy can leave the system very quickly; when D_2 is small, phase 2 acts as an insulator, causing the energy to leave the system slowly. Thus, the larger diffusivity dominates in this case, and the time goes down by a factor of ten as D_2 increase by a factor of ten (which occurs as $|s|$ increases by a factor of 100).

Another trend in Table 8.1 is that the positive ratio solutions take about a factor of ten longer for equivalent negative ratio solutions, which is because the outer boundary, $r = R$, has a smaller diffusivity, which partially insulates the disk. All solutions need six eigenvalues or fewer except the early-stage, $s < 0$ cases. As in the 1D case, the large number of eigenvalues for this case arises because the solution so quickly loses energy through the outer boundary to reach the boundary condition, $T(R, t) = T_R$, leading to a temperature profile with corners rather than smooth curves. An alternative viewpoint to explain this trend is that more eigenvalues are needed simply because the solution is evaluated earlier in time, when the larger eigenvalues terms are not damped as strongly by the exponential term. If we instead compared the different ratio cases at the same time for a given set up (rather than at the same energy characteristic stage), we would find they require an equivalent number of eigenvalues.

8.2.1.2 Selection of Numerical Testing Parameters

We are using the 2D analytical solution to test the validity of the mixed-cell models. We desire to determine the parameters for the test that minimize computational effort while making the mixed cell a critical point (i.e., the temperature profile must have a distinguishing feature in the mixed cell rather than being smooth).

The positive ratio jumps are the best choice for a run-set for the numerical implementation of the 2D analytic solution. Table 8.1 shows that the positive ratio cases needs fewer eigenvalues to resolve the solution, although they do require longer runtimes. However, since the solution must be specified on the boundary at every time step, the solution must be resolved as early as $t = \Delta t$ (the truncated analytic solution is not required at $t = 0$). Since we are defining the time step as a function of the grid size ($\Delta t = \Delta x^3$), we must determine the number of eigenvalues needed for the most refined grid of the study. Next, we desire to run the problem for as little time as possible since this is a computationally intensive problem. We focus our attention on just three jump ratios (10^2, 10^6, 10^{10}), which should be sufficient to discern trends.

Figure 8.3 shows five plots of the radial solution at various points between the initial condition and the steady-state value for $s = 6$. A kink in the solution be-

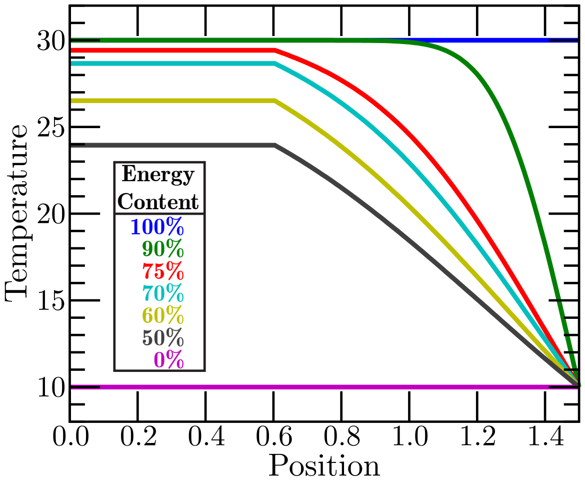
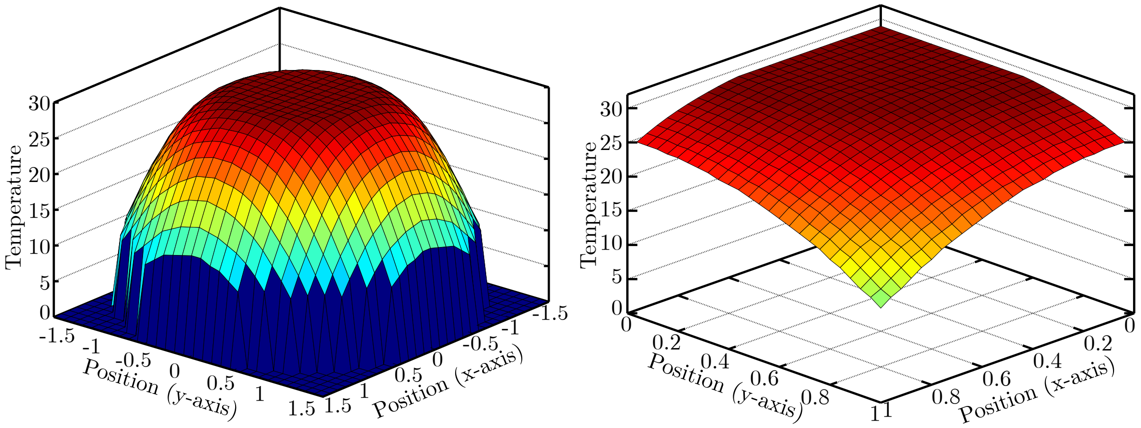


Figure 8.3: Radial plot of 2D analytic solution at various points in the energy evolution. The energy evolution function is defined in Equation (7.50), where 100% means that the (cooling) problem has all of its initial energy, while 0% means the problems has lost all excess energy and equilibrated at steady-state. The jump ratio is $s = 6$, the interface is at $a = 0.6$, and other parameters are given in Equation (8.44).

comes perceptible at about $\xi(t) = 0.75$, after which the quasi-uniform temperature of phase 1 slowly drops while phase 2 becomes linear. Prior to this point, the solution is smooth across the interface, and we desire a kink in order to best test the mixed-cell models. Thus, a simulation can be terminated when $\xi(t) = 0.75$ in order to minimize computation time. Figure 8.4 shows the solution at this point for both the whole domain ($-R \leq x, y \leq R$) and the computational domain ($0 \leq x, y \leq 1$). Table 8.2 gives the times and number of eigenvalues needed to resolve the solution at the stopping point ($\xi(t) = 0.75$), which is a just six eigenvalues. However, we also need enough eigenvalues to create a sufficiently accurate truncated solution at the initial time step ($t = \Delta t$), which depends on grid size. Consequently, Table 8.2 also shows the number of eigenvalues needed for the first time step for grid sizes of 40, 50, and 100.

The number of eigenvalues needed for the truncated solution to have sufficient accuracy varies with time. While the first time step may need on the order of 10^6 eigenvalues, this number quickly drops to the order of 10^2. One cannot use just ten eigenvalues because the solution at early times would be highly inaccurate, and this error would perpetuate. However, using hundreds of thousands of eigenvalues for every time step would be needlessly computationally intensive. One solution to this



(a)

(b)

Figure 8.4: 2D solution at final time ($\xi(t) = 0.75$). A ratio of $s = 6$ is plotted for both the problem domain (a) and the computational domain (b) is shown.

conundrum is to use variable number of eigenvalues by enforcing a criterion,

$$\exp(-\lambda_n^2 t) > \text{tolerance}, \tag{8.47}$$

to truncate the eigenvalue summation (we use a tolerance of $10^{-10}\Delta x^2$).

Two other specific choices are made to speed up implementation. Saving every eigenfunction is an unnecessary use of memory, but recalculating each eigenfunction is likewise inefficient. We therefore save only the first n eigenfunctions and calculate all eigenfunctions greater than n for each time step (n=400 is used). The advantage of this is that the most commonly used eigenfunctions (the first several) are stored, and after some point in time, the number of eigenfunctions used will be less than n. Second, the choice to position the computational domain in the positive quadrant provides symmetry to the solution along the bottom and left edges, as well as along the top and right edges. This means that there are two degenerate edges, so the boundary conditions need only be computed along two edges, and these values can also be used on the remaining two edges. Because each boundary edge is used for two edges of the computational domain, the number of spatial elements in the truncated analytic solution is reduced.

			Solutions at Initial and Final Times			
			Initial Time ($t = \Delta t = \Delta x^3$)			Final Time
		Grid Size	100	50	40	Any
s	t	$\xi(t)$	0.99	0.99	0.99	0.75
2	0.83895	Number	5,255	1,877	1,379	6
6	83.833	of	46,906	16,881	12,034	6
10	8,383.3	Eigenvalues	443,225	160,001	115,234	6

Table 8.2: Number of eigenvalues at the start and the time at the end for 2D MES analytical test for various jump ratios s. The initial time step depends on grid size, and three grid sizes are listed. The value for the final time (t), when $\xi(t) = 0.75$, are also listed for each ratio.

8.3 Results of Mixed-Cell Models for 2D Analytical Test

This section presents the results for six mixed-cell models for solving the 2D analytic test problem discussed in this chapter. The similarities and differences between the six models are summarized in Table 6.1. Three single temperature models are presented: the harmonic mean (S1), the rotated tensor-diffusivity method (S2), and the split-cell method (S3). (See Chapter III for the description of these models.) A multi-temperature model is also presented with three variants, based on the level of the approximation (M1, M2, M3). (See Chapter IV for the description of these models.) Five of these six methods represent new work, with the harmonic mean used as a baseline. The convergence rate of a model compared to the convergence rate of the harmonic mean model determines if a model is advantageous.

8.3.1 Implementation

Implementing this analytical solution on a Cartesian mesh is an involved process. The truncated analytical solution must be computed at every time step to specify the Dirichlet boundary conditions of the computational domain. While the outer boundary conditions are simple for the domain of the problem, $T(R,t) = T_R$, for the computational domain the boundary conditions are $T(r,t) = T_R + \sum_{n=1}^{\infty} c_n e^{-\lambda_n^2 t} v^{(n)}(r)$ where $r = \sqrt{x^2 + y^2}$ and x and y are defined on the perimeter of the unit square ($x, y \in [0, L]$).

The problem is initialized by setting $T(r,0) = T_i$. The time step is conservatively scaled as $\Delta t = \Delta x^3$ to ensure the first-order temporal discretization error is below the second-order spatial discretization error. The mixed-cell models are executed for various grid refinements to characterize convergence trends. These grid refinements are square grids of length 10, 20, 30, and 40 for all three ratios, while the $s = 6$ ratio also has a grid of 50, and the $s = 2$ ratio has additional grids of length 50 and 100.

8.3.2 Results

The calculated solution for every model, ratio, and grid refinement resembles Figure 8.4(b). In other words, even in cases where the convergence rates are near zero, the computed solution for each grid is a qualitative match of the analytical solution.

Table 8.3 presents the convergence rates at the time where $\xi(t) = 0.75$ for each model and ratio. The convergence rates listed are averaged over each grid for each particular ratio. No model performs at second-order accuracy for *all* three error norms. In fact, only the S1 and S3 models achieve near second-order accuracy in *any* error norm, having rates greater than 1.8 in the 1-norm.

Strongest jump ratio. The $s = 10$ ratio does poorly in every model, having near zero convergence rates (constant error regardless of grid resolution) in the single-temperature methods; see Table 8.3. The single-temperature methods have absolute convergence rates on the order of or below 0.1, which is effectively zero. The multi-temperature methods attain positive, nonzero convergence rates, but only about 0.32 on average, which is well below first order. The deficient results of the strongest jump are likely due to one of two primary causes: problems with the eigenvalues or problems with the code. The eigenvalues used for the boundary conditions may have skipped some values or may not be accurate enough. The number of eigenvalues needed to resolve both phases for this ratio is so large that it is impractical to graphically verify that no values were skipped. Since the eigenvalues occur at the asymptotes of Equation (8.45), the results are highly sensitive to slight variations of any digit of the calculated eigenvalues. Accordingly, any inaccuracy in these eigenvalues is greatly magnified in the eigenfunctions, much more than any inaccuracy in the $s = 2$ or $s = 6$ ratios. However, since the results look qualitatively correct it is reasonable to conclude that the eigenvalues are not the problem, especially since the larger the eigenvalue, the smaller its contribution. One might think that the errors occur at early times, when more eigenvalues are needed; however, we show later that the early time solutions are second-order accurate. There could also be a numerical issue with the computation of the Bessel functions. Alternatively, the cause for the poor results for the $s = 10$ ratio may be that the (single- and multi-temperature) codes simply cannot handle such highly discontinuous diffusivities. The 1D MES test also show a performance drop for the single-temperature code for a ratio of $|s| = 10$, however the performance of the multi-temperature code for this ratio was consistent with the smaller ratios. While another test could be used to explore the behavior of these codes with such a strong jump to ascertain the root of the issue, for now we put aside this particular ratio, noting that the multi-temperature models performed better at this ratio, and focus on the jumps of $s = 2$ and $s = 6$.

Case		Single-T			Multi-T		
s	#	q_1	q_2	q_∞	q_1	q_2	q_∞
2	1	**1.90**	1.47	0.89	1.10	1.03	0.65
	2	1.67	1.46	0.88	1.07	1.01	0.85
	3	1.81	1.56	0.68	1.12	1.07	0.90
6	1	1.79	1.41	0.87	1.26	1.10	0.54
	2	1.14	1.04	0.75	1.15	1.06	0.85
	3	1.73	1.53	0.57	1.22	1.13	0.91
10	1	0.06	0.04	0.03	0.41	0.39	0.35
	2	-.08	-.09	-.12	0.32	0.30	0.25
	3	0.05	0.04	-.00	0.31	0.29	0.24

Table 8.3: Convergence rates for 2D MES. Single- and multi-temperature results are given, showing three different diffusivity ratios (10^s) and measured with three error norms. The rates are averaged over all the grids used for each ratio. The model number for each row is indicated in the column labeled #. Bold indicates $q_i \geq 1.90$.

Multi-temperature methods. The multi-temperature results perform, at best, first order. Examination of the convergence of E_∞ in Table 8.3 for the ratios $s = 2$ and $s = 6$ show a clear trend that the convergence rate increases with higher fidelity models (i.e., M3 better than M2, M2 better than M1), which is the sensible and expected result. The M3 method does marginally better than the S1 method in the ∞-norm, and this marks the only measurable instance where any of the multi-temperature results are superior to the single-temperature results for the $s = 2$ and $s = 6$ ratios. Examination of the phase error for phase 2 shows results almost exactly matching the combined-phase error. This is because the phase 1 solution is constant and easily achieves high accuracy, while all of the change and topology occurs in phase 2. Hence, the phase-averaged error is dominated by the phase where the solution is more quickly evolving. Plots of the difference between the analytic and computed solution show a discretization error that is quickly overshadowed by error at the interface; see Figure 8.6. The multi-temperature methods are less accurate than the single-temperature methods for this test problem, and we focus the remainder of this section on the single-temperature results.

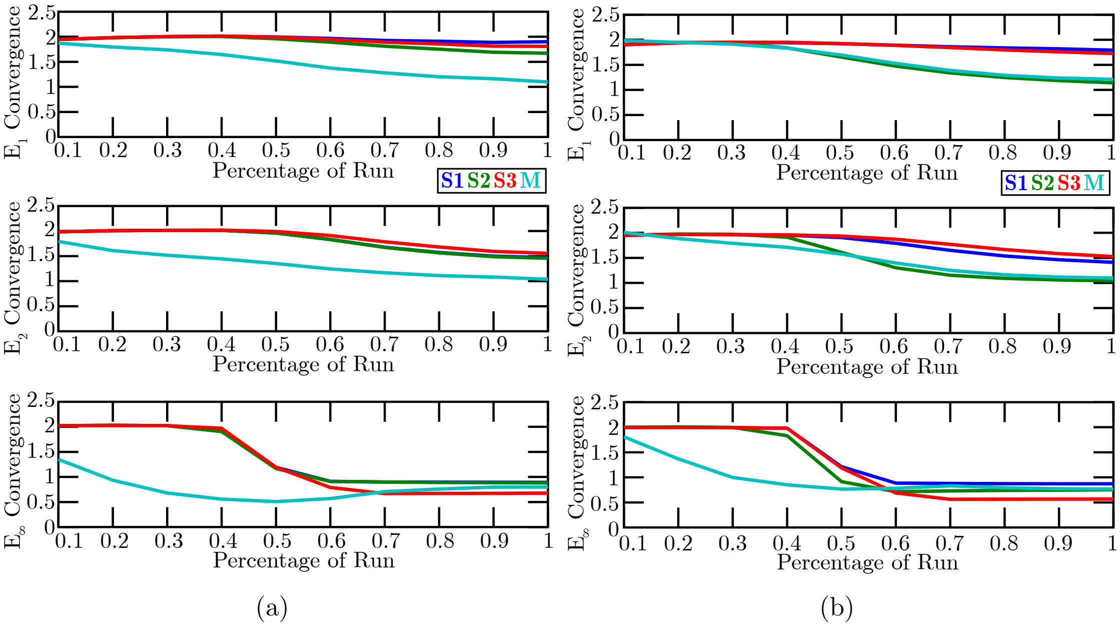
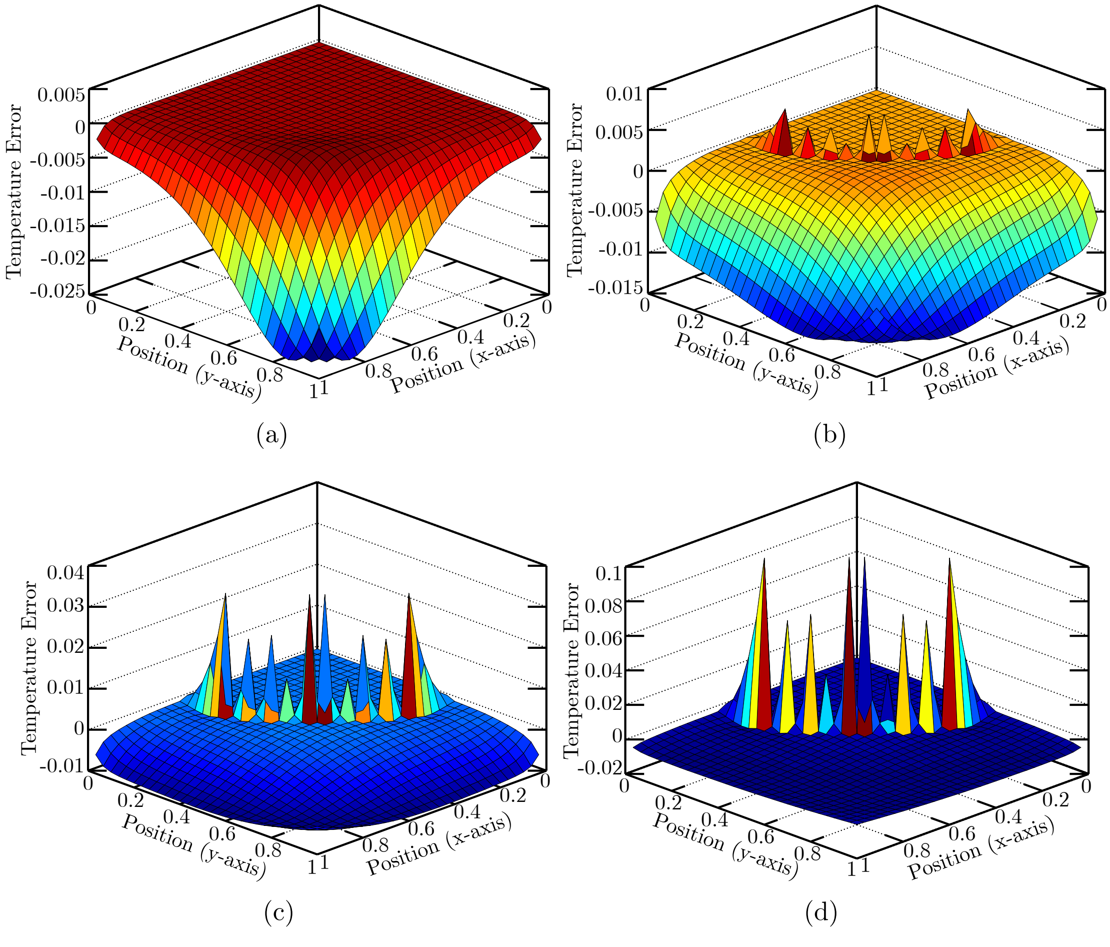


Figure 8.5: Convergence rates versus time for 2D solution. The three single-temperature models are each shown, and the three multi-temperature models are averaged together and labeled as M. Convergence is calculated for three p-norms, with jump ratios of (a) $s = 2$ and (b) $s = 6$. The x-axis shows percent progress to the final time ($\xi(t) = 0.75$).

Single-temperature methods. The single-temperature methods have the best performance for this 2D mixed-cell test for the jump ratios of $s = 2$ and $s = 6$; see Table 8.3. However, the convergence rates at $\xi(t) = 0.75$ are below the expected second-order rate. The S2 model has the worst overall performance for the single-temperature models, however the S3 model has the lowest q_∞. The S1 model has the highest convergence rate for each ratio.

Convergence as a function of time. Figure 8.5 plots the convergence rates versus time for the two lower jumps. The x-axis of the plot is the percentage in time between $t = 0$ and the time where $\xi(t) = 0.75$ (see Table 8.2). The multi-temperature convergence rates are plotted for comparison purposes. Note that the convergence rates for the three multi-temperature models are averaged because they are so similar, so the plot labeled 'M' represents the M1, M2, and M3 models. It is evident from these plots is that the convergence rate begins at second order, but at some point drops to a lower rate. The multi-temperature convergence rates drop sooner than the single-temperature rates in all cases, but especially for q_∞. The single-temperature results appear to have a critical point (most evident in the ∞-norm) about 40% through the run where the convergence rate begins its descent. The S1 and S3 models have similar trends in both cases, near second-order convergence rates in 1-norm, near 1.5 in 2-norm, and between 0.5 and 1 in the ∞-norm. The S2 model follows the trends of the other two models in the $s = 2$ ratio, but is less accurate for the 1-norm and 2-norm for the $s = 6$ ratio.



