DISSERTATION

A FOURTH-ORDER SOLUTION-ADAPTIVE FINITE-VOLUME ALGORITHM FOR COMPRESSIBLE REACTING FLOWS ON MAPPED DOMAINS

Submitted by

Landon Owen

Department of Mechanical Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2019

Doctoral Committee:

Advisor: Xinfeng Gao

Stephen Guzik
Anthony Marchese
Donald Estep

ABSTRACT

A FOURTH-ORDER SOLUTION-ADAPTIVE FINITE-VOLUME ALGORITHM FOR COMPRESSIBLE REACTING FLOWS ON MAPPED DOMAINS

Accurate computational modeling of reacting flows is necessary to improve the design combustion efficiency and emission reduction in combustion devices, such as gas turbine engines. Combusting flows consists of a variety of phenomena including fluid mixing, chemical kinetics, turbulence-chemistry interacting dynamics, and heat and mass transfer. The scales associated with these range from atomic scales up to continuum scales at device level. Therefore, combusting flows are strongly nonlinear and require multiphysics and multiscale modeling. This research employs a fourth-order finite-volume method and leverages increasing gains in modern computing power to achieve high-fidelity modeling of flow characteristics and combustion dynamics. However, it is challenging to ensure that computational models are accurate, stable, and efficient due to the multiscale and multiphysics nature of combusting flows. Therefore, the goal of this research is to create a robust, high-order finite-volume algorithm on mapped domains with adaptive mesh refinement to solve compressible combustion problems in relatively complex geometries on parallel computing architecture.

There are five main efforts in this research. The first effort is to extend the existing algorithm to solve the compressible Navier-Stokes equations on mapped domains by implementing the fourth-order accurate viscous discretization operators. The second effort is to incorporate the species transport equations and chemical kinetics into the solver to enable combustion modeling. The third effort is to ensure stability of the algorithm for combustion simulations over a wide range of speeds. The fourth effort is to ensure all new functionality utilizes the parallel adaptive mesh refinement infrastructure to achieve efficient computations on high-performance computers. The final goal is to utilize the algorithm to simulate a range of flow problems, including a multispecies flow with

Mach reflection, multispecies mixing flow through a planar burner, and oblique detonation waves over a wedge.

This research produces a verified and validated, fourth-order finite-volume algorithm for solving thermally perfect, compressible, chemically reacting flows on mapped domains that are adaptively refined and represent moderately complex geometries. In the future, the framework established in this research will be extended to model reactive flows in gas turbine combustors.

ACKNOWLEDGEMENTS

I would like to thank my advisor, Dr. Gao, for her consistent guidance and reassurance. Your zeal for research has been a constant source of inspiration. I have grown as a person and a researcher more than I thought possible under your advisement. I would also like to thank Dr. Guzik for his counsel and direction. They both have been invaluable during this process.

I would like to thank my dissertation committee, Dr. Estep and Dr. Marchese, for their feedback and recommendations.

I would like to thank my parents for their unending love and support. I would also like to thank my brothers and sister-in-laws for their encouragement and understanding along the way.

I would like to thank my colleagues Jordan, Josh, Justin, Nate O., Nate S., Nelson, Noah, Scott, Sean, Tyler, and Yijun for the advice, stimulating conversation, and, most importantly, friendship. They have made my time here an incredible experience.

I am grateful to Dr. Gao, Dr. Guzik, and Colorado State University for the financial support for this research.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SYMBOLS

Alphanumeric

A_r	Pre-exponential factor in the rate constant for the rth reaction, depends on reaction
c_n	Mass fraction of the nth species
$c_{p,n}$	Specific heat capacity at constant pressure of the nth species, $\mathrm{J}/(\mathrm{kg}\cdot\mathrm{K})$
D	Number of dimensions
D_n	Mass diffusion coefficient of the nth species, m^2/s
$E_{a,r}$	Activation energy for the rth reaction, $\mathrm{cal}/\mathrm{mole}$
$\boldsymbol{e}^d$	Unit normal vector in the dth direction
$\vec{\mathbf{F}}$	Convective flux dyad, i.e., $[\mathbf{F}_x\,,\mathbf{F}_y\,,\mathbf{F}_z]$
$\vec{\boldsymbol{\mathcal{G}}}$	Mapped diffusion flux dyad, i.e., $[\boldsymbol{\mathcal{G}}_\xi\,,\boldsymbol{\mathcal{G}}_\eta\,,\boldsymbol{\mathcal{G}}_\zeta]$
G_n	Molar Gibbs free energy of the nth species, J/mole
H_n	Molar specific enthalpy of the nth species, J/mole
h_n	Total specific enthalpy of the nth species, J/kg
$k_{\mathrm{f},r}$	Forward reaction rate for the rth reaction, depends on reaction
$k_{\mathrm{b},r}$	Backward reaction rate for the rth reaction, depends on reaction
$K_{\mathrm{eq},r}$	Equilibrium constant for the rth reaction, depends on reaction
M_n	Molecular weight of the nth species, $\mathrm{kg}/\mathrm{mole}$
N_r	Total number of reactions
N_s	Total number of species
p	Static pressure, Pa
p_{atm}	Atmospheric pressure, Pa
R_n	Universal gas constant of the nth species, $\mathrm{J}/(\mathrm{kg}\cdot\mathrm{K})$
R_u	Molar universal gas constant, $8.314511\ \mathrm{J}/(\mathrm{mole}\cdot\mathrm{K})$
R_c	Molar universal gas constant (in E_a units), $\mathrm{cal}/(\mathrm{mole}\cdot\mathrm{K})$

$\mathbf{S}$	Source term variable vector
S_n	Molar specific entropy of the nth species, $\mathrm{J}/(\mathrm{mole} \cdot \mathrm{K})$
T	Temperature, K
t	Time, s
$\mathbf{U}$	Conservative variable vector
u_d	Velocity component in the dth direction, $\mathrm{m/s}$
$\mathbf{W}$	Native primitive variable vector
$\widetilde{\mathbf{W}}$	Nonnative primitive variable vector
x_d	Position in physical space in the dth direction
$[X_n]$	Molar concentration of the nth species, $\mathrm{moles/m^3}$

Dimensionless

Le	Lewis number
M	Mach number
Pr	Prandtl number
Re	Reynolds number
Sc	Schmidt number

Greek

$\alpha_{n,r}$	Enhanced third-body efficiency of the nth species for the rth reaction
β_r	Temperature exponent in the rate constant for the rth reaction
γ	Specific heat ratio
κ	Thermal conductivity, $\mathrm{W}/(\mathrm{m} \cdot \mathrm{K})$
μ	Dynamic viscosity, $\mathrm{kg}/(\mathrm{m} \cdot \mathrm{s})$
$\nu'_{n,r}$	Reactant stoichiometric coefficient of the nth species for the rth reaction
$\nu''_{n,r}$	Product stoichiometric coefficient of the nth species for the rth reaction
ξ_d	Position in computational space in the dth direction
ρ	Fluid density, $\mathrm{kg/m^3}$
χ_n	Mole fraction of the nth species

$\dot{\omega}_n$	Chemical production rate of the nth species, s^{-1}

Modifiers

$\lfloor \bullet \rceil$	Round to the nearest integer
$\mathcal{O}\left(\Delta x^m\right)$	Represents an error of the mth order of magnitude
$\vec{\bullet}$	Denotes a vector
$\vec{\nabla}_\xi$	Gradient or divergence operators in computational space
$\vec{\nabla}_x$	Gradient or divergence operators in physical space
1, 2, 3-D	One, two, and three dimensions

Chapter 1

Introduction

1.1 Motivations and Objectives

Combustion devices are ubiquitous to almost every aspect of modern life. The environmental and economic benefits associated with improvements to the design of combustion devices are enormous, whether it is reducing harmful emissions from a jet engine or improving the efficiency of turbo-machinery in a power plant. Due to the complex physical phenomena that occur in such devices, accurately predicting complex reacting physics is essential to improve the design of combustion devices. Experimental modeling is one popular option to achieve this goal. Unfortunately, experimental modeling poses many challenges, especially when modeling flows in extreme operating conditions. To effectively cope with these challenges, Computational Fluid Dynamics (CFD) has become an important tool for predicting nonlinear physical processes in reacting flows and improving fundamental understanding of the flow physics. However, numerical modeling of complex reacting physics poses challenges of its own. Simulations of reacting flows are expensive due to the long integration time [1–3]. Additionally, turbulent combustion involves a wide range of complex physical and chemical behaviors, and care must be taken to ensure the models remain representative of the physical phenomena. In response to these challenges, this work details the development, verification, and validation of a fourth-order finite-volume algorithm for solving compressible combustion problems in relatively complex geometries using mapped domains.

The five main objectives of this research are:

1. implement the fourth-order accurate viscous discretization operators on mapped domains for solving the compressible Navier-Stokes equations;

2. incorporate the chemical kinetics and species transport equations to model thermally perfect, chemically reacting flows;

3. devise novel stability techniques to ensure stability of the algorithm for solving high-speed or strongly discontinuous flows;

4. integrate all new capabilities into the parallel Adaptive Mesh Refinement (AMR) infrastructure to achieve efficient computations on modern supercomputing hardware;

5. apply the algorithm to simulate a multispecies Mach reflection, mixing flow in a planar burner, and oblique detonation waves (ODW) over a wedge.

Any CFD algorithm must be rigorously verified and validated before being applied to simulate realistic reacting flows. AIAA defines verification as [4] "the process of determining that a model implementation accurately represents the developer's conceptual description of the model and the solution to the model." The current algorithm is verified using the Couette flow, Gaussian acoustic pulse, multispecies mass diffusion bubble, and shear problems. Each test case serves a different purpose: the Couette flow verifies the viscous diffusion operators, the Gaussian acoustic pulse verifies the convection operators, the species mass diffusion bubble verifies the operators associated with the multispecies transport processes, and the shear problem verifies the algorithm can accommodate strong gradients present at AMR interfaces. Additionally, freestream preservation on mapped domains and conservation of species concentrations in multispecies, mixing flow are verified.

Validating an algorithm is essential before it can be used to reliably simulate physical phenomena. AIAA defines validation as [4] "the process of determining the degree to which a model is an accurate representation of the real world from the perspective of the intended uses of the model." The algorithm without chemical reaction modeling is first validated using the shock tube, shock box, and lid-driven cavity flow test cases. The shock tube and shock box test cases validate the algorithm's ability to cope with strong discontinuities. The lid-driven cavity flow test case validates the solution of a wall-bounded, thermally perfect flow with both convective and diffusive physics with formations of large vortices and recirculation zones. The combustion algorithm is validated using the reacting hydrogen-oxygen front, shock-driven combustion of a hydrogen bub-

ble, and reacting Richtmyer-Meshkov Instability (RMI) problems. All three test cases involve convective and diffusive physics and chemical reactions. Additionally, the shock bubble and RMI problems involve shock waves and shock induced combustion phenomena; these types of problems can pose severe stability issues. Therefore, these problems are good candidates for validating the novel stability techniques proposed in the present work. Once verified and validated, the algorithm is applied to simulate a multispecies Mach reflection case, mixing flow in a planar burner, and ODWs.

The present work focuses on enabling Chord, the in-house code for the CFD & Propulsion Laboratory at Colorado State University, to model compressible, combusting flows. Chord is a finite-volume algorithm that is fourth-order accurate in both space and time and features AMR and generalized curvilinear coordinate transformation. Specifically, this dissertation is dedicated to the implementation, verification, and validation of the operators for modeling diffusion, species transport, and chemical reactions. With this functionality, Chord can solve thermally perfect, compressible, chemically reacting flows on mapped domains with AMR. Chord will be used to study the nonlinear physical processes of reactive flows in gas turbine combustors and, eventually, to improve the design of combustion devices. The remainder of this chapter is a brief review of the important elements of this research.

1.2 Parallel Fourth-Order Finite-Volume Methods

This research employs the fourth-order finite-volume method (FVM) for solving the fully-coupled, compressible Navier-Stokes equations on structured grids. FVMs that use Cartesian grids for the computational domain, as is the case for this research, are computationally efficient and have well-understood characteristics in terms of solution accuracy. Traditionally, FVMs have been constrained to second-order accuracy, where the flux integrals are approximated using the midpoint rule, because high-order methods are more complicated and lack the significant research investment of second-order methods [5]. However, increasing the order of accuracy of an algorithm from second-order to fourth-order reduces the error as the grid is uniformly refined by twice the expo-

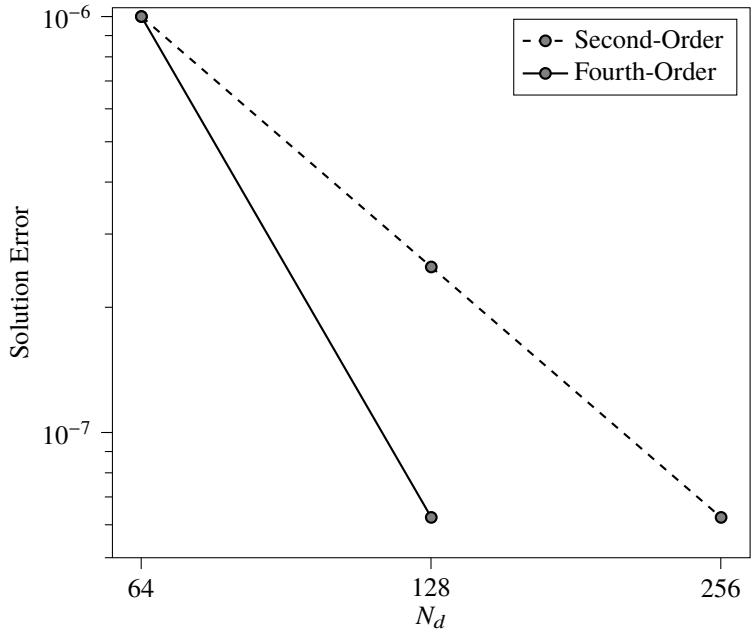


Figure 1.1: Illustration of reduction in solution error for second and fourth-order solutions.

nential rate for smooth flows. This can be illustrated using a simple example, shown in Figure 1.1. In this example, assume the norm of the solution error on a 3-D grid of 64^3 (or 2.6×10^5) cells is 1×10^{-6}. To reduce this error by a factor of 16 to an error of 6.25×10^{-8} using a fourth-order algorithm, the grid must be refined to 128^3 (or 2.1×10^6) cells. However, to achieve the same solution error with a second-order algorithm, the grid must be refined to 256^3 (or 16.8×10^6) cells. Moreover, increasing the order of accuracy improves the accuracy per unit memory and makes better use of modern computer architectures.

1.3 Curvilinear Coordinate Transformation

To model flows in realistic, complex geometries, the present work employs generalized curvilinear coordinate transformations to map a non-Cartesian grid in physical space to a Cartesian grid in computational space. This approach recovers Cartesian methods with some additional complexity associated with grid metrics. Nevertheless, efficient meshing can still be achieved for many geometries. Grid mapping has gained favor in the aerospace community, as wings are often easily meshed by this approach. Figure 1.2 provides an example of a curvilinear coordinate transfor-

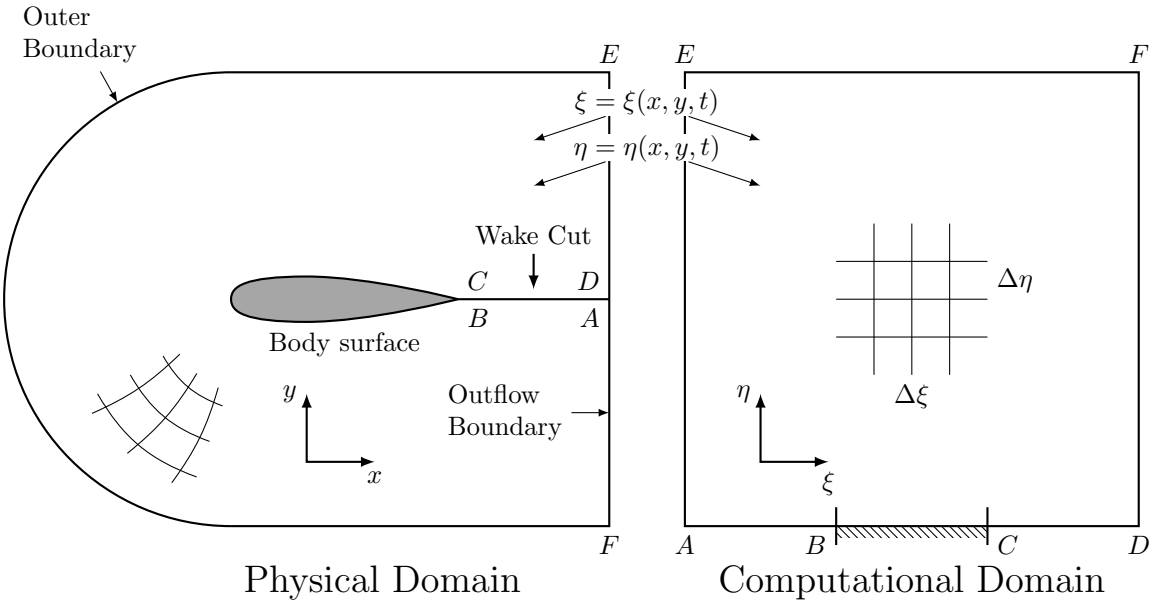


Figure 1.2: An example of a generalized curvilinear coordinate transformation for a C-mesh over an airfoil. Image is adapted from Pulliam and Zingg [6].

mation for a mesh over an airfoil. It is important that freestream preservation is retained when using mapped coordinates. According to Colella et al. [7], freestream preservation "ensures that a uniform flow is unaffected by the choice of mapping and discretization." In the present work, freestream preservation is verified and detailed.

1.4 Adaptive Mesh Refinement

AMR allows for mesh resolution changes to occur in response to the characteristics of the flow. Regions with large errors are adaptively refined to avoid incurring computational costs associated with increased resolution in lower error regions [8]. Care must be taken to ensure the AMR is freestream preserving [9].

Different AMR strategies exist depending on the data structures and partitioning, such as, patch-based, cell-based, and block-based. Interested readers should refer to Gao [10] for a detailed review and comparison of these strategies. This research uses the patch-based AMR method [11, 12]. The patch-based method starts with a coarse, base-level Cartesian grid. During the solution process, individual cells are tagged for refinement based on criteria related to flow physics or

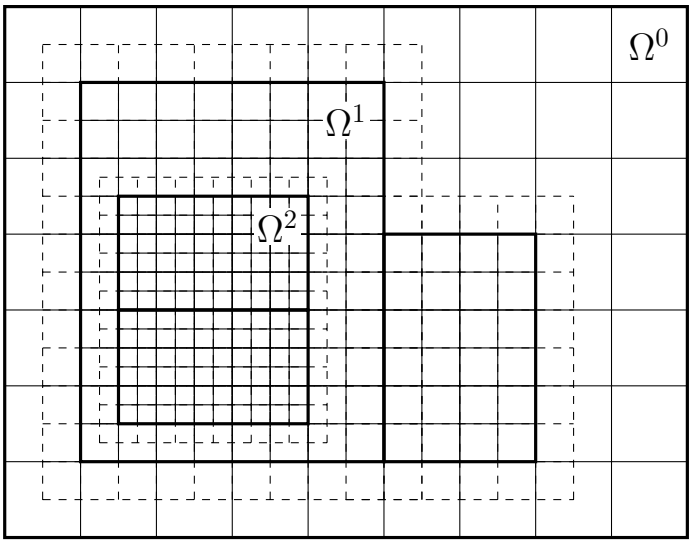


Figure 1.3: Illustration of AMR with two refined levels with refinement ratios of 2 for each level.

error estimation. A collection of these cells are organized into properly nested rectangular patches. Figure 1.3 illustrates a mesh with two refined levels, where Ω_0 is the base mesh, and the successively refined levels are Ω_1 and Ω_2, respectively. The thick lines represent the patches, or boxes; a domain is decomposed into a disjoint union of patches to perform calculations in parallel. The dashed lines represent invalid ghost cells; solution variables in invalid ghost cells are only interpolated from the coarser mesh and are not advanced in time. Readers interested in the detailed logistics of AMR are referred to previous work [13, 14].

1.5 Thermally Perfect, Chemically Reacting Flows

The focus of this research is to develop an algorithm that can simulate thermally perfect, multispecies, compressible flows with chemical reactions. The thermally perfect fluid assumption models the thermodynamic properties of reacting flows better than the calorically perfect fluid assumption. However, the thermally perfect fluid assumption introduces complications to the algorithm. Notable difficulties include:

- Thermodynamic and transport properties are spatially and temporally varying and are approximated using polynomials of temperature. Files of polynomial coefficients must be parsed and applied.
- Temperature must be iteratively calculated from the conservative state using a nonlinear solver.

Additionally, the species mass fractions must be constrained to ensure they adhere to $\sum_{n=1}^{N_s} c_n = 1$ and $0 \leq c_n \leq 1$. However, enforcing these physical constraints introduces stability issues which also motivates the development of stability techniques in this research.

1.6 Stability of Shocks and Detonation Solutions

Flows with shocks or detonations often pose challenges to numerical stability of an algorithm; these challenges are exacerbated for high-order algorithms. The methods employed in literature to stabilize solution of shocks and detonations in FVMs have wide variability. Work by Houim and Kuo [15] provides a demonstration of the extreme steps necessary to model chemically reacting, compressible flows using a FVM with a high-order limiting scheme. Some of the stability techniques mentioned in the paper are briefly listed here for reference: conservation is sacrificed in favor of a more stable quasi-conservative method (the double-flux method [16]); the fifth-order linear weighted essentially non-oscillatory (LWENO) scheme is replaced with a total variation diminishing (TVD) limiter in nonsmooth regions of the flow; a low Mach number adjustment is applied; and characteristic variables are limited and interpolated over primitive variables depending on particular flow conditions. Many of the listed stability methods, such as the double-flux method, are commonly implemented for solving chemically reacting fluid flows using the FVM [17, 18]. Literature has shown that simply adding numerical dissipation is not sufficient to eliminate or mitigate problems that arise as a result of the strong nonlinearities of the thermodynamic system [18]. In fact, inconsistently added dissipation can cause more issues with the solution [19, 20]. Therefore, the development of novel stability techniques that work with the fourth-order extension of the piecewise parabolic method (PPM) limiter [14] for modeling flows with shocks and detona-

tions is an important element of this research. Solutions that utilize the stability techniques retain fourth-order accuracy in smooth regions of the flow.

1.7 Dissertation Organization

This dissertation is structured as follows. In Chapter 2, the system of governing equations and closure models for compressible, thermally perfect, chemically reactive fluids are presented. In Chapter 3, the fourth-order FVM is described. In Chapter 4, the stability issues are demonstrated and novel stability techniques are proposed. In Chapters 5 and 6, the verification and validation of the algorithm are performed, respectively. The numerical results are presented in Chapter 7. Finally, the conclusions, original contributions, and future work are detailed in Chapter 8.

Chapter 2

Mathematical Modeling

In this work, gaseous fluid flow is modeled using the Navier-Stokes equations and a set of species transport equations with reaction modeling if combustion is considered. The computational domain is Cartesian, and the physical space is mapped to computational space using curvilinear coordinate transformation. In the present research, the grid is assumed to not deform over time. Next, the governing equations are described using the curvilinear coordinate transformation.

2.1 Governing Equations

Transformed using grid metrics, the system of governing equations for a compressible gas on a mapped domain, including the continuity, momentum, energy, and a set of species transport equations, is

$$\frac{\partial}{\partial t}\left(J\rho\right)+\vec{\nabla}_{\xi}\cdot\left(\mathrm{N}^{\mathrm{T}}\rho\vec{u}\right)=0\,, \tag{2.1}$$

$$\frac{\partial}{\partial t}\left(J\rho\vec{u}\right)+\vec{\nabla}_{\xi}\cdot\left(\mathrm{N}^{\mathrm{T}}(\rho\vec{u}\vec{u}+p\vec{\vec{I}})\right)=\vec{\nabla}_{\xi}\cdot\left(\mathrm{N}^{\mathrm{T}}\vec{\vec{\mathcal{T}}}\right)+J\rho\vec{f}\,, \tag{2.2}$$

$$\frac{\partial}{\partial t}\left(J\rho e\right)+\vec{\nabla}_{\xi}\cdot\left(\mathrm{N}^{\mathrm{T}}\rho\vec{u}(e+\frac{p}{\rho})\right)=\vec{\nabla}_{\xi}\cdot\left(\mathrm{N}^{\mathrm{T}}(\vec{\vec{\mathcal{T}}}\cdot\vec{u})\right)-\vec{\nabla}_{\xi}\cdot\left(\mathrm{N}^{\mathrm{T}}\vec{\mathcal{Q}}\right)+J\rho\vec{f}\cdot\vec{u}\,, \tag{2.3}$$

$$\frac{\partial}{\partial t}\left(J\rho c_n\right)+\vec{\nabla}_{\xi}\cdot\left(\mathrm{N}^{\mathrm{T}}\rho c_n\vec{u}\right)=-\vec{\nabla}_{\xi}\cdot\left(\mathrm{N}^{\mathrm{T}}\vec{\mathcal{J}}_n\right)+J\rho\dot{\omega}_n\,,\quad n=1\,,\ldots\,,N_s\,. \tag{2.4}$$

The metric Jacobian and transformation grid metrics are defined as

$$J\equiv\det\left(\vec{\nabla}_{\xi}\vec{x}\right),\quad \mathrm{N}^{\mathrm{T}}=J\vec{\nabla}_{x}\vec{\xi},\quad \text{and}\quad \mathrm{N}=J\left(\vec{\nabla}_{x}\vec{\xi}\right)^{\mathrm{T}}\,. \tag{2.5}$$

The identity tensor is denoted by $\vec{\vec{I}}$, and the total specific energy is given by

$$e=\frac{|\vec{u}|^2}{2}+\sum_{n=1}^{N_s}c_nh_n-\frac{p}{\rho}\,. \tag{2.6}$$

Refer to the nomenclature for the meaning of symbols in the equations.

The governing equations on mapped domains can be written in the conservative form as

$$\frac{\partial J\mathbf{U}}{\partial t} + \vec{\nabla}_\xi \cdot \left(\mathrm{N}^{\mathrm{T}}\vec{\mathbf{F}}\right) = \vec{\nabla}_\xi \cdot \left(\mathrm{N}^{\mathrm{T}}\vec{\boldsymbol{\mathcal{G}}}\right) + J\mathbf{S}\,, \tag{2.7}$$

where the solution vector, $\mathbf{U}$, inviscid flux vector, $\vec{\mathbf{F}}$, viscous flux vector, $\vec{\boldsymbol{\mathcal{G}}}$, and source vector, $\mathbf{S}$, are given by

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho\vec{u} \\ \rho e \\ \rho c_n \end{bmatrix}, \vec{\mathbf{F}} = \begin{bmatrix} \rho\vec{u} \\ \rho\vec{u}\vec{u} + p\vec{\vec{I}} \\ \rho\vec{u}\,(e + p/\rho) \\ \rho c_n\vec{u} \end{bmatrix}, \vec{\boldsymbol{\mathcal{G}}} = \begin{bmatrix} 0 \\ \vec{\vec{\mathcal{T}}} \\ (\vec{\vec{\mathcal{T}}} \cdot \vec{u}) - \vec{\mathcal{Q}} \\ -\vec{\mathcal{J}}_n \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 0 \\ \rho\vec{f} \\ \rho\vec{f} \cdot \vec{u} \\ \rho\dot{\omega}_n \end{bmatrix}. \tag{2.8}$$

The primitive variables are $\mathbf{W} = [\rho\,, \vec{u}\,, p\,, c_n]^{\mathrm{T}}$ and $[\widetilde{\mathbf{W}}]^{\mathrm{T}} = [\mathbf{W}, T]^{\mathrm{T}}$. The pressure is determined by the ideal gas law

$$p = \sum_{n=1}^{N_s} \rho c_n R_n T = \sum_{n=1}^{N_s} \rho c_n \frac{R_u}{M_n} T\,. \tag{2.9}$$

The stress tensor and the molecular heat flux vector on mapped domains are represented by $\vec{\vec{\mathcal{T}}}$ and $\vec{\mathcal{Q}}$, respectively. The mapped stress tensor is approximated by

$$\vec{\vec{\mathcal{T}}} = 2\mu \left(\vec{\vec{S}} - \frac{1}{3} J^{-1} \vec{\vec{I}} \vec{\nabla}_\xi \cdot \left(\mathrm{N}^{\mathrm{T}}\vec{u}\right) \right), \tag{2.10}$$

The molecular heat flux is modeled using Fourier's law,

$$\vec{\mathcal{Q}} = -\left(\kappa \frac{\mathrm{N}}{J} \vec{\nabla}_\xi T - \sum_{n=1}^{N_s} h_n \vec{\mathcal{J}}_n \right), \tag{2.11}$$

and the mass diffusion is modeled using Fick's law,

$$\vec{\mathcal{J}}_n = -\rho D_n \frac{\mathrm{N}}{J} \vec{\nabla}_\xi c_n\,. \tag{2.12}$$

2.2 Chemical Kinetics

For reacting flows, chemical reactions are modeled using finite-rate chemistry. The general form of the law of mass action [21] is used in Equation (2.4) to calculate the mean reaction rate for the nth species, defined by

$$\dot{\omega}_n = \frac{M_n}{\rho} \sum_{r=1}^{N_r} \left(\nu_{n,r}^{''} - \nu_{n,r}^{'} \right) \left(k_{\mathrm{f},r} \prod_{j=1}^{N_s} [X_j]^{\nu_{j,r}^{'}} - k_{\mathrm{b},r} \prod_{j=1}^{N_s} [X_j]^{\nu_{j,r}^{''}} \right) , \tag{2.13}$$

and the Arrhenius approach to calculate the forward reaction rate

$$k_{\mathrm{f},r} = A_r T^{\beta_r} \exp\left(\frac{-E_{a,r}}{R_c T} \right) . \tag{2.14}$$

The molar concentration of the nth species is defined by

$$[X_n] = \frac{\rho c_n}{M_n} . \tag{2.15}$$

For reversible reactions, the backward reaction rate is defined by

$$k_{\mathrm{b},r} = \frac{k_{\mathrm{f},r}}{K_{\mathrm{eq},r}} , \tag{2.16}$$

where

$$K_{\mathrm{eq},r} = \exp\left(\sum_{n=1}^{N_s} \nu_{n,r} \frac{-G_n}{R_u T} \right) \left(\frac{p_{\mathrm{atm}}}{R_u T} \right)^{\sum_{n=1}^{N_s} \nu_{n,r}} . \tag{2.17}$$

Additionally, $\nu_{n,r} = \nu_{n,r}^{''} - \nu_{n,r}^{'}$ and the Gibb's free energy is defined by

$$\frac{G_n}{R_u T} = \frac{H_n}{R_u T} - T \frac{S_n}{R_u} . \tag{2.18}$$

Some reactions are three-body reactions, where a third body, M, is required to stabilize the reaction. If three-body reactions are present, Equation (2.13) includes the third body component,

$$\left(\sum_{i=1}^{N_s} \alpha_{i,r} \left[X_i \right] \right) , \tag{2.19}$$

where $\alpha_{i,r} = 1$ for all species unless specified otherwise in the reaction mechanism, and Equation (2.13) becomes

$$\dot{\omega}_n = \frac{M_n}{\rho} \sum_{r=1}^{N_r} \left(\nu_{n,r}^{''} - \nu_{n,r}^{'} \right) \left(\sum_{i=1}^{N_s} \alpha_{i,r} \left[X_i \right] \right) \left(k_{\mathrm{f},r} \prod_{j=1}^{N_s} \left[X_j \right]^{\nu_{j,r}^{'}} - k_{\mathrm{b},r} \prod_{j=1}^{N_s} \left[X_j \right]^{\nu_{j,r}^{''}} \right) . \tag{2.20}$$

2.3 Thermodynamic and Transport Properties

The thermodynamic and transport properties must be approximated to close the system of governing equations.