(a)

(b)

(c)

(d)

Figure 8.6: Difference between 2D analytic and calculated solution versus time. The percentage progress of the plots are (a) 20%, (b) 40%, (c) 60%, and (d) 100%. The S1 model is shown with a jump ratio of 10^6. Error along interface is conspicuous and grows with time.

The reason that the convergence rates for the single-temperature methods decrease with time is apparent from the series of subplots in Figure 8.6: the interfacial error grows with time and eventually dominates the problem. Figure 8.6(a) shows only discretization error, which converges at second-order, with no discernible interfacial error. Examination of the remainder of the subplots in Figure 8.6 shows that the error along the interface grows with time (note the scale of the temperature axis). In fact, this error overshadows the discretization error in the rest of the domain, and will eventually cause the whole problem to drop to first-order accuracy or worse. Note that while these plots are for $s = 6$, the same trends occur in the $s = 2$ ratio. Furthermore, only the S1 model is shown, but the S2 and S3 models look qualitatively equivalent, with the sole exception that the S3 model has both negative and positive spikes at the interface.

8.3.2.1 Alternative Implementations

The poor results for the 2D MES test problem presented in this chapter are likely due to an implementation issue. Two possible causes are (i) the presence of mixed cells on the boundary or (ii) the fact the solution is driven by the boundary values.

Mixed cells on boundary. We repeat the 2D MES test problem as has been presented with one change: the computational domain is expanded from $x, y \in [0, L]$ to $x, y \in [-L, L]$. In other words, we moved the computational domain such that all of phase 1 (and therefore all mixed cells) is contained within the computational domain. By exploiting the symmetry of the problem, this change in domain does not increase or decrease the number of analytic solution elements that needed to be computed for the boundary conditions, although the number of elements in the solution increased by a factor of four.

The convergence rates of the ∞-norm for this problem are presented in Table 8.4, while additional convergence rate information (such as q_1 and q_2) can be found in Tables C.6 and C.7. Comparison between Table 8.3 and Table 8.4 shows that the convergence rates are slightly higher for the single-temperature methods using the larger domain, with q_∞ becoming closer unity. The multi-temperature models increase in convergence rate for M1 while staying about the same for M2 and M3. Therefore, the presence of mixed cells on the boundary is not the cause of the low-order results seen in the previous section. The results of this new domain are equivalent to the $x, y \in [0, L]$ domain: the convergence rates decline from second-order to first-order accuracy with time, and the error is dominated by the mixed cells.

s	grid size	S1	S2	S3	M1c	M2c	M3c	M1	M2	M3
2	20→40	1.19	1.21	1.23	0.84	0.88	0.88	0.83	0.90	0.91
	40→60	0.83	0.79	0.76	1.03	0.95	0.95	1.03	0.92	0.91
	60→80	1.23	1.25	1.23	0.85	0.89	0.89	0.84	0.89	0.89
	mean	1.09	1.08	1.07	0.90	0.91	0.91	0.90	0.90	0.90
6	20→40	1.19	1.32	1.23	0.84	0.86	0.86	0.83	0.88	0.89
	40→60	0.83	0.83	0.75	1.03	1.00	1.06	1.03	0.98	0.97
	60→80	1.22	1.25	1.22	0.84	0.87	0.87	0.83	0.87	0.87
	mean	1.08	1.13	1.07	0.90	0.91	0.92	0.90	0.91	0.91

Table 8.4: Convergence rates of all six mixed-cell models for the 2D MES mixed-cell test problem using the expanded computational domain. The convergence rate is measured using E_∞ for the indicated grid sizes. Results are shown for two different diffusivity ratios, $s = -2$ and $s = -6$, and phase 2 is shown for the multi-temperature models. Tables C.6 and C.7 are expanded versions of this table.

Constant boundary conditions. It is clear from Figures 8.4(b) and 8.6(a) that the temperature in the domain is driven by the corner point, which is undesirable. This set up makes the boundary values for this corner more important than any other point. Running the test problem with the expanded domain, $x, y \in [-L, L]$, exacerbates this issue by making all four corners critical points in driving the solution, as opposed to just one corner. It would be preferable to eliminate the dependence of this solution on the corner values. It would be better if the problem were not driven by changing boundary values, but rather the problem simply evolves from the initial conditions with static boundary conditions.

The 2D MES test problem, with some modifications, could indeed be run using constant boundary conditions. Not only would this eliminate the effect of the corner boundary values, but it would also alleviate much of the computational expense of the problem because the analytical solution would not need to be calculated at any time step, only required as a post-processing step. Furthermore, far less eigenvalues would be needed because the analytical solution would not need to be evaluated until a time much greater than $t = \Delta t$, where $\Delta t \propto \Delta x^3$. By making the outer material (phase 2) have the larger diffusivity ($s < 0$) and setting the initial temperature profile for that phase to the boundary value (T_R), a computational boundary of $x, y \in [-L, L]$ could remain constant in time. The initial temperature in the inner region (phase 1) would need to be different from the outer, constant region so that the problem would not start in equilibrium.

We desire an initial temperature profile that smoothly transitions from zero at

$r = 0$ to T_R at $r = a$. Using a quadratic expression, the initial temperature profile is

$$T(r,0) = \begin{cases} T_R \left(\frac{r}{a}\right)^2 & r < a \\ T_R & a < r. \end{cases} \tag{8.48}$$

The solution expression is the same as before,

$$T(r,t) = T_R + \sum_{n=1}^{\infty} b_n \exp\left(-\lambda_n^2 t\right) v^{(n)}(r), \tag{8.49}$$

with only the constants series (b_n) changing. Integration of Equation (8.32b) using *Maple* [140] with the new initial conditions finds the new Fourier-Bessel constants as

$$b_n = \frac{1}{||v^{(n)}||^2} \frac{2T_R D_1}{\lambda_n^3 \mu a} \left[\lambda_n \mu a - 2\frac{J_1(\lambda_n \mu a)}{J_0(\lambda_n \mu a)} \right], \tag{8.50}$$

where the only contribution occurs in phase 1, since phase 2 set to steady-state.

We find that a diffusivity ratio of $s = -2$ is not strong enough to give constant temperature on the computational boundary, however a ratio of $s \leq -6$ does maintain a constant temperature for parameters $R = 1.5$ and $a = 0.6$. We run the problem to $\xi(t) = 0.5$, which is $t = 41.3$ for a ratio of $s = -6$. Figure 8.7 shows the initial and final temperature profiles for this alternate 2D MES test problem. For this problem, the solution in phase 1 has a nonzero slope, so the phase-centered multi-temperature results are measured to quantify the error (phase 2 results were used in the previous instances of this problem).

While this version of the 2D MES mixed-cell test problem has simple computational boundary values, the initial condition is more complicated. When specifying the initial condition for the mixed cells, the question arises as to whether it is more appropriate to use the value of the solution at the cell center of the mixed cell or to use a value that is averaged over the entire cell, thereby accounting for each phase. A similar issue occurs during the analysis and error computation: should the analytic solution use the solution at the cell-center location or a temperature value that has been averaged over the mixed cell? Two versions of the problem are run to compare these cases: (i) the cell-center version initializes the problem and computes the analytical solution using the cell-centered values (as has been done throughout the thesis), and (ii) the cell-averaged version uses a temperature value for the initial temperature profile and final analytical solution that has been averaged over each mixed cell. The averaging process is done numerically by partitioning each mixed

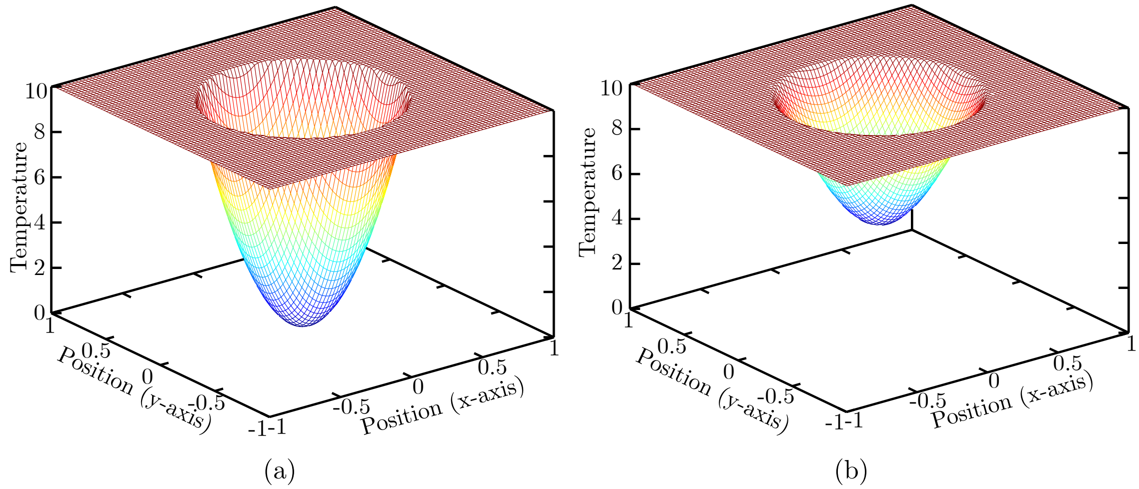


Figure 8.7: 2D MES analytical solution in the computational domain for the quadratic initial condition with $s = -6$. Shown are (a) the initial conditions (t=0) and (b) the final value ($\xi(t) = 0.5$).

cell region into a 5×5 mesh, computing the analytical solution at the center of each of the mesh elements, and then using the average temperature from these 25 values. Centroid-centered average solutions for the multi-temperature models used this same concept, but only the values within each phase contribute to the average.

grid size	S1	S2	S3	M1c	M2c	M3c	M1	M2	M3
20→40	0.52	1.01	1.01	0.01	0.59	0.59	0.09	0.62	0.63
40→60	0.93	0.93	0.93	0.06	0.77	0.76	0.06	0.70	0.69
60→80	1.09	1.11	1.11	-0.02	1.22	1.22	-0.02	1.21	1.22
mean	0.85	1.02	1.02	0.05	0.86	0.86	0.05	0.85	0.84

Table 8.5: Convergence rates for the 2D MES problem with a quadratic initial condition using a cell-centered solution. Phase 1 is shown for the multi-temperature models. Table C.8 is an expanded version of this table. The diffusivity ratio is $s = -6$.

grid size	S1	S2	S3	M1c	M2c	M3c	M1	M2	M3
20→40	0.94	0.88	0.88	0.66	1.01	1.02	0.68	1.03	1.04
40→60	1.00	0.96	0.96	0.48	1.70	1.81	0.48	1.67	1.79
60→80	1.16	1.21	1.21	0.40	-1.45	-1.73	0.40	-1.39	-1.68
mean	1.03	1.02	1.02	0.51	0.42	0.37	0.52	0.44	0.38

Table 8.6: Convergence rates for the 2D MES problem with a quadratic initial condition using a cell-averaged solution. Phase 1 is shown for the multi-temperature models. Table C.9 is an expanded version of this table. The diffusivity ratio is $s = -6$.

The convergence rates using a cell-centered initial temperature profile and analytical solution are presented in Table 8.5 with an expanded version shown in Table C.8. The convergence rates for the cell-averaged versions are presented in Tables 8.6 and C.9. These two options make little change in q_∞ for the S2 and S3 models. The M2 and M3 models perform better using the cell-centered version than the cell-averaged version of this test, while the S1 and M1 models perform better using the cell-averaged version. However, it is clear that, regardless of which choice is made, the models are, at best, first-order accurate. The single-temperature models outperform the multi-temperature models, and the three single temperature models yield about the same convergence rate.

Considering the cell-centered error only, found in Table 8.5 and Table C.8, we see that the S2 and S3 models have equivalent q_∞ (1.02) while the S1 model is lower (0.85). However, the S1 model has the lowest absolute error of these models in all three norms. The S3 error is about twice as large as the S1 error, while the S2 error is five times as high in the ∞-norm and twenty to thirty times as high for the 1-norm and 2-norm. The fact the S2 model has larger error than the S1 and S3 models can be inferred from Table C.8, where it is clear that the S2 model has lower convergence rates than the S1 and S3 models for the 1-norm and 2-norm.

The M2 and M3 models, as seen in Table 8.5, have equivalent convergence rates, as do the S2 and S3 models. However, unlike the S2 and S3 models, the M2 and M3 models have almost identical errors along with their convergence rates. The ratio error between the M2 and M3 models for phase 1 and the S1 model is between five and eight. The error ratio between the M1 and S1 models is between ten and forty. Thus, the S1 model has the lowest error with competitive convergence rates to the S2, S3, M2, and M3 models. While the S2 and S3 models give equivalent convergence rates, the S3 model is much more accurate. Therefore, for this version of the test problem, the S1 model has the highest accuracy, followed by the S3 and M2/3 models, and all have similar convergence rates, between 0.84 and 1.02.

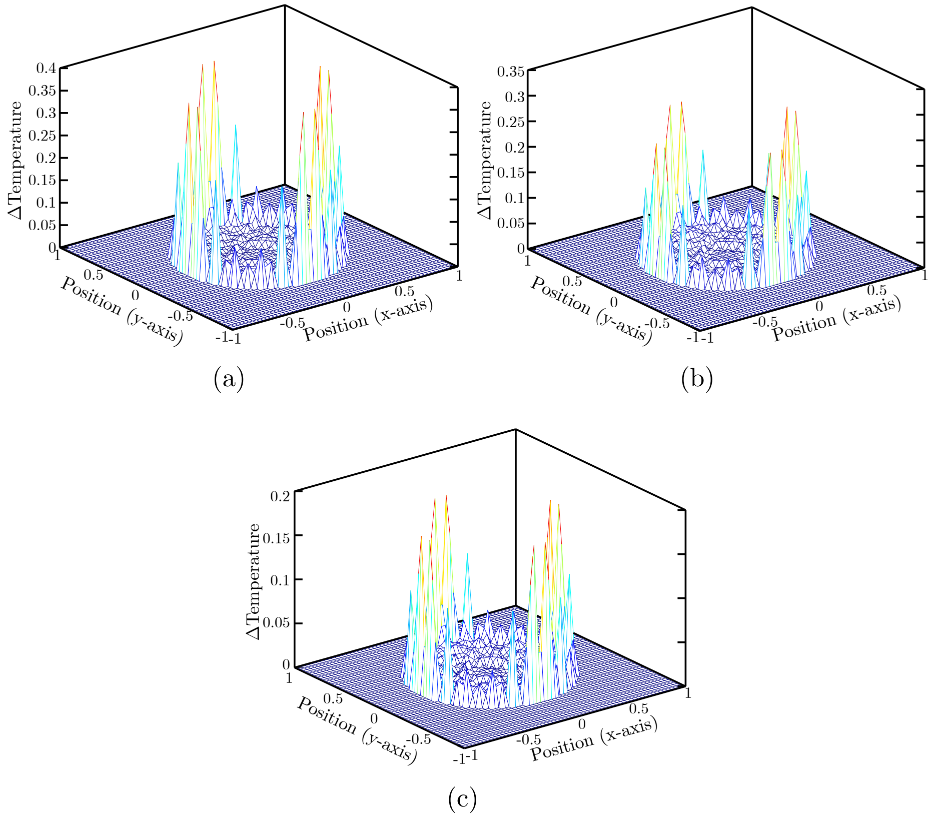


Figure 8.8: Absolute value of difference between analytic and calculated solution for the 2D MES with a quadratic initial profile. The S3 model is shown for a ratio of $s = -6$ at three times between the initial conditions and the final time (when $\xi(t) = 0.5$): (a) 10%, (b) 50%, and (c) 100%.

Unlike the case with a constant initial condition, where the convergence rates start at second-order accuracy and drop to first-order (see Figure 8.5), this version of the test yields convergence rates that are effectively constant. While the mixed cell error does not grow in time, the mixed cell error dominates the error at every time. This is because the problem does not start with constant values in the vicinity of the interface that forms a kink at the interface with time. Instead, this version does the opposite: starts with a kink near the interface that becomes smooth with time. The mixed cell error then is the highest early on (after the initial conditions) and decreases gradually, as shown in Figure 8.8. Thus, this alternate version of this 2D MES test problem removes the boundary effects but adds initialization effects.

What is clear from Figure 8.8 is that the error on either side of the interface is low, indicating that the interfacial error does not spread into phase 1. Thus, the interfacial flux is adequate. The error values (not shown) for the cell-centered version of this problem are lowest for the S1 model and highest for the M1 model. Thus, in terms of E_∞, the models perform (from best to worst) as follows: S1, S3, M3, M2, S2, M1. The ratio between the error for a model and the error for the S1 model are between two and eight for each model except the M1 model, where the ratios are between 300 and 2000. Unlike in the 1D MES test, the multi-temperature models do not have the lowest error for the coarsest grids. It is likely that the error from using a linear approximation of a circular interface has overshadowed the physical accuracy of the M3 model.

8.3.3 Discussion

We run the 2D MES problem using two sets of initial conditions. The first case uses a constant initial condition in the entire domain and requires the analytical solution to be calculated at every time step in order to set the computational boundary values. We refer to this test as the constant case. Two computational domains were tested, $x, y \in [0, L]$ and $x, y \in [-L, L]$, which we will refer to as the quarter-circle and full-circle domain, respectively. The terms quarter- and full-circle refer to the amount of phase 1 (inner phase) that is captured in the computational domain. The quarter-circle problem results in a q_∞ convergence rate below first order for every model, Table 8.3. The S1 and M3 models preform best, having convergence rates of about 0.9 for the $s = 2$ and $s = 6$ ratios. The full-circle problem has better results than the quarter-circle problem, with the convergence rates closer to first order, Table 8.4. The S1, S2, and S3 models each has a convergence rate that average to about 1.1, while the M1, M2, and M3 models each has an average rate of about 0.9. Thus, while using the full-circle removes mixed cells from the boundary, it does not yield second-order convergence rates.

The second version of the problem uses an initial condition that is constant in phase 2, but quadratic in phase 1. We refer to this version as the quadratic case. By setting the phase 2 diffusivity to be larger and the initial phase 2 temperature equal to the boundary value, the temperature in phase 2 away from the interface is constant with time. This makes the computational boundary values constant, which alleviates the necessity to compute the analytical solution at every time step. The quadratic case is only run using the full circle domain. We find q_∞ convergence rates of about 0.8 for the S1, M2, and M3 models, while the S2 and S3 models perform best, with convergence rates of 1.0, Table 8.5. Thus, while the quadratic case alleviates the computational demand for computing the boundary values, the models do not achieve second-order convergence rates for this problem. This problem introduces an ambiguity when defining the initial conditions for mixed cells, where each phase has a unique temperature; this ambiguity was avoided in the previous version of the problem, where the initial conditions were constant.

In the constant case, the mixed-cell error grows with time as the solution evolves and the temperature profile begins to kink near the interface. Given enough time, the problem would reach steady state and give a constant temperature profile once again. The mixed cell error would accordingly diminish and vanish as the system reaches steady state. For the quadratic case, the temperature profile has a kink at the initial condition, and the profile smoothes out over time. We find that the mixed-cell error begins high for this case and declines with time. We expect that the mixed-cell error would vanish as the system reaches steady state, where the temperature profile is constant.

We find that our mixed-cell models are only first-order for this test problem, where we expect convergence rates closer to second-order at least for the S3 model. The most likely reason for this discrepancy is related to the fact that both the single- and multi-temperature codes make an assumption of a linear interface when calculating the parameters necessary for each mixed cell model (e.g., such as surface areas and interface normals). This first-order assumption of a higher-order interface may be the underlying reason why we only see first-order results. If this were indeed the problem, the solution would become more accurate on grids with higher resolution, as the curvature of the interface would be better approximated as linear. Table 8.5 supports this notion as the convergence rates tend to increase with higher resolution for the S1, M2, and M3 models. The expanded version of this table, Table C.8, shows that the S2 and S3 models follow this trend for the q_1 and q_2 convergence rates. However, the S2 and S3 models have a non-monotonic trend in the q_∞ rate, as does the S1 model for the q_1 and q_2 rate.

The multi-temperature models introduce a first-order discretization error by differentiating within the cell with varying grid sizes, as discussed regarding the 2D MMS test problem in Section 6.2.1. This causes a discontinuity in the mesh spacing for adjacent cells, which can lead to a first-order error [93]. Second-order convergence rates for a non-uniform grid requires smoothly varying grids with a change in cell width on the order of $\Delta x_{i+1} - \Delta x_i = \mathcal{O}(\Delta x_i^2)$ [93], whereas the multi-temperature models have a change in cell width on the order of $\Delta x_{i+1} - \Delta x_i = \mathcal{O}(\Delta x_i)$. The advantage of the multi-temperature models is that the physics of heat conduction is modeled more accurately in a mixed cell, particularly when the materials have highly discontinuous diffusivities and different temperatures. The lower convergence rate is offset by having low error at coarser grids (i.e., a single-temperature model will converge at a faster rate but start at a higher error). The point in which a single-temperature model becomes more accurate than a multi-temperature model depends on the material properties (diffusivity) and temperature differences. However, in this test, we do not find that the multi-temperature model begins with a lower error.

While this 2D MES test is an excellent benchmark problem for future mixed-cell models, the circular interface is not reconstructed well enough with the current implementation of the code. This also explains why the 2D MMS results in Section 6.2.1 were better than the 2D results in this section, as the 2D MMS problem utilizes a linear interface. Therefore, while the results for the models for this 2D MES test problem were not second-order, *it is more likely that the models did not receive adequate interfacial information rather than the models being only first-order in 2D.* Future work could retest these models (particularly the S1, S3, and M3 models) for this MES problem using a higher-order interface reconstruction scheme.

The 2D MES test problem described in this chapter makes an excellent benchmark test for future mixed-cell models. The first instance of this problem requires the analytical Fourier-Bessel series solution to be computed for each time step. This version is computationally demanding, and the boundary values drive the solution. We were able to recast the problem to yield constant boundary values. By using a high diffusivity for the outer phase and giving an initial temperature equal to the analytical boundary temperature, the computational boundary conditions remain constant. This makes this benchmark problem much more computationally feasible and less prone to error based on the eigenvalue computations. However, a nonconstant initial conditions introduces an issue for determining what temperature value is appropriate for a mixed cell; this same issue is avoided using a constant initial condition.

Another complication could be added to this problem by using multiple interfaces, as shown by Mikhailov, Özişik, and Vulchanov [147], with the constraint that the outermost phase must have a large diffusivity and the computational boundary is far enough from a interface to remain constant. Thus, the outer phase with high diffusivity would simply act as a boundary phase (similar to an immersed interface method), allowing the inner phases to take any form while keeping the boundary conditions constant.

8.4 Summary

We have shown a detailed derivation of a 1D radial, time dependent, two phase, analytic solution to a composite media heat conduction problem via SoV. The solution is an eigenvalue problem expressed as a Fourier-Bessel series. The cylindrical coordinates of the solution allow the radial solution to be projected onto a 2D Cartesian grid. Because a Cartesian grid cannot resolve a circular interface, this projection generates mixed cells at any resolution. Thus, we convert our 1D radial solution to a 2D problem as a means to test 2D behavior of the mixed-cells models with an analytic solution.

All six mixed-cell models fail to achieve second-order accuracy in the ∞-norm, each having the error dominated by the mixed-cells. We tried three different versions of this problem: (i) constant initial conditions with a quarter-circle domain, (ii) constant initial conditions with a full-circle domain, and (iii) quadratic initial conditions with a full-circle domain. We find that all three test problems gives first-order or lower convergence rates. This is likely due to the linear assumption made by both the single-temperature and multi-temperature codes during interface reconstruction. However, this linear assumption is not an inherent feature of the models, but only the codes that implemented these models. Therefore, if these models were implemented in a code with a higher-order interface reconstruction, they may achieve second-order convergence of E_∞, especially for the S3 and M3 models.

CHAPTER IX

Conclusion

The endeavor of this dissertation is to develop computational schemes for heat transfer with increased accuracy for mixed cells. We have presented five new mixed-cell models for heat transfer (diffusion) on an Eulerian mesh. These models can be used as part of a larger computational scheme for applications such as fluid dynamics or radiation transport.

Traditionally, mixed-cell models have been assessed by comparing their results to those obtained from one of two types of models: (i) other mixed cells models (e.g., taking the harmonic mean of the diffusion coefficients) or (ii) models without mixed cell (e.g., aligning the interface to the cell boundaries or using particle-based methods). We have selected and explored two analytical solutions that function as mixed-cell benchmark tests. These tests improve our ability to quantify the accuracy of mixed-cell models, which provides a consistent basis to which we can compare future mixed-cell models.

Ancillary to the models and analytic solutions, we investigated the Support-Operators Methods (SOM) and provided a complete 2D derivation that will enable future work to use SOM for tensor and/or skewed-mesh diffusion problems. SOM are mimetic finite difference methods (MFD) that discretize operators, such as the gradient or divergence, and then build the discrete analog of the differential equation using these operators while maintaining an integral identity. This integral identity relates the divergence and flux operators (diffusivity multiplied with the gradient), which is expressed in an inner product space such that the divergence and flux operators are adjoint. The diffusion equation is written in terms of these adjoint operators and then discretized. In contrast, standard finite difference methods (FDM) discretize the differential equation directly. These extra steps — defining the diffusion in terms of two operators discretized in an adjoint inner-product space — improve the fidelity of MFD for skewed grids and discontinuous diffusivity problems as compared to standard

FDM, as well as some instances of finite element methods (FEM) and finite volume methods (FVM). However, SOM yield equivalent results to simpler FDM for smooth or orthogonal grids and continuous or constant coefficients [185, 198]. Thus, SOM is best applied to more complicated problems, such as the strongly discontinuous coefficient problems explored in this thesis.

The main contributions of this work are summarized as follows:

- We selected and explored 1D and 2D MES tests to serve as mixed-cell model benchmarks.
- We demonstrated the application of the MMS to create 2D mixed-cell test cases.
- We presented cases where the S3 model achieves second-order convergence rates in three p-norms, including the ∞-norm.
- We described 1D cases where the M3 model achieves second-order convergence rates in three p-norms, including the ∞-norm.
- We demonstrated that the 2D M3 models yields E_∞ lower than any other model on the coarsest grids.
- We demonstrated that the S2 model is not advantageous to the S1 model.
- We showed that the M1 model is numerically inaccurate.

We conclude by reviewing the results of our new models and then discussing future work.

9.1 Performance of Mixed-Cell Models

rank	1DMMS	1DMES	2DMMS	2DMES*
1	M2/3	M2/3	S3	S1
2	M2/3	M2/3	S2	S2
3	S3	S3	S1	S3
4	S1	S1	M3	M1
5	M1	S2	M2	M2
6	S2	M1	M1	M3

Table 9.1: Performance ranking summary of the six mixed-cell models for all four main tests. The 2DMES test has an asterisk to indicate the test is suspect and gives first-order accuracy for every model with only slight variations.

9.1.1 Performance Rankings

The similarities and differences between the six mixed-cell models (five new and one reference model) are summarized in Table 6.1. Four mixed-cell test problems were performed: (i) a 1D steady-state method of manufactured solutions (MMS), (ii) a 1D dynamic method of exact solutions (MES), (iii) a 2D steady-state MMS, and (iv) a 2D dynamic MES.

For each test, we rank the relative performance of each mixed-cell model from best to worst using the average q_∞ as a metric. For multi-temperature models, we consider the phase-centered multi-temperature results (as opposed to the phase-averaged results). We refer to the M2 and M3 models as M2/3 in 1D, where they are identical. The rankings are summarized in Table 9.1.

1D MMS test (Section 6.1). For the 1D MMS test, the model rankings are M2/3, S3, S1, M1, and S2. Only the M2/3 and S3 models yield results that are better than first-order accurate for this test, with average convergence rates of 1.96 and 1.67, respectively. However, when q_∞ of the S3 model is considered for cases where the majority phase has the larger diffusivity, the average convergence rate is 2.01.

1D MES test (Chapter VII). For the 1D MES test, the model rankings are M2/3, S3, S1, S2, and M1. Only the M2/3 and S3 models have convergence rates greater than 1.3, with average rates of 1.81 and 1.59, respectively. However, when the S3 model is averaged using only the volume fractions where the majority phase has the larger diffusivity, the average convergence rate is 1.90.

2D MMS test (Section 6.2). For the 2D MMS test, the model rankings are S3, S2, S1, M3, M2, M1. Only the S2 and S3 models have convergence rates above 1.3, with average rates of 1.43 and 1.50, respectively. However, when the S3 model is tested using the modified implementation discussed in Section 6.2.2, the average convergence rate becomes 1.75. Furthermore, when the modified S3 model is averaged using only the two interfaces that do not cross opposite sides of a mixed cell, the average convergence rate is 1.88. Although the M2 and M3 models are only first-order accurate for this test, they both have the smallest E_∞ for the coarsest grid in each test. Additionally, when the S1 (reference) model is first-order accurate, the M2 and M3 models have smaller error than the S1 model at every resolution.

2D MES test (Chapter VIII). The implementation of this test was challenging, and therefore the results may be significantly affected by errors introduced in the implementation. Three configurations were attempted (quarter-circle domain with constant initial conditions, full-circle domain with constant initial conditions, and full-circle domain with quadratic initial conditions), and the full-circle domain with constant initial conditions is used for these rankings because it is the least likely version to be tainted by implementation error because there are no mixed cells on the boundary and there is no ambiguity in specifying the initial temperature value for a mixed cell. The results for each model were only first-order accurate, and the following rankings occur from the minor differences in these first-order convergence rates: S1, S2, S3, M3, M2, M1.

9.1.2 Discussion of Performance

1D. For the 1D test problems, we find that the M2/3 and S3 models perform best, each having error lower than all the other models while having the highest convergence rates. The average convergence rates, factoring both the MES and MMS 1D problems are 1.89 and 1.63 for the M2/3 and S3 models, respectively. While the M2/3 model performs consistently well for every volume fraction and diffusivity ratio tested, the S3 model varies in its performance. The S3 model (as well as the group of single-temperature models in general) does not perform well when the diffusivity ratio is as large as 10^{10}. When the results using this ratio are ignored, the S3 model is second-order accurate only when the majority phase has the larger diffusivity, while it is only first-order accurate otherwise. The average convergence rate for the S3 model in 1D problems is 1.95 when only factoring in results where the S3 model performs well (i.e., diffusivity ratio is less than 10^{10} and the majority phase has the larger diffusivity).

2D. Two 2D tests were used, a MMS and MES. While the 2D MES test problem was run with a few different implementations, we find that none of the models obtained greater than first-order convergence rates. This first-order result is most likely due to the fact that the codes employing these models assume a linear interface, while the interface for this problem is circular. The 2D MMS test results offer the best available measure of the accuracy for mixed-cell models in 2D because this test has a linear interface. Therefore, this analysis will primarily focus on the 2D MMS test while ignoring the 2D MES test.

Only the modified S3 model has a mean convergence rate significantly greater than first-order in 2D, with an average q_∞ of 1.75. This model is second-order accurate for two of the three interfaces tested (mean $q_\infty = 2.04$), which is an excellent result for the treatment of mixed cells. In the case where the S3 model is only first-order accurate (the L3 case in Section 6.2.1), the interface crosses either opposite or adjacent faces of each mixed cell. The discrepancy between the temperature predicted by the S3 model and the analytic solution is only large in some (but not all) mixed cells where the interface crosses opposite sides. In 1D, the interface can only cross opposite sides of a mixed cell. Therefore, the fact that the S3 model is first-order for the L3 interface appears be related to the first-order results seen in both 1D tests when the majority phase is the smaller diffusivity. Consequently, if this issue can be addressed within the framework of the S3 model, this model would be second-order accurate in any volume fraction in 1D and with any linear interface in 2D (for diffusivity ratios less than 10^{10}). Otherwise, the best option would be to package the S3 model with a new mixed-cell model that obtains second-order convergence rate when the larger diffusivity is the minority phase. The code that implements these models would choose which mixed-cell model to use based on the volume fractions and diffusivities of the mixed cell.

The multi-temperature results are only first-order accurate for the 2D test problems. Although these models take 2D effects into account (such as the centroid position depending on both x and y), first-order convergence rate is expected due to the subcell discretization using the phase centroids. By allowing each phase in a mixed cell to have a unique temperature, these models are more physically accurate. This notion is confirmed by the finding of lower error compared with the single-temperature models for coarsest grids in the 2D MMS test, as seen in Figure 6.4. The multi-temperature models also have the lowest error in the coarsest grids in 1D for both the MMS and MES tests. Further study is needed to investigate the multi-temperature models in 2D problems.

9.1.3 Conclusion for Rankings

We conclude from these results that the S2 and M1 models are the least accurate models tested. The approximations in the M1 model for the calculation of distances between each phase is very inaccurate, and the more refined approximations in the M2 and M3 methods yield superior convergence rates. Compared to the S1 model, the S2 model yields approximately the same first-order accuracy in 1D and moderately higher convergence rates (1.5 versus 1.3) in the 2D MMS problem (with approximately the same error magnitude). Thus, the extra cost to compute a full tensor diffusivity does not lead to sufficiently superior accuracy to be justified. The sub-second-order results of the S2 model suggest that the model does not accurately approximate the physics of an interface with an effective tensor diffusivity. Similar work by Berger and LeVeque [21], which uses a rotated reference frame to compute flux normal and tangent to the interface, is only first-order accurate near the interface. This suggests that techniques of this sort are not second-order accurate.

The S3 and M2/3 models are the most accurate of the new models developed for this thesis. The M2/3 model is second-order accurate in every 1D test. The S3 model is second-order accurate in the majority of the 1D tests as well as the 2D MMS test. The M3 model is more accurate than the M2 model in 2D, but the difference is small. The 2D results for the M3 model are only first-order accurate, but they achieve the lowest relative error compared with the other models on the coarsest grids considered. Furthermore, it is plausible that additional development of this model, such as including higher-order terms when computing mixed-cell fluxes, would yield second-order-accurate results in 2D.

The 1D and 2D MES tests find that the single temperature models (implemented via SOM) are less accurate when the diffusivity ratio is $10^{\pm 10}$. The multi-temperature methods have no decline in convergence rate for this ratio in the 1D case, and while the convergence rates suffer in the 2D case, they perform better than the single-temperature models. Therefore, the multi-temperature models appear to be more robust than the single-temperature models for diffusivities with a large discontinuity. However, none of the models are accurate for diffusivity ratios as large as $10^{\pm 10}$, and approximating one phase to be at constant temperature would treat this case more simply and more accurately.

9.2 Selection & Applications of Methods

We use two metrics to determine if a mixed-cell model has resulted in a more accurate solution: either E_∞ is lower for coarse grids or E_∞ converges at second-order accuracy. The S3 model meets this goal in certain cases, making a positive contribution to the field of mixed cells. The conditions where the S3 model is second-order include instances when the jump strengths are below a ratio of 10^{10} and the majority volume fraction has the larger diffusivity. In 1D, the *cell*-centered M2/3 model performs comparable to the S1 model, signifying that there is not an advantage to using the M2/3 model compared to standard finite differences with a harmonic mean. Yet, the *phase*-centered M2/3 model performs similarly to the S3 model, indicating that that M2/3 is a successful new mixed-cell model. However, the phase-centered M3 model does not match the original intention of this dissertation as it is not cell-centered. The excellent convergence rates of the 1D phase-centered results of the M2/3 model signify another useful contribution to the mixed-cell field. However, because it is not cell-centered, it can only be employed in a computational framework (e.g., hydrodynamics or radiation transport codes) with the capability to treat phase temperatures independently within a mixed cell.

The best single-temperature model tested is the S3 model, even though it is limited in the numerical schemes in which it may be implemented. The S1 and S2 models can be implemented in a variety of numerical methods, but the S3 model must be effectuated using SOM or a very similar method. The S3 model requires a scheme which utilizes a diffusivity-dependent shape matrix for each cell, which allows each cell corner to be treated separately. This is a specific requirement that may not be met in schemes other than SOM. Therefore, the S3 model, although an excellent mixed-cell model, cannot be implemented in most existing numerical methods.

The ideal application of the S3 model would be in situations where SOM is suitable, such as in a coupled hydrodynamic-diffusion code where the interface moves too quickly to create a body-fitted grid at every time step. The S3 model could treat any mixed cells, thereby increasing the accuracy of the code, provided sufficient interfacial knowledge is available (e.g., volume fractions or a level-set function). Additionally, the S3 model could be used in a scheme which also employs an adaptive mesh refinement (AMR) method because SOM (and hence the S3 model) are compatible with AMR methods. In such a scheme, the criteria for the AMR algorithm to resolve interfaces could be relaxed, and the S3 model would maintain second-order accuracy for the mixed-cells at a lower interfacial refinement.

The phase-centered M2/3 model is the next best model; however, it is also limited in applicability. First, it can only be used in a code that tracks both centroids and volume fractions. Second, it requires a temperature for each phase in a mixed cell, a requirement which is not implemented in most codes. Therefore, if one desires to utilize this model in a multiphase computational framework, the code must be written such that each phase has a unique temperature, i.e., LTE is not assumed for adjacent phases. This assumption is more physically accurate but requires an additional temperature variable to be evolved.

An ideal method to use the M2/3 model would be an Arbitrary Lagrangian Eulerian (ALE) method. An ALE method updates the fluid location using a Lagrangian method and then projects the skewed mesh onto an Eulerian mesh to update the fluid parameters. The projection onto the Eulerian mesh typically leads to mixed cells. In such a scheme, the M2/3 model could compute the temperature of each phase separately. The Lagrangian step could be altered such that the individual phase temperatures found in the Eulerian step could be used, thereby increasing the model's accuracy of the physics as well as the remapping process. The centroids for each phase in the Eulerian mesh are needed, however.

The harmonic mean (S1 model) is the best model tested which can be implemented in a single-temperature, cell-centered scheme without using SOM. The harmonic mean emphasizes the smaller diffusivity; thus it is most accurate if the majority phase of a mixed cell is the smaller diffusivity. The S1 model is generally second-order accurate in the 1-norm, around 1.5 in the 2-norm, and only between 1.0 and 1.4 in the ∞-norm. Thus, the standard technique of using the harmonic mean provides ease of implementation while remaining versatile and at least first-order accurate.

9.3 Insights Gained from this Work

In general, the primary insight we gained from this work is that the mixed cells should be split into pure cells, either by cutting or effectively partitioning them. These techniques are superior because each phase is solved independently, which is more physically accurate. This is evident in the analytic solution presented in the 1D and 2D MES test problems, which solve the governing equation for each phase separately, patching the solution from each phase together by continuity of temperature and flux.

For a cell-centered method, we find that the S3 model is the best option amongst the models tested. This model effectively splits the mixed cell by partitioning the calculation of the flux into four regions of the cell. Similarly, the M2/3 model, which obtains lower error for coarse grids but is not cell centered, also calculates the flux and temperature of each phase separately. Alternatively, the cells can be cut into two or more pure cells, creating additional grid elements of irregular size. Another option is to align the cell faces with the interface, creating a body-fitted grid without mixed cells. If a cell-centered method is required that does not treat each phase separately and cannot cut the mesh or fit the grid to the interface, then homogenization is the best remaining option, and one must accept only first-order convergence rates on the interface.

9.4 Future Work

There are a variety of avenues in which the models presented in this thesis could be further explored.

3D. All of the new models discussed can be extended to 3D. The M2 and M3 models are useful mixed-cell models for potential 3D applications. The primary difficulty with a 3D version of these models is tracking of the centroid for each phase; however, there already exists at least one method of doing so (MoF). Thus, a multi-temperature, multiphase, Eulerian mesh, 3D diffusion solver with moving interfaces could employ the M2 or M3 model to improve mixed-cell accuracy.

SOM has already been derived for a variety of 3D grid types [34, 35, 36, 37, 135, 136, 151, 185, 186]. Any existing 3D SOM code could immediately implement the S3 model because each 3D element, most generally a polyhedron, can have a unique value for the diffusivity and volume defined in each corner. Most of the 3D SOM examples have body-fitted grids, but this restriction can be relaxed if using the S3 model. The primary difficulty is determining the interface location and the proper volume fractions, but this is handled by an interface reconstruction scheme, separate from the S3 model. The number of possible interfacial configurations for the S3 model is greater in 3D than 2D, where there are only four (interfaces can cross either two corners, one corner, adjacent faces, or opposite faces). Therefore, implementing the S3 model in 3D would require some additional development.

Additional testing. The mixed-cell models, particularly the S3 and M3, could be tested for more cases of the 1D and 2D analytic solutions. More 2D MMS solutions could be tested. There are many potential 2D MMS tests that could be used; we only explored one at a single diffusivity ratio. A test with an interface orthogonal to the grid, as in the test presented in Section 5.3.4 but with temperature variation in both the x- and y-directions, would be a simple but interesting test, especially to explore the 2D behavior of the M3 model. A MMS test using more than one interface and more than two phases in the domain, but not necessarily in a mixed cell, would also be of interest.

Computational efficiency. The overall benefit of a mixed-cell model is best gauged by weighing the increase in accuracy versus the increase in computational effort. This work does not quantitatively consider computational efficiency, but only compares models in terms of the convergence rate of the relative error with respect to grid resolution. A comparison of computational efficiency would involve first tuning the code such that the model implementation yields near-peak performance, which would include investigating such topics as preconditioners, parallelization, compiler optimization, and the layout of the coefficient matrix (which can be organized to minimize computational effort). After the efficiency is maximized, mixed-cell models can be evaluated for both accuracy and computational cost. One may find, for example, that the benefit of the second-order S3 method only overcomes the extra cost of SOM when the mesh is sufficiently skewed or when the diffusivity discontinuity is suitably large. Alternatively, one may find that the low cost of the M3 model combined with the low error at coarse grids compensates for a lower convergence rate order.

Numerical methods. Codes that use the mixed-cell models can be improved in a number of ways. A more accurate time stepping scheme, such as Crank-Nicholson [54], could be implemented with SOM to increase the time-step size. Maintaining accuracy with larger time steps becomes important for computational speed, to compensate for increases in problem size, grid complexity, and number of phases.