Thermodynamic Properties

Under the thermally perfect fluid assumption, the thermodynamic properties, such as enthalpy and specific heat, must be evaluated based on temperature. For each species, the specific heat at constant pressure is [22]

$$\frac{c_{p,n}}{R_n} = \frac{a_{1,n}}{T^2} + \frac{a_{2,n}}{T} + \sum_{i=3}^{7} a_{i,n} T^{(i-3)} , \tag{2.21}$$

and the specific heat at constant pressure of the mixture is $c_p = \sum_{n=1}^{N_s} c_n c_{p,n}$. The total specific enthalpy is

$$\frac{h_n}{R_n T} = \frac{H_n}{R_u T} = \frac{a_{8,n}}{T} - \frac{a_{1,n}}{T^2} + \frac{a_{2,n}}{T} \ln T + \sum_{i=3}^{7} \frac{a_{i,n}}{i-2} T^{(i-3)} , \tag{2.22}$$

where $a_{8,n}$ is the integration constant for enthalpy. The total specific enthalpy for the fluid mixture is $h = \sum_{n=1}^{N_s} c_n h_n$. The specific molar entropy can also be solved using a fitted polynomial of

$$\frac{S_n}{R_u} = a_{9,n} - \frac{a_{1,n}}{2T^2} - \frac{a_{2,n}}{T} + a_{3,n} \ln T + \sum_{i=4}^{7} \frac{a_{i,n}}{i-3} T^{(i-3)} , \tag{2.23}$$

where $a_{9,n}$ is the integration constant for entropy.

Transport Properties

The values for dynamic viscosity and thermal conductivity are calculated using the curve fit polynomials [22]

$$\ln \mu_n = b_{1,n} \ln T + \frac{b_{2,n}}{T} + \frac{b_{3,n}}{T^2} + b_{4,n} \,, \tag{2.24}$$

and

$$\ln \kappa_n = c_{1,n} \ln T + \frac{c_{2,n}}{T} + \frac{c_{3,n}}{T^2} + c_{4,n} \,, \tag{2.25}$$

where $b_{1-4,n}$ and $c_{1-4,n}$ represent the nth species coefficients for μ and κ, respectively. The coefficients are provided by McBride et al. [23]. The mixture values of μ and κ are calculated using mixture-based formulas [24],

$$\mu = \frac{1}{2} \left[\sum_{n=1}^{N_s} \mu_n \chi_n + \left(\sum_{n=1}^{N_s} \frac{\chi_n}{\mu_n} \right)^{-1} \right] \,, \tag{2.26}$$

and

$$\kappa = \frac{1}{2} \left[\sum_{n=1}^{N_s} \kappa_n \chi_n + \left(\sum_{n=1}^{N_s} \frac{\chi_n}{\kappa_n} \right)^{-1} \right] \,, \tag{2.27}$$

where the mole fractions are

$$\chi_n = \frac{c_n}{M_n \sum_{j=1}^{N_s} \frac{c_j}{M_j}} \,. \tag{2.28}$$

Note that the coefficients in Equations (2.21), (2.22), (2.23), (2.24), and (2.25) represent two sets of coefficients for different ranges of temperature; one set of coefficients for $200 \text{ K} \leq T < 1000 \text{ K}$ and the other for $1000 \text{ K} \leq T < 6000 \text{ K}$.

The mass diffusion coefficient can be obtained from a given Schmidt number, Sc, using the relation

$$D_n = \frac{\mu}{\rho \mathrm{Sc}} \,, \tag{2.29}$$

or from a given Lewis number, Le, using the relation

$$D_n = \frac{\kappa}{\rho c_p \mathrm{Le}} \,. \tag{2.30}$$

It is worth noting that the mixing rule for μ and κ are less accurate when compared to semi-empirical methods or methods that take binary interactions between species into account. Mathur et al. [24] shows that the current mixture-based approach produces an error of 5.4% for the chemicals tested in their paper. However, tests with Chord using the semi-empirical approach demonstrated a three-fold increase in the computational cost of the transport properties when compared to the mixture-based approach.

Currently, the mass diffusion is modeled based solely on concentration gradients. However, mass diffusion can occur in the presence of temperature gradients (called the Soret effect), large pressure gradients, or body forces. Additionally, temperature gradients can form in the presence of concentration gradients, known as the Dufour effect. These effects are generally small and can be neglected. In total, the Soret and Dufour effects, barodiffusion, bulk viscosity, and radiative heat transfer are neglected in the present research. Models for these flow aspects are part of future work.

Chapter 3

Numerical Algorithm

CFD algorithms can employ a number of different spatial discretization methods such as finite-difference, finite-volume, finite-element, discontinuous Galerkin, or lattice-Boltzmann. Each of these methods have their own set of strengths and limitations. Details on these methods can be found in modern CFD textbooks [6, 25–27] and will not be repeated here. The present algorithm uses the FVM. FVMs are inherently conservative and benefit from computational efficiency with structured grids. Maintaining conservation is important for problems of interest to this work. In regions of smooth flow, high-order methods can reduce the numerical error of a solution relative to solutions obtained using low-order methods at the same mesh resolution. However, increasing the order of accuracy of a finite-volume algorithm significantly increases the complexity of the algorithm and the number of operations during the solution, thereby increasing the computational cost. These costs can be reduced or offset through the increase in computation per unit memory provided by high-order methods. This chapter details the fourth-order finite-volume algorithm for solving compressible, combusting flows, including discretization of fourth-order methods, time step calculations, and treatment of physical boundaries.

3.1 Fourth-Order Finite-Volume Methods

The semi-discrete form of the nonlinear system of governing Partial Differential Equations (PDEs) from Equation (2.1) to (2.4) is

$$\begin{aligned} \frac{\mathrm{d}}{\mathrm{d}t}\langle J\mathbf{U}\rangle_{\boldsymbol{i}} &= -\frac{1}{h}\sum_{d=1}^{D}\Bigg[\left(\langle \mathrm{N}_d^{\mathrm{T}}\vec{\mathbf{F}}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle \mathrm{N}_d^{\mathrm{T}}\vec{\mathbf{F}}\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right) \\ &\quad - \left(\langle \mathrm{N}_d^{\mathrm{T}}\vec{\boldsymbol{\mathcal{G}}}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle \mathrm{N}_d^{\mathrm{T}}\vec{\boldsymbol{\mathcal{G}}}\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right)\Bigg] + \langle J\mathbf{S}\rangle_{\boldsymbol{i}}\,, \end{aligned} \tag{3.1}$$

where $\langle \mathrm{N}_d^{\mathrm{T}} \vec{\boldsymbol{\mathcal{G}}} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ is the mapped diffusive fluxes, and $\langle J\mathbf{S} \rangle_{\boldsymbol{i}}$ is the mapped source term. The inviscid flux can be evaluated by simply multiplying the physical inviscid flux by the grid metric terms. Details of the inviscid flux evaluation on mapped domains can be found in the work by Guzik et. al. [9]. The remainder of this section is dedicated to the discretization of the mapped diffusive fluxes.

3.1.1 Deconvolution and Convolution Operations

First, the deconvolution and convolution (DC) operators are described before introducing the procedure for calculating the mapped diffusive fluxes. The DC operations are used frequently during the fourth-order solution procedure. The DC operators can also be a source of instability when solving flows with discontinuities; details regarding these stability issues are provided in Chapter 4.

Most second-order FVMs use the midpoint rule to approximate volume or face-averaged data, meaning $\langle \phi \rangle_{\boldsymbol{i}} = \phi_{\boldsymbol{i}} + \mathcal{O}(\Delta\xi^2)$. However, fourth-order FVMs must use a convolution operation to approximate the average value in a cell or across a face to fourth-order accuracy. The fourth-order convolution of a cell-centered value is used to approximate a cell-averaged value and is defined as

$$\langle \phi \rangle_{\boldsymbol{i}} = \phi_{\boldsymbol{i}} + \frac{\Delta\xi^2}{24} \Delta_\xi^{(2)} \phi_{\boldsymbol{i}} + \mathcal{O}\left(\Delta\xi^4\right) , \tag{3.2}$$

where ϕ can be any variable, $\langle \phi \rangle_{\boldsymbol{i}}$ is a cell-averaged value, $\phi_{\boldsymbol{i}}$ is a cell-centered value, and

$$\Delta_\xi^{(2)} \phi_{\boldsymbol{i}} = \sum_{d=1}^{D} \left(\frac{\partial^2 \phi}{\partial \xi_d^2} \right)_{\boldsymbol{i}} = \sum_{d=1}^{D} \frac{\phi_{\boldsymbol{i}+\boldsymbol{e}^d} - 2\phi_{\boldsymbol{i}} + \phi_{\boldsymbol{i}-\boldsymbol{e}^d}}{\Delta\xi_d^2} . \tag{3.3}$$

Correspondingly, the fourth-order deconvolution of a cell-averaged value is used to approximate a cell-centered value and is defined as

$$\phi_{\boldsymbol{i}} = \langle \phi \rangle_{\boldsymbol{i}} - \frac{\Delta\xi^2}{24} \Delta_\xi^{(2)} \langle \phi \rangle_{\boldsymbol{i}} + \mathcal{O}\left(\Delta\xi^4\right) . \tag{3.4}$$

Equations (3.2) and (3.4) are referred to as the convolution and deconvolution operators in cells, respectively. Clearly, the convolution operator is fourth-order accurate as long as the second derivatives in Equation (3.3) is second-order accurate. Therefore, $\phi_{\boldsymbol{i}}$ in Equation (3.3) can be replaced with the cell-averaged value or even the second-order value, $(\phi|_{(2)})_{\boldsymbol{i}}$. The convolution of a face-centered value is used to approximate a face-averaged value and is defined as

$$\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \phi_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{\Delta\xi^2}{24}\Delta_\xi^{(\perp,d)}\phi_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \mathcal{O}\left(\Delta\xi^4\right) , \tag{3.5}$$

where $\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ is a face-averaged value, $\phi_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}$ is a face-centered value, and

$$\begin{aligned} \Delta_\xi^{(\perp,d)}\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} &= \sum_{d'\neq d}^{D}\left(\frac{\partial^2\langle\phi\rangle}{\partial\xi_{d'}^2}\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \\ &= \sum_{d'\neq d}^{D}\frac{\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d+\boldsymbol{e}^{d'}} - 2\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d-\boldsymbol{e}^{d'}}}{\Delta\xi_{d'}^2} . \end{aligned} \tag{3.6}$$

Similarly, the deconvolution of a face-averaged value is used to a face-centered value and is defined as

$$\phi_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \frac{\Delta\xi^2}{24}\Delta_\xi^{(\perp,d)}\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \mathcal{O}\left(\Delta\xi^4\right) . \tag{3.7}$$

Other operations, such as the product of cell-averaged or face-averaged values, also require convolution operations. The product of two cell-averaged values is evaluated by

$$\langle\phi\psi\rangle_{\boldsymbol{i}} = \langle\phi\rangle_{\boldsymbol{i}}\langle\psi\rangle_{\boldsymbol{i}} + \frac{\Delta\xi^2}{12}\sum_{d=1}^{D}\left(\frac{\partial\langle\phi\rangle}{\partial\xi_d}\right)_{\boldsymbol{i}}\left(\frac{\partial\langle\psi\rangle}{\partial\xi_d}\right)_{\boldsymbol{i}} + \mathcal{O}\left(\Delta\xi^4\right) , \tag{3.8}$$

where

$$\left(\frac{\partial\langle\phi\rangle}{\partial\xi_d}\right)_{\boldsymbol{i}} = \frac{\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}}{2\Delta\xi_d} . \tag{3.9}$$

The product of two face-averaged values is

$$\langle\phi\psi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\langle\psi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{\Delta\xi^2}{12}\sum_{d'\neq d}^{D}\left(\frac{\partial\langle\phi\rangle}{\partial\xi_{d'}}\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\left(\frac{\partial\langle\psi\rangle}{\partial\xi_{d'}}\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \mathcal{O}\left(\Delta\xi^4\right), \tag{3.10}$$

where

$$\left(\frac{\partial\langle\phi\rangle}{\partial\xi_{d'}}\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d+\boldsymbol{e}^{d'}} - \langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d-\boldsymbol{e}^{d'}}}{2\Delta\xi_{d'}}. \tag{3.11}$$

As shown above, the second derivatives in the DC operators must be approximated. These approximations may introduce nonphysical oscillations into the solution near large discontinuities. This is essentially one of the instability sources motivating the formulation of the present fourth-order method.

3.1.2 Fourth-Order Finite-Volume Stencil

Additional fourth-order operations are needed to evaluate the diffusive flux. The operations are detailed with the aid of the fourth-order stencil. Figure 3.1 illustrates the stencil for the diffusive flux evaluation. First, a few notes pertaining to interpreting the figure and the general methodology are provided.

- The boxes represent cells; variables located inside a cell represent cell values and variables placed along an intersection of cells represent a face value.
- The face highlighted in gray at the lower left corner is an interior face where the mapped diffusive flux is evaluated. All cells shown in the figure are interior ghost cells relative to the face of interest.
- All operations are performed for every interior cell and face. However, the number of interior ghost cells and faces vary depending on the operation, and these numbers are provided in the methodology below.
- The dashed lines represent the values that are only used in the derivative terms of Equations (3.2), (3.4), (3.7), (3.5), (3.8), and (3.10).

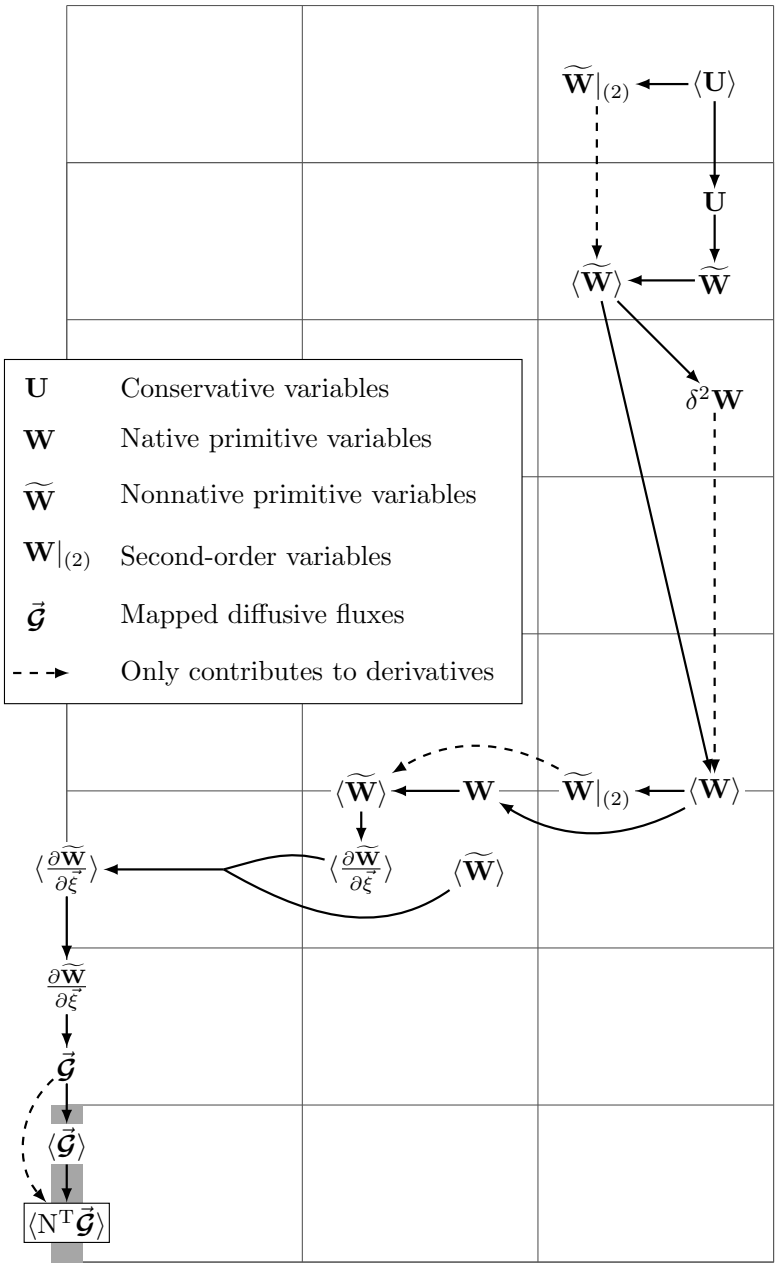


Figure 3.1: Illustration of a fourth-order stencil for evaluating the mapped viscous fluxes in two dimensions.

- The data dependence goes from the bottom-left to the top-right of the figure, meaning information propagates downward and to the left.

- All interior stenciled operations are centered, but the figure only shows dependence on a single variable in a single direction for each stencil. For example, solving for $\langle \frac{\partial \widetilde{\mathbf{W}}}{\partial \xi} \rangle_{\boldsymbol{i}+\boldsymbol{e}^d}$ requires

$\langle\widetilde{\mathbf{W}}\rangle_i$, $\langle\widetilde{\mathbf{W}}\rangle_{i+e^d}$, $\langle\widetilde{\mathbf{W}}\rangle_{i-2e^d}$, $\langle\widetilde{\mathbf{W}}\rangle_{i+2e^d}$, and $\langle\widetilde{\mathbf{W}}\rangle_{i-e^d}$, but the stencil only shows dependence on $\langle\widetilde{\mathbf{W}}\rangle_{i+2e^d}$ since it is the furthest variable in the upper-right direction.

- The normal direction refers to the direction normal to the highlighted face. The tangential direction refers to the direction (or directions in 3-D) tangent to the highlighted face.

The following 18 steps prescribe the solution procedure for evaluating the diffusive fluxes for the highlighted face in Figure 3.1.

1. Calculate the second-order, nonnative primitive state (i.e. temperature),

$$(\widetilde{\mathbf{W}}|_{(2)})_i = \mathbf{W}(\langle\mathbf{U}\rangle_i)\,, \tag{3.12}$$

   in 6 interior ghost cells.

2. Approximate the fourth-order accurate, cell-centered conservative state, $\mathbf{U}$, using

$$\mathbf{U}_i = \langle\mathbf{U}\rangle_i - \frac{\Delta\xi^2}{24}\Delta_\xi^{(2)}\langle\mathbf{U}\rangle_i\,, \tag{3.13}$$

   in 5 interior ghost cells.

3. Directly calculate the cell-centered, nonnative primitive state,

$$\widetilde{\mathbf{W}}_i = \mathbf{W}\left(\mathbf{U}_i\right)\,, \tag{3.14}$$

   in 6 interior ghost cells.

4. Approximate the fourth-order accurate, cell-averaged, nonnative primitive state using

$$\langle\widetilde{\mathbf{W}}\rangle_i = \widetilde{\mathbf{W}}_i + \frac{\Delta\xi^2}{24}\Delta_\xi^{(2)}(\widetilde{\mathbf{W}}|_{(2)})_i\,, \tag{3.15}$$

   in 5 interior ghost cells.

5. Interpolate the fourth-order accurate, face-averaged primitive state using

$$\langle \mathbf{W} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{-\langle \mathbf{W} \rangle_{\boldsymbol{i}-\boldsymbol{e}^d} + 7\langle \mathbf{W} \rangle_{\boldsymbol{i}} + 7\langle \mathbf{W} \rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle \mathbf{W} \rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}}{12} \,. \tag{3.16}$$

on all tangential faces adjacent to 3 and 2 interior ghost cells in the normal and tangential directions, respectively. *Steps 6 and 7 are only employed for flows with discontinuities.*

6. On all faces from the previous step, apply the slope flattening and PPM limiter [28, 29] to the face-averaged primitive state. The PPM limiter modifies the interpolants on each face adjacent to the cell to constrain the parabolic profile in each cell. These techniques require the undivided, second-order second derivative, $\delta^2\mathbf{W}$, in 5 interior ghost cells.

7. Resolve the limited left and right states on all faces from the previous step by solving a Riemann problem. Note that the solution to the Riemann problem depends on which variables of ρ, p, and T are limited and which is calculated using the ideal gas law. Numerical experimentation shows that limiting temperature and pressure and calculating density provides the most stable solution. This is discussed further in Chapter 4.

8. Calculate the second-order, nonnative primitive state from the face-averaged, native primitive state on all faces from step 6.

9. Deconvolve the face-centered, primitive state from the face-averaged, primitive state on all tangential faces adjacent to 2 interior ghost cells using Equation (3.7).

10. Calculate the face-centered, nonnative primitive state from the face-centered primitive state on all faces from the previous step.

11. Convolve the face-averaged, nonnative primitive state using

$$\langle \widetilde{\mathbf{W}} \rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \widetilde{\mathbf{W}}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{\Delta\xi^2}{24}\Delta_\xi^{(\perp,d)}(\widetilde{\mathbf{W}}|_{(2)})_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \,, \tag{3.17}$$

on all faces from step 9.

12. Evaluate the cell-averaged gradients of the nonnative primitive state using

$$\langle\frac{\partial\widetilde{\mathbf{W}}}{\partial\xi_d}\rangle_{\boldsymbol{i}} = \frac{\langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}}{\Delta\xi_d}\,, \tag{3.18}$$

in 2 interior ghost cells.

13. Evaluate the face-averaged normal gradients of the nonnative primitive state using

$$\langle\frac{\partial\widetilde{\mathbf{W}}}{\partial\xi_d}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{\langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}-\boldsymbol{e}^d} - 15\langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}} + 15\langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}}{12\Delta\xi_d}\,, \tag{3.19}$$

on all normal faces adjacent to 2 interior ghost cells in the tangential direction.

14. Evaluate the face-averaged tangential gradients of the nonnative primitive state using

$$\langle\frac{\partial\widetilde{\mathbf{W}}}{\partial\xi_{d'}}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{12}\left(-\langle\frac{\partial\widetilde{\mathbf{W}}}{\partial\xi_{d'}}\rangle_{\boldsymbol{i}-\boldsymbol{e}^d} + 7\langle\frac{\partial\widetilde{\mathbf{W}}}{\partial\xi_{d'}}\rangle_{\boldsymbol{i}} + 7\langle\frac{\partial\widetilde{\mathbf{W}}}{\partial\xi_{d'}}\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\frac{\partial\widetilde{\mathbf{W}}}{\partial\xi_{d'}}\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}\right)\,, \tag{3.20}$$

on all faces from the previous step.

15. Convolve the face-averaged gradients from the face-centered gradients on all normal faces adjacent to 1 interior ghost cell in the tangential direction using Equation (3.5).

16. Calculate the face-centered flux dyad, $\vec{\boldsymbol{\mathcal{G}}}$, on all faces from the previous step using the face-centered, nonnative primitive variables and gradients.

17. Convolve the face-centered flux to find the face-averaged flux dyad, $\langle\vec{\boldsymbol{\mathcal{G}}}\rangle$, on the highlighted face.

18. Solve the mapped face-averaged diffusive fluxes on the highlighted face using

$$\langle\mathrm{N}^{\mathrm{T}}\vec{\boldsymbol{\mathcal{G}}}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \langle\mathrm{N}^{\mathrm{T}}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\langle\vec{\boldsymbol{\mathcal{G}}}\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \frac{\Delta\xi^2}{12}\sum_{d'\neq d}^{D}\left(\frac{\partial\langle\mathrm{N}^{\mathrm{T}}\rangle}{\partial\xi_{d'}}\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\left(\frac{\partial\vec{\boldsymbol{\mathcal{G}}}}{\partial\xi_{d'}}\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\,. \tag{3.21}$$

3.1.3 Physical Boundary Schemes

The procedure above is modified to accommodate cells and faces that are adjacent to a physical boundary. To apply boundary conditions, the inertial physics only depends on boundary face values, but the diffusive physics depends on boundary face values and the boundary ghost cells. Application of boundary conditions are discussed in greater detail in Section 3.3.

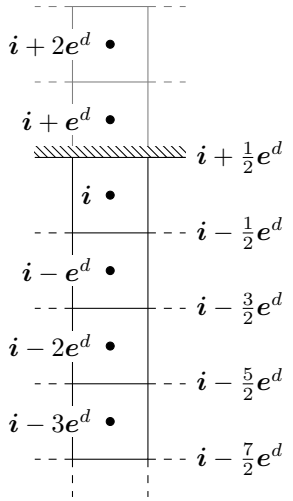


Figure 3.2: Illustration of cells and faces adjacent to an upper physical boundary.

The mesh at an upper physical boundary is illustrated in Figure 3.2. The cross-hatched pattern represents any physical boundary. Specific stencils are necessary when extrapolating to the boundary face or interpolating to the first interior face, labeled in the figure as $\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d$ and $\boldsymbol{i} - \frac{1}{2}\boldsymbol{e}^d$, respectively. The two cells above the boundary represent the exterior ghost cells, labeled $\boldsymbol{i} + \boldsymbol{e}^d$ and $\boldsymbol{i} + 2\boldsymbol{e}^d$ in the figure. The stencils near lower boundaries are similar to the stencils near upper boundaries; therefore, the following is designated for the remainder of the section: $\pm \Rightarrow +$ and $\mp \Rightarrow -$ for solutions near upper boundaries, $\pm \Rightarrow -$ and $\mp \Rightarrow +$ for solutions near lower boundaries, $\boldsymbol{i}$ is the first interior cell, and ϕ represents any primitive variable.

Boundary Face Extrapolation

The stencil for extrapolating the boundary face value, labeled $\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d$ in Figure 3.2, is

$$\langle\phi\rangle_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} = \frac{25\langle\phi\rangle_{\boldsymbol{i}} - 23\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} + 13\langle\phi\rangle_{\boldsymbol{i}\mp2\boldsymbol{e}^d} - 3\langle\phi\rangle_{\boldsymbol{i}\mp3\boldsymbol{e}^d}}{12}\,, \tag{3.22}$$

and the stencil for interpolating the first interior face value, labeled $\boldsymbol{i} - \frac{1}{2}\boldsymbol{e}^d$ in Figure 3.2, is

$$\langle\phi\rangle_{\boldsymbol{i}\mp\frac{1}{2}\boldsymbol{e}^d} = \frac{3\langle\phi\rangle_{\boldsymbol{i}} + 13\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} - 5\langle\phi\rangle_{\boldsymbol{i}\mp2\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}\mp3\boldsymbol{e}^d}}{12}\,. \tag{3.23}$$

After the boundary conditions are applied at the boundary faces, the exterior ghost cell values must be extrapolated depending on the type of boundary condition.

Neumann Boundary Condition

Neumann boundary conditions specify the value of the normal gradient at a boundary face. The exterior ghost cells are extrapolated using the face gradient value, $\left(\frac{\partial\phi}{\partial\xi_d}\right)_{\text{face}}$. If the boundary uses a Neumann boundary condition, the first exterior ghost cell, labeled $\boldsymbol{i} + \boldsymbol{e}^d$ in Figure 3.2, is extrapolated using

$$\langle\phi\rangle_{\boldsymbol{i}\pm\boldsymbol{e}^d} = \left(9\langle\phi\rangle_{\boldsymbol{i}} + 3\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}\mp2\boldsymbol{e}^d} \pm 12\Delta\xi_d\left(\frac{\partial\phi}{\partial\xi_d}\right)_{\text{face}}\right)/11\,, \tag{3.24}$$

and the second exterior ghost cell, labeled $\boldsymbol{i} + 2\boldsymbol{e}^d$ in Figure 3.2, is extrapolated using

$$\langle\phi\rangle_{\boldsymbol{i}\pm2\boldsymbol{e}^d} = 15\langle\phi\rangle_{\boldsymbol{i}\pm\boldsymbol{e}^d} - 15\langle\phi\rangle_{\boldsymbol{i}} + \langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} \mp 12\Delta\xi_d\left(\frac{\partial\phi}{\partial\xi_d}\right)_{\text{face}}\,. \tag{3.25}$$

Dirichlet Boundary Condition

Dirichlet boundary conditions specify the boundary face value. The exterior ghost cells are extrapolated using the boundary face value, ϕ_{face}. If the boundary uses a Dirichlet boundary condition, the first exterior ghost cell, labeled $\boldsymbol{i} + \boldsymbol{e}^d$ in Figure 3.2, is extrapolated using

$$\langle\phi\rangle_{\boldsymbol{i}\pm\boldsymbol{e}^d} = 4\phi_{\text{face}} - \frac{13\langle\phi\rangle_{\boldsymbol{i}} - 5\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}\mp2\boldsymbol{e}^d}}{3}\,, \tag{3.26}$$

and the second exterior ghost cell, labeled $\boldsymbol{i} + 2\boldsymbol{e}^d$ in Figure 3.2, is extrapolated using

$$\langle\phi\rangle_{\boldsymbol{i}\pm 2\boldsymbol{e}^d} = 20\phi_{\text{face}} - \langle\phi\rangle_{\boldsymbol{i}\pm\boldsymbol{e}^d} - \frac{83\langle\phi\rangle_{\boldsymbol{i}} - 37\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} + 8\langle\phi\rangle_{\boldsymbol{i}\mp 2\boldsymbol{e}^d}}{3} \,. \tag{3.27}$$

Boundary Gradient Stencils

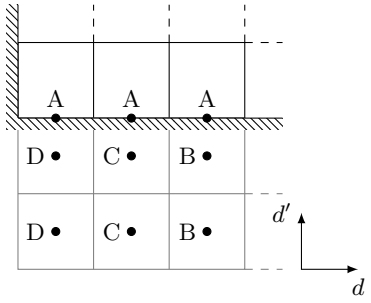


Figure 3.3: Illustration of a mesh at the lower corner of a domain in 2-D.

Using exterior ghost cells in the diffusion operations simplifies the solution logic by allowing face gradient operations to always use centered schemes. The mesh at the lower corner of the domain in two directions is shown in Figure 3.3. Note that the equations in this section are based on the assumption that d' and d are the directions normal and tangential to the boundary of interest, respectively. The face-averaged gradients from Equations (3.19) and (3.20) must be solved on the boundary faces labeled "A" the figure. Equation (3.19) can be solved using the extrapolated ghost cell-averaged values. However, solving Equation (3.20) on the boundary faces requires the tangential gradients to be known in the exterior ghost cells labeled "B", "C", and "D" in Figure 3.3. The cells labeled "B" represent ghost cells that are not adjacent to another physical boundary; the tangential gradient in these cells is calculated using

$$\langle\frac{\partial\phi}{\partial\xi_d}\rangle_{\boldsymbol{i}} = \frac{\langle\phi\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d} - 8\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d} + 8\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}}{12\Delta\xi_d}, \quad \{d \in D \mid d \neq d'\} \,. \tag{3.28}$$

The ghost cells labeled "C" represent cells that are separated by one cell from an adjacent physical boundary; the tangential gradient in these cells is calculated using

$$\langle \frac{\partial \phi}{\partial \xi_d} \rangle_{\boldsymbol{i}} = \pm \left(\frac{2\langle \phi \rangle_{\boldsymbol{i} \pm \boldsymbol{e}^d} + 3\langle \phi \rangle_{\boldsymbol{i}} - 6\langle \phi \rangle_{\boldsymbol{i} \mp \boldsymbol{e}^d} + \langle \phi \rangle_{\boldsymbol{i} \mp 2\boldsymbol{e}^d}}{6\Delta \xi_d} \right), \quad \{ d \in D \mid d \neq d' \} \ . \tag{3.29}$$

The ghost cells labeled "D" represent cells that are adjacent physical boundary; the tangential gradient in these cells is calculated using

$$\langle \frac{\partial \phi}{\partial \xi_d} \rangle_{\boldsymbol{i}} = \pm \left(\frac{11\langle \phi \rangle_{\boldsymbol{i}} - 18\langle \phi \rangle_{\boldsymbol{i} \mp \boldsymbol{e}^d} + 9\langle \phi \rangle_{\boldsymbol{i} \mp 2\boldsymbol{e}^d} - 2\langle \phi \rangle_{\boldsymbol{i} \mp 3\boldsymbol{e}^d}}{6\Delta \xi_d} \right), \quad \{ d \in D \mid d \neq d' \} \ . \tag{3.30}$$

For the signs in Equations (3.29) and (3.30), $\pm \Rightarrow +$ and $\mp \Rightarrow -$ when the adjacent boundary in the d'-direction is the upper boundary, and $\pm \Rightarrow -$ and $\mp \Rightarrow +$ when it is the lower boundary.

3.2 Temporal Integration

Once the fluxes and source terms are evaluated in the right-hand side (RHS) of the semi-discrete form in Equation (3.1), the solution is advanced in time using the standard, four-stage Runge-Kutta (RK4) time marching method [30]. The inertial physics, diffusive physics, and chemical reactions influence the necessary time step size for the explicit time marching method. The global time step calculation is evaluated by

$$\Delta t = \alpha \min \left(\left(\frac{1}{\Delta t_{\text{inertial}}} + \frac{1}{\Delta t_{\text{diffusive}}} + \frac{1}{\Delta t_{\text{chemical}}} \right)_{\boldsymbol{i}}^{-1} \right), \quad \forall \boldsymbol{i} \, , \tag{3.31}$$

where α is a constant for scaling the time step.

3.2.1 Inertial Time Scales

The inertial time step for a mapped solution is computed using

$$(\Delta t_{\text{inertial}})_{\boldsymbol{i}} = \frac{1.3925 \Delta \xi}{\sum_{d=1}^{D} \left| \vec{u} \cdot \boldsymbol{e}^d \right|} \, , \tag{3.32}$$

where the constant is based on stability analysis performed by Guzik et al. [9] and

$$\left|\vec{u}\cdot\boldsymbol{e}^d\right| = \frac{1}{2J}\sum_{\pm=+,-}\left(\left|\mathrm{N}_d^{\mathrm{T}}\vec{u}\right| + c\left\|\mathrm{N}_d^{\mathrm{T}}\right\|\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\,. \tag{3.33}$$

3.2.2 Diffusive Time Scales

The diffusive time step for a Cartesian mesh at a single point in space is calculated as

$$(\Delta t_{\text{diffusive}})_{\boldsymbol{i}} = \frac{2.5\rho_{\boldsymbol{i}}\Delta x^2}{\lambda_{\max} D\mu_{\boldsymbol{i}}}\,, \tag{3.34}$$

where $\lambda_{\max} = 16/3$ is from the Cartesian stability analysis performed by Gao et al. [31]. When the mesh is mapped, the physical grid spacing is no longer homogeneous and isotropic; therefore, a single location and direction of the physical grid spacing is no longer sufficient to estimate the time step size. Instead, the following is used in place of the physical grid spacing

$$\Delta x_{\boldsymbol{i}} = \min\left(\Delta x_1, \ldots, \Delta x_D\right)_{\boldsymbol{i}}\,, \tag{3.35}$$

which is the minimum physical grid spacing between all directions. The transformation from physical grid spacing to computational grid spacing is

$$\frac{\Delta\xi}{(\Delta x_d)_{\boldsymbol{i}}} = \left\|\frac{\mathrm{N}_d^{\mathrm{T}}}{J}\right\|_{\boldsymbol{i}}\,, \tag{3.36}$$

where $\Delta\xi = (\Delta\xi_d)_{\boldsymbol{i}}$ since the computational grid spacing is homogeneous and isotropic. Since the grid metrics are known on the faces, the adjacent face values are averaged in the cell, and the RHS of Equation (3.36) expands to

$$\left\|\frac{\mathrm{N}_d^{\mathrm{T}}}{J}\right\|_{\boldsymbol{i}} = \frac{1}{2}\sqrt{\sum_{m=1}^{D}\sum_{q=1}^{D}\left(\frac{\mathrm{N}_{m,q}^{\mathrm{T}}}{J}\right)^2_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}} + \frac{1}{2}\sqrt{\sum_{m=1}^{D}\sum_{q=1}^{D}\left(\frac{\mathrm{N}_{m,q}^{\mathrm{T}}}{J}\right)^2_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}}\,. \tag{3.37}$$

Substituting Equations (3.37) and (3.36) into Equation (3.35) makes

$$\Delta x_{\boldsymbol{i}} = \frac{\Delta\xi}{\max\left(\left\|\frac{\mathrm{N}_1^{\mathrm{T}}}{J}\right\|, \ldots, \left\|\frac{\mathrm{N}_D^{\mathrm{T}}}{J}\right\|\right)_{\boldsymbol{i}}} . \tag{3.38}$$

Similarly, the dynamic viscosity is generally known on the faces and is averaged into a cell using

$$\mu_{\boldsymbol{i}} = \frac{1}{2D}\sum_{d=1}^{D}\left(\mu_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \mu_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}\right) . \tag{3.39}$$

Finally, the diffusive time step in cell $\boldsymbol{i}$ on mapped domains becomes

$$(\Delta t_{\text{diffusive}})_{\boldsymbol{i}} = \frac{2.5\rho_{\boldsymbol{i}}\Delta\xi^2}{\lambda_{\max} D\mu_{\boldsymbol{i}} \max\left(\left\|\frac{\mathrm{N}_1^{\mathrm{T}}}{J}\right\|, \ldots, \left\|\frac{\mathrm{N}_D^{\mathrm{T}}}{J}\right\|\right)_{\boldsymbol{i}}^2} . \tag{3.40}$$

3.2.3 Chemical Time Scales

The time step based on the characteristic time of the chemical reactions is calculated as

$$(\Delta t_{\text{chemical}})_{\boldsymbol{i}} = \min\left(\frac{[X_1]}{\dot{\Phi}_1}, \frac{[X_2]}{\dot{\Phi}_2}, \ldots, \frac{[X_{N_s}]}{\dot{\Phi}_{N_s}}\right)_{\boldsymbol{i}} , \tag{3.41}$$

where the destruction rate for the nth species, $\dot{\Phi}_n$, is [32]

$$\dot{\Phi}_n = \sum_{r=1}^{N_r}\left(\nu_{n,r}^{'} k_{\mathrm{f},r}\prod_{i=1}^{N_s}[X_i]^{\nu_{i,r}^{'}} + \nu_{n,r}^{''} k_{\mathrm{b},r}\prod_{i=1}^{N_s}[X_i]^{\nu_{i,r}^{''}}\right) . \tag{3.42}$$

For three-body reactions, Equation (3.42) is multiplied by Equation (2.19).

3.3 Boundary Conditions

Boundary conditions are necessary to define a computational domain. Typical boundary conditions include slip wall, no-slip wall, inflow, outflow, and farfield boundary types.

3.3.1 Wall Boundaries

In Chord, walls are always assumed to be impermissible ($u = 0$ and $(\partial c_n/\partial \vec{n})_{\text{face}} = 0$) but vary based on velocity constraints (such as slip or no-slip) and thermal constraints (such as adiabatic or isothermal). To improve stability near walls, acoustic and isentropic corrections are applied.

The primitive state in the first interior cell and at the boundary face are used to apply an acoustic and isentropic wall correction. The following denotations are made for clarity: $\phi_{\boldsymbol{i}}$ is any variable ϕ in the cell adjacent to the boundary, $\phi_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}$ is any variable ϕ extrapolated to the boundary face from the interior, ϕ_{wall} is the corrected or set values of any variable ϕ on the wall boundary face, and u is the velocity component normal to the boundary face.

The acoustic and isentropic wall correction procedure is as follows:

1. Solve for $\gamma_{\boldsymbol{i}}$ and $\gamma_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}$ then solve for the speed of sound in the cell using

$$a_{\boldsymbol{i}} = \sqrt{\gamma_{\boldsymbol{i}} p_{\boldsymbol{i}}/\rho_{\boldsymbol{i}}}\,. \tag{3.43}$$

2. Assign a temporary variable that is the limited normal velocity on the face to be between zero and the velocity in the first interior cell using

$$u^*_{\text{wall}} = \begin{cases} 0, & \text{if } u_{\boldsymbol{i}} u_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d} < 0\,, \\ u_{\boldsymbol{i}}, & \text{if } \left|u_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}\right| > \left|u_{\boldsymbol{i}}\right|\,, \\ u_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}, & \text{otherwise}\,. \end{cases} \tag{3.44}$$

3. Define another temporary variable

$$\Delta u = u_{\boldsymbol{i}} - u^*_{\text{wall}}\,. \tag{3.45}$$

4. Calculate the pressure and temperature limits using

$$p_{\text{lim}} = p_{\boldsymbol{i}} \pm \rho_{\boldsymbol{i}} \Delta u a_{\boldsymbol{i}}\,, \tag{3.46}$$

and

$$T_{\text{lim}} = T_{\boldsymbol{i}} \left(\frac{p_{\text{lim}}}{p_{\boldsymbol{i}}} \right)^{\frac{\gamma_{\boldsymbol{i}} - 1}{\gamma_{\boldsymbol{i}}}} , \tag{3.47}$$

respectively.

5. Determine if the wave approaching the wall is a compression or expansion wave, then assign temporary variables accordingly.

   - If $\pm u^*_{\text{wall}} > 0$, then the approaching wave is a compression wave. This means that the pressure and temperature should increase as they approach the wall. The temporary variables are set as

   $$\begin{aligned} p^*_{\text{wall}} &= \max\left[p_{\text{lim}}, p_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}\right], \\ T^*_{\text{wall}} &= \max\left[T_{\text{lim}}, T_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}\right]. \end{aligned} \tag{3.48}$$

   - Otherwise, the approaching wave is an expansion wave. The temporary variables are set as

   $$\begin{aligned} p^*_{\text{wall}} &= \min\left[p_{\text{lim}}, p_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}\right], \\ T^*_{\text{wall}} &= \min\left[T_{\text{lim}}, T_{\boldsymbol{i}\pm\frac{1}{2}\boldsymbol{e}^d}\right]. \end{aligned} \tag{3.49}$$

   The density is solved at the wall as $\rho^*_{\text{wall}} = p^*_{\text{wall}}/(R_{\boldsymbol{i}} T^*_{\text{wall}})$.

6. Calculate the speed of sound at the wall using

$$a^*_{\text{wall}} = \sqrt{\gamma_{\boldsymbol{i}} p^*_{\text{wall}}/\rho^*_{\text{wall}}}\,. \tag{3.50}$$

7. Calculate the acoustical correction to pressure on the boundary face using

$$p_{\text{wall}} = p^*_{\text{wall}} \pm \rho^*_{\text{wall}} a^*_{\text{wall}} u^*_{\text{wall}}\,. \tag{3.51}$$

8. Calculate the isentropic correction to temperature on the boundary face using

$$T_{\text{wall}} = T^*_{\text{wall}} \left(\frac{p_{\text{wall}}}{p^*_{\text{wall}}} \right)^{\frac{\gamma_i - 1}{\gamma_i}} , \tag{3.52}$$

and solve for the density on the face using the ideal gas law.

9. Assign the velocity variables depending on the type of wall. For slip walls, assign the velocity normal to the wall to zero ($u_{\text{wall}} = 0$) and leave the tangential velocities as the extrapolated values. For no-slip walls, assign every velocity component to zero.

The exterior ghost cells must be computed after the acoustic and isentropic corrections are applied and the velocity values at the wall are assigned. For adiabatic walls, the temperature in the ghost cells are set using Equations (3.24) and (3.25) where $(\partial T/\partial \vec{n})_{\text{face}} = 0$. For isothermal walls, the temperature on the face is assigned using $T_{\text{face}} = T_{\text{wall}}$, and the ghost cells are extrapolated using Equations (3.26) and (3.27). Similarly, when the wall values are set to zero, the ghost cells are extrapolated using Equations (3.26) and (3.27).

3.3.2 Inflow and Outflow Boundaries

Characteristic analysis of the Euler equations shows that one piece of information leaves the domain and the rest of the information enters the domain for a subsonic inflow. Similarly, one piece of information enters the domain and the rest of the information leaves the domain for a subsonic outflow. However, for a supersonic inflow and outflow, all information enters or exits the domain, respectively.

For these boundary conditions, an interior and exterior state is set on the boundary face. The stability of the boundary is impacted based on which of the variables are assigned, extrapolated, or calculated using the ideal gas law. The interior state is always assigned to the values extrapolated to the boundary faces. For the exterior state at an inflow boundary, temperature, mass fractions, and velocity are prescribed and pressure is extrapolated from the interior. For the exterior state at an outflow boundary, temperature, mass fractions, and velocity are extrapolated from the interior and

pressure is prescribed. Density is always calculated using the ideal gas law. A Riemann solution is performed between the interior and exterior states to find the final boundary state on the face.

3.4 Multispecies and Thermally Perfect Treatments

A few points must be addressed regarding treatment of multispecies and thermally perfect fluids. Lookup tables are used to reduce the computational costs associated with computing the transport properties, a nonlinear iterative process is used to compute temperature and pressure from the conservative state, and the physical constraints on the mass fractions and species concentrations must be enforced to avoid nonphysical values.

3.4.1 Lookup Tables

Lookup tables for values of μ_n and κ_n are used to reduce the computational cost associated with the relations provided by Equations (2.24) and (2.25). Lagrange interpolating polynomials are used to extract the values from the lookup tables. The upper and lower bounds on temperature (denoted as T_{Hi} and T_{Lo}, respectively) depend on the empirical data being used. The temperature interval for the data by Gordon and McBride [22] is $T_{\text{Lo}} = 200\,\text{K}$ to $T_{\text{Hi}} = 6000\,\text{K}$. The lookup table increments are based on a set value of ΔT_T. The total size of the lookup table is $N_{\text{table}} = \lfloor (T_{\text{Hi}} - T_{\text{Lo}})/\Delta T_T \rceil$. The values for μ_n and κ_n for each species are solved at each temperature interval and stored in the lookup table. Another vector, $\vec{\mathbb{H}}$, is sized as the number of points used during interpolation, N_{interp}, and lists the location of the interpolation points. By default, values are interpolated with a 5 point centered scheme; therefore, $N_{\text{interp}} = 5$, and the vector $\vec{\mathbb{H}}$ would be $\vec{\mathbb{H}} = [-2, -1, 0, 1, 2]$. This interpolation scheme is illustrated in Figure 3.4. The denominators, $\vec{\mathbb{D}}$,

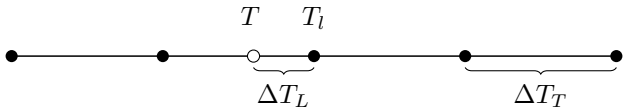


Figure 3.4: Illustration of interpolation stencil for lookup tables. The white dot is the temperature at which the properties are interpolated and the black dots are temperature values at given intervals in the lookup table.

are also calculated and stored using

$$\mathbb{D}_k = \prod_{i \neq k}^{N_{\text{interp}}} (\mathbb{H}_k - \mathbb{H}_i)\, \Delta T_T \,, \quad k = 1, \ldots, N_{\text{interp}} \,. \tag{3.53}$$

The procedure above takes place once before the start of the solution.

The following procedure occurs during the solution to interpolate the values of μ_n or κ_n at a temperature value using a Lagrange polynomial.

- Determine the index of the closest temperature value to the actual temperature using

$$l = \lfloor (T - 200)/\Delta T_T \rceil \,, \tag{3.54}$$

  and the temperature value at this index using $T_l = l \Delta T_T$.

- Calculate the vector of the polynomial coefficients, $\vec{\mathbb{P}}$, using

$$\mathbb{P}_k = \frac{1}{\mathbb{D}_k} \prod_{i \neq k}^{N_{\text{interp}}} -(\Delta T_L + \mathbb{H}_i \Delta T_T) \,, \quad k = 1, \ldots, N_{\text{interp}} \,, \tag{3.55}$$

  where $\Delta T_L = T_l - T$.

- Solve for the interpolated dynamic viscosity at T using

$$\mu = \sum_{k}^{N_{\text{interp}}} \mu_{j(k)} \mathbb{P}_k \,, \quad j(k) \equiv l + \mathbb{H}_k \,, \tag{3.56}$$

  where $\mu_{j(k)}$ is the value of μ_n at the $j(k)$ index in the lookup table. Similarly, κ and κ_n are plugged into Equation (3.56) to find the thermal conductivity.

3.4.2 Conservative to Primitive Operator

The primitive variables ρ, $\vec{u}$, and c_n are linearly related to the conservative variables ρ, $\rho\vec{u}$, and ρc_n. For calorically perfect fluids, the primitive variable p is the following linear function of the

conservative variables

$$p = (\gamma - 1)\left(\mathbf{U}_{(D+2)} - \sum_{d=1}^{D} \frac{\mathbf{U}_{(d+1)}^2}{2\mathbf{U}_{(1)}}\right), \tag{3.57}$$

where $\mathbf{U}_{(i)}$ represents the ith component of the $\mathbf{U}$ vector in Equation (2.8). For thermally perfect fluids, the relationship between the primitive variable p and the conservative variables is nonlinear. Instead of solving for pressure directly, temperature is computed from the nonlinear equation

$$f(\mathbf{U}, T) = \sum_{d=1}^{D} \frac{\mathbf{U}_{(d+1)}^2}{2\mathbf{U}_{(1)}} + \sum_{n=1}^{N_s} \left(\mathbf{U}_{(D+2+n)}\left(h_n(T) - \frac{R_u}{M_n}T\right)\right) - \mathbf{U}_{(D+2)}\,, \tag{3.58}$$

then pressure is calculated using the ideal gas law. Brent's method [33] is utilized to solve for temperature as $f(\mathbf{U}, T)$ approaches a set tolerance. The solution in Brent's method is converged if the following is true

$$\frac{|T_{\text{new}} - T_{\text{old}}|}{2} \leq 2\epsilon_1 |T_{\text{new}}| + \frac{\epsilon_2}{2}\,, \tag{3.59}$$

where ϵ_1 and ϵ_2 are 3×10^{-15} and 1×10^{-12}, respectively. This nonlinear iterative solution procedure may diverge if numerical oscillations occur in the flow variables; more information on this topic is provided in Chapter 4.

3.4.3 Physical Constraints

The physical constraints on mass fractions are

$$\sum_{n=1}^{N_s} c_n = 1 \quad \text{and} \quad 0 \leq c_n \leq 1, \quad \forall n \in N_s\,. \tag{3.60}$$

The first physical constraint in Equation (3.60) is adhered to as a result of conservation; however, positivity of c_n is not inherently ensured due to numerical errors; therefore, the physical constraints must be explicitly enforced. There are many different ways to enforce Equation (3.60). One common practice is to simply modify the mass fraction of a single inert species to meet the physical constraints. However, this method can result in significant changes to enthalpy estimates and lead to divergence of Equation (3.58). Instead, the physical constraints are enforced by modifying all

present species depending on the relative quantity of each species, so the greater the species concentration or mass fraction in a cell, the more that species is adjusted. This method is implemented using

$$(\rho c_n)^f = \frac{(\widetilde{\rho c_n})\,\rho}{\sum_{j=1}^{N_s} \widetilde{\rho c_j}}\,, \tag{3.61}$$

where

$$\widetilde{\rho c_n} = \max\left(0, \min(\rho, \rho c_n)\right), \tag{3.62}$$

and $(\rho c_n)^f$ is the adjusted species concentration. Equation (3.60) can also be enforced on the primitive mass fractions using

$$(c_n)^f = \frac{\widetilde{c_n}}{\sum_{j=1}^{N_s} \widetilde{c_j}}\,, \tag{3.63}$$

where

$$\widetilde{c_n} = \max\left(0, \min(1, c_n)\right), \tag{3.64}$$

and $(c_n)^f$ is the adjusted species concentration.

3.5 Adaptive Mesh Refinement

Although this research contains no new work on AMR, all functionality in the present work is implemented in the AMR framework to allow solutions to utilize AMR. Parallel AMR in Chord makes use of the Chombo library [13, 14]. Since the detailed AMR methodology has been well-documented in literature, only a typical work-flow for the AMR framework is briefly summarized here for convenience.

1. Regrid finer levels using a refinement tagging method. Regridding means to generate a new grid hierarchy. Interpolate the coarser level solution to regions that are newly refined.

2. Advance the solution in the coarser level using the methodology detailed in Section 3.1.2.

3. Interpolate the invalid ghost cells surrounding the finer level in both space and time. Invalid ghost cells are cells that are used to couple a finer grid level to a coarser grid level; they are

filled by interpolation from the coarser level. By contrast, valid ghost cells are cells that are filled by means of exchange. Referring back to Figure 1.3, valid ghost cells would be any solid lined cells adjacent to another box on the same level, while invalid ghost cells would be the cells shown with dashed lines.

4. Restart this procedure at step 1 for the finer level. The finer level can employ sub-cycling, meaning the solution on the finer level advances multiple smaller time steps relative to the coarser level solution. The time step size and the number of time steps on the finer level depend on the refinement ratio between the levels.

5. Average the solution from the finer level to the coarser level and correct the fluxes at the coarse-fine interface to ensure conservation.

AMR allows increased mesh resolution at areas of interest without the increased computational costs associated with having increased mesh resolution throughout the entire domain. Areas of interest, such as combustion flame fronts, should use a finer mesh to reduce the error. A strategy must be devised to ensure cells in these areas of interest are tagged for refinement. In the present work, the mesh is refined based on a normalized gradient of a variable (e.g. density), arbitrarily denoted as ϕ, using

$$\delta\phi_{\boldsymbol{i}} = \sqrt{\sum_{d=1}^{D} \left(\frac{\phi_{\boldsymbol{i}+\boldsymbol{e}^d} - \phi_{\boldsymbol{i}-\boldsymbol{e}^d}}{\phi_{\boldsymbol{i}+\boldsymbol{e}^d} + \phi_{\boldsymbol{i}-\boldsymbol{e}^d}} \right)^2}, \tag{3.65}$$

and a refinement threshold, δ_t. If $\delta\phi_{\boldsymbol{i}} > \delta_t$, then that particular cell is tagged for refinement.

Chapter 4

Numerical Instabilities

As previously mentioned, numerical stability is a common issue with high-order algorithms. The PPM limiter [14] has sufficiently suppressed oscillations for flows with strong discontinuities in previous research with Chord [34, 35]. However, severe stability issues become apparent when modeling flows with detonations. Efforts to extend the PPM limiter to resolve these issues were not successful, and new stability techniques were devised for flows with detonations. In general, the stability techniques must meet the following criteria: retain fourth-order accuracy in smooth regions, maintain the existing number of ghost cells, and improve stability for flows containing shocks and detonations.

The following stability techniques have been devised, implemented, and tested:

- face value limiting;
- primitive variable limiting selection;
- cell and face based DC flattening;
- face construction order reduction.

This chapter details the methodology and demonstrates the enhanced stability on problems with shocks. Verification of the stability techniques is detailed in Chapter 5.

4.1 Face Value Limiting

As mentioned in Section 3.1.2, the PPM limiter extended by McCorquodale and Colella [14] is employed for in the present work for solving flows with strong discontinuities. The PPM limiting mechanism can be viewed as three steps:

1. limit the interpolated face values;

2. limit the parabolic interpolant construction;

3. apply slope flattening to the interpolant.

McCorquodale and Colella recommend only limiting the parabolic interpolant construction and applying the slope flattening; they state that limiting the interpolated face values is overly dissipative. However, numerical experiments as part of this research demonstrated that face value limiting is necessary to stabilize solutions containing shocks and detonations.

4.1.1 Methodology

The fourth-order face interpolation in step 5 in Section 3.1.2 is replaced with the following face value limiting technique. First, a few terms in the face value limiting methodology are defined for convenience.

- The undivided first derivatives to the left and right of a face are

$$(\delta\phi)^L = \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}} - \left(\delta^2\phi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} , \tag{4.1}$$

  and

$$(\delta\phi)^R = \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}} + \left(\delta^2\phi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} , \tag{4.2}$$

  respectively.

- The fourth-order, undivided second derivatives are calculated at a face using

$$\left(\delta^2\phi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \left(\langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}} + \langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d}\right)/2 . \tag{4.3}$$

- The undivided third derivatives are calculated at a face using

$$\left(\delta^3\phi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \left(\delta^2\phi\right)_{\boldsymbol{i}+\boldsymbol{e}^d} - \left(\delta^2\phi\right)_{\boldsymbol{i}} , \tag{4.4}$$

where

$$\left(\delta^2\phi\right)_{\boldsymbol{i}} = \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - 2\langle\phi\rangle_{\boldsymbol{i}} + \langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d} \,. \tag{4.5}$$

- The error is estimated in a cell using [15]

$$e_{\boldsymbol{i}} = \left| \frac{-\langle\phi\rangle_{\boldsymbol{i}-2\boldsymbol{e}^d} + 4\langle\phi\rangle_{\boldsymbol{i}-\boldsymbol{e}^d} + 4\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}+2\boldsymbol{e}^d}}{6\langle\phi\rangle_{\boldsymbol{i}}} - 1 \right| . \tag{4.6}$$

The face value limiting methodology, applied at a face $\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d$ for each primitive variable, is described by the following 4 steps:

1. Interpolate the fourth-order face value using

$$\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{\langle\phi\rangle_{\boldsymbol{i}} + \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d}}{2} - \frac{\left(\delta^2\phi\right)_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d} + 18\left(\delta^2\phi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} + \left(\delta^2\phi\right)_{\boldsymbol{i}+\frac{3}{2}\boldsymbol{e}^d}}{120} \,, \tag{4.7}$$

   where the undivided second derivatives are computed using Equation (4.3).

2. If $(\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}})(\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}) > 0$, then the face value is not a local extremum and face value limiting is not applied. Similarly, if $e_{\boldsymbol{i}} < 1\times10^{-3}$ and $e_{\boldsymbol{i}+\boldsymbol{e}^d} < 1\times10^{-3}$, then the values around the face are smooth and face value limiting is not applied. Otherwise, move onto the next step.

3. Check that the interpolated face value is not a small perturbation of a cubic interpolation using

$$0.1\max\left(\left|\left(\delta^3\phi\right)^{\min}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\right|, \left|\left(\delta^3\phi\right)^{\max}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\right|\right) \leq \left(\delta^3\phi\right)^{\max}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \left(\delta^3\phi\right)^{\min}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} \,, \tag{4.8}$$

   where

$$\left(\delta^3\phi\right)^{\min}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \min\left(\left(\delta^3\phi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}, \left(\delta^3\phi\right)_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}, \left(\delta^3\phi\right)_{\boldsymbol{i}+\frac{3}{2}\boldsymbol{e}^d}\right), \tag{4.9}$$

   and

$$\left(\delta^3\phi\right)^{\max}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \max\left(\left(\delta^3\phi\right)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}, \left(\delta^3\phi\right)_{\boldsymbol{i}-\frac{1}{2}\boldsymbol{e}^d}, \left(\delta^3\phi\right)_{\boldsymbol{i}+\frac{3}{2}\boldsymbol{e}^d}\right). \tag{4.10}$$

If Equation (4.8) is false, do not apply any face value limiting. Otherwise, move onto the next step.