Research into mixed-cell models on more complicated grids is a logical next step. The S3 model can be incorporated with an AMR scheme since SOM is compatible AMR schemes for general quadrilaterals [132]. SOM also works naturally with meshes from Lagrangian hydrodynamics methods. The multi-temperature model could be explored through implementation with more advanced methods, such as potential integration with SOM or FVM to achieve grid flexibility. As a consequence, both single-temperature and multi-temperature mixed-cell models could be explored using general grids, which is of interest to the hydrodynamics community.

A higher-order interface reconstruction method could be used. Our models have only been tested with a linear interface approximation. The performance of these models may increase with a more accurate interface reconstruction method, particularly with the 2D problem in Chapter VIII. The Moment of Fluid (MoF) method is an excellent option for accurate interface reconstruction. It is capable of 2D and 3D implementation, it is compatible with AMR grids, and it is successful with reconstructing multiple interfaces [2, 3, 6, 69, 68, 70]. The multi-temperature methods would benefit most from MoF. An alternative interface tracking scheme of interest is the level set methods (LSM), which can provide curvature information in addition to more accurate knowledge of interfacial position.

Physical problems. This thesis focused on a linear heat conduction problem between multiple phases with a fixed interface. This simplification was imposed in order to focus on the accuracy of mixed cells by eliminating all other effects. However, testing these models on problems with other physical effects, such as fluid motion, would be of greater interest to the community. Future work could involve coupling these models with a hydrodynamics code. The models could be applied to more accurately simulate nonlinear diffusion as an approximation to radiation transport. Inclusion of position- and temperature-dependent heat capacities, densities, and opacities would represent a more physically realistic case.

Expanding the mixed-cell models to handle three or more phases would be of great importance. A two-phase case is the logical starting point, but generalizing to n phases is more applicable to a wider range of problems and cases. The CRASH code, for example, includes five phases.

Additional models. A combination of the S1 and S3 models is a potentially better model. The S1 model, which uses the harmonic mean to give an effective diffusivity, is more accurate when the majority phase of a mixed cell has the *smaller* diffusivity. The S3 model, as shown in the 1D MMS and 1D MES tests, has better performance when the majority phase of the mixed cell has the *larger* diffusivity. Accordingly, an intermediate model would define the mixed cell choosing either the S1 or S3 model, depending on the diffusivity of the majority phase. The downside to this combination is that the S1 model is not second-order accurate.

Another potential model would combine the rotated tensor diffusivity (S2) model with the S3 model. In such a case, the tensor diffusivity could be assigned to some of the corners of a mixed cell. The failure of the S2 model as implemented may be due to the fact that every face on a mixed cell uses the same effective diffusivity tensor, despite the fact that the interface only crosses two or less faces. Thus, combining the S2 model with the S3 model would allow the diffusion of a pure face (a face not crossed by an interface) to be defined by that phase and not some anisotropic mixture.

An additional model could modify SOM to place a face-centered unknown on the interface of a mixed cell, similar to 1D work by Winters and Shashkov [216]. SOM already places face-centered unknowns on the exterior faces of every cell to calculate flux, and thus including one additional unknown for each mixed cell would not significantly increase the overall cost. Additional complexity lies in determining how to recalculate the cell-centered temperatures in mixed cells from the face-centered values.

Finding a model that gives second-order accuracy in E_∞ (or gives lower error than the standard treatments for coarse grids) for all test cases is the consummate goal of a mixed-cell model. While we found the M3 and S3 models indeed give this accuracy in *some* cases, they are not second-order accurate in *all* cases. Incorporating previous work with the S3 and M3 models, such as an improved Taylor expansion near interfaces, or a predictor-corrector method, may be a useful approach to consider. Fully determining where and why the S3 and M3 models fail to give second-order accuracy may guide future research as well. A logical starting point for the multi-temperature models would be to explore why they give different results depending on the interface location when the diffusivities equal. A logical starting point for the S3 model would be to explain why it requires the majority phase to be the larger diffusivity in order to achieve second-order accuracy in 1D.

Investigations can be performed using other error norms, such as E_1 and E_2. We chose the E_∞ as an error metric in order to focus on the accuracy of the solution on or near the interface. However, if one is more interested in the accuracy of the solution in the bulk (i.e., far from the interface), then another error metric may prove to be more useful. The accuracy of the interface is only important to a bulk solution if the interfacial error spreads into the bulk region. If the interfacial error (which is typically higher than the bulk error) remains local to the interface, the accuracy of the bulk solution is best measured with error norms that do not isolate the largest error in the domain (such as E_1 and E_2). Alternatively, the E_∞ can also serve as a measure of the error in the bulk region if it is calculated using only data from the bulk regions (i.e., filtering out data from the interfacial region). This may be useful since it will provide a stringent assessment of the bulk error; by using E_∞, any localized error would not be diluted due to averaging that occurs with other means.

APPENDICES

APPENDIX A

Split-Zone (S3) Model Implementation Details

This appendix derives the subcell volumes and diffusivities used for the split-zone (S3) model. We first note that the interface is assumed to be sufficiently resolved for nonlinear cases. This implies that the discretized representation of a curved or complicated interface must be represented by sufficient number of mesh points that when viewing a 3×3 stencil, the interface can be approximated as linear. Pember *et al.* [170] discuss such resolution requirements in further detail. Note that the subcells discussed in this section are virtual subcells, meaning that the subcells quantities are used as intermediate step with the shape matrix; we do not replace a mixed-cell with four smaller cells. In this section, the term ‘virtual’ is implied whenever ‘subcell’ is used.

The expressions for the subcell volumes for the S3 model varies for four interface cases: crosses opposite faces, crosses one corner, crosses adjacent faces, and crosses two corners. Figures A.1 and A.2 show the interfacial configurations and the resulting split-cell configurations for the particular orientations we derived here. The other orientations have analogous expressions. All cases are identified by the volume fractions of the nearest neighbors with respect to the volume fraction of the mixed cell in question. The surface fractions are approximated from the volume fractions, where a surface fraction is the fraction of the surface area that belongs to a particular phase with respect the total surface area of that face of the mixed cell. Let the surface fractions of the left, right, top, and bottom face be denoted a_L, a_R, a_T, and a_B, respectively. We take the volume and surface fractions to be of the majority phase, which in this case is phase 1 (blue). With this definition, the majority phase is phase 1, which is render as blue in all figures. Let the volume fraction be denoted as f, which is written in terms of majority phase (phase 1), and the volume fraction

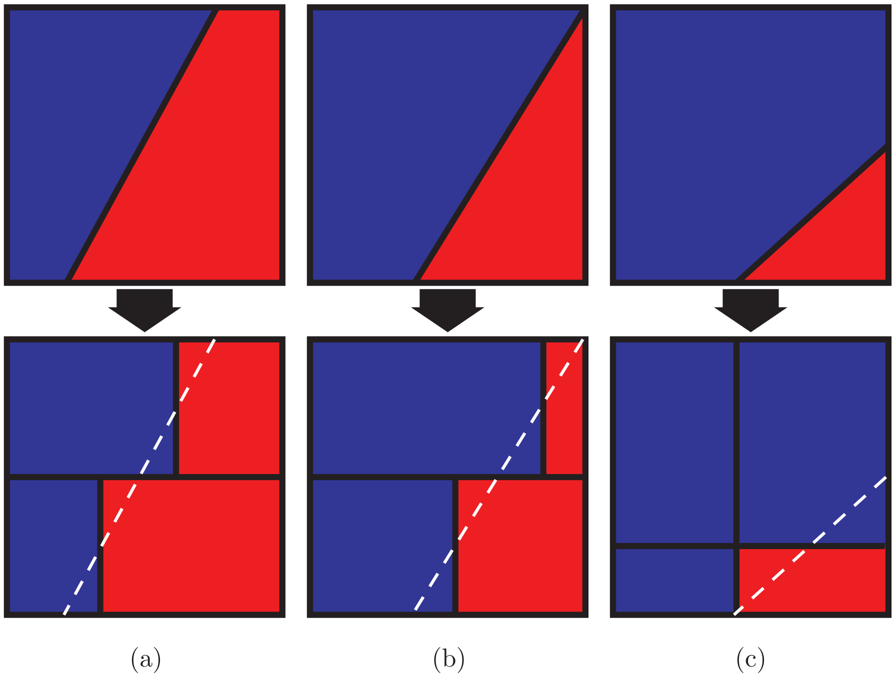


Figure A.1: Three of the four classes of configurations that arise in the S3 (split-zone) model for mixed cells in the single-temperature method. The S3 model divides a mixed cell into four virtual subcells associated with each corner of the cell such that the volume of each phase is conserved. Note that only the phase and volume of each virtual subcell determine the diffusivity assigned at the associated cell corner in the S3 model; the orientation is for visualization purposes only. The top row shows the mixed cell and the bottom row shows the corresponding virtual subcells. The classes are distinguished by how the interface crosses the cell: the interface (a) crosses opposite faces, (b) crosses one corner, and (c) crosses adjacent faces. The majority phase, phase 1, is rendered in blue, while the minority phase, phase 2, is rendered in red.

of the minority phase is $(1 - f)$. A volume fraction without a subscript indicates the volume fraction of the center cell, while a volume fraction with two subscripts refers the fraction of a phase in a split-zone corner (i.e., f_{LT} is the amount of phase 1 in the upper-left subcell relative to the total amount of phase 1 in the mixed cell).

Crosses opposite faces. For this case, Figure A.1(a), we cut each trapezoid in half by height. The upper-left volume fraction is the volume of top half of the trapezoid

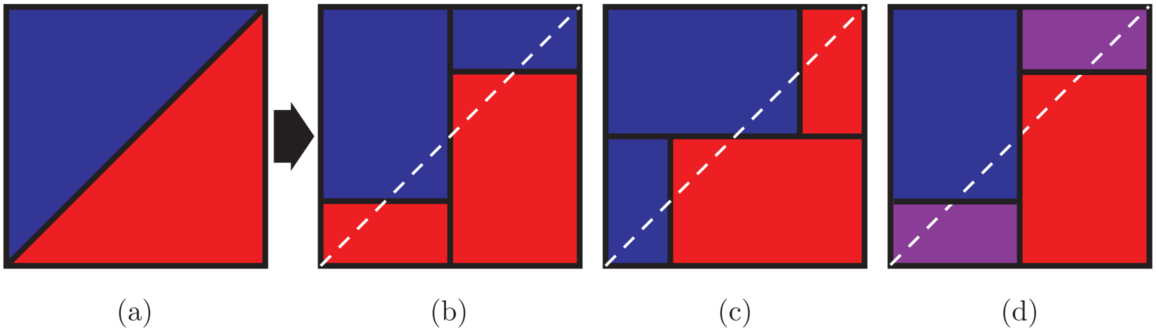


Figure A.2: The fourth class of configurations for the S3 (split-zone) model for mixed cells in the single-temperature method. This case is when the interface crosses two corners, as shown in (a). Note that only the phase and volume of each virtual subcell determine the diffusivity assigned at the associated cell corner in the S3 model; the orientation is for visualization purposes only. Figures (b) and (c) show two ways to split this cell while preserving phase volumes and minimizing the ambiguous regions. Both options result in a large and small rectangle (3/8 and 1/8 of the cell volume) of each phase. The LT and RB virtual subcells clearly belong to one of the two phases, defining the diffusivities at these corners, but the LB and RT corners have two possible assignments of the phases. Case (d) resolves the ambiguity of phase assignments in the RT and LB virtual subcells via a harmonic-averaged diffusivity (shown as purple).

divided by the total trapezoid volume,

$$f_{LT} = \frac{\left[\frac{1}{2}a_L\right]\left[\frac{1}{2}\left(a_T + \frac{a_T + a_B}{2}\right)\right]}{[a_L]\left[\frac{a_T + a_B}{2}\right]} \tag{A.1a}$$

$$= \frac{1}{4}\frac{3a_T + a_B}{a_T + a_B}, \tag{A.1b}$$

where this volume fraction is with respect to the trapezoid and must be multiplied by $f\Delta x\Delta y$ to give the physical volume. Similarly, the volume fraction for the lower-left face is found to be

$$f_{LB} = \frac{1}{4}\frac{a_T + 3a_B}{a_T + a_B}. \tag{A.2}$$

Thereby, the four subcell volumes for the case where the interface crosses the top and bottom sides are

$$V_{LT} = \frac{1}{4}\frac{3a_T + a_B}{a_T + a_B} f\Delta x\Delta y \tag{A.3a}$$
$$V_{LB} = \frac{1}{4}\frac{a_T + 3a_B}{a_T + a_B} f\Delta x\Delta y \tag{A.3b}$$
$$V_{RT} = \frac{1}{4}\frac{3(1-a_T) + (1-a_B)}{(1-a_T) + (1-a_B)}(1-f)\Delta x\Delta y \tag{A.3c}$$
$$V_{RB} = \frac{1}{4}\frac{(1-a_T) + 3(1-a_B)}{(1-a_T) + (1-a_B)}(1-f)\Delta x\Delta y, \tag{A.3d}$$

where there are similar expressions for when the interface crosses the left and right faces. Note that these four corner volumes sum to $\Delta x\Delta y$. The diffusivities are $D_{LT} = D_{LB} = D_1$ and $D_{RT} = D_{RB} = D_2$.

Crosses one corner. For this case, Figure A.1(b), we work first with the triangle of phase 2 (red), cutting this triangle into a triangle and a trapezoid of equal height. We introduce two dummy variables, p and q to describe the width of the rectangles to which we map this triangle and trapezoid. Since terms will cancel, let us describe the full triangle with a generic height h and base width w. The upper-right volume fraction is the volume of the smaller triangle divided by the volume of the larger triangle,

$$f_{RT} = \frac{\frac{1}{2}\left[\frac{1}{2}w\right]\left[\frac{1}{2}h\right]}{\frac{1}{2}\left[w\right]\left[h\right]} = \frac{1}{4}. \tag{A.4}$$

This corresponds to a square of height $\frac{1}{2}$ and width p, where

$$[\tfrac{1}{2}][p] = f_{RT}(1-f) \tag{A.5a}$$
$$p = \tfrac{1}{2}(1-f). \tag{A.5b}$$

Similarly, the lower-right volume fraction is the volume of the trapezoid divided by the volume of the larger triangle,

$$f_{RB} = \frac{\left[\frac{1}{2}h\right]\left[\frac{1}{2}(w+\frac{1}{2}w)\right]}{\frac{1}{2}\left[w\right]\left[h\right]} = \frac{3}{4}. \tag{A.6}$$

This corresponds to a square of height $\frac{1}{2}$ and width q, where

$$[\tfrac{1}{2}][q] = f_{RB}(1-f) \tag{A.7a}$$
$$q = \tfrac{3}{2}(1-f). \tag{A.7b}$$

As a consequence, the four subcell volumes for the case where the interface crosses a face and the upper-right corner are

$$V_{RT} = \tfrac{1}{2}p\Delta x\Delta y = \frac{1}{4}(1-f)\Delta x\Delta y \tag{A.8a}$$

$$V_{RB} = \tfrac{1}{2}q\Delta x\Delta y = \frac{3}{4}(1-f)\Delta x\Delta y \tag{A.8b}$$

$$V_{LT} = \tfrac{1}{2}(1-p)\Delta x\Delta y = \frac{1}{4}(1+f)\Delta x\Delta y \tag{A.8c}$$

$$V_{LB} = \tfrac{1}{2}(1-q)\Delta x\Delta y = \frac{1}{4}(3f-1)\Delta x\Delta y, \tag{A.8d}$$

where there are similar expression for the case in which the interface crosses a face and one of the other three corners. Note that these four corner volumes sum to $\Delta x\Delta y$, and, as mentioned earlier, the volume fractions are of the majority phase ($f \geq \frac{1}{2}$). The diffusivities are $D_{LT} = D_{LB} = D_1$ and $D_{RT} = D_{RB} = D_2$.

Crosses adjacent faces. For this case, Figure A.1(c), we map the small triangle of the minority phase (volume fraction $(1-f)$) to a rectangle of height p and width q. We approximate the values of p and q through the following substitutions,

$$pq = 1-f \tag{A.9a}$$

$$= \frac{(1-a_R)(1-a_B)(1-f)}{(1-a_R)(1-a_B)} \tag{A.9b}$$

$$= \sqrt{\frac{(1-a_R)^2(1-f)}{(1-a_R)(1-a_B)}}\sqrt{\frac{(1-a_B)^2(1-f)}{(1-a_R)(1-a_B)}} \tag{A.9c}$$

$$= \sqrt{\frac{(1-a_R)(1-f)}{(1-a_B)}}\sqrt{\frac{(1-a_B)(1-f)}{(1-a_R)}}. \tag{A.9d}$$

Therefore, we define

$$p = \sqrt{\frac{(1-a_R)(1-f)}{(1-a_B)}} \tag{A.10a}$$

$$q = \sqrt{\frac{(1-a_B)(1-f)}{(1-a_R)}}. \tag{A.10b}$$

These expressions give the correct limiting behavior: both $(1-a_R)$ and p increase in value as the height of the triangle grows, and both $(1-a_B)$ and q increase in value as the base of the triangle grows. For that reason, in the case where the interface crosses the bottom and right faces of the cell, we express the four subcell volumes in terms of p and q as,

$$V_{RB} = pq\Delta x\Delta y \tag{A.11a}$$

$$V_{LB} = p(1-q)\Delta x\Delta y \tag{A.11b}$$

$$V_{LT} = (1-p)q\Delta x\Delta y \tag{A.11c}$$

$$V_{LB} = (1-p)(1-q)\Delta x\Delta y, \tag{A.11d}$$

where there are similar expressions for the other three cases in which the interfaces cross adjacent faces. Again, these four volumes sum to $\Delta x\Delta y$. The diffusivities are $D_{LT} = D_{LB} = D_{RT} = D_1$ and $D_{RB} = D_2$.

Crosses two corners. As explained in Figure A.2, there are a variety of ways to express the subcell volumes in the case where the interface crosses two corners. Both case (b) and (c) in Figure A.2 are equally valid and yield a upper-left and lower-right subcell volume of $\frac{3}{8}\Delta x\Delta y$. However, while both (b) and (c) also yield a lower-left and upper-right subcell volume of $\frac{1}{8}\Delta x\Delta y$, these smaller subcells correspond to different phases; see Figure A.2(b-c). Unlike the upper-left and lower-right subcells, it is not clear which phase the lower-left and upper-right subcells should have. Therefore, we use the harmonic mean to combine the two diffusivities for these subcells, as shown in case (d), giving an effective diffusivity for those subcells. Note that the S3 model only specifies a volume and diffusivity in each subcell. Thus, the upper-left subcell is equivalent for cases (b) and (c) because they both define this subcell as phase 1 with volume of $\frac{3}{8}\Delta x\Delta y$. This same argument applies to the lower-right subcell, where both (b) and (c) define it as phase 2 with volume of $\frac{3}{8}\Delta x\Delta y$. Consequently, only the phase and volume of each subcell is important, and it is irrelevant that the larger rectangles are shown in case (d) as being tall rather than wide. If (d) showed the larger rectangles as being wide, the following would still be true: the upper-left subcell is phase 1 with volume $\frac{3}{8}\Delta x\Delta y$, the bottom-right subcell is phase 2 with volume $\frac{3}{8}\Delta x\Delta y$, and both the upper-right and lower-left subcells are the harmonic mean of phases 1 and 2 with volume $\frac{1}{8}\Delta x\Delta y$.

Unlike all other cases in the S3 method, which yield only pure cells, this case only reduces the mixed-cell volume from the full mixed cell to one quarter of the mixed cell. Splitting the mixed cell into four equally sized virtual subcells would only reduces the mixed-cell volume to one half of the mixed cell, so our choice of volumes further reduces mixed cell effects. (See Equations (A.4) and (A.6) and their associated discussions justification of why the triangles are cut into squares containing one- and three-quarter fractions of the triangle's volume.) The volumes of each subcell are given as

$$V_{RB} = \frac{3}{8}\Delta x \Delta y \tag{A.12a}$$

$$V_{LB} = \frac{1}{8}\Delta x \Delta y \tag{A.12b}$$

$$V_{LT} = \frac{3}{8}\Delta x \Delta y \tag{A.12c}$$

$$V_{LB} = \frac{1}{8}\Delta x \Delta y, \tag{A.12d}$$

The diffusivities are $D_{LT} = D_1$, $D_{RB} = D_2$, and $D_{LB} = D_{RT} = \frac{2D_1D_2}{D_1+D_2}$.