4. Set the face value based on the undivided first derivatives to the left and right side of the face. If the first derivatives change signs, set the face value to a linear average of the adjacent cells. Otherwise, check the magnitudes of the first derivatives. If the first derivative on one side of the face is twice the first derivative on the other side of the face, fit a quadratic through both cells that ensures the first derivative on the side of the face with a smaller first derivative becomes zero. This can be shown in equation form as

$$\langle\phi\rangle_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{\langle\phi\rangle_{\boldsymbol{i}} + \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d}}{2} - \frac{1}{6}\begin{cases} 0, & \text{if } (\delta\phi)^L (\delta\phi)^R < 0\,, \\ \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}}, & \text{if } (\delta\phi)^L > 2\,(\delta\phi)^R\,, \\ \langle\phi\rangle_{\boldsymbol{i}} - \langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d}, & \text{if } (\delta\phi)^R > 2\,(\delta\phi)^L\,, \\ (\delta^2\phi)_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}\,, & \text{otherwise}\,. \end{cases} \tag{4.11}$$

Modified Stencils Near Physical Boundaries

Different stencils are necessary to accommodate faces near boundary physical boundaries. Figure 3.2 is referenced to demonstrate a mesh near an upper physical boundary. The undivided second derivative at the first interior face, labeled $\boldsymbol{i} - \frac{1}{2}\boldsymbol{e}^d$ in Figure 3.2, modifies Equation (4.3) to

$$\left(\delta^2\phi\right)_{\boldsymbol{i}\mp\frac{1}{2}\boldsymbol{e}^d} = \frac{3\langle\phi\rangle_{\boldsymbol{i}} - 7\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} + 5\langle\phi\rangle_{\boldsymbol{i}\mp2\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}\mp3\boldsymbol{e}^d}}{2}\,. \tag{4.12}$$

The undivided second derivative at the first interior cell, labeled $\boldsymbol{i}$ in Figure 3.2, modifies Equation (4.5) to

$$\left(\delta^2\phi\right)_{\boldsymbol{i}} = 2\langle\phi\rangle_{\boldsymbol{i}} - 5\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} + 4\langle\phi\rangle_{\boldsymbol{i}\mp2\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}\mp3\boldsymbol{e}^d}\,. \tag{4.13}$$

Similarly, the error in the first interior cell modifies Equation (4.6) to

$$e_{\boldsymbol{i}} = \left|\frac{4\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d} - 6\langle\phi\rangle_{\boldsymbol{i}\mp2\boldsymbol{e}^d} + 4\langle\phi\rangle_{\boldsymbol{i}\mp3\boldsymbol{e}^d} - \langle\phi\rangle_{\boldsymbol{i}\mp4\boldsymbol{e}^d}}{\langle\phi\rangle_{\boldsymbol{i}}} - 1\right|\,, \tag{4.14}$$

and the error at the second interior cell, labeled $\boldsymbol{i} - \boldsymbol{e}^d$ in Figure 3.2, modifies Equation (4.6) to

$$e_{\boldsymbol{i}\mp\boldsymbol{e}^d} = \left| \frac{\langle\phi\rangle_{\boldsymbol{i}} + 6\langle\phi\rangle_{\boldsymbol{i}\mp 2\boldsymbol{e}^d} - 4\langle\phi\rangle_{\boldsymbol{i}\mp 3\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}\mp 4\boldsymbol{e}^d}}{4\langle\phi\rangle_{\boldsymbol{i}\mp\boldsymbol{e}^d}} - 1 \right| . \tag{4.15}$$

The interpolation of the faces near and at boundaries remains the same as detailed in Section 3.1.3. No face value limiting is applied to the boundary face value.

4.1.2 Demonstration of Stability

As briefly mentioned in Section 3.1.2, selecting which variables to limit and which calculate has a significant impact on the stability of the solution. If all three variables (ρ, p, and T) are limited independently, the state on the face could violate the ideal gas law. Therefore, only two variables can be limited, and the third must be calculated using the ideal gas law.

A shock tube case with a shock strength of $\mathrm{M} \approx 4.9$ is used to demonstrate the capability of the face value limiting and the impact of the calculated primitive variable selection. The shock tube domain is $L_x \times L_y = 31.25\,\mathrm{cm} \times 0.4883\,\mathrm{cm}$ and the initial discontinuity is at $x = 15.625\,\mathrm{cm}$. The mesh is $N_x \times N_y = 512 \times 8$. The domain is periodic in the y-direction since the shock tube is essentially 1-D. The x-direction boundary conditions are slip walls. Half of the shock tube is composed of air and the other half is composed of helium. A large gradient of the mixture gas constants form at the initial discontinuity due to disparate molecular weights of He and $\mathrm{O_2}$/$\mathrm{N_2}$. The diffusive physics are ignored for this solution, and artificial viscosity is utilized. The shock tube is initialized with the following left and right properties, denoted by subscripts L and R, respectively,

$$\begin{bmatrix} p_L \\ T_L \\ (c_{\mathrm{N_2}})_L \\ (c_{\mathrm{O_2}})_L \\ (c_{\mathrm{He}})_L \end{bmatrix} = \begin{bmatrix} 7.27\times10^6\ \mathrm{Pa} \\ 2\times10^3\ \mathrm{K} \\ 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} p_R \\ T_R \\ (c_{\mathrm{N_2}})_R \\ (c_{\mathrm{O_2}})_R \\ (c_{\mathrm{He}})_R \end{bmatrix} = \begin{bmatrix} 1\times10^5\ \mathrm{Pa} \\ 300\ \mathrm{K} \\ 0.767 \\ 0.233 \\ 0 \end{bmatrix} . \tag{4.16}$$

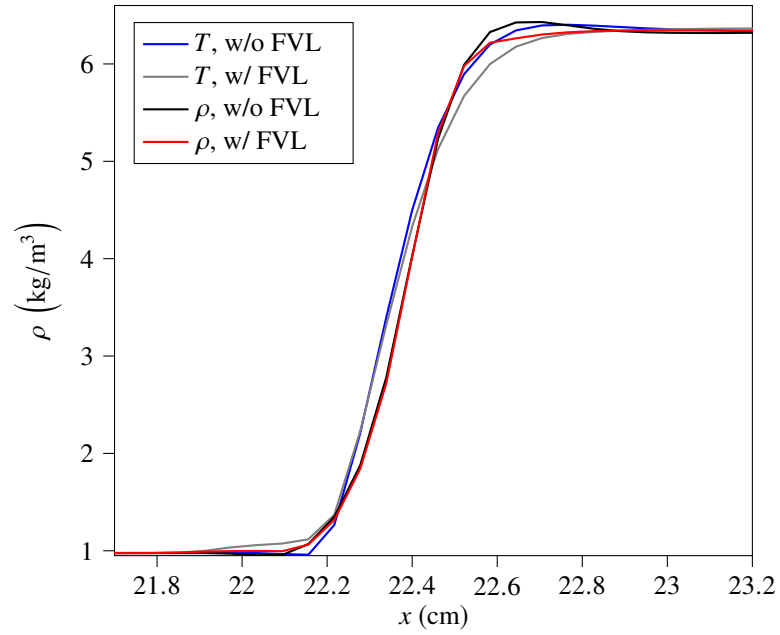


Figure 4.1: Density profiles of the contact surface at solution time $t = 49$ µs. The blue line: temperature is calculated and face value limiting is not applied; black line: density is calculated and face value limiting is not applied; gray line: temperature is calculated and face value limiting is applied; red line: density is calculated and face value limiting is applied. Pressure is always limited.

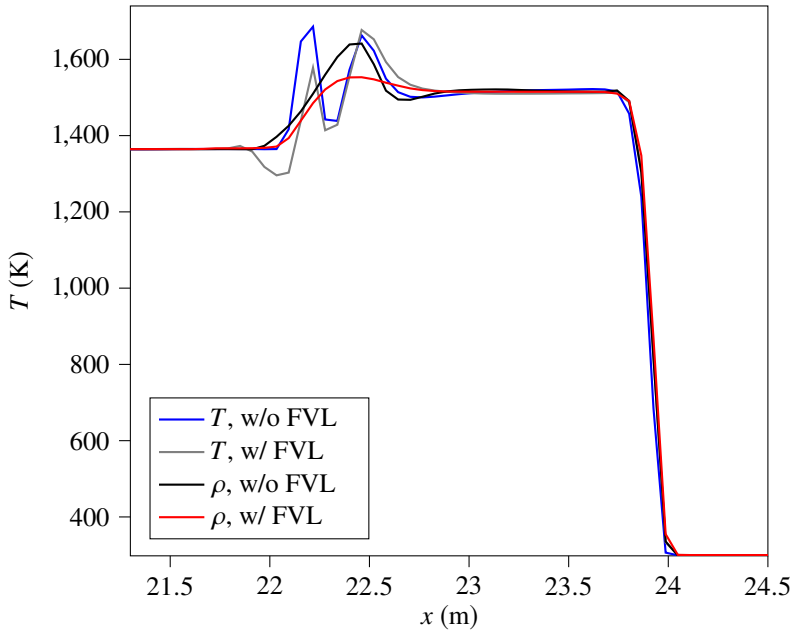


Figure 4.2: Shock tube temperature profiles at solution time $t = 49$ µs. The blue line: temperature is calculated and face value limiting is not applied; black line: density is calculated and face value limiting is not applied; gray line: temperature is calculated and face value limiting is applied; red line: density is calculated and face value limiting is applied. Pressure is always limited.

The test problem is run with four different implementations to demonstrate the stability techniques described above. The results are shown in Figures 4.1 and 4.2. The density profiles in Figure 4.1 show that the overshoot at the contact surface is eliminated when face value limiting is applied. Since temperature is a function of density, energy, and mass fractions, the temperature profiles in Figure 4.2 are a better indicator of instabilities than the density profiles. As with density, stability issues arise mainly around the contact surface. Oscillations around the contact surface are eliminated when density is calculated based on the limited temperature and pressure. The results suggest the most stable scheme applies face value limiting, the PPM limiter, and slope flattening to all primitive states except density, which is calculated using the ideal gas law.

4.2 Deconvolution/Convolution Stability

Existing stabilization techniques for FVMs [14, 28, 29] focus on reducing oscillations at the faces through means such as interpolant limiting and slope flattening. These existing stabilization techniques are often sufficient for second-order FVMs, where the cell and flux integrals are approximated using the midpoint rule. However, fourth-order FVMs use the integral of a cubic polynomial to compute a cell or face-averaged value, which contains first or second derivatives. As seen in Section 3.1.1, the finite-difference approximation of these derivatives require information from neighboring cells or faces. Near large gradients or discontinuous regions, these numerical approximations can introduce severe nonphysical oscillations in the flow. Furthermore, the impact of these nonphysical oscillations can be exacerbated by the nonlinearities of the thermodynamic system. To provide theoretical scenario of this, assume that severe oscillations are introduced during the deconvolution of $\langle \mathbf{U} \rangle_i$ in Equation (3.13). Traditionally, the face value limiting, interpolant limiting, or slope flattening mechanisms would be relied on to eliminate these oscillations. However, due to the nonlinear relationship between the conservative variables and temperature, these oscillations must be addressed before the limiting or flattening on the faces. Otherwise, the nonlinear solution for the cell-centered primitive variables could diverge. Additionally, the limiting and slope flattening are unable to eliminate oscillations introduced during the face-based DC op-

erations due to the order of operations. Unfortunately, literature regarding techniques to alleviate oscillations in high-order FVMs [36] is scarce. To solve the issues in the present algorithm, the DC flattening and face construction order reduction (FCOR) techniques are developed and presented here.

4.2.1 Methodology

The DC stability techniques involve checks between low-order and high-order variables on faces and in cells. In this section, "high-order values" refer to averaged or centered values in cells or at faces, such as $\langle\widetilde{\mathbf{W}}\rangle$, or $\widetilde{\mathbf{W}}$, and "low-order values" refer to values that result from operations directly on cell or face-averaged values directly, such as $(\widetilde{\mathbf{W}}|_{(2)})_{\boldsymbol{i}}$.

Deconvolution/Convolution Flattening

The DC flattening technique selectively reduces the order of accuracy of the DC operations based on the normalized difference between the fourth-order approximation and the second-order approximation of a variable. The normalized difference is calculated using

$$(\Delta\phi)^{DC} = \frac{\left|\phi|_{(2)} - \phi|_{(4)}\right|}{C_1 + C_2\left|\phi|_{(4)}\right| + \left|\phi|_{(2)}\right|}\,, \tag{4.17}$$

where the constants $C_1 = 1\times10^{-20}$ and $C_2 = 0.02$. These constants serve different purposes. C_1 prevents the denominator from going to zero, and C_2 ensures the denominator does not skew the normalization when the values are different orders of magnitude. The flattening coefficient, η^{DC}, is then solved as

$$\eta^{DC} = \begin{cases} 0, & \text{if } (\Delta\phi)^{DC} > \epsilon^{DC}\,, \\ 1, & \text{otherwise}\,, \end{cases} \tag{4.18}$$

where $\epsilon^{DC} = 0.2$ is the tolerance for the DC order reduction. The values for C_1, C_2, and ϵ^{DC} are the result of numerical experimentation in this research. The flattening coefficient is applied using

$$\phi^f = (1-\eta^{DC})\phi|_{(2)} + \eta^{DC}\phi|_{(4)}\,, \tag{4.19}$$

where ϕ^f represents the final flattened value in a cell or at a face. If the nonlinear solution in Equation (3.58) fails to converge to a solution, the values for T and p are set to 1.2×10^{300} so that $(\Delta\phi)^{DC} \approx 1 > \epsilon^{DC}$. This ensures the order of accuracy is dropped when the nonlinear solution fails to converge. If the second-order values of $T|_{(2)}$ and $p|_{(2)}$ are greater than 1×10^{100}, the solution has diverged and is halted.

Regarding Equation (4.17), special care must be taken when velocity values approach zero. To address this issue, the constant $C_1 = 1$ when $\phi = \vec{u}$. Special care must also be taken when mass fractions oscillate near values much less than one. To address this issue, the constant $C_2 = 1$ when $\phi = c_n, \forall n \in N_s$.

Face Construction Order Reduction

In addition to applying the DC flattening, it is necessary to selectively modify the order of accuracy of the values in the face value construction and limiting. Two different checks are required to see if FCOR is necessary. The first test involves the percent difference between the linear face construction using fourth-order average values and second-order values. The equation for the first test is

$$\epsilon_1^{FR} < \frac{\phi^{\max}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} - \phi^{\min}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}}{\phi^{\max}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d}}, \tag{4.20}$$

where $\epsilon_1^{FR} = 1\times10^{-4}$,

$$\phi^{\max}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{2}\max\left(\left|\left(\phi|_{(2)}\right)_{\boldsymbol{i}+\boldsymbol{e}^d} + \left(\phi|_{(2)}\right)_{\boldsymbol{i}}\right|, \left|\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}}\right|\right), \tag{4.21}$$

and

$$\phi^{\min}_{\boldsymbol{i}+\frac{1}{2}\boldsymbol{e}^d} = \frac{1}{2}\min\left(\left|\left(\phi|_{(2)}\right)_{\boldsymbol{i}+\boldsymbol{e}^d} + \left(\phi|_{(2)}\right)_{\boldsymbol{i}}\right|, \left|\langle\phi\rangle_{\boldsymbol{i}+\boldsymbol{e}^d} + \langle\phi\rangle_{\boldsymbol{i}}\right|\right). \tag{4.22}$$

The second check evaluates the normalized difference between the undivided, second-order face derivatives. The second check at the $\boldsymbol{i} + \frac{1}{2}\boldsymbol{e}^d$ face is

$$\epsilon_2^{FR} < \frac{\left|\delta^1\left(\phi|_{(2)}\right) - \delta^1\left(\langle\phi\rangle\right)\right|}{\max\left(\left|\delta^1\left(\phi|_{(2)}\right)\right|, \left|\delta^1\left(\langle\phi\rangle\right)\right|\right)}, \tag{4.23}$$

where $\epsilon_2^{FR} = 1 \times 10^{-3}$ and the second-order, first derivative on the face is calculated using

$$\delta^1(\phi) = \phi_{\boldsymbol{i}+\boldsymbol{e}^d} - \phi_{\boldsymbol{i}} \,. \tag{4.24}$$

The tolerances for both tests are determined through numerical experimentation conducted as part of this research. If Equations (4.20) and (4.23) are both true, then the face values are constructed and limited using the second-order values.

Implementation

The following details when the DC flattening and FCOR techniques are used relative to the solution procedure in Section 3.1.2.

- After step 4, the flattening coefficient is calculated using the normalized difference of

$$\left(\Delta\langle\widetilde{\mathbf{W}}\rangle\right)^{DC} = \frac{\left|\widetilde{\mathbf{W}}|_{(2)} - \langle\widetilde{\mathbf{W}}\rangle\right|}{C_1 + C_2\left|\langle\widetilde{\mathbf{W}}\rangle\right| + \left|\widetilde{\mathbf{W}}|_{(2)}\right|} \,. \tag{4.25}$$

  The flattening coefficient is applied using

$$\langle\widetilde{\mathbf{W}}\rangle^f = (1 - \eta^{DC})\widetilde{\mathbf{W}}|_{(2)} + \eta^{DC}\langle\widetilde{\mathbf{W}}\rangle \,, \tag{4.26}$$

  where $\langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}} \rightarrow \langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}}^f$ throughout the remainder of the solution procedure in Section 3.1.2. This step is then repeated for the cell-centered primitive variables, $\widetilde{\mathbf{W}}$.

- If Equations (4.20) and (4.23) are both true, then the face value limiting technique in Section 4.1 uses the second-order cell values, $(\widetilde{\mathbf{W}}|_{(2)})_{\boldsymbol{i}}$, instead of $\langle\widetilde{\mathbf{W}}\rangle_{\boldsymbol{i}}$.

- Following the solution of the primitive states on the faces in the 9th step in Section 3.1.2, Equation (4.19) is applied to the following:

  - The face-centered, native primitive variables are compared to the face-averaged, native primitive variables. Since no low-order primitive state exists at during this stage,

$\phi|_{(2)}$ in Equations (4.17) and (4.19) is replaced by the face-averaged, native primitive variables, which are considered the more stable primitive state.

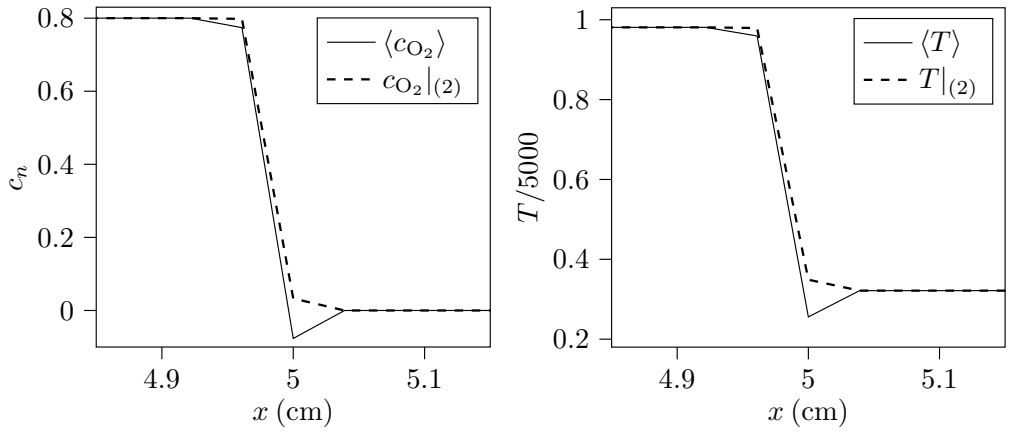
– The face-centered, nonnative primitive variables are compared to the second-order, nonnative primitive variables on the face from step 8.

– The face-averaged, nonnative primitive variables are compared to the second-order, nonnative primitive variables on the face from step 8.

4.2.2 Demonstration of Stability

Deconvolution/Convolution Flattening

A shock tube problem demonstrates the numerical oscillations that occur due to the DC operations. The shock tube domain is $L_x \times L_y = 10\,\mathrm{cm} \times 0.625\,\mathrm{cm}$ and the initial discontinuity is at $x = 5\,\mathrm{cm}$. The mesh is $N_x \times N_y = 128 \times 8$ cells with two refined levels, each with a refinement ratio of 2. The mesh refinement is based on the gradient of density. This shock tube problem differs from the one in Section 4.1 because each case is deliberately designed to expose the different stability issues. The shock tube is initialized with left and right properties shown below.



(a) O_2 mass fraction profile. **(b)** Normalized temperature profile.

Figure 4.3: Demonstration of oscillations produced during solution procedure. The solid, black lines are the fourth-order, cell-averaged values and the dashed, black lines are the second-order, cell-averaged values.

$$\begin{bmatrix} p_L \\ T_L \\ (c_{\mathrm{N}_2})_L \\ (c_{\mathrm{O}_2})_L \\ (c_{\mathrm{CO}})_L \\ (c_{\mathrm{CO}_2})_L \end{bmatrix} = \begin{bmatrix} 1.177\times10^6\ \mathrm{Pa} \\ 4.905\times10^3\ \mathrm{K} \\ 0.1 \\ 0.8 \\ 0 \\ 0.1 \end{bmatrix}, \begin{bmatrix} p_R \\ T_R \\ (c_{\mathrm{N}_2})_R \\ (c_{\mathrm{O}_2})_R \\ (c_{\mathrm{CO}})_R \\ (c_{\mathrm{CO}_2})_R \end{bmatrix} = \begin{bmatrix} 1\times10^5\ \mathrm{Pa} \\ 1.607\times10^3\ \mathrm{K} \\ 0.5 \\ 0 \\ 0.25 \\ 0.25 \end{bmatrix}. \tag{4.27}$$

Figure 4.3 shows the profiles of c_{O_2} and the normalized temperature on the first refined mesh level during the first time step. The solid, gray line is the solution of Equation (3.12), and the dashed, black line is the solution of Equation (3.15). Figure 4.3a shows a nonphysical, negative cell-averaged mass fraction at the discontinuity. The corrections detailed in Section 3.4.3 would eliminate this oscillation, but would significantly modify the mass fractions of the other species. Figure 4.3b shows the cell-averaged temperature is 1278 K at $x = 5\ \mathrm{cm}$, while the second-order, cell-averaged temperature is 1744 K; this produces a value of $(\Delta T)^{DC} = 0.26$ from Equation (4.17). If c_{CO} is replaced with c_{H_2} in the initialization, the oscillations introduced during deconvolution cause the nonlinear solution of the cell-centered primitive variables to diverge during the first time step. The DC flattening technique would reduce the order of the DC operations at the discontinuity and eliminate the oscillations in Figure 4.3.

Face Construction Order Reduction

An ODW over a wedge is used to demonstrate the improvements provided by the FCOR technique. In this problem, high Mach number flow of a reactive mixture encounters a ramp, causing an oblique shock to form. The heating from the oblique shock induces deflagration and ultimately produces a detonation wave. In the present work, a Schwarz-Christoffel mapping transformation is used to model the wedge as illustrated in Figure 4.4, where l_r represents the length of the ramp, and l_s represents the length of the lead up to the ramp.

For the current problem, $\theta = 23.8°$, $\mathrm{M}_\infty = 8$, $p_\infty = 34\ \mathrm{kPa}$, and $T_\infty = 300\ \mathrm{K}$. The mixture is a stoichiometric mixture of hydrogen and air. The current setup is partly based on a case run in

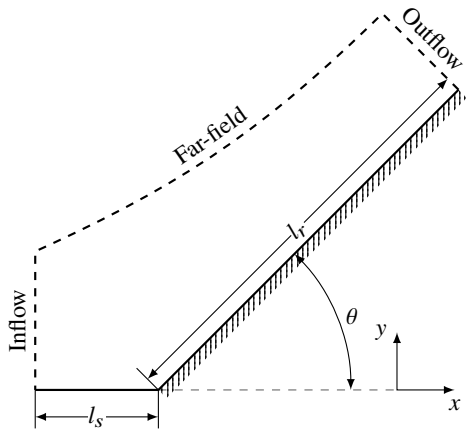


Figure 4.4: Demonstration of computational domain.

Figueria da Silva and Deshaies [37]. The reaction mechanism is the 9 species, 19-step mechanism from Billet et al. [38]. The lengths of the ramp and lead up to the ramp are $l_r = 7\,\mathrm{cm}$ and $l_s = 0.5\,\mathrm{cm}$, respectively. The base mesh is 320×128. The computational mesh has three additionally refined levels; the first two levels have refinement ratios of 4 and the third level has a refinement ratio of 2. Refinement is based on gradients of temperature, and there is fixed refinement in the layer of cells adjacent to the wall. Sub-cycling is not used for this problem.

Figure 4.5 shows temperature profiles taken along the direction of the lower boundary in the rectangular computational domain at a fixed point of $\eta = 3\times10^{-4}$, or 0.075% of the domain in the wall normal direction. The profile is taken at solution time $t = 11.4\ \mu\mathrm{s}$. The profiles show the temperature far behind the oblique shock wave (OSW), which forms at $\xi \approx 0.049$. The rise in temperature is due to the heat release from the reactions which are initiatiated due to the numerical overheating at the wall. Two different profiles are shown, the dashed line is the solution with FCOR applied, and the solid line is the solution without FCOR applied. Both solutions utilize face value limiting and DC flattening. The solution without FCOR contains severe oscillations in this overheated region. These oscillations cause the solution to diverge shortly after the solution time $t = 11.4\ \mu\mathrm{s}$. The solution with FCOR increases the amount of numerical overheating at the wall, resulting in increased heat release above the wall. However, the oscillations are eliminated when the FCOR is utilized. Reducing the severe oscillations in the flow but increasing the overheating

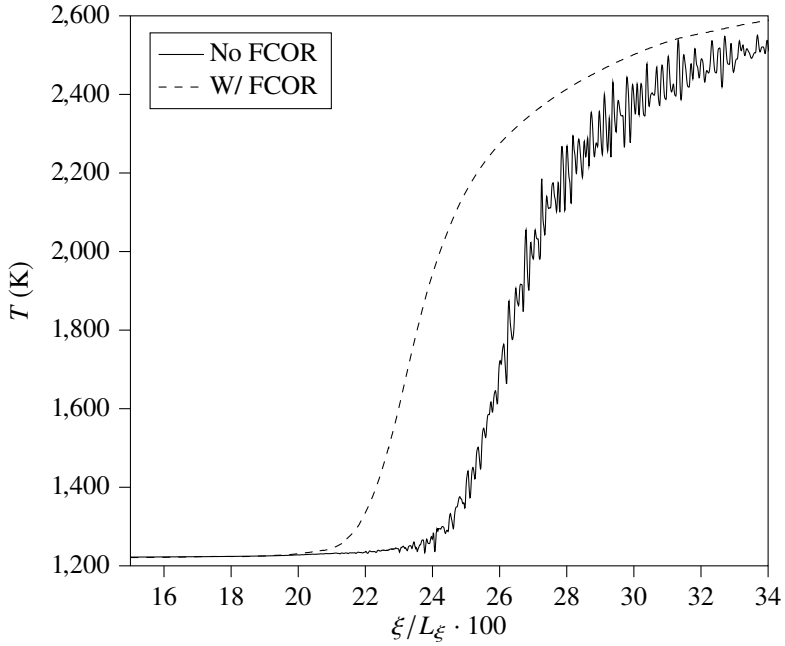


Figure 4.5: Shock tube temperature profiles at solution time $t = 11.4$ μs. The profile is taken at a constant value of $\eta = 3\times10^{-4}$.

near the wall is deemed an acceptable trade-off to solve problems involving hypersonic flows with chemical reactions.

Chapter 5

Verification

The goal of this work is to develop a fourth-order accurate, finite-volume algorithm for solving fluid flow problems. Therefore, it is essential to verify the resulting algorithm is fourth-order accurate for smooth flows. This chapter will verify the order of accuracy of the algorithm by measuring the solution errors for the multispecies Couette flow, Gaussian acoustic pulse, mass diffusion bubble, and shear problem. Finally, freestream preservation and conservation of species concentrations are verified.

5.1 Order of Accuracy

Algorithm accuracy is verified by examining the solution errors. Errors are measured with the L_∞, L_1, and L_2-norms of the difference between the "exact" solution and the numerical solution. The "exact" solution is the analytical solution, when one exists, or determined using Richardson extrapolation. The norms are computed using

$$L_m = \begin{cases} \max(|\phi_{\boldsymbol{i}}^{\text{exact}} - \phi_{\boldsymbol{i}}|), & \text{if } m = \infty\,, \\ \left(\sum_{\boldsymbol{i}} |\phi_{\boldsymbol{i}}^{\text{exact}} - \phi_{\boldsymbol{i}}|^m\right)^{1/m} \left(\prod_{d=1}^{D} N_d \Delta\xi_d\right)^{-1/m}, & \text{otherwise}\,, \end{cases} \tag{5.1}$$

where $\phi_{\boldsymbol{i}}$ is the numerical solution of any conservative variable (such as ρ, ρu, or ρe), $\phi_{\boldsymbol{i}}^{\text{exact}}$ is the exact solution of the conservative variable, and N_d is the number of cells in the dth direction.

The order of accuracy n, as in $\mathcal{O}(\Delta\xi^n)$, is

$$n = \log\left(\frac{L_m(r\Delta\xi)}{L_m(\Delta\xi)}\right) / \log(r)\,, \tag{5.2}$$

where $L_m(\Delta\xi)$ is the m-norm of the error from a solution with a mesh of resolution $\Delta\xi$, and r is the refinement ratio between the meshes. For verification presented below, $r = 2$.

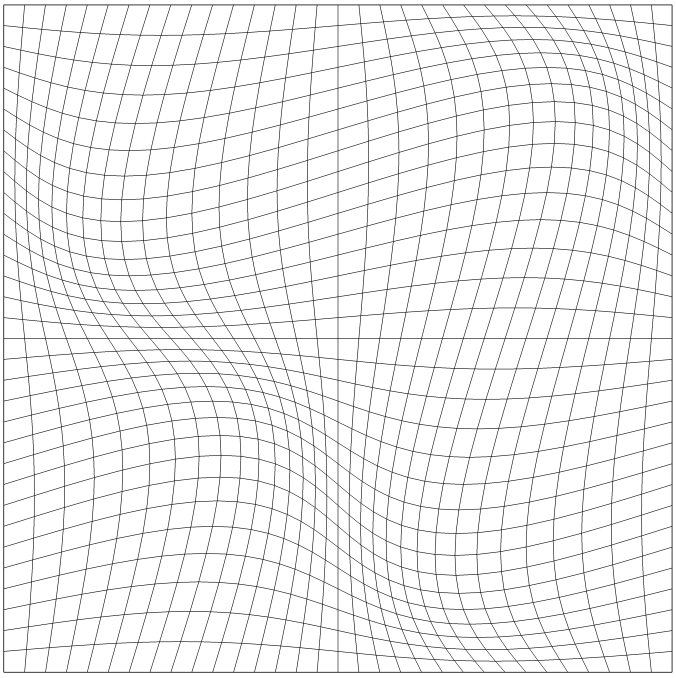


Figure 5.1: Demonstration of a warped mesh with $\vec{S} = (0.075, 0.075)$.