APPENDIX B

SOM Shape Matrix Derivation

This section discretizes the LHS of the following integral identity,

$$\int_V \vec{J}\cdot(\mathbf{D}^{-1}\vec{F})dV = \int_V U(\vec{\nabla}\cdot\vec{J})dV - \oint_{\partial V}(U\vec{J})\cdot\hat{n}dS. \tag{B.1}$$

The discretization of the two terms on the RHS, Equations (3.33) and (3.35), occur in the face-normal coordinate system, where a vector component J_s points in the outward normal direction to side s. Since the vector contributions of these integrals reduce to surface contributions, it is not necessary to specify elements in terms of the x-y coordinate system. However, the LHS of Equation (B.1) contains dot products. A coordinate transform is required to form the discrete analog of this term. This is because the continuum functions existed in the x-y coordinate system, where $\hat{x}\cdot\hat{y}=0$; while the discrete variables are written in a face normal coordinate system, where $\hat{n}_s\cdot\hat{n}_{s'}=\cos\theta_{ss'}$ and $\theta_{ss'}$ is the angle between sides s and s'. For a rectangular cell, $\theta_{ss'}=\frac{\pi}{2}$ for all corners, giving zero for the cosine term. Accounting for the nonzero dot products for a general quadrilateral takes the next few pages.

We first jump to the answer,

$$\begin{aligned}\int_V \vec{J}\cdot(\mathbf{D}^{-1}\vec{F})dV \approx &(\vec{J}_{LT}\cdot\mathbf{S}_{LT}\vec{F}_{LT})V_{LT} + (\vec{J}_{RT}\cdot\mathbf{S}_{RT}\vec{F}_{RT})V_{RT}\\ &+(\vec{J}_{LB}\cdot\mathbf{S}_{LB}\vec{F}_{LB})V_{LB} + (\vec{J}_{RB}\cdot\mathbf{S}_{RB}\vec{F}_{RB})V_{RB},\end{aligned} \tag{B.2}$$

where $\mathbf{S}_{ss'}$ is a 2×2 'shape' matrix for terms at the vertex at the intersection of side s and side s'. Similarly, $\vec{J}_{ss'}$ and $\vec{F}_{ss'}$ are two-element vectors pointing in the $\hat{n}_s$ and $\hat{n}_{s'}$ directions. The volume, $V_{ss'}$, represents the volumetric weight of the ss' corner. For a general quadrilateral, each corner weight is one quarter of the area defined by the parallelogram formed by mirroring sides s and s' [152]. Note that defining the volumetric weights in this manner may require normalizing the four corner volumes such that their sum is equal to the volume of the cell. For a rectangle, the four volumetric weights are all simply $\frac{1}{4}\Delta x \Delta y$.

We now introduce a coordinate transform matrix ($\mathbf{G}_{ss'}$) in order to fully explain Equation (B.2) and the $\mathbf{S}_{ss'}$ matrices. We denote the vectors in the x-y system using the standard arrow (e.g., $\vec{F}$), while using the bar notation (e.g., $\bar{F}$) to indicate the face-normal coordinate system. The face-normal vector can be expressed in terms of an x-y vector as follows:

$$\bar{F}_{ss'} = \mathbf{G}_{ss'}\vec{F}_{ss'} \tag{B.3a}$$

$$\begin{bmatrix} F_s \\ F_s' \end{bmatrix} = \begin{bmatrix} \hat{x}\cdot\hat{n}_s & \hat{y}\cdot\hat{n}_s \\ \hat{x}\cdot\hat{n}_s' & \hat{y}\cdot\hat{n}_s' \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix} \tag{B.3b}$$

$$\begin{bmatrix} F_s \\ F_s' \end{bmatrix} = \begin{bmatrix} \cos\theta_{xs} & \cos\theta_{ys'} \\ \cos\theta_{xs'} & \cos\theta_{ys} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix} \tag{B.3c}$$

$$\begin{bmatrix} F_s \\ F_s' \end{bmatrix} = \begin{bmatrix} \cos\theta_{xs} & \sin\theta_{xs} \\ \cos\theta_{xs'} & \sin\theta_{xs'} \end{bmatrix} \begin{bmatrix} F_x \\ F_y \end{bmatrix}, \tag{B.3d}$$

where we use the fact that $\cos\theta_{ys} = \sin\theta_{xs}$ on the last step, which follows because $\theta_{ys} + \theta_{xs} = \frac{\pi}{2}$. This is a useful step because it reduces the number of angles from four to two. Hence, $\mathbf{G}_{ss'}$ for a general vertex is

$$\mathbf{G}_{ss'} = \begin{bmatrix} \cos\theta_{xs} & \sin\theta_{xs} \\ \cos\theta_{xs'} & \sin\theta_{xs'} \end{bmatrix}. \tag{B.4}$$

The inverse is

$$\mathbf{G}_{ss'}^{-1} = \frac{1}{\sin\theta_{xs'}\cos\theta_{xs} - \sin\theta_{xs}\cos\theta_{xs'}} \begin{bmatrix} \sin\theta_{xs'} & -\sin\theta_{xs} \\ -\cos\theta_{xs'} & \cos\theta_{xs} \end{bmatrix}. \tag{B.5}$$

We can now write the dot product between x-y vectors in terms of face-normal vectors by using the inverse of the transform matrix $\mathbf{G}_{ss'}$,

$$(\vec{J}_{ss'}, \vec{F}_{ss'}) = (\mathbf{G}_{ss'}^{-1}\bar{J}_{ss'}, \mathbf{G}_{ss'}^{-1}\bar{F}_{ss'}). \tag{B.6}$$

Since $\mathbf{G}_{ss'}$ is real, we can move it to other side of the inner product by taking the transpose,

$$(\mathbf{G}_{ss'}^{-1}\bar{J}_{ss'}, \mathbf{G}_{ss'}^{-1}\bar{F}_{ss'}) = (\bar{J}_{ss'}, (\mathbf{G}_{ss'}^{-1})^T\mathbf{G}_{ss'}^{-1}\bar{F}_{ss'}). \tag{B.7}$$

The product $(\mathbf{G}_{ss'}^{-1})^T\mathbf{G}_{ss'}^{-1}$ is easily evaluated in 2D:

$$
\begin{aligned}
(\mathbf{G}_{ss'}^{-1})^T\mathbf{G}_{ss'}^{-1} &= \frac{1}{(\sin\phi\cos\theta - \sin\theta\cos\phi)^2}\begin{bmatrix} \sin\phi & -\cos\phi \\ -\sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} \sin\phi & -\sin\theta \\ -\cos\phi & \cos\theta \end{bmatrix} \\
&= \frac{1}{(\sin(\phi-\theta))^2}\begin{bmatrix} \sin^2\phi + \cos^2\phi & -\sin\phi\sin\theta - \cos\phi\cos\theta \\ -\sin\phi\sin\theta - \cos\phi\cos\theta & \sin^2\theta + \cos^2\theta \end{bmatrix} \\
&= \frac{1}{(\sin(\phi-\theta))^2}\begin{bmatrix} 1 & -\cos(\phi-\theta) \\ -\cos(\phi-\theta) & 1 \end{bmatrix} \\
&= \frac{1}{\sin^2\theta_{ss'}}\begin{bmatrix} 1 & -\cos\theta_{ss'} \\ -\cos\theta_{ss'} & 1 \end{bmatrix},
\end{aligned} \tag{B.8}
$$

where ϕ and θ correspond to θ_{xs} and $\theta_{xs'}$. Note that the square of the sine term and the fact that cosine is an even function means both $\theta - \phi$ and $\phi - \theta$ give the same result. Equation (B.8) defines the shape matrix when the diffusivity is a scalar. Also note that for $\theta_{ss'} = \frac{\pi}{2}$, Equation (B.8) reduces to the identity. Accordingly, for orthogonal grids, the $\mathbf{S}$ matrices in Equation (B.2) are all equal to the identity, and the integral is approximated as

$$
\int_V \vec{J}\cdot(D^{-1}\vec{F})dV \approx \frac{\Delta x\Delta y}{2D}(J_LF_L + J_RF_R + J_TF_T + J_BF_B). \tag{B.9}
$$

However, for a tensor diffusivity, the LHS of Equation (B.1) requires the inner product of $(\vec{J}, \mathbf{D}^{-1}\vec{F})$. Using the same steps, we find that

$$
\begin{aligned}
(\vec{J}_{ss'}, \mathbf{D}^{-1}\vec{F}_{ss'}) &= (\mathbf{G}_{ss'}^{-1}\bar{J}_{ss'}, \mathbf{D}_{ss'}^{-1}\mathbf{G}_{ss'}^{-1}\bar{F}_{ss'}) && \text{(B.10a)} \\
&= (\bar{J}_{ss'}, (\mathbf{G}_{ss'}^{-1})^T\mathbf{D}_{ss'}^{-1}\mathbf{G}_{ss'}^{-1}\bar{F}_{ss'}), && \text{(B.10b)}
\end{aligned}
$$

where $\mathbf{G}_{ss'}$ is defined exactly as in Equation (B.4). The diffusivity is already written in the x-y coordinate system, so it does not require any factors of $\mathbf{G}$. We then define the shape matrix as,

$$
\mathbf{S}_{ss'} = (\mathbf{G}_{ss'}^{-1})^T\mathbf{D}_{ss'}^{-1}\mathbf{G}_{ss'}^{-1}. \tag{B.11}
$$

The evaluation of this product proceeds similarly to that in Equation (B.8), with the one additional matrix product from the 2×2 diffusivity tensor. The presence of this tensor prevents the use of trigonometric identities to give a simple, compact answer.

Therefore, we write the shape matrix in component form as

$$\mathbf{S}_{ss'} = \begin{bmatrix} S^{s,s}_{ss'} & S^{s,s'}_{ss'} \\ S^{s',s}_{ss'} & S^{s',s'}_{ss'} \end{bmatrix}, \tag{B.12}$$

where each component is

$$S^{s,s}_{ss'} = \frac{1}{\sin^2\theta_{ss'}}[K^{xx}_{ss'}\sin^2\theta_{xs'} + K^{yy}_{ss'}\cos^2\theta_{xs'} - K^{xy}_{ss'}\sin(2\theta_{xs'})] \tag{B.13a}$$

$$S^{s',s'}_{ss'} = \frac{1}{\sin^2\theta_{ss'}}[K^{xx}_{ss'}\sin^2\theta_{xs} + K^{yy}_{ss'}\cos^2\theta_{xs} - K^{xy}_{ss'}\sin(2\theta_{xs})] \tag{B.13b}$$

$$\begin{aligned} S^{s,s'}_{ss'} = S^{s',s}_{ss'} = \frac{1}{\sin^2\theta_{ss'}}[&K^{xy}_{ss'}\sin(\theta_{xs}+\theta_{xs'}) \\ &- (K^{xx}_{ss'}\sin\theta_{xs}\sin\theta_{xs'} + K^{yy}_{ss'}\cos\theta_{xs}\cos\theta_{xs'})] \end{aligned} \tag{B.13c}$$

where $\mathbf{K} = \mathbf{D}^{-1}$. We have $S^{s',s}_{ss'} = S^{s,s'}_{ss'}$ because $D^{xy} = D^{yx}$, since $\mathbf{D}$ is physically required to be SPD [16, 111, 159, 163, 164]. From Equation (B.13) it is clear that $(\mathbf{S}_{ss'})^{-1}$ acts as an effective diffusivity tensor that converts the x-y diffusivity and angular dependence with respect to the x-y grid into the face-normal system. We note that setting the diffusivity to a scalar recovers Equation (B.8), while using a tensor diffusivity on an orthogonal grid yields $\mathbf{S}_{ss'} = \mathbf{K}_{ss'} = \mathbf{D}^{-1}_{ss'}$.

We now repeat Equation (B.2) where, with Equation (B.12) and Equation (B.13), we have specified all terms,

$$\begin{aligned} \int_V \vec{J}\cdot(\mathbf{D}^{-1}\vec{F})dV \approx &(\vec{J}_{LB}\cdot\mathbf{S}_{LB}\vec{F}_{LB})V_{LB} + (\vec{J}_{RB}\cdot\mathbf{S}_{RB}\vec{F}_{RB})V_{RB} \\ &+(\vec{J}_{LT}\cdot\mathbf{S}_{LT}\vec{F}_{LT})V_{LT} + (\vec{J}_{RT}\cdot\mathbf{S}_{RT}\vec{F}_{RT})V_{RT}. \end{aligned} \tag{B.14}$$

Via the shape matrices, we specify the diffusivity and the corner volumes as being unique to each corner, which is an important distinction for the S3 model. The assumption of a homogeneous phase would imply that the diffusivity is the same in each corner within a cell, and a typical rectangular discretization would give equal quarter weights to the four corner volumes. We use the same values in all four corners for diffusivity and volumetric weight in pure cells. However, we modify these values for mixed cells in the S3 method.

APPENDIX C

Supplemental Tables

This appendix contains tables of convergence rates that supplement the main text. These tables shows the convergence rates of E_1, E_2 and E_∞, whereas the main text primarily shows only E_∞. The tables in this appendix also show the convergence rates for each grid, while the main text often gives only the convergence rate averaged over all grids. Each table caption indicates the associated section or table. All convergence rates equal to or above 1.900 will be marked in bold.

$s = 2$		v1 (33%)			v2 (50%)			v3 (85%)		
Model	Grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
S1	10→20	**1.902**	1.357	0.878	**1.918**	1.372	0.939	**1.923**	1.505	0.931
	20→30	**1.951**	1.418	0.938	**1.961**	1.427	0.969	**1.960**	1.492	0.966
	30→40	**1.968**	1.441	0.958	**1.975**	1.448	0.979	**1.972**	1.490	0.977
	mean	**1.940**	1.405	0.924	**1.951**	1.416	0.962	**1.952**	1.496	0.958
S2	10→20	1.662	1.348	0.430	1.620	1.331	0.461	1.726	1.566	0.871
	20→30	1.863	1.367	0.721	1.840	1.357	0.707	1.849	1.518	0.801
	30→40	**1.936**	1.393	0.873	**1.921**	1.386	0.868	1.899	1.501	0.900
	mean	1.821	1.369	0.675	1.794	1.358	0.679	1.825	1.528	0.857
S3	10→20	**1.979**	1.465	0.999	**2.098**	**2.034**	**1.976**	**1.997**	**1.982**	**2.047**
	20→30	**1.992**	1.482	1.000	**2.058**	**2.023**	**1.987**	**1.997**	**1.981**	**2.029**
	30→40	**1.995**	1.488	1.000	**2.033**	**2.012**	**1.990**	**1.994**	**1.974**	**2.020**
	mean	**1.989**	1.479	1.000	**2.063**	**2.023**	**1.984**	**1.996**	**1.979**	**2.032**
M1	10→20	0.970	1.067	0.991	0.658	0.644	0.539	0.604	0.636	0.588
	20→30	0.958	0.996	0.771	0.771	0.772	0.723	0.779	0.793	0.749
	30→40	0.965	0.985	0.834	0.831	0.836	0.796	0.845	0.854	0.814
	mean	0.964	1.016	0.865	0.753	0.751	0.686	0.743	0.761	0.717
M2	10→20	**2.431**	**2.443**	**2.110**	**2.059**	**2.153**	**1.926**	1.736	1.763	1.674
	20→30	**2.433**	**2.514**	**2.173**	**2.062**	**2.158**	**1.954**	1.856	1.869	1.809
	30→40	**2.385**	**2.527**	**2.209**	**2.048**	**2.144**	**1.966**	1.891	**1.903**	1.860
	mean	**2.416**	**2.495**	**2.164**	**2.056**	**2.152**	**1.949**	1.827	1.845	1.781
M3	10→20	**2.431**	**2.443**	**2.110**	**2.059**	**2.153**	**1.926**	1.736	1.763	1.674
	20→30	**2.433**	**2.514**	**2.173**	**2.062**	**2.158**	**1.954**	1.856	1.869	1.809
	30→40	**2.385**	**2.527**	**2.209**	**2.048**	**2.144**	**1.966**	1.891	**1.903**	1.860
	mean	**2.416**	**2.495**	**2.164**	**2.056**	**2.152**	**1.949**	1.827	1.845	1.781

Table C.1: Full table of convergence rates of all six mixed-cell models for the 1D MMS mixed-cell test problem from Section 6.1. The convergence rate is measured in three p-norms for the indicated grid sizes. The interface is near the center of the domain and makes the volume fractions (v1, v2, and v3) indicated for the mixed cells. The diffusivity ratio is 10^2. Bold indicates $q_i \geq 1.90$.

$s = 2$		$y = x$			$y = 2x$			$y = \frac{1}{\sqrt{5}} - \frac{x}{\sqrt{2}}$		
Model	Grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
S1	10→20	**1.943**	1.825	1.344	**1.946**	**1.988**	**2.039**	**1.904**	**1.938**	1.583
	20→30	**1.976**	1.759	0.908	**1.964**	**1.967**	1.898	**1.943**	**1.916**	1.829
	30→40	**1.986**	1.720	0.998	**1.980**	**1.956**	1.147	**1.962**	1.862	-0.233
	mean	**1.969**	1.768	1.083	**1.963**	**1.970**	1.694	**1.937**	**1.905**	1.060
S2	10→20	1.882	1.803	1.370	**1.912**	**1.992**	**2.041**	1.870	**1.925**	**1.966**
	20→30	**1.961**	1.746	0.884	**1.958**	**1.967**	**2.014**	1.853	1.855	1.248
	30→40	**1.978**	1.710	0.982	**1.963**	**1.943**	**2.021**	1.889	1.809	0.996
	mean	**1.940**	1.753	1.079	**1.944**	**1.967**	**2.025**	1.871	1.863	1.404
S3	10→20	**1.968**	**2.006**	**2.083**	**1.945**	**1.956**	1.550	**1.973**	**1.930**	1.113
	20→30	**1.988**	**1.980**	**2.042**	1.887	1.873	1.014	**2.032**	**1.993**	1.438
	30→40	**1.994**	**1.968**	**2.016**	1.829	1.798	1.025	**1.979**	1.874	0.616
	mean	**1.983**	**1.985**	**2.047**	1.887	1.876	1.196	**1.995**	**1.932**	1.056
M1	10→20	1.242	1.106	0.756	0.698	0.794	1.249	1.640	1.605	1.010
	20→30	1.097	1.001	0.781	0.805	0.810	0.862	**1.942**	1.868	1.762
	30→40	1.047	0.987	0.831	0.851	0.858	0.851	1.218	1.114	0.109
	mean	1.129	1.031	0.790	0.785	0.821	0.987	1.600	1.529	0.960
M2	10→20	1.5170	1.436	1.015	0.703	0.769	0.832	1.770	1.719	1.447
	20→30	1.310	1.205	0.893	0.808	0.819	0.893	1.549	1.470	0.729
	30→40	1.221	1.119	0.863	0.857	0.866	0.857	1.456	1.395	1.332
	mean	1.349	1.253	0.924	0.790	0.818	0.861	1.591	1.528	1.170
M3	10→20	1.579	1.516	1.262	0.710	0.774	0.892	1.870	1.848	1.633
	20→30	1.370	1.270	0.917	0.808	0.823	0.895	1.653	1.639	1.295
	30→40	1.274	1.166	0.863	0.855	0.868	0.855	1.570	1.540	1.050
	mean	1.408	1.317	1.014	0.791	0.822	0.881	1.697	1.676	1.326

Table C.2: Full table of convergence rates of all six mixed-cell models for the 2D MMS mixed-cell test problem from Section 6.2. The convergence rate is measured in three p-norms. Results for three different linear interfaces are shown. The diffusivity ratio is 10^2. Bold indicates $q_i \geq 1.90$.

$s=-2$		v1 (33%)			v2 (50%)			v3 (85%)		
Model	Grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
S1	10→20	**2.041**	**1.921**	1.386	**1.971**	1.824	1.310	**1.960**	**2.053**	1.613
	20→30	**2.004**	1.811	1.214	**1.976**	1.729	1.179	**1.947**	**1.920**	1.479
	30→40	**2.063**	1.762	1.147	**2.032**	1.687	1.123	**2.022**	**1.908**	1.326
	40→50	**2.008**	1.692	1.100	**1.975**	1.638	1.084	**1.960**	1.820	1.224
	50→100	**2.015**	1.630	1.056	**2.010**	1.585	1.046	**1.988**	1.754	1.123
	mean	**2.026**	1.763	1.181	**1.993**	1.693	1.148	**1.975**	1.891	1.353
S2	10→20	**2.041**	**1.921**	1.386	**1.971**	1.824	1.310	**1.960**	**2.053**	1.613
	20→30	**2.004**	1.811	1.214	**1.976**	1.729	1.179	**1.947**	**1.920**	1.479
	30→40	**2.063**	1.762	1.147	**2.032**	1.687	1.123	**2.022**	**1.908**	1.326
	40→50	**2.008**	1.692	1.100	**1.975**	1.639	1.084	**1.960**	1.820	1.224
	50→100	**2.015**	1.630	1.056	**2.010**	1.585	1.046	**1.988**	1.754	1.123
	mean	**2.026**	1.763	1.181	**1.993**	1.693	1.148	**1.975**	1.891	1.353
S3	10→20	**2.074**	**2.344**	**2.280**	**2.070**	**2.389**	**2.264**	1.754	1.517	0.949
	20→30	**2.053**	**2.070**	**2.139**	**2.075**	**2.084**	**2.137**	1.855	1.533	0.999
	30→40	**2.055**	**2.123**	**2.157**	**2.074**	**2.103**	**2.156**	1.897	1.537	1.007
	40→50	**1.982**	**2.003**	**1.905**	1.612	1.660	1.720	**1.925**	1.537	1.010
	50→100	**1.999**	**2.017**	**2.008**	**2.140**	**2.127**	**2.067**	**1.956**	1.530	1.009
	mean	**2.033**	**2.111**	**2.098**	**1.994**	**2.073**	**2.069**	1.877	1.531	0.995
M1	10→20	0.546	0.298	-0.069	0.729	0.289	-0.241	1.090	0.712	0.138
	20→30	0.774	0.701	0.622	0.833	0.688	0.613	1.343	1.231	0.914
	30→40	0.846	0.815	0.796	0.864	0.805	0.798	1.237	1.178	0.952
	40→50	0.876	0.857	0.861	0.880	0.844	0.859	1.234	1.167	0.971
	50→100	0.921	0.914	0.929	0.918	0.906	0.928	1.157	1.116	0.983
	mean	0.792	0.717	0.628	0.845	0.706	0.591	1.212	1.081	0.792
M2	10→20	1.624	1.512	1.090	1.499	1.340	0.740	1.808	**1.902**	1.845
	20→30	**1.949**	**1.977**	**2.033**	1.836	1.883	**2.041**	1.878	1.869	**1.930**
	30→40	**1.987**	**2.044**	**2.097**	**1.961**	**1.998**	**1.924**	**1.962**	**1.977**	**1.958**
	40→50	**2.003**	**1.998**	**1.916**	**1.981**	**1.971**	**1.970**	**1.954**	**1.953**	**2.051**
	50→100	**2.006**	**2.009**	**2.012**	**1.991**	**1.996**	**1.998**	**1.981**	**1.985**	**1.974**
	mean	**1.914**	**1.908**	1.830	1.854	1.838	1.735	**1.917**	**1.937**	**1.952**
M3	10→20	1.624	1.512	1.090	1.499	1.340	0.740	1.808	**1.902**	1.845
	20→30	**1.949**	**1.977**	**2.033**	1.836	1.883	**2.041**	1.878	1.869	**1.930**
	30→40	**1.987**	**2.044**	**2.097**	**1.961**	**1.998**	**1.924**	**1.962**	**1.977**	**1.958**
	40→50	**2.003**	**1.998**	**1.916**	**1.981**	**1.971**	**1.970**	**1.954**	**1.953**	**2.051**
	50→100	**2.006**	**2.009**	**2.012**	**1.991**	**1.996**	**1.998**	**1.981**	**1.985**	**1.974**
	mean	**1.914**	**1.908**	1.8230	1.854	1.838	1.735	**1.917**	**1.937**	**1.952**

Table C.3: Full table of convergence rates for the 1D MES mixed-cell test problem in Section 7.3.2. Diffusivity ratio is 10^{-2}. Bold indicates $q_i \geq 1.90$.

$s=-6$		v1 (33%)			v2 (50%)			v3 (85%)		
Model	Grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
S1	10→20	**1.991**	**1.929**	1.361	**1.981**	1.874	1.359	**1.971**	**2.111**	1.784
	20→30	**2.082**	1.843	1.171	**2.042**	1.753	1.161	**2.026**	**1.994**	1.513
	30→40	**2.051**	1.737	1.087	**2.033**	1.672	1.089	**2.014**	**1.908**	1.295
	40→50	**2.035**	1.680	1.051	**2.024**	1.624	1.054	**2.011**	1.851	1.190
	50→100	**2.013**	1.613	1.021	**2.008**	1.572	1.023	**2.000**	1.762	1.086
	mean	**2.034**	1.760	1.138	**2.017**	1.699	1.137	**2.004**	**1.925**	1.374
S2	10→20	**1.991**	**1.929**	1.361	**1.981**	1.874	1.359	**1.971**	**2.111**	1.784
	20→30	**2.082**	1.843	1.171	**2.042**	1.753	1.161	**2.026**	**1.994**	1.513
	30→40	**2.051**	1.737	1.087	**2.033**	1.672	1.089	**2.014**	**1.908**	1.295
	40→50	**2.035**	1.680	1.051	**2.024**	1.624	1.054	**2.011**	1.851	1.190
	50→100	**2.013**	1.613	1.021	**2.008**	1.572	1.023	**2.000**	1.762	1.086
	mean	**2.034**	1.760	1.138	**2.017**	1.699	1.137	**2.004**	**1.925**	1.374
S3	10→20	**1.903**	**2.264**	**2.179**	**2.029**	**2.317**	**2.179**	1.790	1.558	0.973
	20→30	**2.107**	**2.153**	**2.156**	**2.095**	**2.144**	**2.155**	**1.904**	1.544	0.991
	30→40	**2.044**	**2.075**	1.897	**2.052**	**2.065**	1.897	**1.937**	1.539	0.996
	40→50	**2.060**	**2.047**	**2.047**	**2.036**	**2.039**	**2.047**	**1.958**	1.534	0.997
	50→100	**2.011**	**2.024**	**2.019**	**2.004**	**2.018**	**2.019**	**1.976**	1.525	0.999
	mean	**2.025**	**2.113**	**2.060**	**2.043**	**2.117**	**2.059**	**1.913**	1.540	0.991
M1	10→20	0.575	0.307	-0.059	0.796	0.314	-0.263	1.173	0.857	0.338
	20→30	0.792	0.720	0.664	0.831	0.702	0.652	1.325	1.252	0.980
	30→40	0.854	0.822	0.829	0.863	0.807	0.828	1.258	1.190	0.988
	40→50	0.887	0.870	0.896	0.887	0.858	0.896	1.214	1.154	0.993
	50→100	0.927	0.923	0.955	0.922	0.915	0.955	1.148	1.104	0.997
	mean	0.807	0.728	0.657	0.860	0.719	0.614	1.223	1.111	0.859
M2	10→20	1.651	1.553	1.199	1.507	1.371	0.699	1.849	**1.900**	1.787
	20→30	**2.009**	**2.030**	**1.958**	1.885	**1.936**	**1.998**	1.875	**1.902**	**2.131**
	30→40	**1.997**	**2.032**	**2.105**	**1.974**	**1.993**	**2.109**	**1.960**	**1.961**	1.837
	40→50	**2.023**	**2.025**	**2.039**	**2.007**	**2.004**	1.871	**1.986**	**1.980**	**1.979**
	50→100	**2.015**	**2.015**	**2.004**	**2.001**	**2.007**	**2.016**	**1.981**	**1.993**	**2.009**
	mean	**1.939**	**1.931**	1.861	1.875	1.862	1.739	**1.930**	**1.947**	**1.949**
M3	10→20	1.651	1.553	1.199	1.507	1.371	0.699	1.849	**1.900**	1.787
	20→30	**2.009**	**2.030**	**1.958**	1.885	**1.936**	**1.998**	1.875	**1.902**	**2.131**
	30→40	**1.997**	**2.032**	**2.105**	**1.974**	**1.993**	**2.109**	**1.960**	**1.961**	1.837
	40→50	**2.023**	**2.025**	**2.039**	**2.007**	**2.004**	1.871	**1.986**	**1.980**	**1.979**
	50→100	**2.015**	**2.015**	**2.004**	**2.001**	**2.007**	**2.016**	**1.981**	**1.993**	**2.009**
	mean	**1.939**	**1.931**	1.861	1.875	1.862	1.739	**1.930**	**1.947**	**1.949**

Table C.4: Full table of convergence rates for the 1D MES mixed-cell test problem in Section 7.3.2. Diffusivity ratio is 10^{-6}. Bold indicates $q_i \geq 1.90$.

$s = -10$		v1 (33%)			v2 (50%)			v3 (85%)		
Model	Grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
S1	10→20	1.878	1.865	1.387	1.863	1.832	1.375	1.805	**1.951**	1.775
	20→30	1.693	1.651	1.173	1.753	1.668	1.169	1.567	1.632	1.511
	30→40	1.505	1.506	1.095	1.500	1.516	1.092	1.361	1.431	1.314
	40→50	1.279	1.345	1.054	1.247	1.374	1.054	1.057	1.139	1.191
	mean	1.589	1.592	1.177	1.591	1.598	1.172	1.448	1.538	1.447
S2	10→20	**1.942**	1.897	1.361	**1.938**	1.855	1.358	**1.914**	**2.051**	1.779
	20→30	**1.910**	1.763	1.144	**1.920**	1.726	1.163	1.830	1.851	1.513
	30→40	1.698	1.597	1.063	1.809	1.614	1.067	1.740	1.737	1.319
	40→50	1.432	1.450	1.043	1.594	1.516	1.028	1.711	1.668	1.098
	mean	1.746	1.677	1.153	1.815	1.678	1.154	1.799	1.827	1.427
S3	10→20	1.753	**2.028**	**1.955**	1.882	**2.120**	**1.972**	1.796	1.566	0.973
	20→30	1.575	1.653	1.742	1.653	1.701	1.774	1.881	1.540	0.991
	30→40	1.396	1.447	1.439	1.235	1.323	1.335	1.879	1.5145	0.996
	40→50	0.800	0.904	0.987	0.683	0.847	0.970	1.837	1.472	0.997
	mean	1.381	1.508	1.531	1.363	1.498	1.513	1.848	1.523	0.989
M1	10→20	0.590	0.307	-0.064	0.799	0.309	-0.272	1.182	0.871	0.350
	20→30	0.799	0.724	0.666	0.823	0.703	0.654	1.319	1.232	0.955
	30→40	0.857	0.824	0.831	0.861	0.808	0.829	1.254	1.179	0.975
	40→50	0.888	0.871	0.895	0.885	0.858	0.897	1.211	1.147	0.985
	mean	0.783	0.681	0.582	0.842	0.670	0.527	1.241	1.107	0.816
M2	10→20	1.641	1.543	1.173	1.473	1.343	0.664	**1.969**	**2.047**	**1.952**
	20→30	**2.017**	**2.034**	**1.996**	1.893	**1.932**	**2.025**	**1.910**	**1.917**	**2.086**
	30→40	**1.999**	**2.033**	**2.118**	**1.976**	**1.990**	**2.061**	**1.990**	**1.985**	1.850
	40→50	**2.024**	**2.021**	**1.999**	**2.002**	**1.989**	1.879	**1.995**	**1.988**	**2.021**
	mean	**1.920**	**1.908**	1.821	1.836	1.814	1.657	**1.966**	**1.984**	**1.977**
M3	10→20	1.641	1.543	1.173	1.473	1.343	0.664	**1.969**	**2.047**	**1.952**
	20→30	**2.017**	**2.034**	**1.996**	1.893	**1.932**	**2.025**	**1.909**	**1.917**	**2.086**
	30→40	**1.999**	**2.033**	**2.118**	**1.976**	**1.990**	**2.061**	**1.990**	**1.985**	1.851
	40→50	**2.024**	**2.021**	**1.999**	**2.002**	**1.989**	1.879	**1.995**	**1.988**	**2.021**
	mean	**1.920**	**1.908**	1.821	1.836	1.814	1.657	**1.966**	**1.984**	**1.977**

Table C.5: Full table of convergence rates for the 1D MES mixed-cell test problem in Section 7.3.2. Diffusivity ratio is 10^{-10}. Bold indicates $q_i \geq 1.90$.

$s = 2$		Single T			Multi-T			Multi-T (phase 2)		
#	grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
1	20→40	**2.320**	**1.990**	1.193	1.569	1.597	0.837	1.473	1.462	0.832
	40→60	1.772	1.334	0.833	1.085	1.347	1.026	1.798	1.616	1.031
	60→80	1.807	1.678	1.228	1.420	1.842	0.846	1.289	0.943	0.839
	mean	**1.966**	1.667	1.085	1.358	1.595	0.903	1.520	1.341	0.900
2	20→40	**2.241**	**1.979**	1.207	1.530	1.531	0.880	1.473	1.462	0.896
	40→60	1.581	1.328	0.791	0.148	0.766	0.946	1.798	1.616	0.920
	60→80	1.779	1.678	1.249	1.675	1.844	0.893	1.289	0.943	0.893
	mean	1.867	1.661	1.082	1.118	1.380	0.906	1.520	1.341	0.903
3	20→40	**2.313**	**1.988**	1.225	1.582	1.550	0.882	1.473	1.462	0.909
	40→60	1.694	1.317	0.764	0.147	0.818	0.950	1.798	1.616	0.908
	60→80	1.762	1.673	1.226	1.732	1.885	0.892	1.289	0.943	0.892
	mean	**1.923**	1.659	1.072	1.154	1.418	0.908	1.520	1.341	0.903

Table C.6: Full table of convergence rates for the 2D MES mixed-cell test problem in Section 8.3.2.1 with the expanded computational domain. The diffusivity ratio is 10^2. Bold indicates $q_i \geq 1.90$.

$s = 6$		Single T			Multi-T			Multi-T (phase 2)		
#	grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
1	20→40	**2.255**	**1.984**	1.192	1.626	1.599	0.835	1.733	1.771	0.829
	40→60	1.681	1.328	0.830	1.100	1.349	1.029	1.787	1.603	1.031
	60→80	1.631	1.666	1.222	1.416	1.835	0.838	1.251	0.909	0.831
	mean	1.856	1.659	1.081	1.381	1.594	0.900	1.590	1.428	0.897
2	20→40	1.432	1.821	1.318	1.549	1.536	0.860	1.733	1.771	0.875
	40→60	0.649	0.882	0.828	0.478	0.979	1.004	1.787	1.603	0.981
	60→80	1.084	1.233	1.250	1.767	**1.981**	0.870	1.251	0.909	0.871
	mean	1.055	1.312	1.132	1.265	1.499	0.911	1.590	1.428	0.909
3	20→40	**2.252**	**1.980**	1.228	1.606	1.555	0.863	1.733	1.771	0.888
	40→60	1.560	1.305	0.752	0.496	1.023	1.007	1.787	1.603	0.968
	60→80	1.824	1.677	1.220	1.875	**2.030**	0.874	1.251	0.909	0.874
	mean	1.878	1.654	1.067	1.326	1.536	0.915	1.590	1.428	0.910

Table C.7: Full table of convergence rates for the 2D MES mixed-cell test problem in Section 8.3.2.1 with the expanded computational domain. The diffusivity ratio is 10^6. Bold indicates $q_i \geq 1.90$.

$s = 6$		Single T			Multi-T			Multi-T (phase 1)		
#	grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
1	20→40	1.569	1.207	0.522	0.421	0.296	0.090	0.095	0.109	0.090
	40→60	**2.038**	1.608	0.932	0.257	0.173	0.062	0.087	0.081	0.062
	60→80	1.509	1.432	1.089	0.022	-0.008	-0.017	-0.019	-0.011	-0.017
	mean	1.705	1.416	0.848	0.233	0.153	0.045	0.055	0.060	0.045
2	20→40	0.883	0.933	1.005	1.442	1.339	0.585	1.081	0.999	0.622
	40→60	0.961	1.026	0.930	1.409	1.312	0.767	1.262	1.236	0.701
	60→80	1.315	1.302	1.111	0.343	0.535	1.215	0.481	0.824	1.214
	mean	1.053	1.087	1.015	1.065	1.062	0.855	0.941	1.020	0.846
3	20→40	1.491	1.226	1.009	1.481	1.359	0.587	1.120	1.027	0.631
	40→60	1.643	1.623	0.930	1.512	1.365	0.763	1.360	1.289	0.685
	60→80	1.766	1.429	1.111	0.198	0.510	1.215	0.409	0.836	1.215
	mean	1.633	1.426	1.016	1.064	1.078	0.855	0.963	1.051	0.844

Table C.8: Full table of convergence rates for the 2D MES problem with a quadratic initial condition using a cell-centered solution corresponding to Table 8.5. The diffusivity ratio is 10^6. Bold indicates $q_i \geq 1.90$.

$s = 6$		Single T			Multi-T			Multi-T (phase 1)		
#	grid	q_1	q_2	q_∞	q_1	q_2	q_∞	q_1	q_2	q_∞
1	20→40	**2.088**	1.674	0.938	1.258	1.101	0.661	0.918	0.672	0.676
	40→60	1.255	1.038	0.996	-0.119	0.106	0.475	0.694	0.709	0.480
	60→80	**2.197**	**2.190**	1.161	**1.910**	1.459	0.395	0.932	0.613	0.395
	mean	1.847	1.634	1.032	1.016	0.889	0.510	0.848	0.665	0.517
2	20→40	0.947	0.972	0.876	1.341	1.385	1.013	0.775	0.766	1.033
	40→60	0.906	0.914	0.963	1.307	1.332	1.696	0.940	1.091	1.667
	60→80	1.344	1.338	1.205	0.037	-0.070	-1.445	0.019	-0.084	-1.392
	mean	1.066	1.075	1.015	0.895	0.883	0.422	0.578	0.591	0.436
3	20→40	**2.183**	1.607	0.877	1.377	1.427	1.020	0.852	0.819	1.039
	40→60	1.243	1.269	0.962	1.417	1.448	1.813	1.061	1.230	1.786
	60→80	**2.383**	**2.012**	1.205	-0.197	-0.310	-1.726	-0.186	-0.299	-1.676
	mean	**1.936**	1.629	1.015	0.866	0.855	0.369	0.576	0.583	0.383

Table C.9: Full table of convergence rates for the 2D MES problem with a quadratic initial condition using a cell-averaged solution corresponding to Table 8.6. The diffusivity ratio is 10^6. Bold indicates $q_i \geq 1.90$.

REFERENCES

REFERENCES

[1] S. N. Acharya. Multigrid Conjugate Gradient Method. Master's thesis, Fraunhofer Institut (Friedrich-Alexander- Universitat Erlangen-Nurnberg), 2006.

[2] H. T. Ahn and M. Shashkov. Multi-Material Interface Reconstruction on Generalized Polyhedral Meshes. *Journal of Computational Physics*, 226(2):2096 – 2132, 2007.

[3] H. T. Ahn and M. Shashkov. Adaptive Moment-of-Fluid Method. *Journal of Computational Physics*, 228(8):2792 – 2821, 2009.

[4] R. E. Alcouffe, A. Brandt, J. J. E. Dendy, and J. W. Painter. The Multi-Grid Method for the Diffusion Equation with Strongly Discontinuous Coefficients. *SIAM Journal on Scientific and Statistical Computing*, 2(4):430–454, December 1981.

[5] G. Allaire and G. Bal. Homogenization of the Criticality Spectral Equation in Neutron Transport. *ESAIM: Mathematical Modelling and Numerical Analysis*, 33(04):721–746, 1999.

[6] H. Anbarlooei and K. Mazaheri. Moment of Fluid Interface Reconstruction Method in Multi-Material Arbitrary Lagrangian Eulerian (MMALE) Algorithms. *Computer Methods in Applied Mechanics and Engineering*, 198(47):3782–3794, 2009.

[7] T. Arbogast, M. F. Wheeler, and I. Yotov. Mixed Finite Elements for Elliptic Problems with Tensor Coefficients as Cell-centered Finite Differences. *SIAM Journal on Numerical Analysis*, 34(2):828–852, April 1997.

[8] G. Arfken, H. J. Weber, and F. E. Harris. *Mathematical Methods for Physicists*. Academic Press Inc, seventh edition, 2012. ISBN 978-0-12-384654-9.

[9] M. Arioli. A Stopping Criterion for the Conjugate Gradient Algorithm in a Finite Element Method Framework. *Numerische Mathematik*, 97(1):1–24, 2004.

[10] M. Arioli, I. Duff, and D. Ruiz. Stopping Criteria for Iterative Solvers. *SIAM Journal on Matrix Analysis and Applications*, 13(1):138–144, 1992.

[11] M. Arioli and S. Gratton. Linear Regression Models, Least-Squares Problems, Normal Equations, and Stopping Criteria for the Conjugate Gradient Method. *Computer Physics Communications*, 2012.

[12] D. N. Arnold. Mixed Finite Element Methods for Elliptic Problems. *Computer Methods in Applied Mechanics and Engineering*, 82(1):281–300, 1990.

[13] S. R. Arridge and J. C. Hebden. Optical Imaging in Medicine: II. Modelling and Reconstruction. *Physics in Medicine and Biology*, 42(5):841, 1997.

[14] K. Aziz and A. Settari. *Petroleum Reservoir Simulation*, volume 476. Applied Science Publishers London, 1979.

[15] G. Bal and L. Ryzhik. Diffusion Approximation of Radiative Transfer Problems with Interfaces. *SIAM Journal on Applied Mathematics*, 60(6):1887–1912, 2000.

[16] R. W. Balluffi, S. M. Allen, and W. C. Carter. *Kinetics of Materials.* Wiley-Interscience Publications, 2005.

[17] D. Balsara. Fast and Accurate Discrete Ordinates Methods for Multidimensional Radiative Transfer. Part I: Basic Methods. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 69(6):671 – 707, 2001.

[18] D. J. Benson. Volume of Fluid Interface Reconstruction Methods for Multi-Material Problems. *Applied Mechanics Reviews*, 55(2):151–165, 2002.

[19] A. Bensoussan, J.-L. Lions, and G. C. Papanicolaou. Boundary Layers and Homogenizatlon of Transport Processes. *Research Institute for Mathematical Sciences*, 15(1):53–157, 1979.

[20] M. J. Berger and R. J. LeVeque. Stable Boundary Conditions for Cartesian Grid Calculations. *Computing Systems in Engineering*, 1(2):305–311, 1990.

[21] M. J. Berger and R. J. LeVeque. A Rotated Difference Scheme for Cartesian Grids in Complex Geometries. *AIAA Paper CP-91-1602*, pages 1–9, 1991.