The algorithm uses curvilinear coordinate transformations for solving problems where geometry is not confined to a rectangular domain or a non-Cartesian mesh is present. To verify and validate the mapping functionality, some test problems are run on a warped mesh. This warped mesh is created using a mapping function [7] described by

$$x_d = \xi_d + S_d \prod_{p=1}^{D} \sin\left(\frac{2\pi\xi_p}{L_p}\right),\ d = 1, \ldots, D\,, \tag{5.3}$$

where L_p is the length of the domain in the pth direction and S_d is the scaling factor in the dth direction. The scaling factor must be bounded by $0 \leq 2\pi S_d \leq L_d$ for all directions to ensure the mesh does not tangle. A warped mesh with $\vec{S} = (0.075, 0.075)$ is demonstrated in Figure 5.1.

For convenience, acronyms are used to denote mesh types. "PC" stands for the precoarse mesh, "C" for coarse, "M" for medium, "F" for fine, and "PF" for postfine. The convergence rates between two adjacent meshes are denoted as "PC-C", "C-M", etc.

All solutions in this chapter utilize the PPM limiter and artificial viscosity and calculates density from the limited temperature and pressure, as detailed in Section 4.1.2. Additionally, the

stability techniques presented in Chapter 4 are verified to ensure they are not engaged in regions of smooth flow and do not negatively impact the accuracy of the underlying numerical algorithm. If this is the case, the limiter is determined to be well devised. Solutions that employ the face value limiting described in Section 4.1 will be labeled as "FVL". Results labeled as "DCS" employ the DC flattening and FCOR described in Section 4.2 as well as the face value limiting. Solutions that only use the PPM limiter and artificial viscosity for stability are labeled with "PPM".

5.1.1 Multispecies Couette Flow

Couette flow is used to verify the order of accuracy of the molecular viscous operators. Coutte flow is defined as flow between two parallel no-slip walls; one wall is stationary and another wall is moving at a fixed tangential velocity. The Couette flow has a time-dependent analytical solution that is used as the "exact" solution in Equation (5.1). Eventually, the flow reaches steady state.

The domain length $L_x \times L_y = 0.5456\,\mathrm{mm} \times 0.5456\,\mathrm{mm}$. The domain boundaries are periodic in the x-direction and adiabatic walls in the y-direction. The fluid is a mixture of $c_{\mathrm{O}_2} = 0.233$ and $c_{\mathrm{N}_2} = 0.767$. The lower wall is stationary, and the velocity of the upper wall, denoted as U_w, is calculated using

$$U_w = \frac{\mathrm{Re}\mu}{\rho L_y},$$

where $\mu = 1.7894 \times 10^{-5}\ \mathrm{kg/(m \cdot s)}$, $\rho = 1\ \mathrm{kg/m^3}$, and $\mathrm{Re} = 1000$. The mesh is warped according to Equation (5.3), where $S_d = 0.075$. Only diffusive physics are considered in this test. The fluid is initialized to the analytical solution at time $t = 0.2\,\mathrm{ms}$, and the final solution time is $t = 0.20108\,\mathrm{ms}$.

The time step for each consecutively coarsened mesh solution is scaled based on the resolution change, so the coarse grid solution time step is $\Delta t_{\mathrm{C}} = \Delta t_{\mathrm{PF}} \Delta x_{\mathrm{C}} / \Delta x_{\mathrm{PF}}$. The PC mesh size is 64^2 and the F mesh size is 512^2.

The convergence rates for the momentum are shown in Table 5.1. The table lists convergence rates between four consecutively refined grids with refinement ratios of two between each grid. The convergence rates for all norms and solutions are above 4, indicating the discrete operators

Table 5.1: Convergence rates between consecutively refined grid resolutions for the Couette flow based on the L_∞, L_1, and L_2-norms.

		PPM			FVL			DCS		
Variable	Grids	L_∞	L_1	L_2	L_∞	L_1	L_2	L_∞	L_1	L_2
$J\rho u$	PC-C	4.13	4.89	4.71	4.13	4.87	4.71	4.13	4.87	4.71
	C-M	4.94	4.85	5.36	4.94	4.82	5.36	4.94	4.82	5.36
	M-F	5.01	4.37	5.16	5.02	4.35	5.14	5.02	4.35	5.14
$J\rho v$	PC-C	4.62	4.68	5.03	4.54	4.47	4.78	4.54	4.47	4.78
	C-M	4.99	4.21	4.68	4.86	4.14	4.45	4.86	4.14	4.45
	M-F	4.33	4.09	4.28	4.50	4.07	4.18	4.50	4.07	4.18

for momentum diffusion are fourth-order accurate. The limiting and stability techniques have not interfered with the order of accuracy.

5.1.2 Gaussian Acoustic Pulse

The Gaussian acoustic pulse problem verifies the order of accuracy of the thermally perfect convective operators. The density, temperature, and pressure are each initialized as a smooth Gaussian profile. The resulting pressure wave introduces a smooth extrema. It is important that instances of smooth extrema are preserved throughout the limiting and flattening process. Also, the new stability techniques should not impair the preservation of the smooth extrema.

The domain length is $L_x \times L_y = 1\,\mathrm{m} \times 1\,\mathrm{m}$. The domain boundaries are periodic all directions. The fluid is a mixture of $c_{\mathrm{O}_2} = 0.5$, $c_{\mathrm{He}} = 0.25$, and $c_{\mathrm{H}_2} = 0.25$. The density is initialized using

$$\rho_{\boldsymbol{i}} = \rho_0 + \begin{cases} \Delta\rho \exp\left(-16 r_{\boldsymbol{i}}^2\right) \cos^6\left(\pi r_{\boldsymbol{i}}\right), & \text{if } r_{\boldsymbol{i}} \leq 1/2\,, \\ 0, & \text{otherwise}\,, \end{cases} \tag{5.4}$$

where $r_{\boldsymbol{i}}$ is the radius from the center at cell $\boldsymbol{i}$, given by

$$r_{\boldsymbol{i}} = \sqrt{\bar{x}_{\boldsymbol{i}}^2 + \bar{y}_{\boldsymbol{i}}^2}\,, \quad (\bar{x}_{\boldsymbol{i}}, \bar{y}_{\boldsymbol{i}}) = (x_{\boldsymbol{i}} - x_c, y_{\boldsymbol{i}} - y_c)\,. \tag{5.5}$$

The center of the pulse is at the center of the domain, $(x_c, y_c) = (0.5, 0.5)$. The pressure is initialized using the isentropic relation

$$p_{\boldsymbol{i}} = \rho_0 R T_0 \left(\frac{\rho_{\boldsymbol{i}}}{\rho_0} \right)^{\gamma_0} , \tag{5.6}$$

where γ_0 is the initial specific heat ratio for the fluid mixture at T_0. The initial velocity is zero. The initial values are $\rho_0 = 1.2\,\mathrm{kg/m^3}$, $\Delta\rho = 0.1\,\mathrm{kg/m^3}$, and $T_0 = 300\,\mathrm{K}$. Only inertial physics are considered in the solutions. The final solution time for the tests is $t = 0.1\,\mathrm{ms}$.

Richardson extrapolation is performed to project a more accurate solution based on the numerical solutions obtained at different spatial resolutions. The PF solution is used to extrapolate an "exact" solution for a coarser grid solution. The grid level of interest is designated as GL; GL can be PC, C, M, or F. The "exact" solution for a given GL, $\phi^{\mathrm{exact}}(\mathrm{GL})_{\boldsymbol{i}}$, is evaluated by

$$\phi^{\mathrm{exact}}(\mathrm{GL})_{\boldsymbol{i}} = \frac{r(\mathrm{GL})^P \phi(\mathrm{PF})_{\boldsymbol{i}} - \phi(\mathrm{GL})_{\boldsymbol{i}}}{r(\mathrm{GL})^P - 1} , \tag{5.7}$$

where P is the expected order of accuracy, $\phi(\mathrm{GL})_{\boldsymbol{i}}$ is the numerical solution on the GL, $\phi(\mathrm{PF})_{\boldsymbol{i}}$ is the PF solution averaged down to the GL, and $r(\mathrm{GL})$ is the ratio of the number of cells in the PF grid, $N(\mathrm{PF})$, to the number of cells in the current GL, $N(\mathrm{GL})$, or

$$r(\mathrm{GL}) = \frac{N(\mathrm{PF})}{N(\mathrm{GL})} . \tag{5.8}$$

When Richardson extrapolation is used, the left-hand side of Equation (5.7) becomes the "exact" solution in Equation (5.1). Theoretically, the "exact" solution from Equation (5.7) has a fifth-order error. For solutions that use AMR, the resolution on the refined level of each AMR grid matches the resolution of a corresponding single-level solution. More information is provided by Guzik et al. [9].

The convergence rates are measured at solution time $t = 0.1\,\mathrm{ms}$ and tabulated for density, x-momentum, and energy-density in Table 5.2. The PC mesh size is 64^2, and the PF mesh size is 1024^2. The face value limiting modifies the convergence rates for the PC-C grids but does not diminish the convergence rates for any variable. Is it likely that the face value limiting does not treat the PC solution as smooth. The convergence rates for the all solutions are unaffected by the

Table 5.2: Convergence rates between consecutively refined grid resolutions for the Gaussian acoustic pulse problem based on the L_∞, L_1, and L_2-norms.

		PPM			FVL			DCS		
Variable	Grids	L_∞	L_1	L_2	L_∞	L_1	L_2	L_∞	L_1	L_2
$J\rho$	PC-C	3.98	3.99	3.99	3.98	4.08	4.02	3.97	4.08	4.03
	C-M	4.00	4.02	4.03	3.99	4.01	4.02	3.99	4.01	4.02
	M-F	4.00	4.07	4.09	4.00	4.06	4.08	4.00	4.06	4.08
$J\rho u$	PC-C	3.97	3.97	3.97	3.98	4.20	4.08	3.98	4.20	4.08
	C-M	3.99	3.99	3.99	3.99	3.99	3.99	3.99	3.99	3.99
	M-F	4.01	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
$J\rho e$	PC-C	3.98	3.98	3.98	3.97	4.07	4.02	3.97	4.07	4.02
	C-M	4.00	4.00	4.00	3.99	4.00	4.00	3.99	4.00	4.00
	M-F	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00

DC flattening techniques. This demonstrates that solutions with face value limiting, DC flattening, and FCOR retain fourth-order accuracy in the presence of smooth pressure waves.

5.1.3 Multispecies Mass Diffusion Bubble Problem

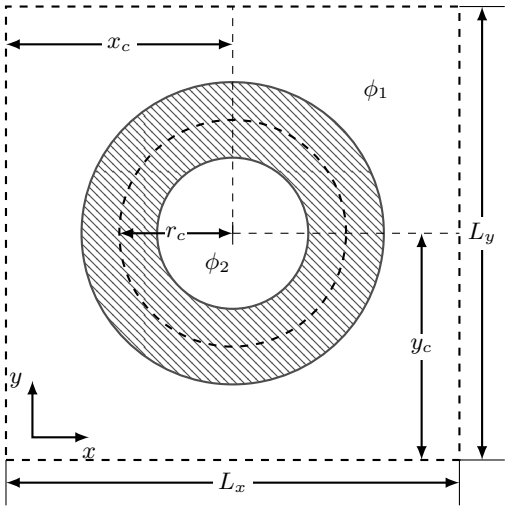


Figure 5.2: Demonstration of the initialization of the two regions for the diffusion bubble problem. The shaded region represents the smooth interface between region 1 and 2.

The mass diffusion bubble problem is used to verify the order of accuracy of the total diffusive operations, including thermal and species diffusion. The inertial and viscous physics are both

modeled. The domain is initialized with a circular bubble of nitrogen and oxygen at a given temperature surrounded by a mixture of oxygen and carbon dioxide at a different temperature.

The initialization is demonstrated in Figure 5.2. The domain length is $L_x \times L_y = 1\,\mathrm{m} \times 1\,\mathrm{m}$, and the domain boundaries are all periodic. In the figure, a circle of radius r_c is centered at location (x_c, y_c). The cross-hatched region indicates the smooth transition from one set of initial values to another. The initial values for given variables in the domain are designated using the subscript 1, and the initial values for the variables inside the circle are designated using the subscript 2. The initialization depends on a smoothing parameter, $\alpha_{\boldsymbol{i}}$, that is calculated using

$$\alpha_{\boldsymbol{i}} = \frac{1}{2}\left(1 + \tanh\left(\frac{\beta(r_c - r_{\boldsymbol{i}})}{L_x}\right)\right), \tag{5.9}$$

where β determines the sharpness of the interface and $r_{\boldsymbol{i}}$ is the radius, given by

$$r_{\boldsymbol{i}} = \sqrt{\bar{x}_{\boldsymbol{i}}^2 + \bar{y}_{\boldsymbol{i}}^2}\,, \quad (\bar{x}_{\boldsymbol{i}}, \bar{y}_{\boldsymbol{i}}) = (x_{\boldsymbol{i}} - x_c, y_{\boldsymbol{i}} - y_c)\,, \tag{5.10}$$

where $(x_c, y_c) = (0.5, 0.5)$ is the location of the center of the bubble. The initial mass fractions at cell $\boldsymbol{i}$ are calculated using

$$(c_n)_{\boldsymbol{i}} = (c_n)_1 + \alpha_{\boldsymbol{i}}\left((c_n)_2 - (c_n)_1\right)\,. \tag{5.11}$$

The initial temperature at cell $\boldsymbol{i}$ is calculated using

$$T_{\boldsymbol{i}} = \left(\frac{\sigma_{\boldsymbol{i}}}{T_1} + \frac{1 - \sigma_{\boldsymbol{i}}}{T_2}\right)^{-1}, \tag{5.12}$$

where

$$\sigma_{\boldsymbol{i}} = \frac{1}{N_s}\sum_{n=1}^{N_s} \frac{(c_n)_{\boldsymbol{i}} - (c_n)_2}{(c_n)_1 - (c_n)_2}\,. \tag{5.13}$$

The mass diffusion coefficient is modeled using Equation (2.30) with a constant Lewis number of $\mathrm{Le} = 0.7$. The domain is initialized to a uniform pressure of p_{atm}. The initialization values are:

$\beta = 80$, $T_1 = 298\,\mathrm{K}$, $T_2 = 310\,\mathrm{K}$, $(c_{\mathrm{O}_2})_1 = (c_{\mathrm{N}_2})_1 = 0.5$, $(c_{\mathrm{CO}_2})_1 = (c_{\mathrm{N}_2})_2 = 0$, $(c_{\mathrm{O}_2})_2 = 0.3$, $(c_{\mathrm{CO}_2})_2 = 0.7$, and $r_c = 0.2\,\mathrm{m}$. The initial density is given by the ideal gas law.

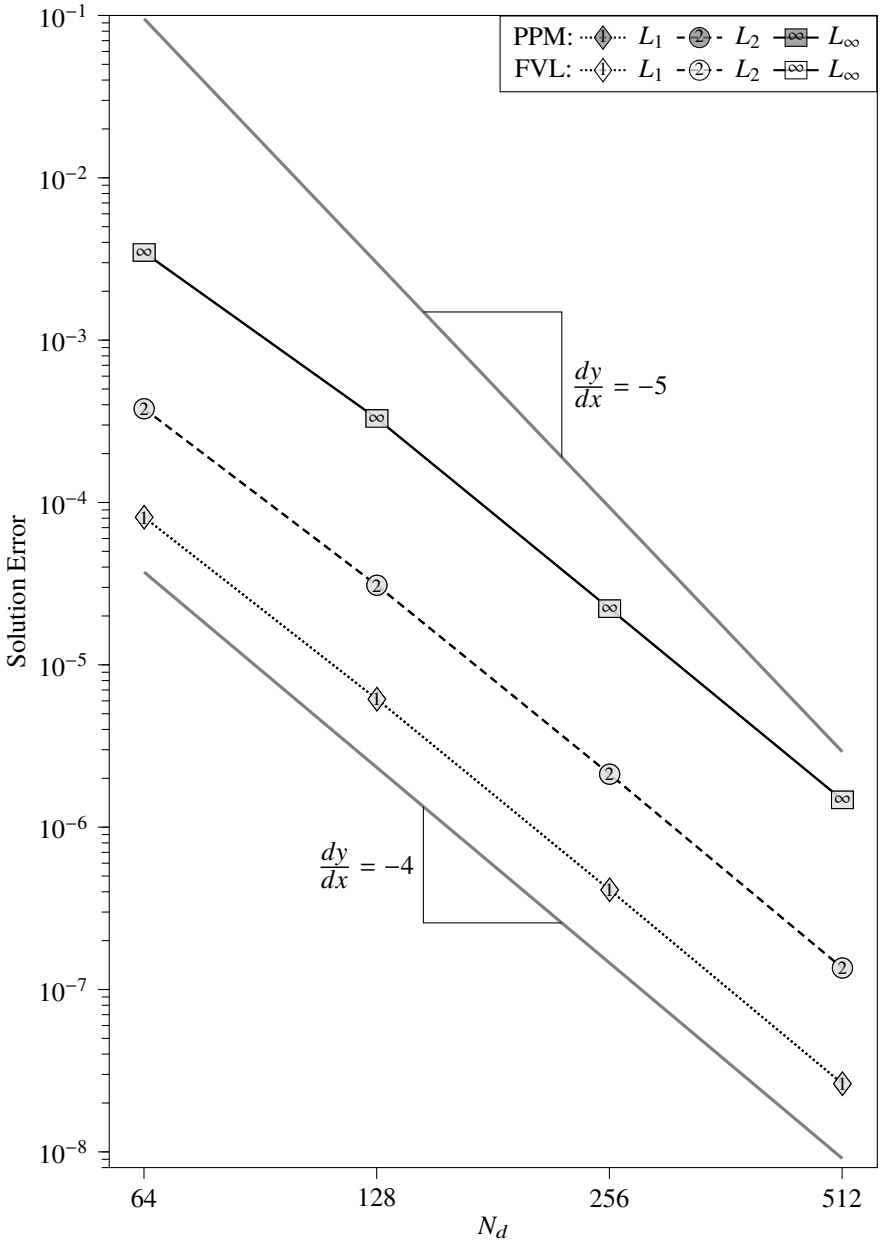


Figure 5.3: Solution accuracy of $J\rho c_{\mathrm{O}_2}$ for the mass diffusion bubble problem with and without face value limiting.

The order of accuracy is determined using a PC mesh of 64^2 and a PF mesh size of 1024^2. The "exact" solution is computed following the Richardson extrapolation methodology. The solution

error of $J\rho c_{\mathrm{O}_2}$ for the mass diffusion bubble problem at solution time $t = 0.25\ \mathrm{ms}$ is shown in Figure 5.3. The error for solutions without and with face value limiting are shown to be identical for all grid levels and norms.

Table 5.3: Convergence rates between consecutively refined grid resolutions for the mass diffusion bubble problem based on the L_∞, L_1, and L_2-norms.

		PPM			FVL			DCS		
Variable	Grids	L_∞	L_1	L_2	L_∞	L_1	L_2	L_∞	L_1	L_2
$J\rho$	PC-C	3.39	3.71	3.60	3.39	3.71	3.60	3.39	3.71	3.60
	C-M	3.89	3.90	3.87	3.89	3.90	3.87	3.89	3.90	3.87
	M-F	3.91	3.97	3.97	3.91	3.96	3.97	3.91	3.96	3.97
$J\rho u$	PC-C	2.15	2.80	2.57	2.08	2.76	2.55	2.07	2.76	2.55
	C-M	2.21	2.82	2.63	1.35	2.52	2.38	1.34	2.52	2.38
	M-F	3.05	3.89	3.59	1.11	2.77	2.60	1.12	2.77	2.59
$J\rho e$	PC-C	3.40	3.73	3.62	3.40	3.73	3.62	3.40	3.73	3.62
	C-M	3.89	3.90	3.87	3.89	3.90	3.87	3.89	3.90	3.87
	M-F	3.91	3.97	3.97	3.91	3.97	3.97	3.91	3.97	3.97
$J\rho c_{\mathrm{O}_2}$	PC-C	3.40	3.72	3.61	3.40	3.72	3.61	3.40	3.72	3.61
	C-M	3.89	3.90	3.87	3.89	3.90	3.87	3.89	3.90	3.87
	M-F	3.91	3.97	3.97	3.91	3.97	3.97	3.91	3.97	3.97
$J\rho c_{\mathrm{N}_2}$	PC-C	3.41	3.73	3.62	3.41	3.73	3.62	3.41	3.73	3.62
	C-M	3.88	3.90	3.87	3.88	3.90	3.87	3.88	3.90	3.87
	M-F	3.92	3.97	3.97	3.92	3.97	3.97	3.92	3.97	3.97
$J\rho c_{\mathrm{CO}_2}$	PC-C	3.40	3.73	3.62	3.40	3.73	3.62	3.40	3.73	3.62
	C-M	3.89	3.90	3.87	3.89	3.90	3.87	3.89	3.90	3.87
	M-F	3.91	3.97	3.97	3.91	3.97	3.97	3.91	3.97	3.97

The convergence rates for the density, momentum, energy, and species concentrations are shown in Table 5.3. The table lists convergence rates between three consecutively refined grids with refinement ratios of two between each grid. The convergence rates for all variables approach 4 as the mesh is refined, except momentum. The convergence rates for momentum do not demonstrate fourth-order convergence, likely due to the difference in magnitude between the pressure and the velocity; more specifically, the error in the pressure term dominates the error in the momentum since the momentum flux contains a pressure term.

Table 5.4: Convergence rates between consecutively refined grid resolutions for the shear flow problem based on the L_∞, L_1, and L_2-norms.

			PPM			FVL			DCS		
AMR	Variable	Grids	L_∞	L_1	L_2	L_∞	L_1	L_2	L_∞	L_1	L_2
None	$J\rho u$	PC-C	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
		C-M	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
		M-F	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
	$J\rho v$	PC-C	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
		C-M	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
		M-F	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
	$J\rho e$	PC-C	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
		C-M	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
		M-F	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00	4.00
1-Level	$J\rho u$	C-M	3.50	3.97	3.96	3.48	3.97	3.95	3.48	3.97	3.95
		M-F	3.82	3.97	3.95	3.87	3.97	3.94	3.87	3.97	3.94
	$J\rho v$	C-M	3.50	3.97	3.96	3.48	3.97	3.95	3.48	3.97	3.95
		M-F	3.82	3.97	3.95	3.87	3.97	3.94	3.87	3.97	3.94
	$J\rho e$	C-M	4.05	3.98	4.00	4.04	3.98	4.00	4.04	3.98	4.00
		M-F	3.82	3.97	3.96	3.83	3.97	3.97	3.83	3.97	3.97

5.1.4 Shear Problem

Verifying the order of accuracy of the shear problem ensures the algorithm can accommodate strong gradients present at AMR interfaces. The domain is initialized to a uniform pressure and temperature of $p = 1\ \mathrm{atm}$ and $T = 300\ \mathrm{K}$, respectively. The initial fluid comprised of mass fractions $c_{\mathrm{O}_2} = 0.233$ and $c_{\mathrm{N}_2} = 0.767$. The velocity is initialized using

$$u(x, y) = 100 \cos(2\pi y)\ ,\ v(x, y) = 100 \cos(2\pi x)\ . \tag{5.14}$$

The mesh is warped according to Equation (5.3), where $S_d = 0.1$. When AMR is used in the solution, the grid on the finer level is fixed in space from $(0.25, 0.25)$ to $(0.75, 0.75)$. The intention is for large gradients to occur at the AMR interfaces. The refinement ratio between levels is 2 for solutions that use AMR.

The convergence rates for x-momentum, y-momentum, and energy-density are measured at solution time $t = 44.8\ \mu\mathrm{s}$ and tabulated in Table 5.4. The top of the table compares the convergence

rates between solutions for single-level solutions, i.e., no AMR. The bottom of the table compares convergence rates for solutions that use an additional refinement level. The convergence rates on the solutions with AMR are smaller than the single-level solution. As described by Guzik et al. [9], a loss of up to one-order of accuracy can occur in coarse cells adjacent to the interfaces between coarse and fine levels; this impact is made more apparent when the interface is exposed to strong gradients. Nonetheless, the momentum order of accuracy from the L_1 and L_2-norms for solutions with AMR converge to ≈ 3.95, and the L_∞-norm continues to increase from 3.5 to 3.82 as the grid is refined. Similarly, the AMR solutions with the stability techniques increase from 3.48 to 3.87. This demonstrates that the face value limiting, DC flattening, and FCOR do not impair convergence to fourth-order accuracy for nonlinear flow solutions on warped grids with or without AMR.

5.2 Freestream Preservation

Freestream preservation must be maintained when mapping and AMR are employed. The freestream preservation test is conducted on a doubly periodic domain initialized with a uniform velocity, density, pressure, and species mass fractions. To fully test freestream preservation, a warped mapping with two levels of moving AMR. The base grid is 64^2. Over this period of time, the refinement region makes one circular rotation in the computational space about the center of the domain.

Table 5.5: Comparison of the volume-averaged L_1-norm of initial and final states of a multispecies freestream case, and the calculated solution difference.

Var	Initial	Final	Difference
ρ	1.22500000000000120E+0	1.22500000000000098E+0	0.00000000000000022E+0
ρu	2.45000000000000079E–1	2.44999999999982288E–1	0.00000000000017791E–1
ρv	2.45000000000000079E–1	2.44999999999979318E–1	0.00000000000020761E–1
ρe	1.10183777676502243E+5	1.10183777676502243E+5	0.00000000000000000E+5
ρc_{O_2}	6.12500000000000600E–1	6.12500000000000488E–1	0.00000000000000111E–1
ρc_{N_2}	6.12500000000000600E–1	6.12500000000000488E–1	0.00000000000000111E–1

For each of the conservative variables, the initial state and the final state at the chosen run time are compared using an L_1-norm. The difference between the values is computed to quantify error present in the solution, since the freestream case will, ideally, have no change in states. Table 5.5 clearly demonstrates that the freestream condition is preserved. The difference between the L_1-norms the solution time are all close to machine zero. This verifies that freestream preservation is maintained.

5.3 Conservation of Species Concentration

One important benefit to using FVMs is that they ensure conservation. However, the species concentration corrections described in Section 3.4.3 can violate the conservation of species concentrations. Therefore, minimizing the impact of the species concentration correction is of great importance. To test this, the shock tube problem described in Section 4.1.2 is used to evaluate the stability techniques impact on the conservation of the species concentration. The test does not use any reaction source terms, and the boundaries are walls in the x-direction and periodic in the y-direction to ensure no mass, energy, or species are added or removed at the boundaries.

Table 5.6: The percent change of the L_1-norm of the conservative solution variables from the initial state to the state at solution time $t = 49$ µs for a shock tube problem.

Variable	Test 1	Test 2	Test 3	Test 4
$J\rho$	3.402×10^{-12}	3.540×10^{-12}	2.891×10^{-12}	2.911×10^{-12}
$J\rho e$	7.118×10^{-12}	8.088×10^{-12}	7.549×10^{-12}	7.075×10^{-12}
$J\rho c_{\mathrm{N}_2}$	1.318×10^{-1}	6.312×10^{-3}	4.165×10^{-13}	3.844×10^{-13}
$J\rho c_{\mathrm{O}_2}$	1.318×10^{-1}	6.312×10^{-3}	1.147×10^{-12}	1.094×10^{-12}
$J\rho c_{\mathrm{He}}$	8.796×10^{-2}	4.211×10^{-3}	2.115×10^{-12}	3.131×10^{-12}

To measure conservation, the L_1-norm of the conserved variables $\langle J\mathbf{U}\rangle$ are calculated at the start of the simulation and at solution time $t = 49$ µs for four different test cases. The values listed in Table 5.6 are the difference between the L_1-norms of the variable initially and the at the end of the run as a percentage of the initial L_1-norm. The test cases use the different stability techniques as follows: Test 1 does not use face value limiting, DC flattening, or FCOR; Test 2 uses face value

limiting only; Test 3 uses face value limiting and DC flattening; and Test 4 uses face value limiting, DC flattening, and FCOR. All test cases limit temperature and solve for density as recommended in Section 4.1.2. Additionally, all test cases utilize the species correction detailed in Section 3.4.3.

The table demonstrates that mass and energy are always conserved because the percent change approaches numerical error for all test cases. However, the species concentrations for individual species are evidently not conserved in Tests 1 and 2. This loss of conservation is due repeated corrections of the species concentrations with Equation (3.61). The species corrections are used in Tests 3 and 4, but the effects appear to be negligible. This test demonstrates how the DC stability techniques can significantly improve the conservation of species concentrations.

Chapter 6

Validation

6.1 Multispecies Shock Tube

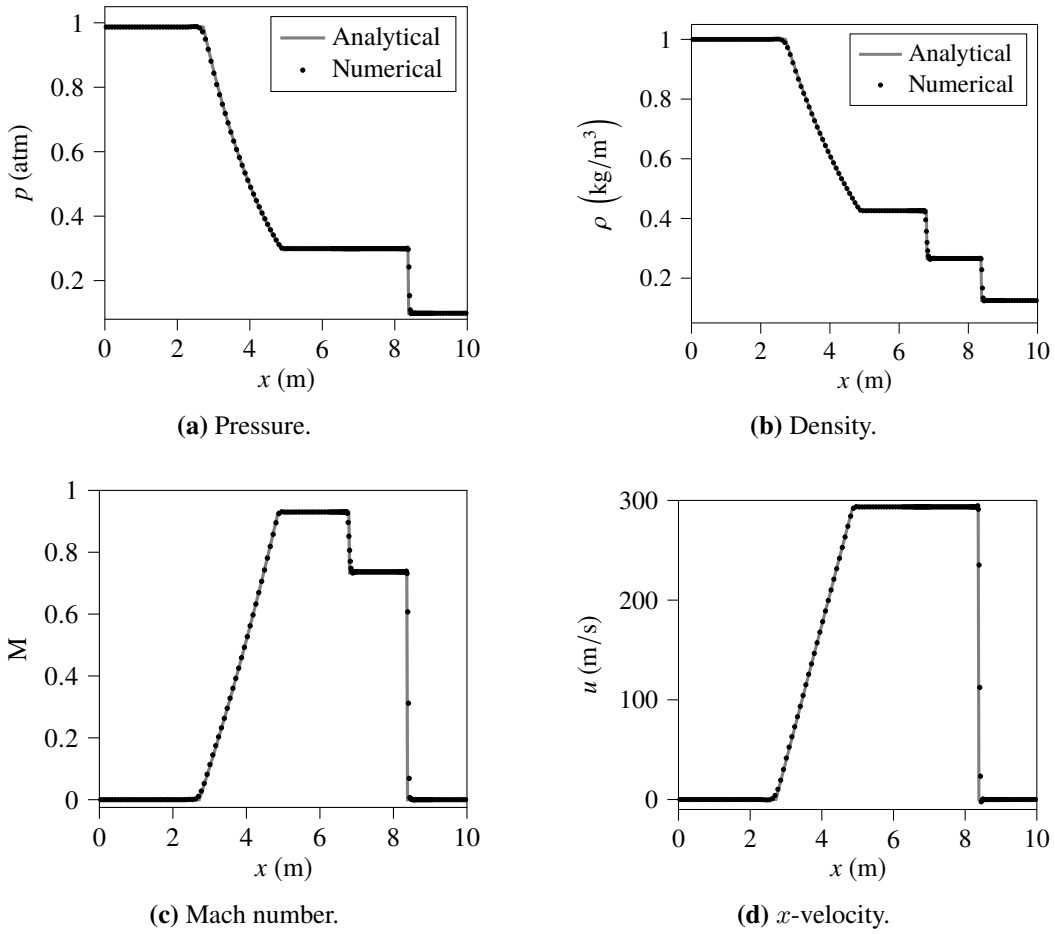
The 1-D shock tube problem is a classic case for validating convective flux operators because it contains strong discontinuities and has an analytical solution. The problem set-up consists of a long tube of a stagnant gas or gaseous mixture is separated into two regions of equal size by a thin diaphragm. The pressure in one region is higher than in the other. When the diaphragm is removed, the high pressure gas expands through a rarefaction wave and flows into the low pressure region; the compression of the low pressure gas creates a shock wave. The shock tube case in the present work uses a $10\ \mathrm{m} \times 0.625\ \mathrm{m}$ domain. The right portion of the domain is initialized as the low pressure region with $\rho_R = 0.125\ \mathrm{kg/m^3}$ and $p_R = 1 \times 10^4\ \mathrm{Pa}$, and the left portion is initialized as the high pressure region with $\rho_L = 8\rho_R$ and $p_L = 10p_R$. The fluid in the domain is a uniform mixture of $c_{\mathrm{O}_2} = 0.233$ and $c_{\mathrm{N}_2} = 0.767$.

A base grid of 128×8 cells is created. Two additional refinement levels, each with refinement ratios of 2, are applied. The grid is dynamically refined using Equation (3.65) based on gradients of density with $\delta_t = 0.05$. Sub-cycling is used during the solution. The boundaries are periodic in the y-direction and slip walls in the x-direction.

The test case is run to solution time $t = 6.1\ \mathrm{ms}$ and compared with the analytical solution. Figure 6.1 shows good agreement between the analytical and numerical solutions with slight oscillations occurring at the discontinuities, validating the algorithm accurately captures shock physics.

6.2 Multispecies Shock Box

The shock box extends the shock tube problem to two dimensions; multidimensional shocks convect and interact within the domain. The domain length is $L_x \times L_y = 1\ \mathrm{m} \times 1\ \mathrm{m}$. Initially, the fluid in the computational domain is a quiescent, uniform mixture of $c_{\mathrm{O}_2} = 0.233$ and $c_{\mathrm{N}_2} =$



(a) Pressure.

(b) Density.

(c) Mach number.

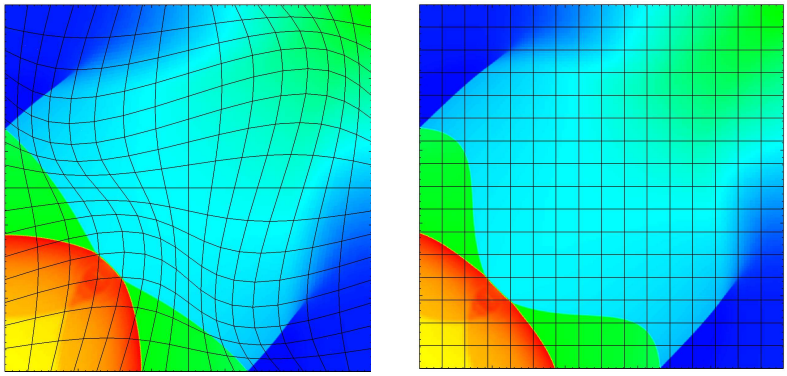
(d) x-velocity.

Figure 6.1: Shock tube solutions at $t = 6.1$ ms.

0.767. The initial state in the lower left quarter of the domain, designated with the subscript L, is $\rho_L = 1.225\,\mathrm{kg/m^3}$ and $p_L = p_{\mathrm{atm}}$ and $\rho_U = 4\rho_L$ and $p_U = 4p_L$ in the rest of the domain. The boundaries are slip walls.

The computational mesh uses a base grid of 128×128 with two additional refinement levels, each with refinement ratios of 2. The grid is dynamically refined using Equation (3.65) based on gradients of density and pressure with $\delta_t = 0.1$. Again, sub-cycling is used during the solution. To validate the mapping functionality, the mesh is warped according to Equation (5.3) with $S_d = 0.075$.

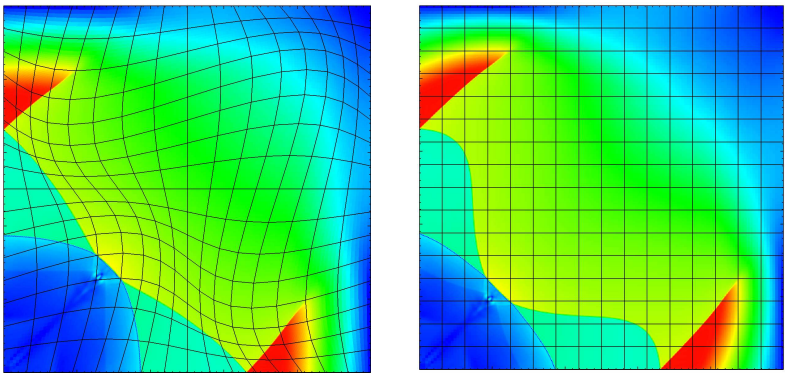
The test case is run to solution time $t = 2$ ms. At this solution time, the shocks in the x and y-directions have converged in the lower left corner and reflected back into the domain. Figure 6.2



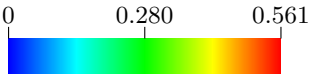
(a) Physical space. **(b)** Computational space.

(c) Pressure legend, (atm).

Figure 6.2: Pseudo-color plot of pressure in physical and computational space at solution time $t = 2\ \mathrm{ms}$ with overlay to demonstrate the mesh.



(a) Physical space. **(b)** Computational space.



(c) Mach number legend.

Figure 6.3: Pseudo-color plot of Mach number in physical and computational space at solution time $t = 2\ \mathrm{ms}$ with overlay to demonstrate the mesh.

shows the pressure solution with an overlay of the mesh. Figures 6.2a and 6.2b show the solutions in physical and computational spaces, respectively. The mesh in Figure 6.2a demonstrates the warping imposed by Equation (5.3). Similarly, Figure 6.3 shows the solution of the Mach number in physical and computational space. The solutions agree well with literature results [39].

6.3 Multispecies Lid-Driven Cavity Flow

The lid-driven cavity flow problem is a classic case that demonstrates intricate flow physics manifested by multiple counter-rotating and recirculating regions. The domain is a square surrounded by four no-slip walls. The upper wall in the y-direction moves at a constant velocity, denoted as U_w, in the x-direction. The fluid mixture in the left half of the domain consists of $c_{\mathrm{O}_2} = 0.233$ and $c_{\mathrm{N}_2} = 0.767$, and the fluid mixture in the right half of the domain consists of $c_{\mathrm{O}_2} = 0.1$ and $c_{\mathrm{N}_2} = 0.9$. The velocity of the top wall is calculated from a given Mach number and initial speed of sound using

$$U_w = \mathrm{M}\sqrt{\gamma p_0/\rho_0}\,. \tag{6.1}$$

Once the wall velocity is calculated, the length of the domain is calculated from a given Reynolds number, initial density, and dynamic viscosity using

$$L_x = L_y = \frac{\mu \mathrm{Re}}{U_w \rho_0}\,. \tag{6.2}$$

The initial values for the test case presented here are $\mathrm{M} = 0.1$, $\mathrm{Re} = 1000$, $\mu = 1.7894\times10^{-5}$ $\mathrm{kg}/(\mathrm{m}\cdot\mathrm{s})$, $\kappa = 2.5326\times10^{-2}$ $\mathrm{W}/(\mathrm{m}\cdot\mathrm{K})$, $\rho_0 = 1\,\mathrm{kg}/\mathrm{m}^3$, and $p_0 = p_{\mathrm{atm}}$. The initial temperature is calculated using the ideal gas law. The mass diffusion is modeled using Equation (2.30) with a constant Lewis number of $\mathrm{Le} = 1$.

The base grid is 128×128 with two additional refinement levels, each with refinement ratios of 2. The refined levels are fixed in space and illustrated in Figure 6.4. The lengths in Figure 6.4 are $L_1 = 0.85L_y$, $L_2 = 0.9L_y$, and $L_3 = 0.125L_y$. The mesh is warped according to Equation (5.3) with $S_d = 0.075$.

The numerical results are compared to the literature data by Ghia et al. [40]. Figure 6.5 shows the profile of normalized v taken across the domain at a constant $y = L_y/2$, and Figure 6.6 is the profile of normalized u taken across the domain at a constant $x = L_x/2$. The profiles show good agreement with the literature, validating the convective and diffusive operations within Chord.

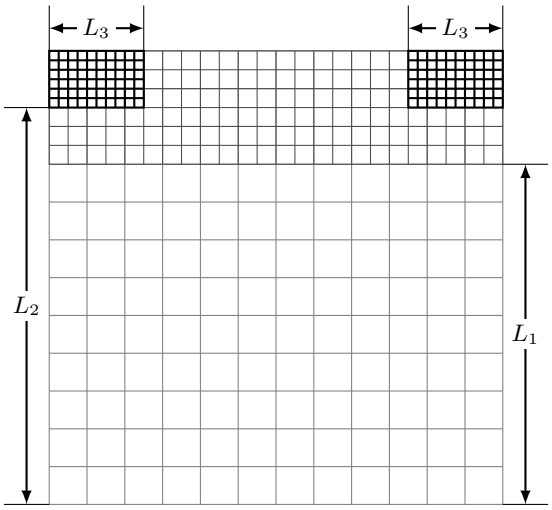


Figure 6.4: Illustration of the mesh refinement for the lid-driven cavity flow problem.

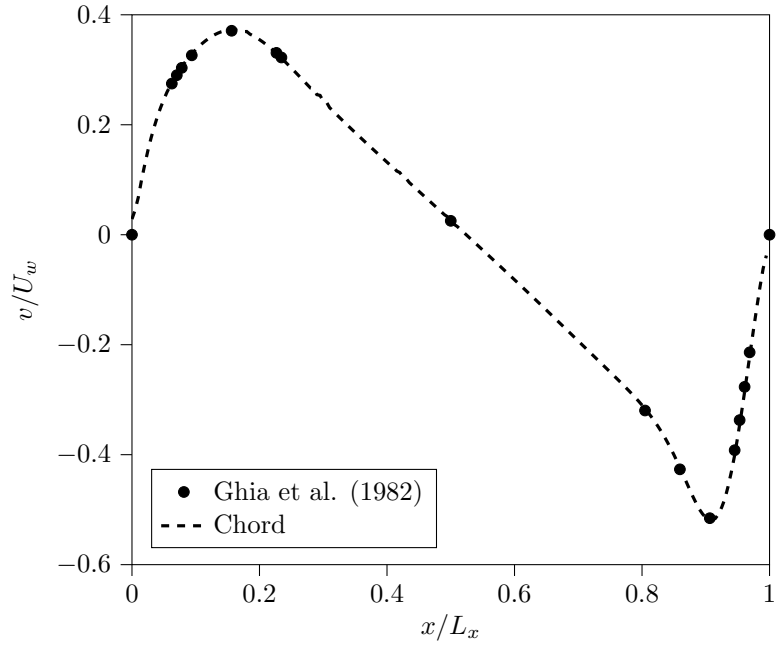


Figure 6.5: Normalized v profile in the x-direction.

6.4 Advection of a Reacting Front

The 1-D advection of a reacting H_2-O_2 front problem involves the advection of a plateau of H_2 through a domain of O_2 at a higher temperature. The species mass fractions are initialized using

$$c_{H_2} = \frac{1}{2}\left(1 + \tanh\left(\beta\left(\frac{L_p}{2} - |x - x_0|\right)\right)\right), \tag{6.3}$$

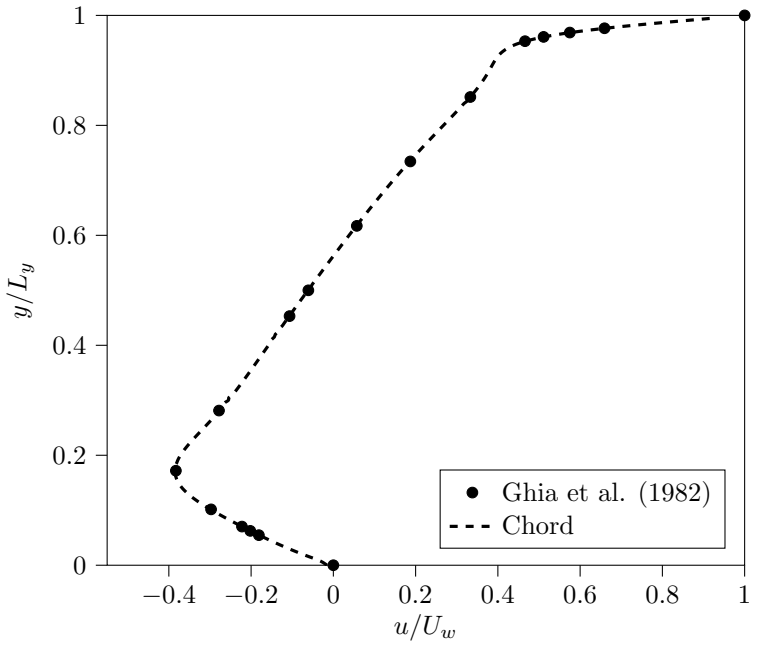


Figure 6.6: Normalized u profile in the y-direction.

and

$$c_{\mathrm{O}_2} = 1 - c_{\mathrm{H}_2} \,, \tag{6.4}$$

where β determines the sharpness of the front, x_0 is the location of the center of the hydrogen plateau, and L_p is the width of the hydrogen plateau. The density is initialized using

$$\rho = \left(\frac{c_{\mathrm{H}_2} R_{\mathrm{H}_2} T_{\mathrm{H}_2} + c_{\mathrm{O}_2} R_{\mathrm{O}_2} T_{\mathrm{O}_2}}{p_{\mathrm{atm}}} \right)^{-1} . \tag{6.5}$$

The problem parameters for this test case are $\beta = 80\,\mathrm{cm}^{-1}$, $x_0 = 3.7\,\mathrm{cm}$, $L_p = 0.6\,\mathrm{cm}$, $T_{\mathrm{H}_2} = 1000\,\mathrm{K}$, and $T_{\mathrm{O}_2} = 2000\,\mathrm{K}$. The entire domain is initialized to p_{atm}. The solutions models the reactions with the 9 species, 19-step reaction mechanism from Billet [38].

The base grid has a grid spacing of $\Delta x = 0.02\,\mathrm{cm}$ with three additional refinement levels, each with refinement ratios of 2. The grid is dynamically refined using Equation (3.65) based on gradients of temperature with $\delta_t = 0.05$.

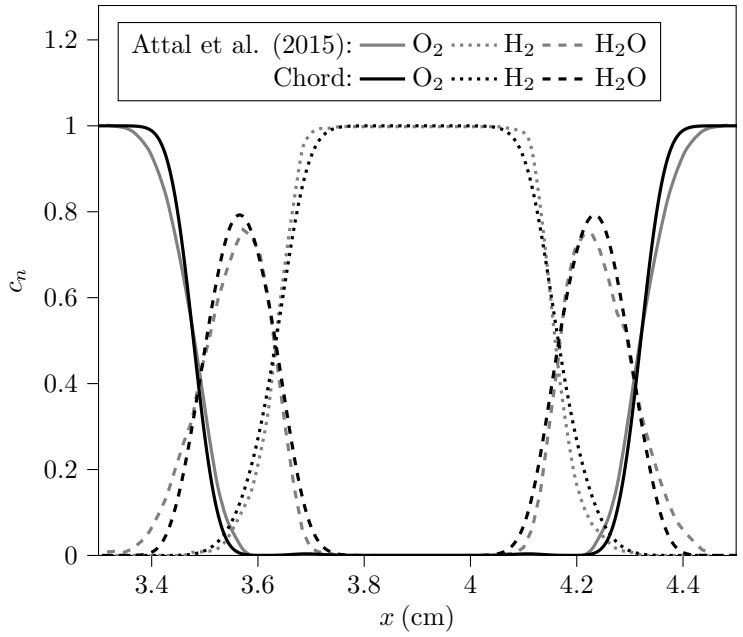
The numerical solution from Chord is compared to the numerical solution from Attal et al. [41]. The algorithm utilized by Attal et al. is called FLASH. First, it is worth noting that Chord and

FLASH do not use all the same fluid assumptions. Chord assumes the fluid is thermally perfect, whereas FLASH models the thermodynamic properties using a gamma-law fluid assumption. The gamma-law fluid model approximates the specific heats and specific heat ratio based on species and temperature. Additionally, FLASH calculates the mass diffusion coefficient for this test case by assuming all species interactions mimic a binary H_2-O_2 system and models the dynamic viscosity with the semi-empirical formula from Wilke [42]. Chord, however, uses the mixture-based formula in Equation (2.26) and does not take into account binary species interactions when modeling the mass diffusion coefficient. FLASH calculates the thermal conductivity from the mass diffusion coefficients and species specific Lewis numbers.

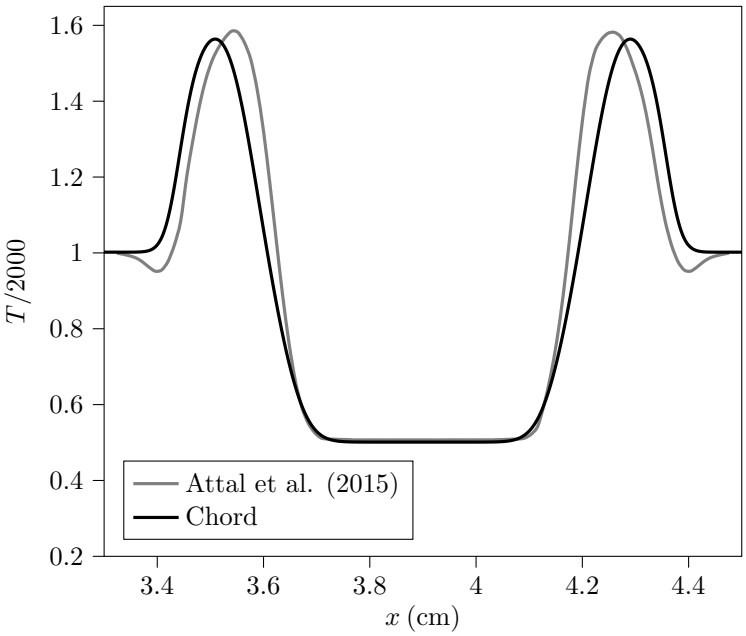
Species mass fractions and temperature profiles at $t = 100$ μs are compared in Figure 6.7. Overall, the profile shape and interface locations are in good agreement with the literature data. However, Figure 6.7a shows that Chord predicts a greater amount of O_2 and H_2O near the leading and trailing fronts and predicts less H_2 at the edges of the plateau in the center region. Figure 6.7b shows that Chord predicts a smaller maximum temperature at the trailing and leading fronts and a greater length between temperature peaks. The Attal et al. solution shows undershoots in the temperature at the leading and trailing fronts. The solution discrepancies could be due to the different methods for modeling the thermodynamic properties of the fluid.

6.5 Shock Bubble

The shock-driven combustion of a 2-D H_2 bubble is used to validate Chord's capability to solve reacting flows with shock waves on a warped grid. This test case involves a bubble of H_2 in surrounding air, traversing into a planar shock. The problem configuration and boundary conditions, shown in Figure 6.8, follow that by Billet et al. [43] and Attal et al. [41] for validation purposes. The solution uses the 9 species, 19-step reaction mechanism from Billet [38]. The domain is initialized with a circular hydrogen bubble near a Mach 2 planar shock in air. Figure 6.8 shows the domain is split into three regions labeled I, II, and III. The H_2 mass fraction is initialized using



(a) Mass fraction profiles.



(b) Temperature profiles.

Figure 6.7: Mass fraction and temperature profiles for the reacting flame front problem at solution time $t = 100$ µs.

$$c_{\mathrm{H}_2} = \frac{1}{2}\left(1 + \tanh\left(\frac{r_c - r}{\beta}\right)\right), \quad r = \sqrt{(x - x_c)^2 + (y - y_c)^2}, \tag{6.6}$$

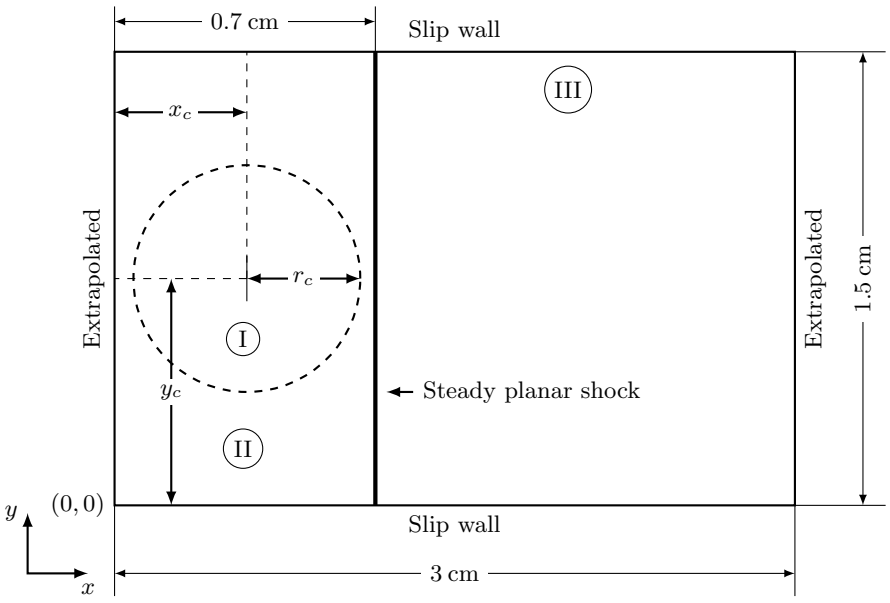


Figure 6.8: Diagram of the shock bubble configuration.

where β determines the sharpness of the interface and is set to $3\times10^{-3}\ \mathrm{cm}^{-1}$ for this case. The characteristics of these regions, denoted by the subscript for the corresponding region, are:

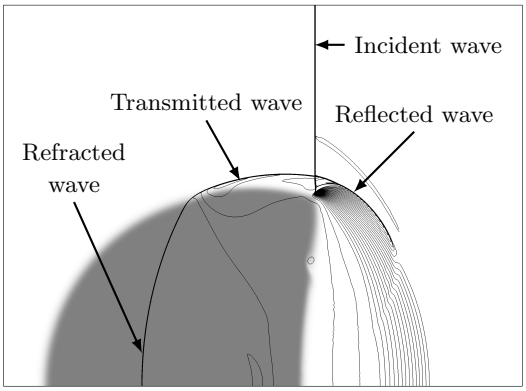
- $(c_{\mathrm{H_2}})_{\mathrm{I}} = 1$, $(c_{\mathrm{O_2}})_{\mathrm{II}} = (c_{\mathrm{O_2}})_{\mathrm{III}} = 0.233$, $(c_{\mathrm{N_2}})_{\mathrm{II}} = (c_{\mathrm{N_2}})_{\mathrm{III}} = 0.767$;
- $u_{\mathrm{I}} = u_{\mathrm{II}} = 1.24\times10^5\ \mathrm{cm/s}$, $u_{\mathrm{III}} = 4.34\times10^4\ \mathrm{cm/s}$;
- $T_{\mathrm{I}} = T_{\mathrm{II}} = 1000\ \mathrm{K}$, $T_{\mathrm{III}} = 1565\ \mathrm{K}$;
- $p_{\mathrm{I}} = p_{\mathrm{II}} = 1\ \mathrm{atm}$, $p_{\mathrm{III}} = 4.45\ \mathrm{atm}$;
- $r_c = 0.28\ \mathrm{cm}$, $(x_c, y_c) = (0.4, 0.75)\ \mathrm{cm}$.

The domain has a base grid of 1024×512 with three levels of refinement, each with refinement ratios of 2. The grid is dynamically refined using Equation (3.65) based on gradients of density and pressure with $\delta_t = 0.03$ and 0.008, respectively. The mesh is warped according to Equation (5.3), where $S_d = 0.08$.

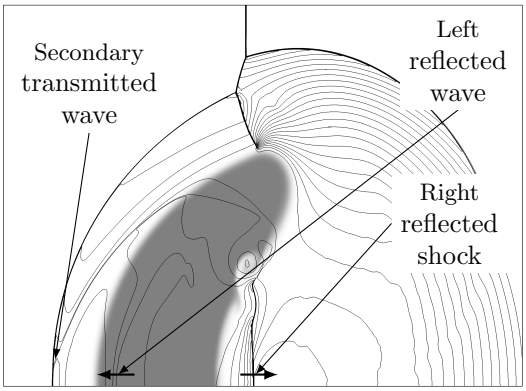
The numerical solution from Chord is compared to the numerical solutions from Billet et al. [43] and Attal et al. [41]. All three solutions use different methods and assumptions pertaining to the thermodynamic and transport properties. In addition to the differences between FLASH

and Chord mentioned in Section 6.4, Attal et al. use Equations (2.26), (2.27), and Fick's law for mass diffusion and assume that the mass diffusion coefficient for all species is equal to the binary diffusion coefficient for a N_2-O_2 system. Billet et al. implicitly solve the species mass and viscous diffusion fluxes, thus taking the binary interactions and Soret effects into account when modeling mass diffusion.

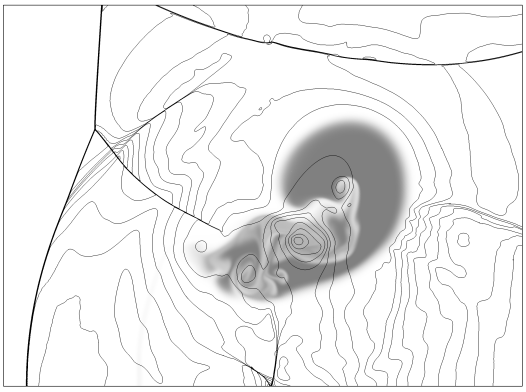
Figure 6.9 shows the pressure contour lines superposed on the H_2 mass fraction pseudo-color plot at various solution times. At $t = 1.5$ μs in Figure 6.9a, the H_2 bubble compresses as it collides with the stationary shock. This interaction produces transmitted, reflected, and refracted pressure waves that are labeled in the figure. At $t = 3.5$ μs in Figure 6.9b, three different waves can be identified: a right reflected shock forms in the air downstream of the bubble, a secondary transmitted wave forms in the air upstream of the bubble, and a reflected wave from the right reflected shock forms inside of the bubble. At $t = 10$ μs in Figure 6.9c, two counter rotating vortices exist in the H_2 region. At the same time, a wave is reflected from the top boundary of the domain. Figure 6.10 shows comparisons of the axial distribution of pressure at solution time $t = 3.5$ μs along $y = y_c$ from Figure 6.8. Additionally, two lines in the center of the figure compare the size of the compressed H_2 bubble predicted by Chord and Billet et al. The numerical predictions from Chord are in good agreement with the location of the upstream transmitted wave (labeled "Secondary transmitted wave"). However, there are discrepancies between the Chord solution and the literature solutions with respect to the size of the H_2 bubble, the locations of the wave reflected from the left side of the compressed bubble (labeled "Left reflected wave"), and the shock reflected from the interface on the right side of the compressed bubble (labeled "Right reflected shock"). Chord predicts a smaller compressed bubble size, further movement of the left and right waves, and a larger pressure increase downstream of the bubble. Again, the solution discrepancies could be due to the variation in thermodynamic models between the solutions.



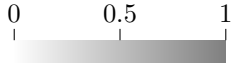
(a) $t = 1.5$ µs.



(b) $t = 3.5$ µs.



(c) $t = 10$ µs.



(d) H_2 mass fraction legend.

Figure 6.9: Pressure contour lines ($1.1 - 7.37$ atm) superposed on H_2 mass fraction pseudo-color plot (grayscale) with labeled waves.

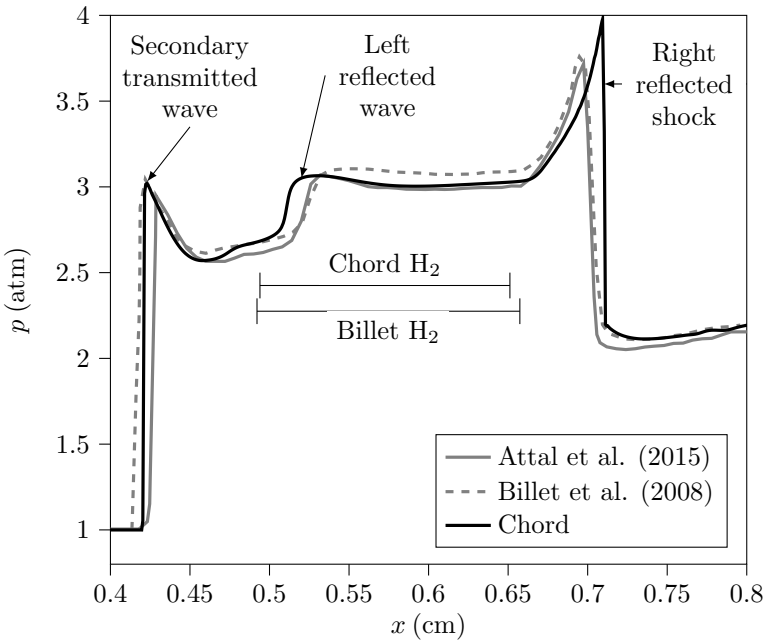


Figure 6.10: Pressure across the line at $y = y_c$ at $t = 3.5$ µs

6.6 Reacting Richtmyer-Meshkov Instability

The RMI occurs when a perturbed interface between heavy and light density fluids is accelerated by a shock wave [44, 45]. For the current test case, the transmitted shock rebounds from the right wall boundary and subjects the interface and flame surface to a second compression, a phenomenon referred to as reshock. The problem configuration follows that by Attal et al. [41] for validation purposes. The two interacting fluids are H_2 and O_2.