[22] M. Berndt, K. Lipnikov, J. D. Moulton, and M. J. Shashkov. Convergence of Mimetic Finite Difference Discretizations of the Diffusion Equation. *East–West Journal of Numerical Mathematics*, 9:253–316, 2001.

[23] M. Berndt, K. Lipnikov, M. J. Shashkov, M. F. Wheeler, and I. Yotov. A Mortar Mimetic Finite Difference Method on Non-Matching Grids. *Numerishe Mathematik*, 102(2):203–230, September 2005.

[24] M. Berndt, K. Lipnikov, M. J. Shashkov, M. F. Wheeler, and I. Yotov. Superconvergence of the Velocity in Mimetic Finite Difference Methods on Quadrilaterals. *SIAM Journal on Numerical Analysis*, 43(4):1728–1749, February 2005.

[25] C. M. Bethke. A Numerical Model of Compaction-Driven Groundwater Flow and Heat Transfer and its Application to the Paleohydrology of Intracratonic Sedimentary Basins. *Journal of Geophysical Research: Solid Earth*, 90(B8):6817–6828, 1985.

[26] P. B. Bochev and J. M. Hyman. Principles of Mimetic Discretizations of Differential Operators. In D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, and M. Shashkov, editors, *Compatible Spatial Discretizations*, volume 142 of *Mathematics and Statistics*, pages 89–119. Springer, New York, 2006. ISBN 978-0-387-38034-5-5.

[27] J. Bolz, I. Farmer, E. Grinspun, and P. Schröoder. Sparse Matrix Solvers on the GPU: Conjugate Gradients and Multigrid. In *ACM SIGGRAPH 2003 Papers*, pages 917–924. ACM, 2003.

[28] T. B. Boving and P. Grathwohl. Tracer Diffusion Coefficients in Sedimentary Rocks: Correlation to Porosity and Hydraulic Conductivity. *Journal of Contaminant Hydrology*, 53(1–2):85 – 100, 2001.

[29] R. L. Bowers and J. R. Wilson. *Numerical Modeling in Applied Physics and Astrophysics*. Jones & Bartlett, Boston, 1991.

[30] D. Braess, P. Deuflhard, and K. Lipnikov. A Subspace Cascadic Multigrid Method for Mortar Elements. *Computing*, 69(3):205–225, 2002.

[31] A. Brandt. Multi-Level Adaptive Technique for Fast Numerical Solution to Boundary Value Problems. In H. Cabannes and R. Temam, editors, *Proceedings of the 3rd International Conference on Numerical Methods in Fluid Dynamics*, volume 18 of *Lecture Notes in Physics*, pages 82–89, Springer Verlag, New York, 1972.

[32] A. Brandt. Multi-Level Adaptive Solutions to Boundary-Value Problems. *Mathematics of Computation*, 31(138):333–390, 1977.

[33] F. Brezzi. On the Existence, Uniqueness and Approximation of Saddle-Point Problems Arising from Lagrangian Multipliers. *ESAIM: Mathematical Modelling and Numerical Analysis-Modélisation Mathématique et Analyse Numérique*, 8(R2):129–151, 1974.

[34] F. Brezzi, A. Fuffa, and K. Lipnikov. Mimetic Finite Differences for Elliptic Problems. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(2):277–295, 2008.

[35] F. Brezzi, K. Lipnikov, M. Shashkov, and N. Bellomo. Convergence of Mimetic Finite Difference Method for Diffusion Problems on Polyhedral Meshes with Curved Faces. *Mathematical Models and Methods in Applied Sciences*, 16(2):275–298, 2006.

[36] F. Brezzi, K. Lipnikov, M. J. Shashkov, and V. Simoncini. A New Discretization Methodology for Diffusion Problems on Generalized Polyhedral Meshes. *Computer Methods in Applied Mechanics and Engineering*, 196(37-40):3682 – 3692, 2007.

[37] F. Brezzi, K. Lipnikov, and V. Simoncini. A Family of Mimetic Finite Difference Methods on Polygonal and Polyhedral Meshes. *Mathematical Models and Methods in Applied Sciences*, 15(10):1533–1552, 2005.

[38] P. Brown and B. Chang. On the Use of Diffusion Synthetic Acceleration in Parallel 3D Neutral Particle Transport Calculations. Technical Report UCRL-JC-130877, Lawrence Livermore National Laboratory, 1998.

[39] P. N. Brown, B. Chang, and U. R. Hanebutte. Spherical Harmonic Solutions to the 3D Kobayashi Benchmark Suite. In *PHYSOR 2000*. ANS International Topial Meetings on the Advances in Reactor Physics and Mathematics and Computation into the Next Millennium, American Nuclear Society, May 2000.

[40] P. N. Brown, A. C. Hindmarsh, and L. R. Petzold. Using Krylov Methods in the Solution of Large-Scale Differential-Algebraic Systems. *SIAM Journal on Scientific Computing*, 15(6):1467–1488, 1994.

[41] P. N. Brown and C. S. Woodward. Preconditioning Strategies for Fully Implicit Radiation Diffusion with Material-Energy Transfer. *SIAM Journal on Scientific Computing*, 23(2):499–516, 2001.

[42] T. Brunner. Forms of Approximate Radiation Transport. Technical Report SAND2002-1778, Sandia National Laboratories, PO Box 5800, Albuquerque, New Mexico 87185, July 2002.

[43] Z. Cai, J. Jones, S. McCormick, and T. Russell. Control-Volume Mixed Finite Element Methods. *Computational Geosciences*, 1(3):289–315, September 1997.

[44] J. C. Campbell, J. M. Hyman, and M. J. Shashkov. Mimetic Finite Difference Operators for Second-Order Tensors on Unstructured Grids. *Computers & Mathematics with Applications*, 44(1-2):157 – 173, 2002.

[45] J. C. Campbell and M. J. Shashkov. A Tensor Artificial Viscosity Using a Mimetic Finite Difference Algorithm. *Journal of Computational Physics*, 172(2):739 – 765, April 2001.

[46] K. M. Case and P. F. Zweifel. *Linear Transport Theory*. Plenum Press, New York, 1967.

[47] J. E. Castillo, J. M. Hyman, M. Shashkov, and S. Steinberg. Fourth- and Sixth-Order Conservative Finite Difference Approximations of the Divergence and Gradient. *Applied Numerical Mathematics*, 37(1-2):171 – 187, 2001.

[48] J. E. Castillo, J. M. Hyman, M. J. Shashkov, and S. Steinberg. High-order mimetic finite difference methods on nonuniform grids. In *International Conference on Spectral and High Order Methods*, Third. Houston Journal of Mathematics, 1996.

[49] J. Castor. *Radiation Hydrodynamics*. Cambridge University Press, 32 Avenue of the Americas, New York, NY 10013-2473, 1st edition, 2004. ISBN 0-521-54062-3.

[50] B. Cheng, J. Glimm, and D. H. Sharp. A Multi-Temperature Multiphase Flow Model. *Zeitschrift für Angewandte Mathematik und Physik (ZAMP)*, 53:211–238, 2002. 10.1007/s00033-002-8153-8.

[51] C. Chueh, M. Secanell, W. Bangerth, and N. Djilali. Multi-Level Adaptive Simulation of Transient Two-Phase Flow in Heterogeneous Porous Media. *Computers & Fluids*, 39(9):1585–1596, 2010.

[52] M.-H. Chung. An Adaptive Cartesian Cut-Cell/Level-Set Method to Simulate Incompressible Two-Phase Flows with Embedded Moving Solid Boundaries. *Computers & Fluids*, 71, 2013.

[53] B. C. Craft, M. F. Hawkins, and R. E. Terry. *Applied Petroleum Reservoir Engineering*. Old Tappan, NJ (USA); Prentice Hall Inc., second edition, 1991.

[54] J. Crank and P. Nicolson. A Practical Method for Numerical Evaluation of Solutions of Partial Differential Equations of the Heat-Conduction Type. In *Mathematical Proceedings of the Cambridge Philosophical Society*, volume 43, pages 50–67. Cambridge Univ Press, 1947.

[55] J. C. Crittenden, N. J. Hutzler, D. G. Geyer, J. L. Oravitz, and G. Friedman. Transport of Organic Compounds With Saturated Groundwater Flow: Model Development and Parameter Sensitivity. *Water Resources Research*, 22(3):271–284, 1986.

[56] P. I. Crumpton, G. J. Shaw, and A. F. Ware. Discretisation and Multigrid Solution of Elliptic Equations with Mixed Derivative Terms and Strongly Discontinuous Coefficients. *Journal of Computational Physics*, 116(2):343 – 358, 1995.

[57] W. Dai and P. R. Woodward. Numerical Simulations for Nonlinear Heat Transfer in a System of Multimaterials. *Journal of Computational Physics*, 139:158–178, 1998.

[58] M. D'amico. A Newton-Raphson Approach for Nonlinear Diffusion Equations in Radiation Hydrodynamics. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 54(4):655–669, 1995.

[59] B. Das, S. Steinberg, S. Weber, and S. Schaffer. Finite Difference Methods for Modeling Porous Media Flows. *Transport in Porous Media*, 17(2):171–200, August 1994.

[60] R. Dautray and J. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology: Volume 3 - Spectral Theory and Applications*. Springer-Verlag, Berlin, 1993.

[61] F. De Monte. Transient Heat Conduction in One-Dimensional Composite Slab: A 'Natural' Analytic Approach. *International Journal of Heat and Mass Transfer*, 43(19):3607–3619, 2000.

[62] F. De Monte. An Analytic Approach to the Unsteady Heat Conduction Processes in One-Dimensional Composite Media. *International Journal of Heat and Mass Transfer*, 45(6):1333–1343, 2002.

[63] J. D. Densmore and E. W. Larsen. Asymptotic Equilibrium Diffusion Analysis of Time-Dependent Monte Carlo Methods for Grey Radiative Transfer. *Journal of Computational Physics*, 199(1):175 – 204, 2004.

[64] A. D'Hueppe. *Heat Transfer Modeling at an Interface Between a Porous Medium and a Free Region*. PhD thesis, Ecole Centrale Paris, 2011.

[65] R. P. Drake. *High-Energy-Density Physics*. Springer Publishing Company, 11 West 42nd Street, 15th Floor New York, NY 10036, 1st edition, 2006. ISBN 3-540-29314-0.

[66] D. A. Drew. Mathematical Modeling of Two-Phase Flow. *Annual Review of Fluid Mechanics*, 15:261–291, 1983.

[67] J. J. Duderstadt and L. J. Hamilton. *Nuclear Reactor Analysis*. John Wiley and Sons, Inc., New York, 1976.

[68] V. Dyadechko and M. Shashkov. Moment-of-Fluid Interface Reconstruction. Technical Report LA-UR-05-7571, Los Alamos National Laboratory, 2006.

[69] V. Dyadechko and M. Shashkov. Multi-Material Interface Reconstruction From the Moment Data. Technical Report LA-UR-06-5846, Los Alamos National Laboratory, 2006.

[70] V. Dyadechko and M. Shashkov. Reconstruction of Multi-Material Interfaces From Moment Data. *Journal of Computational Physics*, 227(11):5361 – 5384, 2008.

[71] M. G. Edwards. Symmetric, Flux Continuous, Positive-Definite Approximation of the Elliptic Full-Tensor Pressure Equation in Local Conservation Form. In *Proceedings of the 13th SPE Reservoir Simulation Symposium*, pages 553–562. Society of Petroleum Engineers, February 1995.

[72] M. G. Edwards. A Higher-Order Godunov Scheme Coupled with Dynamic Local Grid Refinement for Flow in a Porous Medium. *Computer Methods in Applied Mechanics and Engineering*, 131(3):287–308, 1996.

[73] M. G. Edwards. Elimination of Adaptive Grid Interface Errors in the Discrete Cell Centered Pressure Equation. *Journal of Computational Physics*, 126(2):356–372, 1996.

[74] M. G. Edwards. Cross-Flow, Tensors and Finite Volume Approximation with Deferred Correction. *Computer Methods in Applied Mechanics and Engineeringe*, 151(1):143–161, 1998.

[75] M. G. Edwards and C. F. Rogers. Finite Volume Discretization with Imposed Flux Continuity for the General Tensor Pressure Equation. *Computational Geosciences*, 2(4):259–290, 1998.

[76] A. Einstein. Investigations on the Theory of Brownian Movement. *Annalen der Physik*, 17(8):549–560, 1905.

[77] H. I. Ene and D. Poliwevski. *Thermal Flow in Porous Media*. Kluwer Academic Publishers, Norwell, MA, 1987.

[78] R. Ewing, Y. Yuan, and G. Li. Timestepping Along Characteristics for a Mixed Finite-Element Approximation for Compressible Flow of Contamination from Nuclear Waste in Porous Media. *SIAM Journal on Numerical Analysis*, 26(6):1513–1524, 1989.

[79] T. J. Farrell, M. S. Patterson, and B. Wilson. A Diffusion Theory Model of Spatially Resolved, Steady-State Diffuse Reflectance for the Noninvasive Determination of Tissue Optical Properties in Vivo. *Medical physics*, 19:879, 1992.

[80] R. Fedorenko. A Relaxation Method for Solving Elliptic Difference Equations. *USSR Computational Mathematics and Mathematical Physics*, 1:1092–1096, 1962.

[81] R. Fedorenko. The Speed of Convergence of an Iterative Process. *USSR Computational Mathematics and Mathematical Physics*, 4:227–235, 1964.

[82] A. Fick. On Liquid Diffusion. *Philosophical Magazine Series 4*, 10(63):30–39, July 1855. Reprinted in Journal of Membrane Sciences 100(1):33-38: 1995.

[83] J. J. Fleck and J. Cummings. An Implicit Monte Carlo Scheme for Calculating Time and Frequency Dependent Nonlinear Radiation Transport. *Journal of Computational Physics*, 8(3):313–342, 1971.

[84] S. T. Flock, M. S. Patterson, B. C. Wilson, and D. R. Wyman. Monte Carlo Modeling of Light Propagation in Highly Scattering Tissues: I. Model Predictions and Comparison with Diffusion Theory. *Biomedical Engineering, IEEE Transactions on*, 36(12):1162–1168, 1989.

[85] D. R. Fokkema, G. L. Sleijpen, and H. A. Van der Vorst. Accelerated Inexact Newton Schemes for Large Systems of Nonlinear Equations. *SIAM Journal on Scientific Computing*, 19(2):657–674, 1998.

[86] R. V. Garimella and K. Lipnikov. Solutions of the Diffusion Equation in Multi-Material Domains by Subdivision of Elements Along Reconstructed Interfaces. *(preprint)*, 2010.

[87] N. Gentile. Implicit Monte Carlo Diffusion: An Acceleration Method for Monte Carlo Time-Dependent Radiative Transfer Simulations. *Journal of Computational Physics*, 172(2):543–571, 2001.

[88] W. Gray. A Derivation of the Equations for Multi-Phase Transport. *Chemical Engineering Science*, 30(2):229–233, 1975.

[89] W. Gray and P. Lee. On the Theorems For Local Volume Averaging of Multiphase Systems. *International Journal of Multiphase Flow*, 3(4):333 – 340, 1977.

[90] J. Guevara-Jordan, S. Rojas, M. Freites-Villegas, and J. Castillo. A New Second Order Finite Difference Conservative Scheme. *Divulgaciones Matemáticas*, 13(1):107–122, 2005.

[91] V. Gvozdev, Y. Kuznetsov, and M. Shashkov. Mimetic Finite-Difference Method for Diffusion Equations on Polyhedral Meshes with Mixed Cells. Technical Report LA-UR-06-4596, Los Alamos National Laboratory, 2006.

[92] M. R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *Journal of Research of the National Bureau of Standards*, 49(6), December 1952.

[93] C. Hirsch. *Numerical Computation of Internal & External Flows: Fundamentals of Computational Fluid Dynamics.* Elsevier, Linacre House, Jordan Hill, Oxford OX2 8DP, 2nd edition, 2007. ISBN 978-0-7506-6594-0.

[94] C. W. Hirt and B. D. Nichols. Volume of Fluid (VoF) Method for the Dynamics of Free Boundaries. *Journal of Computational Physics*, 39:201–225, Jan. 1981.

[95] J. M. Hyman, J. E. Morel, M. J. Shashkov, and S. Steinberg. Mimetic Finite Difference Methods for Diffusion Equations. *Computational Geosciences*, 6(3-4):333–325, September 2002.

[96] J. M. Hyman and M. J. Shashkov. Natural Discretizations for the Divergence, Gradient, and Curl on Logically Rectangular Grids. *Computers & Mathematics with Applications*, 33(4):81 – 104, 1997.

[97] J. M. Hyman and M. J. Shashkov. Approximation of Boundary Conditions for Mimetic Finite-Difference Methods. *Computers & Mathematics with Applications*, 36(5):79 – 99, 1998.

[98] J. M. Hyman and M. J. Shashkov. The Orthogonal Decomposition Theorems for Mimetic Finite Difference Methods. *SIAM Journal on Numerical Analysis*, 36(3):788–818, 1999.

[99] J. M. Hyman, M. J. Shashkov, and S. Steinberg. The Numerical Solution of Diffusion Problems in Strongly Heterogeneous Non-Isotropic Materials. *Journal of Computational Physics*, 132(1):130–148, September 1997.

[100] J. M. Hyman and S. Steinberg. The Convergence of Mimetic Discretization for Rough Grids. *Computers & Mathematics with Applications*, 47(10-11):1565 – 1610, 2004.

[101] J. D. Jackson. *Classical Electrodynamics*. John Wiley & Sons, third edition, 1999.

[102] Z. Jin and K. Stamnes. Radiative Transfer in Nonuniformly Refracting Layered Media: Atmosphere-Ocean System. *Appl. Opt.*, 33(3):431–442, 1994.

[103] S. R. Johnson. *Anisotropic Diffusion Approximations for Time-Dependent Particle Transport*. PhD thesis, University of Michigan, 2012.

[104] J. Jung, H. Chijiwa, K. Kobayashi, and H. Nishihara. Discrete Ordinate Neutron Transport Equation Equivalent to $\mathrm{P_L}$ Approximation. *Nuclear Science and Engineering*, 49:1, 1972.

[105] S. Y. Kadioglu, R. R. Nourgaliev, and V. A. Mousseau. Comparative Study of the Harmonic and Arithmetic Averaging of Diffusion Coefficients for Non-linear Heat Conduction Problems. Technical Report INL/EXT-08-13999, Idaho National Laboratory, Idaho Falls, Idaho 83415, March 2008.

[106] M. Keijzer, S. L. Jacques, S. A. Prahl, and A. J. Welch. Light Distributions in Artery Tissue: Monte Carlo Simulations for Finite-Diameter Laser Beams. *Lasers in Surgery and Medicine*, 9(2):148–154, 1989.

[107] M. Khandelwal and M. Mench. Direct Measurement of Through-Plane Thermal Conductivity and Contact Resistance in Fuel Cell Materials. *Journal of Power Sources*, 161(2):1106–1115, 2006.

[108] A. D. Kim and A. Ishimaru. Optical Diffusion of Continuous-Wave, Pulsed, and Density Waves in Scattering Media and Comparisons with Radiative Transfer. *Applied Optics*, 37(22):5313–5319, 1998.

[109] T.-K. Kim and H. S. Lee. Radiative Transfer in Two-Dimensional Anisotropic Scattering Media with Collimated Incidence. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 42(3):225 – 238, 1989.

[110] M. J. King. Application and Analysis of Tensor Permeability to Crossbedded Reservoirs. In *Proceedings of the SPE Western Regional Meeting*. Society of Petroleum Engineers, May 1993.

[111] A. Kleidon, R. Lorenz, and R. D. Lorenz. *Non-Equilibrium Thermodynamics and the Production of Entropy: Life, Earth, and Beyond*. Springer, 2005.

[112] D. Knoll and W. Rider. A Multigrid Preconditioned Newton–Krylov Method. *SIAM Journal on Scientific Computing*, 21(2):691–710, 1999.

[113] D. Knoll, W. Rider, and G. Olson. Nonlinear Convergence, Accuracy, and Time Step Control in Non-Equilibrium Radiation Diffusion. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 70(1):25 – 36, 2001.

[114] D. Knoll, W. J. Rider, and G. Olson. An Efficient Nonlinear Solution Method For Non-Equilibrium Radiation Diffusion. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 63:15–29, Sept. 1999.

[115] O. Kolditz, R. Ratke, H.-J. G. Diersch, and W. Zielke. Coupled Groundwater Flow and Transport: 1. Verification of Variable Density Flow and Transport Models. *Advances in Water Resources*, 21(1):27 – 46, 1998.

[116] D. B. Kothe and W. J. Rider. A Comparison of Interface Tracking Methods. In *12th AIAA CFD Conference*, number LA-UR–95-1145; CONF-950634–1 in 12, pages 19–22. LANL, USDOE, Mar. 1995.

[117] P. K. Kundu, I. M. Cohen, and D. R. Dowling. *Fluid Mechanics*. Elsevier Amsterdam, fifth edition, 2011. ISBN 978-0-12-3821003.