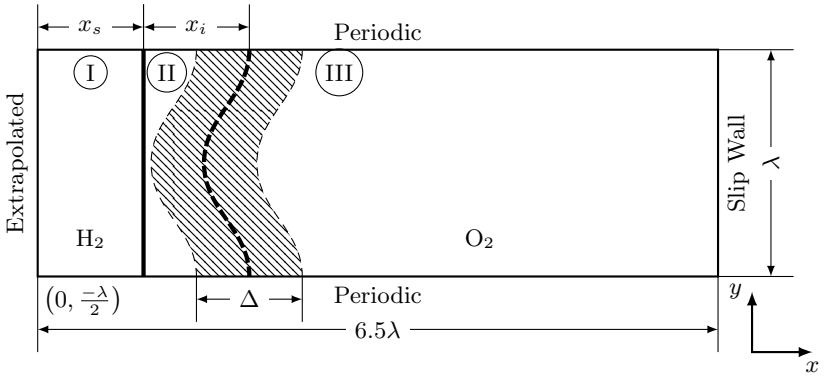


Figure 6.11: Initial configuration of the RMI test case. The "downstream" area includes the regions labeled II and III and the shaded region.

The problem configuration shown in Figure 6.11 indicates the boundary conditions, domain size, and schematic of the problem setup. In the figure, the shaded region represents the interface where Z varies from ϵ to $1-\epsilon$. The thick, dashed line in the shaded region represents the surface where $Z = 0.5$, defined as Z_{50}. The interface is initialized and evaluated using the mixture fraction Z, defined as

$$Z \equiv \frac{8c_{\mathrm{H_2}} - c_{\mathrm{O_2}} + 1}{9}, \tag{6.7}$$

where $Z = 1$ corresponds to a region of pure fuel and $Z = 0$ corresponds to a region of pure oxidizer. Region I is upstream of the shock, and all other regions are downstream of the shock. The shock is represented by the thick black line separating regions I and II. The problem parameters and corresponding values for this test case are:

- wavelength of the interface, $\lambda = 6\,\mathrm{cm}$;
- wavenumber of the interface, $k = 2\pi/\lambda$;
- initial amplitude of the interface, $h_0 = 0.2/k$;
- threshold for Z variation, $\epsilon = 1 \times 10^{-5}$;
- thickness of the interface where Z varies from ϵ to $1-\epsilon$, $\Delta = 2h_0$;
- initial location of the shock, $x_s = 4\,\mathrm{cm}$. The domain is extended slightly in the x-direction so the left boundary is $\lambda/2$ further from the shock than in the reference solution;
- downstream initial density of the H_2-O_2 and O_2 regions, $\rho_{\mathrm{H_2}} = 0.08\,\mathrm{kg/m^3}$ and $\rho_{\mathrm{O_2}} = 0.24\,\mathrm{kg/m^3}$, respectively;
- downstream initial pressure, $p_{\mathrm{II}} = p_{\mathrm{III}} = p_{\mathrm{atm}}$;
- Mach number of the shock, $\mathrm{M} = 1.2$, where the pressure ratio ≈ 1.51 and the density ratio ≈ 1.34;
- uses the 9 species, 19-step reaction mechanism from Billet [38].

The interface location as it varies in the y-direction is defined by

$$x_i(y) = x_s + \frac{\Delta}{2} + h_0 \left(1 - \cos\left(\frac{2\pi y}{\lambda}\right)\right) . \tag{6.8}$$

The mixture fraction is initialized as

$$Z(x,y) = \frac{1}{2}\left(1 - \operatorname{erf}\left(\left(x - x_i(y)\right)\frac{W_s}{\Delta}\right)\right) , \tag{6.9}$$

where W_s is a scaling factor defined as

$$W_s = 2\left|\operatorname{erf}^{-1}(1-2\epsilon)\right| \approx 6.03 . \tag{6.10}$$

The initial mass fractions are determined from the initial mixture fraction. The initial downstream density of the mixture is defined by

$$\rho = \left(\frac{c_{\mathrm{H}_2}}{\rho_{\mathrm{H}_2}} + \frac{c_{\mathrm{O}_2}}{\rho_{\mathrm{O}_2}}\right)^{-1} . \tag{6.11}$$

The pressure downstream of the shock is initialized to a uniform p_{atm}. The initial downstream temperature is calculated using the ideal gas law. It is common in RMI problems to define an Atwood number, A_t, which is

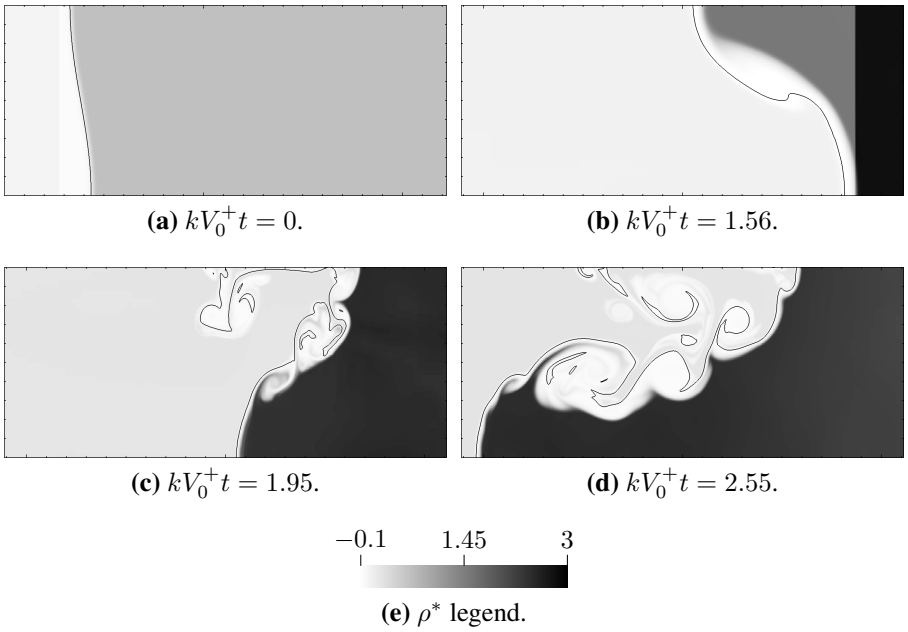
$$A_t \equiv \frac{\rho_{\mathrm{O}_2} - \rho_{\mathrm{H}_2}}{\rho_{\mathrm{O}_2} + \rho_{\mathrm{H}_2}} . \tag{6.12}$$

The domain has a base grid of 416×64 with three levels of refinement, each with refinement ratios of 2. The grid is dynamically refined using Equation (3.65) based on gradients of temperature with $\delta_t = 8 \times 10^{-3}$.

RMI solutions are presented using a scaled time parameter based on the growth of the interface amplitude over time. The growth is determined using the post-shock value V_0^+ which is expected to grow linearly according to

$$V_0^+ = kU^+A_t^+h_0^+ , \tag{6.13}$$

where the post-shock Atwood number is $A_t^+ \approx 0.51$, the post-shock amplitude is $h_0^+ \approx 0.8h_0$, and the velocity imparted to the interface by the shock, U^+, is calculated using the Rankine-Hugoniot conditions (where $U^+ \approx 3.06 \times 10^4$ cm/s).



(a) $kV_0^+t = 0.$ **(b)** $kV_0^+t = 1.56.$

(c) $kV_0^+t = 1.95.$ **(d)** $kV_0^+t = 2.55.$

(e) ρ^* legend.

Figure 6.12: Pseudo-color plot of the scaled density, ρ^*, at multiple scaled times. The solid line represents the iso-contour of Z_{50}. Figure 6.12a shows the solution at location $(3\text{ cm}, -3\text{ cm})$ to $(11\text{ cm}, 0\text{ cm})$. The other figures show the solution at location $(17\text{ cm}, -3\text{ cm})$ to $(25\text{ cm}, 0\text{ cm})$.

Figure 6.12 is a pseudo-color plot of the scaled density, $\rho^* = \frac{\rho - \rho_{\mathrm{H}_2}}{\rho_{\mathrm{O}_2} - \rho_{\mathrm{H}_2}}$, at various scaled times with an iso-contour showing $Z = Z_{50}$. Figure 6.12a shows the initial solution before the incident shock interacts with the interface. Figure 6.12b shows the solution immediately prior to reshock; the iso-contour of Z_{50} is still smooth at this point in the solution. Figures 6.12c and 6.12d show how reshock causes the interface to undergo an indirect phase inversion. These figures demonstrate the significant enhancement of mixing and the loss of interface smoothness as a result of reshock.

A mixing layer width based on the mixture fraction is calculated from the current solution and compared with the literature. The mixing layer width, W_{mix}, measures the extent of penetration of one fluid into another as

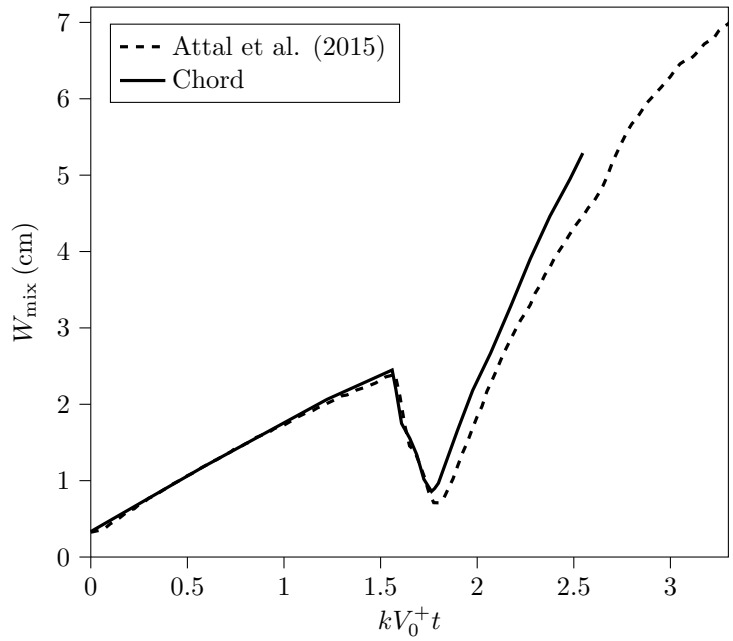


Figure 6.13: Evolution of the mixing layer width over time plotted against scaled time.

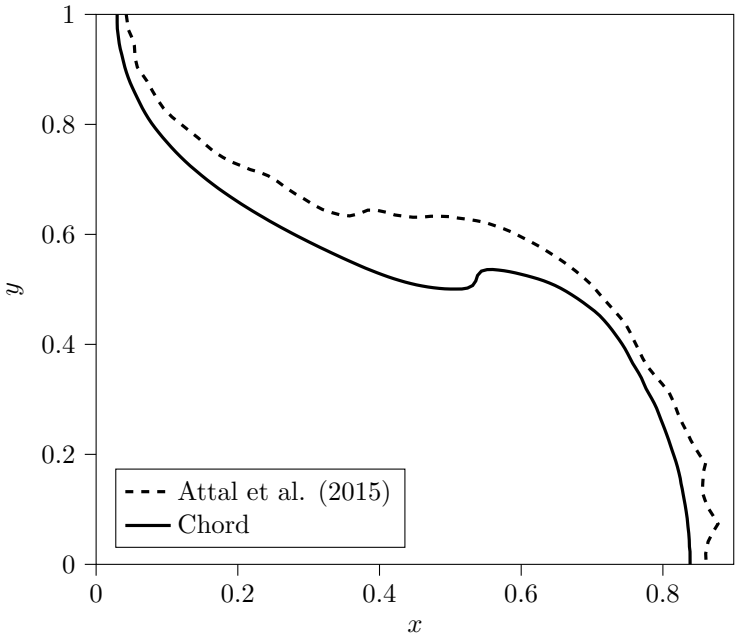


Figure 6.14: Comparison of the Z_{50} profiles at scaled time immediately prior to reshock, $kV_0^+t = 1.56$.

$$W_{\text{mix}} \equiv \int_0^\infty 4Z_{\text{avg}}(x)\left(1 - Z_{\text{avg}}(x)\right)\mathrm{d}x\,, \tag{6.14}$$

where

$$Z_{\text{avg}}(x) \equiv \frac{1}{\lambda} \int_{-\lambda/2}^{\lambda/2} Z(x, y) \mathrm{d}y \,. \tag{6.15}$$

The mixing layer widths over the scaled time are compared in Figure 6.13. Figure 6.13 shows similar magnitudes and trends of the mixing layer widths between the two solutions, especially during the reshock phase ($kV_0^+ t \approx 1.56$). The Chord solution predicts slightly greater mixing layer widths during and after reshock relative to the reference. These discrepancies could be due to the difference in diffusion modeling between the solutions. Specifically, Attal et al. rely on numerical dissipation to mimic the diffusion physics instead of explicitly solving the diffusive fluxes for the RMI case. This method underpredicts the diffusive fluxes, resulting in shorter mixing layer widths when compared to explicitly employing diffusion. The comparisons of the Z_{50} profiles in Figure 6.14 further supports this claim, since the predicted from Chord is smooth and the profile predicted by Attal et al. using FLASH is rough. Roughness in the Attal et al. profile is due to the formation of hydrodynamic instabilities that the diffusive physics suppress in the Chord solution.

Chapter 7

Numerical Results

Now that the algorithm is verified and validated, it can be applied to simulate multispecies mixing and reacting flows, including the Mach reflection shock ramp, mixing flow through a planar burner, and ODWs over a wedge. These test problems are chosen to demonstrate three different aspects of the algorithm. First, they demonstrate the ability of the algorithm to capture strong shock waves and the new stability techniques to suppress oscillations. Second, they demonstrate that the algorithm is capable of solving flow problems with non-rectangular physical domains. Lastly, they demonstrate the algorithm's ability to model flow configurations with more realistic boundary conditions.

7.1 Multispecies Mach Reflection Problem

A Mach reflection problem is considered to demonstrate the capability of the algorithm to solve a thermally perfect, multispecies flow with strong shocks on a non-rectangular physical domain. The present solution is compared to the experimental data published by Ben-Dor and Glass [46]. For this particular case, the thermally perfect, multispecies solution should not significantly differ from the single species, calorically perfect solution by Gao et al. [47].

The ramp geometry, physical and computational domain, and flow conditions in the present study are the same as those in Gao et al. [47], except the gas in the present study is composed of $c_{\mathrm{Ar}} = 0.99$ and $c_{\mathrm{N_2}} = 0.01$ to validate the implementation of the multispecies functionality.

The base grid is 96×24 with two additional refinement levels, each with a refinement ratio of 4. Cells are tagged for refinement based on density gradients. All cells near the wall boundary are also tagged for refinement to ensure the boundary layer is properly resolved.

Figure 7.1 compares the density contours from the numerical solution with the experimental results published by Ben-Dor and Glass [46] at a time of $t = 0.107\ \mathrm{ms}$. Note, the density contours in Figure 7.1 are relative to the freestream density, $\rho_1 = \rho_0 = 0.04354\ \mathrm{kg/m^3}$. The present

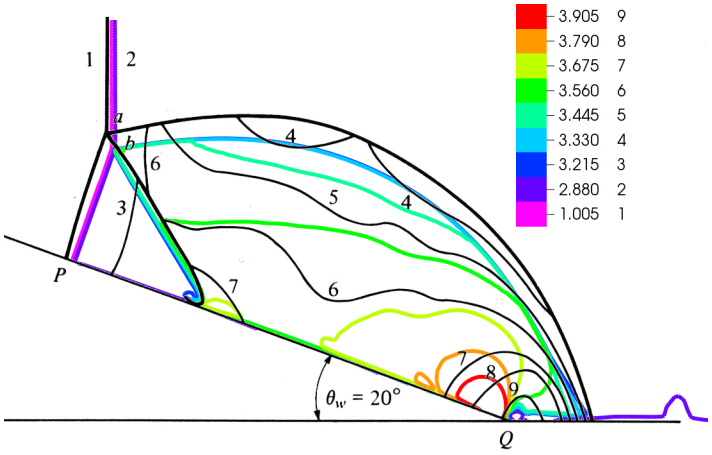


Figure 7.1: Relative density (ρ/ρ_0) contours. Computed results are shown in color, and experimental results, shown in black, are reproduced from Ben-Dor and Glass [46].

simulation results agree well with the previous study [47] and show general agreement with the experiment. Due to the similarities between the calorically perfect and thermally perfect numerical solutions, a detailed analysis of the simulation can be found in Gao et al. [47] and will not be repeated here.

7.2 Planar Burner Simulation

Flow mixing inside a 2-D planar burner geometry [48] is considered to demonstrate the capability of the algorithm to solve thermally perfect, multispecies flow in a configuration with more realistic boundary conditions. Figure 7.2 shows the 2-D burner geometry and the computational domain. A fuel-air mixture flows in the positive y-direction between two vertical walls, while air is injected horizontally from jets located on each wall. The height and width of the burner geometry are denoted by L_y and L_x, respectively, and are $L_y = L_x = 0.1016\,\mathrm{m}$. $L_w = 0.0492\,\mathrm{m}$ is the distance from the bottom of the burner to the bottom of the jet, and $L_j = 1.6\times10^{-3}$ is the height of the jets. The computational domain has dimensions of $L_x \times 8L_y$. The computational domain extends beyond the top of the burner geometry in the y-direction to set the outlet boundary far from the top of the burner geometry, ensuring the outflow boundary minimally interferes with the interior flow.

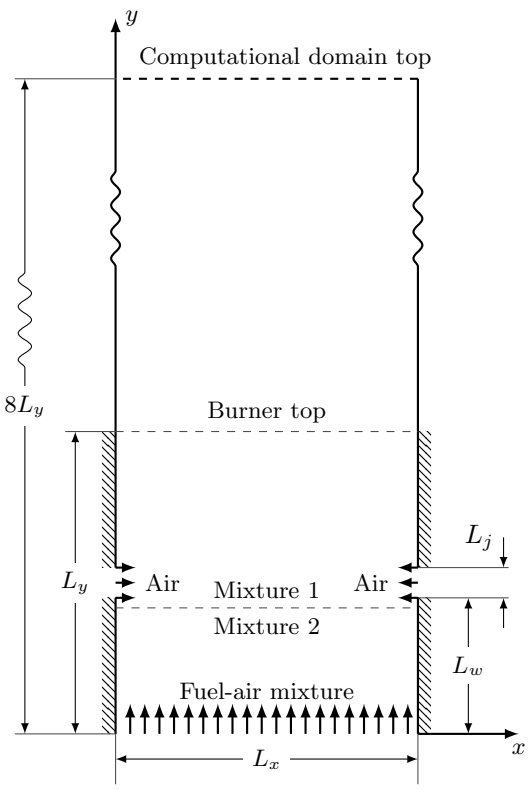


Figure 7.2: The burner and computational domain geometry.

The fuel-air mixture consists of $c_{\mathrm{CH_4}} = 0.0551$, $c_{\mathrm{O_2}} = 0.2202$, and $c_{\mathrm{N_2}} = 0.7447$ and flows into the domain from the lower y-direction boundary at $0.075\ \mathrm{m/s}$ and $313\ \mathrm{K}$. Air, consisting of $c_{\mathrm{O_2}} = 0.233$ and $c_{\mathrm{N_2}} = 0.767$, is horizontally injected into the domain from the jets at $4.96\ \mathrm{m/s}$ and $293\ \mathrm{K}$. At both boundaries, pressure is extrapolated from the interior, and density is computed using the ideal gas law. The walls are no-slip for $y < L_y$ but slip for $y \geq L_y$. The outlet uses a zero gradient Neumann condition for all variables.

The initial mixture in the domain is quiescent. Where $y \geq 0.04\ \mathrm{m}$, labeled "Mixture 1" in Figure 7.2, the initial fluid consists of $c_{\mathrm{CO_2}} = 0.1514$, $c_{\mathrm{H_2O}} = 0.1239$ and $c_{\mathrm{N_2}} = 0.7246$ at $298\ \mathrm{K}$. The remainder of the initial fluid in the domain, labeled "Mixture 2" in Figure 7.2, is set to the same composition and temperature as the air from the jets. Both "Mixture 1" and "Mixture 2" are set to atmospheric pressure. The Reynolds number is 610 based on the jet inlet stream condition and 493 based on the syngas inflow condition.

The base grid is 32×256 with two additional refinement levels. The refinement ratio of the first level is 2, and the refinement ratio of the second level is 4. Cells are dynamically tagged for refinement based on gradients of density. Sub-cycling is used during the solution. The solution is run for one convective time scale, i.e. the time required for the inflow to reach the top of the burner, which is ≈ 1.345 s.

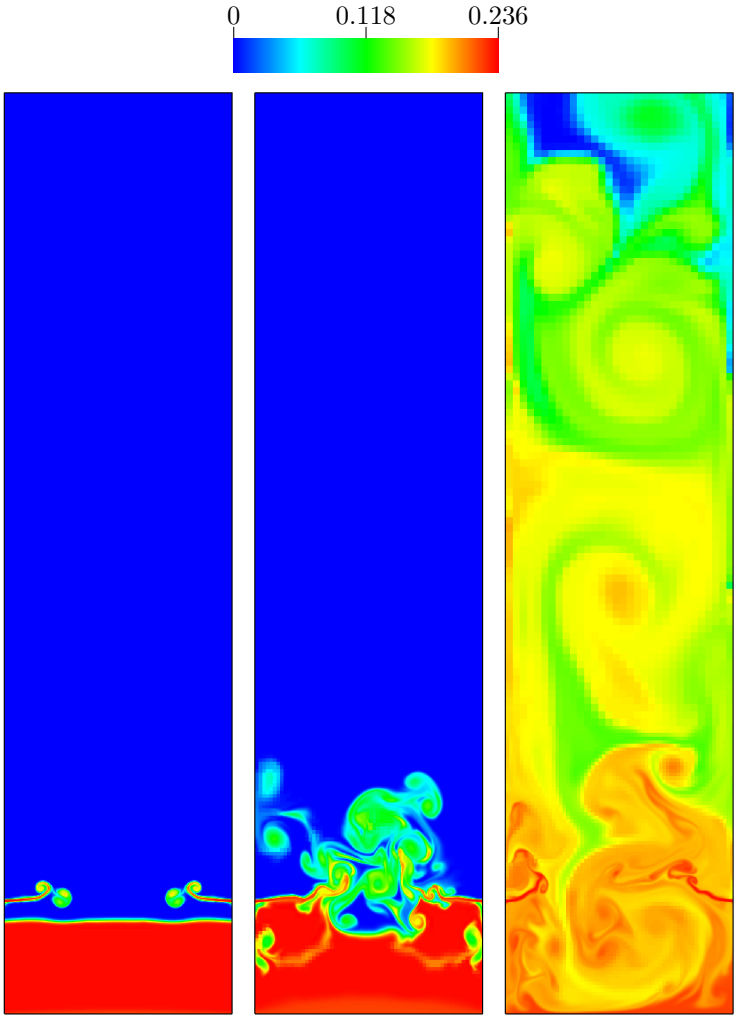


Figure 7.3: Pseudo-color plot of c_{O_2} in the lower half of the computational domain at $t = 0.02$ s, 0.101 s, and 1.652 s, respectively.

Figure 7.3 shows the distribution and time evolution of c_{O_2} in the lower half of the computational domain. The mesh adaption is demonstrated in Figure 7.4 for two solution times, $t = 0.02$ s and 1.652 s. For demonstration purposes, the meshes are only shown for the coarse levels. Fig-

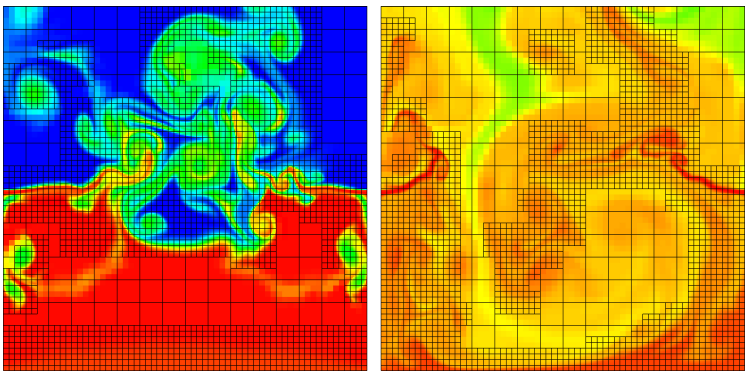


Figure 7.4: Demonstration of the grid adaption at $t = 0.02$ s and 1.652 s, respectively. The mesh in the figure is coarsened for display purposes.

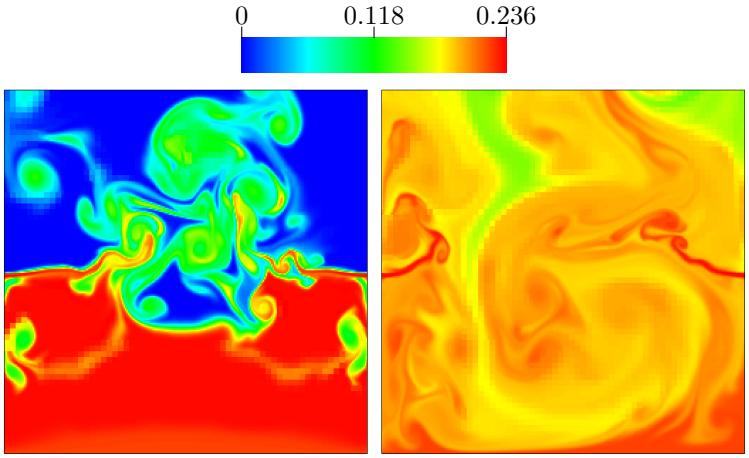


Figure 7.5: Pseudo-color plot of c_{O_2} in the burner geometry at $t = 0.101$ s and 1.652 s, respectively.

ure 7.5 is a view of the burner geometry and shows the fluid structures produced by the jets and the fluid interactions between the bottom inlet flow and initial flow field. At solution time $t = 0.02$ s, the fluid from both jets form a symmetric shape. At solution time $t = 0.101$ s, the symmetry begins to break down as the two jets interact with each other. Much later in the solution, at $t = 1.652$ s, Figure 7.5 clearly shows no symmetry, and the jets appear to overlap one another; the O_2 begins to mix more uniformly throughout the domain. This asymmetric process has been observed in experiments [48]. Although quantitative profiles are not available from the experiment for detailed comparison, the numerical predictions at various times capture the experimental structure fairly well.

7.3 Oblique Detonation Waves

Solutions of standing ODWs are obtained using Chord and compared with literature. In these solutions, a stoichiometric mixture of hydrogen and air ($H_2 : O_2 : N_2/2 : 1 : 3.76$) flows from left to right at high Mach number and encounters a ramp. Heating from the resultant oblique shock induces deflagration after an induction delay. Because the combustion is constrained by the wall of the ramp, compression waves are produced that propagate inwards while coalescing to ultimately produce a detonation wave. The transition in steady 2-D space is analogous to deflagration-to-detonation (DDT) observed in unsteady 1-D space when deflagration is initiated at a wall, except that in an ODW thermal diffusion is not required for flame propagation. The observed structures match those described by Li et al. [49].

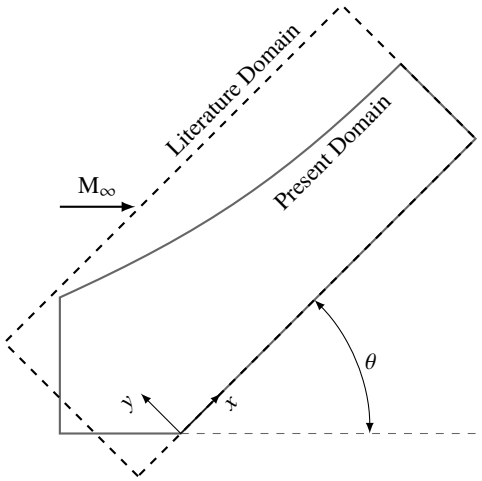


Figure 7.6: Comparison between wedge domains.

For all cases studied herein, the computational domain is shown in Figures 4.4 and 7.6, with the wedge angle denoted by θ and the freestream Mach number denoted by M_∞. As with the ODW results shown in Section 4.2.2, a Schwarz-Christoffel mapping transformation is used to model the wedge. By contrast, studies from the literature commonly use a Cartesian grid with an angled velocity vector, as shown in Figure 7.6. In all simulations, the top boundary is sufficiently far from the wall as to not interact with waves produced in the simulations. Supersonic inflow and outflow boundary conditions are used along with slip boundaries at the walls. Solutions ob-

Table 7.1: List of each reference ODW case and relevant details.

Case	θ	M_∞	p_∞ (atm)	T_∞ (K)	l_r (mm)	l_s (mm)	N_r-Ref	N_r-Chord	Ref
1	29°	8	1	300	4	0.5	2[a]	12[b]	[49]
2	27°	9.3	1	300	5	1	12[b]	12[b]	[50]
3	15°	4.3	0.5527	1021	60	5	23[c]	23[c]	[51]

[a] Two-step reaction mechanism involving 5 species: H_2, O_2, N_2, H_2O, and a representation for intermediate species [49];
[b] 12-step reaction mechanism involving 8 species: H_2, O_2, N_2, H_2O, HO_2, OH, O, and H [50];
[c] 23-step reaction mechanism involving 11 species: H_2, O_2, N_2, H_2O, HO_2, OH, H_2O_2, O, H, N, and NO [32].

tained with Chord use adaptive mesh refinement to reduce errors where required, especially near discontinuities, while minimizing computational cost elsewhere. Refinement is based on gradients of temperature using Equation (3.65) with $\delta_t = 0.05$; additionally, refinement is fixed in the layer of cells adjacent to the wall.