[118] Y. A. Kuznetsov. Mixed Finite Element Method for Diffusion Equations on Polygonal Meshes with Mixed Cells. *Journal of Numerical Mathematics*, 14(4):305–315, 2006.

[119] Y. A. Kuznetsov, K. Lipnikov, and M. J. Shashkov. The Mimetic Finite Difference Method on Polygonal Meshes for Diffusion-Type Problems. *Computational Geosciences*, 8(4):301–324, 2004.

[120] E. W. Larsen. Neutron Transport and Diffusion in Inhomogeneous Media. I. *Journal of Mathematical Physics*, 16:1421–1427, 1975.

[121] E. W. Larsen. Neutron Transport and Diffusion in Inhomogeneous Media. II. *Nuclear Science and Engineering*, 60:357–368, 1976.

[122] E. W. Larsen and J. B. Keller. Asymptotic Solution of Neutron Transport Problems for Small Mean Free Paths. *Journal of Mathematical Physics*, 15:75–81, 1974.

[123] R. J. LeVeque and Z. Li. The Immersed Interface Method for Elliptic Equations with Discontinuous Coefficients and Singular Sources. *SIAM Journal on Numerical Analysis*, 31(4):1019–1044, 1994.

[124] R. J. LeVeque and Z. Li. Immersed Interface Methods for Stokes Flow with Elastic Boundaries or Surface Tension. *SIAM Journal on Scientific Computing*, 18(3):709–735, 1997.

[125] R. J. LeVeque, D. Mihalas, E. Dorfi, and E. Müller. *Computational Methods for Astrophysical Fluid Flow*. Springer, 1998. ISBN 3-540-64448-2.

[126] C. Levermore. Chapman-Enskog Approach to Flux-Limited Diffusion Theory. Technical Report UCID-18229, California University, Lawrence Livermore Lab, Livermore, CA 94551, 1979.

[127] C. Levermore and G. Pomraning. A Flux-Limited Diffusion Theory. *The Astrophysical Journal*, 248:321–334, 1981.

[128] E. Lewis and W. Miller. *Computational Methods of Neutron Transport*. American Nuclear Society, La Grange Park, Illinois, 1993. ISBN 0-89448-452-4.

[129] Z. Li. A Fast Iterative Algorithm for Elliptic Interface Problems. *SIAM Journal on Numerical Analysis*, 35(1):230–254, 1998.

[130] Z.-Y. Li. Magnetohydrodynamic Disk-Wind Connection: Magnetocentrifugal Winds from Ambipolar Diffusion-Dominated Accretion Disks. *The Astrophysical Journal*, 465:855, 1996.

[131] B. T. Liou and C. Y. Wu. Radiative Transfer in a Multi-Layer Medium with Fresnel Interfaces. *Heat and Mass Transfer*, 32(1):103–107, 11 1996.

[132] K. Lipnikov, J. E. Morel, and M. Shashkov. Mimetic Finite Difference Methods for Diffusion Equations on Non-Orthogonal Non-Conformal Meshes. *Journal of Computational Physics*, 199(2):589 – 597, 2004.

[133] K. Lipnikov, J. D. Moulton, and D. Svyatskiy. A Multilevel Multiscale Mimetic (M3) Method for Two-Phase Flows in Porous Media. *Journal of Computational Physics*, 227(14):6727 – 6753, 2008.

[134] K. Lipnikov and M. J. Shashkov. A Mimetic Tensor Artificial Viscosity for Lagrangian Hydrocodes on Arbitrary Polygonal Meshes. Unlimited Release LA-UR-10-00006, Los Alamos National Laboratory, MS B284, Los Alamos, NM 87545, January 2010.

[135] K. Lipnikov, M. J. Shashkov, and D. Svyatskiy. The Mimetic Finite Difference Discretization of Diffusion Problem on Unstructured Polyhedral Meshes. *Journal of Computational Physics*, 211(2):473 – 491, 2006.

[136] K. Lipnikov, M. J. Shashkov, and I. Yotov. Local Flux Mimetic Finite Difference Methods. *Numerische Mathematik*, 112(1):115–152, March 2009.

[137] R. Liska, M. J. Shashkov, and A. V. Solovjov. Support-Operators Method for PDE Discretization: Symbolic Algorithms and Realization. *Mathematics and Computers in Simulation*, 35(2):173 – 183, 1993.

[138] C. Liu and W. P. Ball. Analytical Modeling of Diffusion-Limited Contamination and Decontamination in a Two-Layer Porous Medium. *Advances in Water Resources*, 21(4):297 – 313, 1998.

[139] R. MacKinnon and G. Carey. Analysis of Material Interface Discontinuities and Superconvergent Fluxes in Finite Difference Theory. *Journal of Computational Physics*, 75(1):151–167, 1988.

[140] Maple. *Version 14.00*. Maplesoft, Waterloo ON, Canada, 2010.

[141] A. Mayo. The Fast Solution of Poisson's and the Biharmonic Equations on Irregular Regions. *SIAM Journal on Numerical Analysis*, 21(2):285–299, 1984.

[142] A. Mayo. The Rapid Evaluation of Volume Integrals of Potential Theory on General Regions. *Journal of Computational Physics*, 100(2):236–245, 1992.

[143] A. Mayo and A. Greenbaum. Fast Parallel Iterative Solution of Poisson's and the Biharmonic Equations on Irregular Regions. *SIAM Journal on Scientific and Statistical Computing*, 13(1):101–118, 1992.

[144] R. G. McClarren. *Spherical Harmonics Methods for Thermal Radiation Transport*. PhD thesis, University of Michigan, 2007.

[145] R. G. McClarren, J. P. Holloway, and T. A. Brunner. On Solutions to the P_N Equations for Thermal Radiative Transfer. *Journal of Computational Physics*, 227(5):2864–2885, 2008.

[146] D. Mihalas and B. Weibel-Mihalas. *Foundations of Radiation Hydrodynamics*. Dover Publications, Inc., 31 East 2nd Street, Mineola, NY 11501, 1st edition, 1984. ISBN 0-486-40925-2.

[147] M. Mikhailov, M. Özişik, and N. Vulchanov. Diffusion in Composite Layers with Automatic Solution of the Eigenvalue Problem. *International Journal of Heat and Mass Transfer*, 26(8):1131–1141, 1983.

[148] W. Miller Jr and W. H. Reed. Ray-Effect Mitigation Methods for Two-Dimensional Neutron Transport Theory. *Nuclear Science and Engineering*, 62(3):391, 1977.

[149] O. Montilla, C. Cadenas, and J. Castillo. Matrix Approach to Mimetic Discretizations for Differential Operators on Non-Uniform Grids. *Mathematics and Computers in Simulation*, 73(1-4):215–225, August 2006.

[150] J. E. Morel, J. E. Dendy, M. L. Hall, and S. W. White. A Cell-Centered Lagrangian-Mesh Diffusion Differencing Scheme. *Journal of Computational Physics*, 103(2):286 – 299, 1992.

[151] J. E. Morel, M. L. Hall, and M. J. Shashkov. A Local Support-Operators Diffusion Discretization Scheme for Hexahedral Meshes. *Journal of Computational Physics*, 170(1):338 – 372, 2001.

[152] J. E. Morel, R. M. Roberts, and M. J. Shashkov. A Local Support-Operators Diffusion Discretization Scheme for Quadrilateral r-z Meshes. *Journal of Computational Physics*, 144(1):17–51, January 1998.

[153] V. A. Mousseau and D. A. Knoll. Temporal Accuracy of the Nonequilibrium Radiation Diffusion Equations Applied to Two-Dimensional Multimaterial Simulations. *Nuclear Science and Engineering*, 154(2):174–189, 2006.

[154] G. Müller, editor. *Medical Optical Tomography: Functional Imaging and Optical Technologies*. SPIE Series vol IS11, SPIE Press, Bellingham, WA, 1993.

[155] W.-C. Müller and D. Carati. Dynamic Gradient-Diffusion Subgrid Models for Incompressible Magnetohydrodynamic Turbulence. *Physics of Plasmas*, 9:824, 2002.

[156] I. Neretnieks. Diffusion in the Rock Matrix: An Important Factor in Radionuclide Retardation? *Journal of Geophysical Research: Solid Earth*, 85(B8):4379–4397, 1980.

[157] J. Ni and C. Beckermann. A Volume-Averaged Two-Phase Model for Transport Phenomena During Solidification. *Metallurgical and Materials Transactions B*, 22(3):349–361, 1991.

[158] I. Nozad, R. Carbonell, and S. Whitaker. Heat Conduction in Multiphase Systems–I: Theory and Experiment for Two-Phase Systems. *Chemical Engineering Science*, 40(5):843 – 855, 1985.

[159] J. F. Nye. *Physical Properties of Crystals: Their Representation by Tensors and Matrices*. Oxford University Press, 1985.

[160] G. L. Olson. Efficient Solution of Multi-Dimensional Flux-Limited Nonequilibrium Radiation Diffusion Coupled to Material Conduction with Second-Order Time Discretization. *Journal of Computational Physics*, 226(1):1181–1195, 2007.

[161] G. L. Olson, L. H. Auer, and M. L. Hall. Diffusion, P1, and Other Approximate Forms of Radiation Transport. *Journal of Quantitative Spectroscopy and Radiative Transfer*, 64:619–634, Mar. 2000.

[162] K. O'Neill and G. Pinder. A Derivation of the Equations for Transport of Liquid and Heat in Three Dimensions in a Fractured Porous Medium. *Advances in Water Resources*, 4(4):150–164, 1981.

[163] L. Onsager. Reciprocal Relations in Irreversible Processes: I. *Physical Review*, 37(4):405, 1931.

[164] L. Onsager. Reciprocal Relations in Irreversible Processes: II. *Physical Review*, 38(12):2265, 1931.

[165] S. J. Osher and J. A. Sethian. Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations. *Journal of Computational Physics*, 79(1):12 – 49, 1988.

[166] S. V. Patankar. *Numerical Heat Transfer and Fluid Flow.* Hemisphere Pub, 1980.

[167] D. Peaceman. *Fundamentals of Numerical Reservoir Simulation.* Elsevier Amsterdam, 1977.

[168] M. H. Pedras and M. J. de Lemos. On the Definition of Turbulent Kinetic Energy for Flow in Porous Media. *International Communications in Heat and Mass Transfer*, 27(2):211–220, 2000.

[169] O. Pedrosa Jr and K. Aziz. Use of a Hybrid Grid in Reservoir Simulation. *SPE Reservoir Engineering*, 1(6):611–621, 1986.

[170] R. B. Pember, J. B. Bell, P. Colella, W. Y. Crutchfield, and M. L. Welcome. An Adaptive Cartesian Grid Method for Unsteady Compressible Flow in Irregular Regions. *Journal of Computational Physics*, 120(2):278–304, 1995.

[171] C. S. Peskin. Numerical Analysis of Blood Flow in the Heart. *Journal of Computational Physics*, 25(3):220–252, 1977.

[172] C. S. Peskin. Lectures on Mathematical Aspects of Physiology. *Lectures in Applied Math*, 19:69 – 107, 1981.

[173] C. S. Peskin and D. M. McQueen. Modeling Prosthetic Heart Valves for Numerical Analysis of Blood Flow in the Heart. *Journal of Computational Physics*, 37(1):113–132, 1980.

[174] C. Pflaum. A Multigrid Conjugate Gradient Method. *Applied Numerical Mathematics*, 58(12):1803–1817, 2008.

[175] G. Pomraning. *The Equations Of Radiation Hydrodynamics.* Dover Publications, Inc., 31 East 2nd Street, Mineola, NY 11501, 1st edition, 1973. ISBN 0-486-44599-2.

[176] G. C. Pomraning and G. M. Foglesong. Transport-Diffusion Interfaces in Radiative Transfer. *Journal of Computational Physics*, 32(3):420 – 436, 1979.

[177] J. D. Ramshaw and C. H. Chang. Multicomponent Diffusion in Two-Temperature Magnetohydrodynamics. *Phys. Rev. E*, 53:6382–6388, Jun 1996.

[178] R. Raviart and J. Thomas. A Mixed Finite Element Method for Second-Order Elliptic Problems. *Lecture Notes of Mathematics*, 606:292–315, 1977.

[179] W. H. Reed. Spherical Harmonic Solutions of the Neutron Transport Equation from Discrete Ordinate Codes. *Nuclear Science and Engineering*, 49:10, 1972.

[180] J. Renn. Einstein's Invention of Brownian Motion. *Annalen der Physik*, 14(S1):23–37, 2005.

[181] W. Rider and D. Knoll. Solving Nonlinear Heat Conduction Problems with Multigrid Preconditioned Newton-Krylov Methods. In *IMACS (Iterative Methods in Scientific Computation) International Symposium*, number LA-UR-97-2929 in 3, page 8, Laramie, WY (United States), September 1997. Los Alamos National Laboratory.

[182] W. Rider, D. Knoll, and G. Olson. A Multigrid Newton-Krylov Method for Flux-Limited Radiation Diffusion. In *Copper Mountain Conference On Iterative Methods*, number LA-UR-98-1075 in 5, page 5, Copper Mountain, CO, September 1998. Los Alamos National Laboratory.

[183] W. Rider and D. Kothe. Reconstructing Volume Tracking. *Journal of Computational Physics*, 141(2):112–152, 1998.

[184] W. J. Rider, D. Knoll, and G. Olson. A Multgrid Newton-Krylov Method for Multimaterial Equilibrium Radiation Diffusion. *J. Comput. Phys.*, 152(1):164–191, 1999.

[185] S. Runnels, T. Gianakon, M. Kenamond, M. Berry, K. Lipnikov, and M. Shashkov. Implementation of the Local Support Operator Method for the Diffusion Equation on General Polyhedral Grids. Technical Report LA-UR-04-8510, Los Alamos National Laboratory, 2004.

[186] S. Runnels, K. Lipnikov, and M. Shashkov. Mathematic and Algorithmic Details for Applying the Local Support Operator Method on Polyhedral Grids. Technical Report LA-UR 10-07498, Los Alamos National Laboratory, July 2006.

[187] J. Rutqvist, Y. Wu, C. Tsang, and G. Bodvarsson. A Modeling Approach for Analysis of Coupled Multiphase Fluid Flow, Heat Transfer, and Deformation in Fractured Porous Rock. *International Journal of Rock Mechanics and Mining Sciences*, 39(4):429–442, 2002.

[188] Y. Saad and M. H. Schultz. GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems. *SIAM Journal on Scientific and Statistical Computing*, 7(3):856–869, 1986.

[189] K. Salari and P. Knupp. Code Verification by the Method of Manufactured Solutions. Technical Report SAND2000-1444, Sandia National Labs., Albuquerque, NM (US); Sandia National Labs., Livermore, CA (US), 2000.

[190] A. Samarskii, V. Tishkin, A. Favorskii, and M. J. Shashkov. Operational Finite-Difference Schemes. *Differential Equations*, 17(7):854–862, 1981.

[191] A. Samarskii, V. Tishkin, A. Favorskii, and M. J. Shashkov. Employment of the Reference Operator Method in the Construction of Finite Difference Analogs of Tensor Operations. *Differential Equations*, 18(7):881–885, 1983.

[192] C. Santos, J. Quaresma, and A. Garcia. Determination of Transient Interfacial Heat Transfer Coefficients in Chill Mold Castings. *Journal of Alloys and Compounds*, 319:174 – 186, 2001.

[193] R. Sentis. Study of the Corrector of the Eigenvalue of a Transport Operator. *SIAM Journal on Mathematical Analysis*, 16(1):151–166, 1985.

[194] A. Settari and K. Aziz. Use of Irregular Grid in Reservoir Simulation. *Old SPE Journal*, 12(2):103–114, 1972.

[195] C. Shackelford and D. Daniel. Diffusion in Saturated Soil. I: Background. *Journal of Geotechnical Engineering*, 117(3):467–484, 1991.

[196] M. J. Shashkov. *Conservative Finite-Difference Methods on General Grids. Ed. by Stanly Steinberg.* CRC Press, 1996.

[197] M. J. Shashkov and S. Steinberg. Support-Operator Finite-Difference Algorithms for General Elliptic Problems. *Journal of Computational Physics*, 118(1):131 – 151, 1995.

[198] M. J. Shashkov and S. Steinberg. Solving Diffusion Equations with Rough Coefficients in Rough Grids. *Journal of Computational Physics*, 129(2):383–405, August 1996.

[199] G. Shubin and J. Bell. An Analysis of the Grid Orientation Effect in Numerical Simulation of Miscible Displacement. *Computer Methods in Applied Mechanics and Engineering*, 47(1):47–71, 1984.

[200] K. Stamnes, S. Tsay, W. Wiscombe, and K. Jayaweera. Numerically Stable Algorithm for Discrete-Ordinate-Method Radiative Transfer in Multiple Scattering and Emitting Layered Media. *Applied Optics*, 27:2502–2509, June 1988.

[201] D. P. Starinshak. *Level Set Methods for Multimaterial Radiative Shock Hydrodynamics.* PhD thesis, University of Michigan, 2012.

[202] J. M. Stone, D. Mihalas, and M. L. Norman. ZEUS-2D: A Radiation Magnetohydrodynamics Code for Astrophysical Flows in Two Space Dimensions. III. The Radiation Hydrodynamic Algorithms and Tests. *The Astrophysical Journal Supplement Series*, 80:819–845, 1992.

[203] G. Strang. Conjugate Gradient Method (Lecture 19). In *Mathmatical Methods for Engineers II (18.086).* MIT OpenCourseWare, 2008.

[204] M. Sussman, P. Smereka, and S. Osher. A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow. *Journal of Computational Physics*, 114(1):146 – 159, 1994.

[205] F. D. Swesty and E. S. Myra. A Numerical Algorithm for Modeling Multigroup Neutrino-Radiation Hydrodynamics in Two Spatial Dimensions. *The Astrophysical Journal Supplement Series*, 181(1):1, 2009.

[206] O. Tatebe. The Multigrid Preconditioned Conjugate Gradient Method, 1993.

[207] L. N. Trefethen and D. B. III. *Numerical Linear Algebra.* Society for Industrial Mathematics, 1997.

[208] V. V. Tuchin and V. Tuchin. *Tissue Optics: Light Scattering Methods and Instruments for Medical Diagnosis*, volume 13. SPIE press Bellingham, 2007.

[209] B. Van der Holst, G. Toth, I. V. Sokolov, K. G. Powell, J. P. Holloway, E. S. Myra, Q. Stout, M. L. Adams, J. E. Morel, S. Karni, et al. CRASH: Block-Adaptive-Mesh Code for Radiative Shock Hydrodynamics - Implementation and Verification. *The Astrophysical Journal Supplement Series*, 194(2):23, 2011.

[210] G. Wang and E. Matthys. Numerical Modelling of Phase Change and Heat Transfer During Rapid Solidification Processes: Use of Control Volume Integrals with Element Subdivision. *International Journal of Heat and Mass Transfer*, 35(1):141–153, 1992.

[211] T. Washio and C. Oosterlee. Error Analysis for a Potential Problem on Locally Refined Grids. *Numerische Mathematik*, 86(3):539–563, July 2000.

[212] B. Wendroff and A. B. White Jr. Some Supraconvergent Schemes for Hyperbolic Equations on Irregular Grids. In *Proceedings of the 2nd International Conference on Nonlinear Hyperbolic Equations*, pages 671–677. Springer, 1989.

[213] M. F. Wheeler and I. Yotov. Mixed Finite Element Methods for Modeling Flow and Transport in Porous Media. Technical Report CRPC-TR95555-S, Rice University, July 1995.

[214] A. Wiegmann and K. P. Bube. The Immersed Interface Method for Nonlinear Differential Equations with Discontinuous Coefficients and Singular Sources. *SIAM Journal on Numerical Analysis*, 35(1):177–200, 1998.

[215] A. Wiegmann and K. P. Bube. The Explicit-Jump Immersed Interface Method: Finite Difference Methods for PDEs with Piecewise Smooth Solutions. *SIAM Journal on Numerical Analysis*, 37(3):827–862, 2000.

[216] A. R. Winters and M. J. Shashkov. Support Operators Method for the Diffusion Equation in Multiple Materials. Technical Report LA-UR-12-24117, Los Alamos National Laboratory, 2012.

[217] A. B. Wollaber. *Advanced Monte Carlo Methods for Thermal Radiation Transport.* PhD thesis, University of Michigan, 2008.

[218] T. Ye, W. Shyy, and J. N. Chung. A Fixed-Grid, Sharp-Interface Method for Bubble Dynamics and Phase Change. *Journal of Computational Physics*, 174(2):781–815, 2001.

[219] H. D. Young and R. A. Freedman. *University Physics with Modern Physics*. Addison-Wesley, San Francisco, CA, 13th edition, 2012. ISBN 978-0-321-69686-1.

[220] D. Youngs. Time-Dependent Multi-Material Flow with Large Fluid Distortion. *Numerical Methods for Fluid Dynamics*, 24:273–285, 1982.

[221] E. Zauderer. *Partial Differential Equations of Applied Mathematics*. Wiley-Interscience Publications, second edition, 1989.

[222] O. C. Zienkiewicz and R. L. Taylor. *Finite Element Method: Volume 1 The Basis*, volume 1. Butterworth-Heinemann, fifth edition, 2000. ISBN 0-7506-5049-4.