Comparisons are made with three different ODW cases from the literature. The wedge angle, freestream conditions, domain lengths, number of reactions in the reference reaction mechanism, number of reactions in the Chord reaction mechanism, and literature reference are provided in Table 7.1. The footnotes list the species considered in the reaction mechanism. In all cases, the fluid is a stoichiometric mixture of hydrogen and air, and diffusion is neglected. The same reaction mechanisms used in the references are also used in Chord except for Case 1, where the 12-step model from Thaker and Chelliah [50] is used. The Chord results in this section are taken from a rotated physical domain to make comparisons with literature possible. See the axis in Figure 7.6 for reference.

7.3.1 Case 1: $\mathrm{M}_\infty = 8$, $\theta = 29^\circ$

The first case is taken from the work by Li et al. [49] with $\mathrm{M}_\infty = 8$, $\theta = 29^\circ$. The reference study uses a uniform grid spacing of $\Delta x = \Delta y = 6\ \mu\mathrm{m}$ and a two-step reaction mechanism. The solution by Chord is obtained with one additional refinement level with a refinement ratio of 4. The base mesh size is 192×96 (finest level grid spacing: $(\Delta x\,, \Delta y) \approx (5.86\ \mu\mathrm{m}\,, 5.86\ \mu\mathrm{m})$). The 12-step mechanism described by Thaker and Chelliah [50] models the reaction.

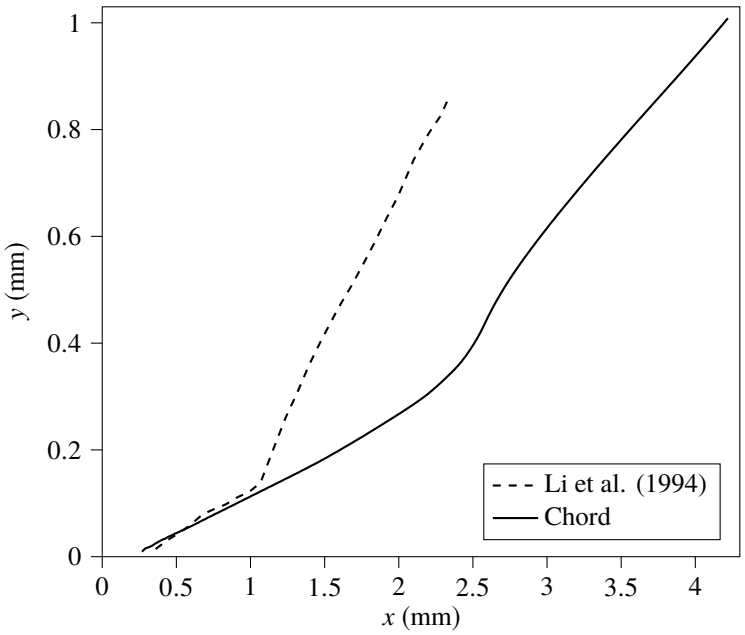


Figure 7.7: Oblique shock location comparison for Case 1.

Comparisons of the OSW and ODW profiles are shown in Figure 7.7. There is poor agreement in the length of the induction region and the slopes of the detonation waves. Increased heat addition should result in a shorter induction region and steeper detonation wave. The disparate rates of heat release behind the detonation, caused by the different reaction models, account for the observed differences. The transition is more abrupt in the reference, but the geometry of the detonation is otherwise similar. In particular, the same steepening of the detonation, just after transition, can be observed in the reference and leads to a small region of subsonic flow.

Attempts to reproduce any examples from the reference that use a 23° wedge angle for stoichiometric mixtures of hydrogen and air were unsuccessful. Li et al. [49] describe temperatures of $1200\ \mathrm{K}$ behind the oblique shock, similar to the $1150\ \mathrm{K}$ predicted by Chord. However, these temperatures were insufficient to support reactions in time or lengths scales of interest. Again, differences in heat release rates from different reaction models are the likely cause.

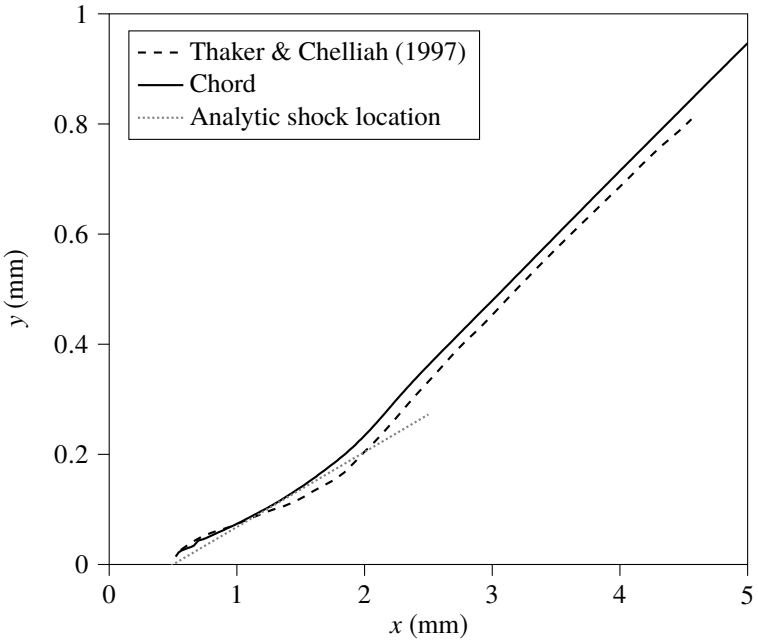


Figure 7.8: Oblique shock location comparison for Case 2.

7.3.2 Case 2: $\mathrm{M}_\infty = 9.3$, $\theta = 27^\circ$

The second case reproduces a problem studied by Thaker and Chelliah [50]. The parameters of the set-up from the reference are $\theta = 27^\circ$, $\mathrm{M}_\infty = 9.3$, and a grid spacing of $(\Delta x, \Delta y) = (25\ \mu\mathrm{m}, 16.6\ \mu\mathrm{m})$. In Chord, the base mesh is 192×48 with one additional refinement level with a refinement ratio of 4 (finest level grid spacing: $(\Delta x, \Delta y) \approx (7.8\ \mu\mathrm{m}, 7.8\ \mu\mathrm{m})$). In both the reference and the Chord simulation, the 8 species, 12-step reaction mechanism described by Thaker and Chelliah [50] is used to model the chemical reactions. For boundary conditions, the reference enforces $\partial T/\partial y = 0$ and $\partial p/\partial y = 0$ whereas no such restrictions are used in Chord.

The OSW and ODW profiles are compared in Figure 7.8. Interestingly, the oblique shocks have different slopes while the slopes of the detonations are in good agreement. In all other comparisons, Chord matches the oblique shock angle from the literature. Furthermore, oblique shock predictions should reasonably match analytical theory for a calorically perfect gas with frozen composition. This curve is also plotted as a dotted line which is in much better agreement with Chord. The reference predicts an abrupt transition whereas Chord predicts a much more gradual one.

Temperature profiles across the wedge, shown in Figure 7.9, provide more information. Across

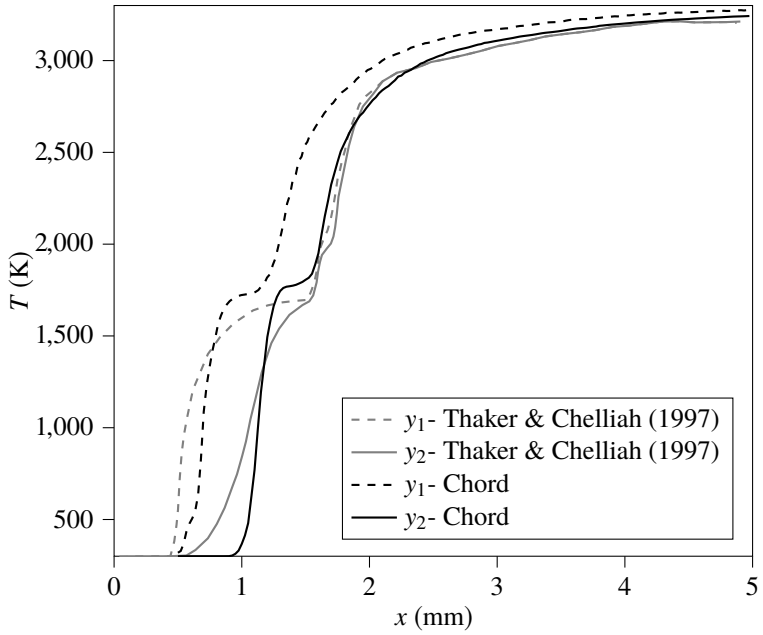


Figure 7.9: Temperature profiles along varying wall normal planes for Case 2; $y_1 = 33.2\ \mu\text{m}$ and $y_2 = 83\ \mu\text{m}$.

an ODW, one expects that the temperature profile would consist of the following: a nonreactive shock, signaled by an initial discontinuous temperature rise; an induction zone, signaled by a temperature plateau; and finally, a region of heat release indicated by monotonic temperature rise, either in the form of a deflagration wave for regions near the wall or a more energetic release following a detonation. Accordingly, the temperature profile predicted by Chord in Figure 7.9 consists of an oblique shock, induction zone, and heat release. The temperature profiles from the reference solution instead smear the oblique shock and heat release with no distinct induction region. In summary, there is some doubt about the numerically accuracy of the reference solution in the vicinity of the oblique shock and induction zone.

7.3.3 Case 3: $\text{M}_\infty = 4.3$, $\theta = 15^\circ$

The third ODW reference was simulated by Wang et al. [51]. In this problem, $\theta = 15^\circ$ and $\text{M}_\infty = 4.3$. The reference also utilizes AMR with a grid spacing of $(\Delta x\,, \Delta y) = (25\ \mu\text{m}\,, 25\ \mu\text{m})$ on the finest grid level. A base mesh size of 200×80 is used in Chord with two additional refinement levels, each with a refinement ratio of 4 (finest level grid spacing: $(\Delta x\,, \Delta y) \approx (20.3$

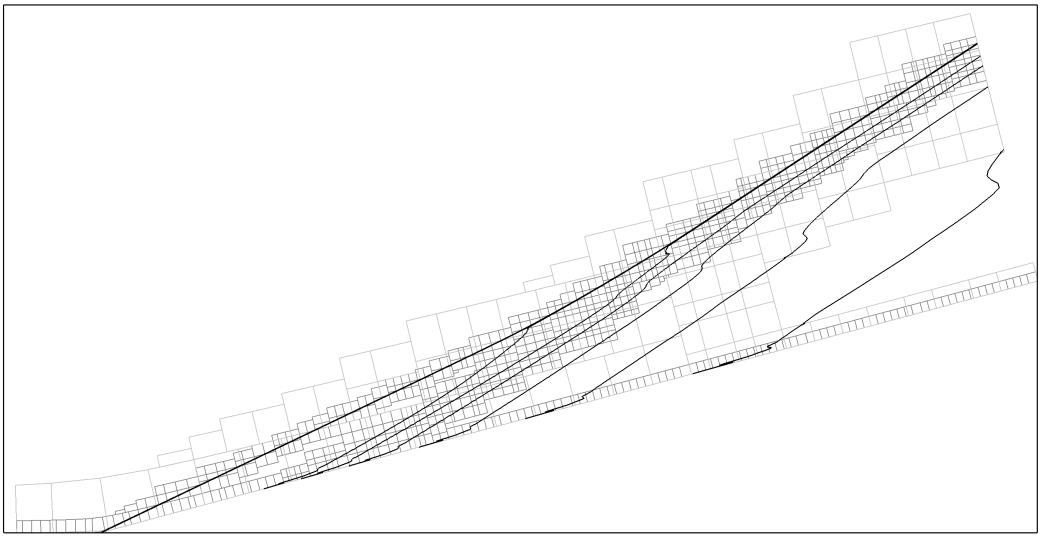


Figure 7.10: Mesh patches overlying temperature contour lines. The lighter boxes are the first refined level and the darker boxes are the second refined level. The darker lines are contours of temperature. The mesh refines around the temperature increases associated with the oblique shock and the regions of significant heat release in the flow.

$\mu\mathrm{m}\,, 20.3\ \mu\mathrm{m}$)). When the shock structures are developed, the AMR grid appears as shown in Figure 7.10. In both the reference and the Chord simulation, the 11 species, 23-step reaction mechanism from CHEMKIN [32] is employed to model the chemical reactions.

The nonreactive shock and flame profiles are compared in Figure 7.11. The flame location is identified based on a critical temperature of $T = 2070\ \mathrm{K}$; this temperature corresponds to the temperature at the end of the Zel'dovich-von Neumann-Döring (ZND) induction zone. The shock profiles from the present study are in very good agreement with slight deviations near the outlet. The flame locations differ slightly in the near wall region. The reference predicts a flame that appears almost perpendicular to the wall whereas Chord predicts a steep angle leading into the wall.

Given the slip walls, the most reasonable expectation for the flame shape near the wall is a straight line with a slope similar to the ODW. Positive curvature is expected as one moves away from the wall due to compression waves resulting from constrained deflagration. The solutions from both codes deviate from an expected profile. The reference does not detail how temperature is handled at the wall, but enforcement of $\partial T/\partial y = 0$ at the wall is consistent with the shape

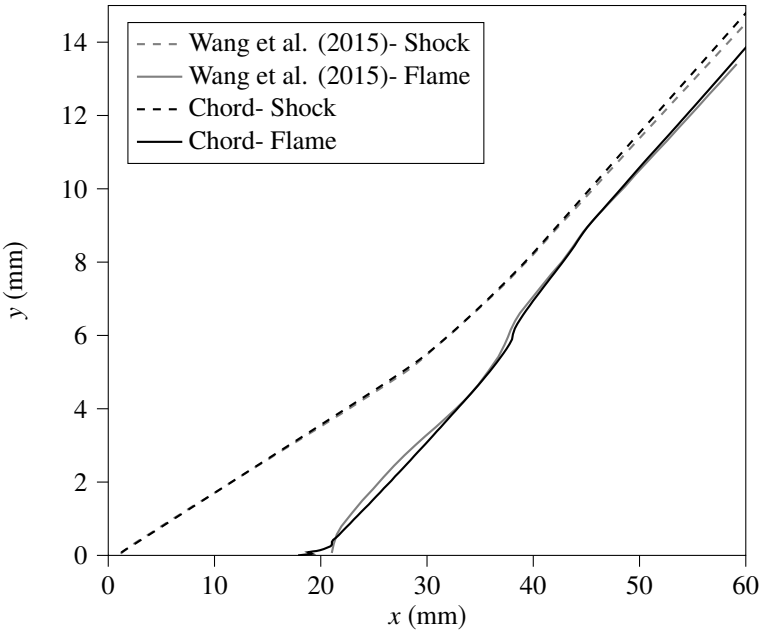
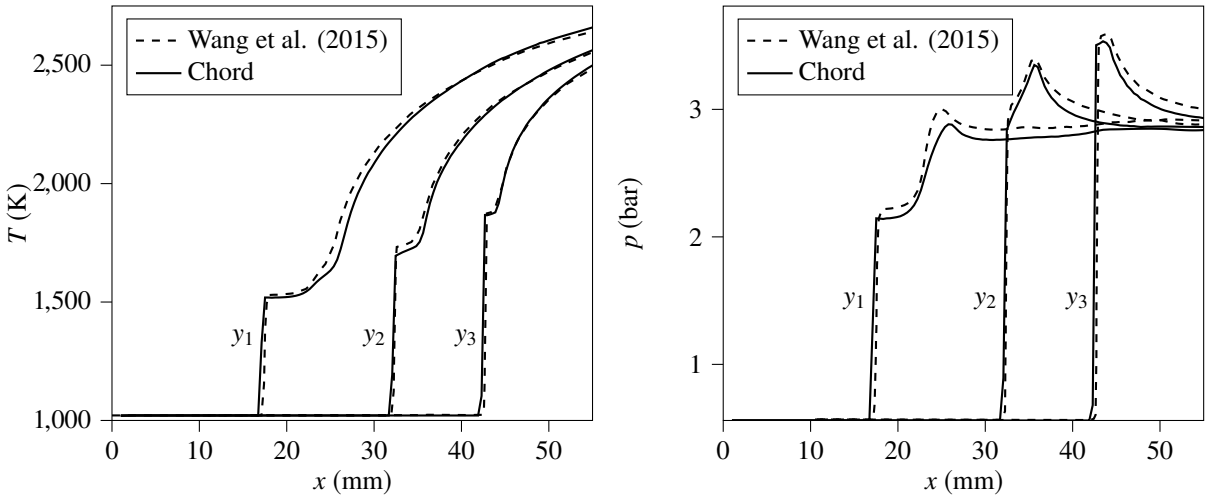


Figure 7.11: Shock and flame location comparison for Case 3.

of the flame profile near the wall in Figure 7.11. In Chord, cells adjacent to the wall suffer from overheating at the oblique shock, shortening the induction zone length adjacent to the wall. When solving the jump conditions across a shock, the interior scheme relies on an error cancellation property in the fluxes on each side of a cell. However, the exact momentum flux at the wall is used as a boundary condition so the errors do not cancel. The error manifests as the observed heating; this effect is described in detail by Woodward [52]. Overheating at the wall is purely numerical and not representative of any physical phenomena; it scales with mesh resolution and does not affect convergence. The overheating at the wall would vanish if the viscous scales were resolved near the wall. Development of limiting strategies for suppressing near wall oscillations is ongoing.

Temperature and pressure profiles at varying wall normal planes are shown in Figure 7.12. From the pressure profiles, it is evident that the y_1 curve shows deflagration, y_2 is near transition, and at y_3 a single pressure rise of a detonation is observed. The Chord solution agrees well with the reference solution, and the profiles are consistent with the observations of Li et al. [49], except the detonation induction zone is clearly seen by both predictions for the present case (y_3 in Figure 7.12a). The induction zone is the plateau in the temperature profile following the shock.



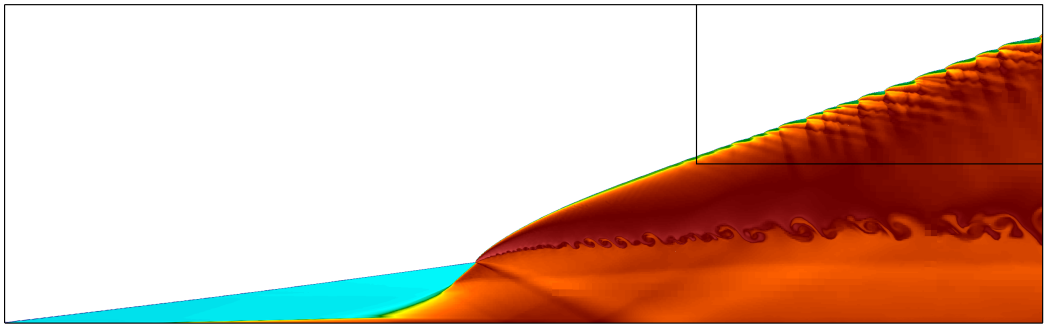
(a) Temperature **(b)** Pressure

Figure 7.12: Temperature and pressure profiles along varying wall normal planes for Case 3; $y_1 = 3\,\mathrm{mm}$, $y_2 = 6\,\mathrm{mm}$, and $y_3 = 9\,\mathrm{mm}$.

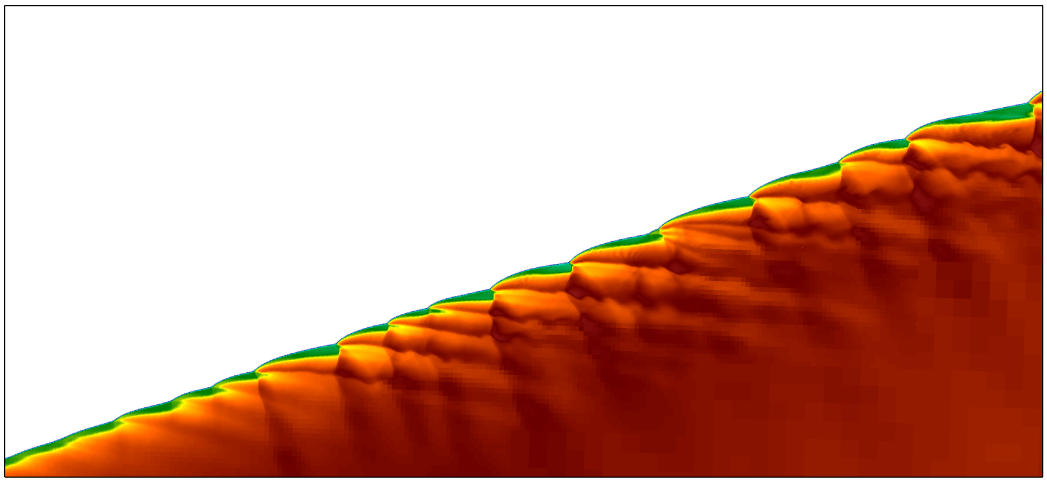
This induction zone becomes smaller as the distance from the wall grows and is quite brief in the detonation (y_3). The peaks in pressure correspond to the heat release from the reactions. For the y_1 profile, the separation between the initial pressure rise and the pressure peak means the nonreactive shock is followed by a set of deflagration waves. The y_3 profile captures the oblique detonation since the oblique shock and the heat release are fully coupled. The profile at y_2 corresponds to an intermediate structure between the decoupled and fully coupled nonreactive and reactive shocks.

7.3.4 Oblique Cellular Detonation

A final case provides a brief demonstration of detonation cells forming in an ODW. For this problem, $\theta = 23.8°$, $\mathrm{M}_\infty = 8$, $p_\infty = 34\,\mathrm{kPa}$, and $T_\infty = 300\,\mathrm{K}$. The mixture is a stoichiometric mixture of hydrogen and air. The current setup is partly based on a case run as part of a parametric study by Figueria da Silva and Deshaies [37]. The ramp and lead lengths are changed to $l_r = 10\,\mathrm{cm}$ and $l_s = 0.8\,\mathrm{cm}$ to ensure the detonation cells are captured. The base mesh is 160×64 and has three additionally refined levels; the first two levels have refinement ratios of 4, and the third level has a refinement ratio of 2. The reaction mechanism is 9 species, 19-step mechanism from Billet et al. [38].



(a) Rotated temperature pseudo-color plot.



(b) Magnified temperature pseudo-color plot.

(c) Temperature legend, (K).

Figure 7.13: Temperature (K) pseudo-color plot at solution time $t = 61.1$ μs.

A pseudo-color plot of temperature is shown in Figure 7.13a. The plots are rotated so that the wedge surface is horizontal. The color legend is shown in Figure 7.13c. The blue region in the lower left corner is the induction zone behind the nonreactive shock. From the wall, a deflagration wave forms and steepens until it transitions into a detonation wave, where the slope becomes straight and the contours become discontinuous. The detonation wave then intersects with the initial OSW. Detonation cells form further downstream from the triple point along the leading shock front. A box is placed around the area of visible detonation cells to represent the region that is shown in Figure 7.13b. Figure 7.13b is a magnified image of the cells forming along the ODW.

The cells produce both left and right propagating waves, but the tangential velocity convects the transverse waves downstream.

Chapter 8

Conclusions

8.1 Conclusions

This work produces a stable, verified, and validated fourth-order, solution-adaptive finite-volume algorithm for solving the compressible Navier-Stokes equations with chemical reaction modeling on mapped domains.

Significant effort is dedicated to improving the stability of the high-order FVMs for solving flows with strong shock waves or detonations. The face value limiting, limited primitive variable selection, DC flattening, and FCOR are devised and implemented into Chord to improve the stability of the algorithm. The DC flattening stabilization technique prevents divergence of the solution during DC operations. The stability improvements provided by the face value limiting, limited primitive variable selection, and FCOR are demonstrated on a shock tube case. The shock tube solution indicates that limiting temperature and pressure, applying face value limiting, and using FCOR provides the least oscillatory solution with the most conservation of species concentrations.

Fourth-order accuracy is verified with the Couette flow, Gaussian acoustic pulse, multispecies mass diffusion bubble, and shear problems. These problems verify that the algorithm retains fourth-order accuracy for smooth solutions and that the stability techniques do not impair the order of accuracy of the underlying algorithm.

The thermally perfect, multispecies capability is validated through the shock tube, shock box, and lid-driven cavity problems. The chemical reaction modeling is validated through the reacting-advecting hydrogen-oxygen front, shock-driven combustion of a hydrogen bubble, and reacting RMI problems.

The resulting algorithm is used to simulate a multispecies Mach reflection case, mixing flow through a planar burner, and ODWs over a wedge. These cases demonstrate the ability of the

algorithm to model reacting, multiscale physics and strong shock dynamics in relatively complex geometries with realistic boundary conditions.

8.2 Original Contributions

This research has considerable novel contributions toward the development of a fourth-order finite-volume algorithm for solving the thermally perfect, compressible Navier-Stokes equations with chemical reaction modeling on mapped domains. The key features of the proposed algorithm are below.

1. Developed and optimized the fourth-order finite-volume methodology for solving the mapped diffusive fluxes. This task entailed optimizing the operations with consideration of the convolution, deconvolution, and gradient operations.

2. Compared different primitive variable limiting for face value construction. It was determined that limiting temperature and pressure and calculating density produced the greatest stability and diminished numerical oscillations.

3. Devised, implemented, and calibrated the DC flattening and FCOR stability techniques.

4. Devised, implemented, and calibrated the face value limiting methodology to improve stability and reduce severity of numerical oscillations near the strong discontinuities.

8.3 Future Work

Throughout this research, potential improvements to the algorithm have been recognized. This future work is briefly suggested below.

- Stability of the algorithm can be further improved through more robust boundary conditions. Outflow boundary conditions do not remain stable in the presence of outgoing subsonic pressure waves, and pressure waves tangent to the boundary surface pose another set of stability issues for the outflow boundary.

- While current mass diffusion models are proven sufficient for the present work, more accurate mass diffusion models that account for the Soret effect, Dufour effect, and the binary interactions between species may be considered.

- Gradient based refinement tagging is sufficient for the present work; however, refinement tagging based on error estimates using adjoint based methods is widely considered to be superior. Implementing adjoint based refinement tagging would significantly improve the utilization of AMR.

- Assessing and improving the parallel performance of the algorithm should be performed on a modern high-performance computing architecture.

Bibliography

[1] S. B. Pope, "Computations of turbulent combustion: Progress and challenges," *Proc. Combust. Inst.*, pp. 591–612, 1990.

[2] T. Poinsot, S. Candel, and A. Trouve, "Applications of direct numerical simulation to premixed turbulent combustion," *Prog. Energy Combust. Sci.*, vol. 21, pp. 531–576, 1995.

[3] C. K. Westbrook, Y. Mizobuchi, T. J. Poinsot, P. J. Smith, and J. Warantz, "Computational combustion," *Proc. Combust. Inst.*, vol. 30, no. 1, pp. 125–157, 2005.

[4] AIAA, "Guide for the verification and validation of computational fluid dynamics simulations," Tech. Rep. G-077-1998, AIAA, 1998.

[5] Z. J. Wang, K. Fidkowski, R. Abgrall, F. Bassi, D. Caraeni, A. Cary, H. Deconinck, R. Hartmann, K. Hillewaert, H. T. Huynh, N. Kroll, G. May, P.-O. Persson, B. van Leer, and M. Visbal, "High-order CFD methods: current status and perspective," *Int. J. Numer. Methods Fluids*, vol. 72, no. 8, pp. 811–845, 2013.

[6] T. H. Pulliam and D. W. Zingg, *Fundamental Algorithms in Computational Fluid Dynamics*. Scientific Computation, New Delhi, India: Spinger International Publishing, 1 ed., 2014.

[7] P. Colella, M. R. Dorr, J. A. F. Hittinger, and D. F. Martin, "High-order finite-volume methods in mapped coordinates," *J. Comput. Phys.*, vol. 230, pp. 2952–2976, 2011.

[8] M. Barad and P. Colella, "A fourth-order accurate local refinement method for Poisson's equation," *J. Comput. Phys.*, vol. 209, no. 1, pp. 1–18, 2005.

[9] S. M. Guzik, X. Gao, L. D. Owen, P. McCorquodale, and P. Colella, "A freestream-preserving fourth-order finite-volume method in mapped coordinates with adaptive-mesh refinement," *Comput. Fluids*, vol. 123, pp. 202–217, 2015.

[10] X. Gao, *A parallel solution-adaptive method for turbulent non-premixed combusting flows*. PhD thesis, University of Toronto, 2008.

[11] M. J. Berger and J. Oliger, "Adaptive mesh refinement for hyperbolic partial differential equations," *J. Comput. Phys.*, vol. 53, no. 3, pp. 484–512, 1984.

[12] M. J. Berger and P. Colella, "Local adaptive mesh refinement for shock hydrodynamics," *J. Comput. Phys.*, vol. 82, pp. 64–84, May 1989.

[13] M. Adams, P. Colella, D. T. Graves, J. N. Johnson, H. S. Johansen, N. D. Keen, T. J. Ligocki, D. F. Martin, P. W. McCorquodale, D. Modiano, P. O. Schwartz, T. D. Sternberg, and B. Van Straalen, "Chombo Software Package for AMR Applications - Design Document," Tech. Rep. LBNL-6616E, Lawrence Berkeley National Laboratory, 2014.

[14] P. McCorquodale and P. Colella, "A high-order finite-volume method for conservation laws on locally refined grids," *Comm. App. Math. Comput. Sci.*, vol. 6, no. 1, pp. 1–25, 2011.

[15] R. W. Houim and K. K. Kuo, "A low-dissipation and time-accurate method for compressible multi-component flow with variable specific heat ratios," *J. Comput. Phys.*, vol. 230, no. 23, pp. 8527–8553, 2011.

[16] S. Karni, "Hybrid multifluid algorithms," *SIAM J. Sci. Comput.*, vol. 17, no. 5, pp. 1019–1039, 1996.

[17] R. Abgrall and S. Karni, "Computations of compressible multifluids," *J. Comput. Phys.*, vol. 169, no. 2, pp. 594 – 623, 2001.

[18] P. C. Ma, Y. Lv, and M. Ihme, "An entropy-stable hybrid scheme for simulations of transcritical real-fluid flows," *J. Comput. Phys.*, vol. 340, pp. 330–357, 2017.

[19] H. Terashima and M. Koshi, "Approach for simulating gas–liquid-like flows under supercritical pressures using a high-order central differencing scheme," *J. Comput. Phys.*, vol. 231, no. 20, pp. 6907–6923, 2012.

[20] T. Schmitt, L. Selle, A. Ruiz, and B. Cuenot, "Large-eddy simulation of supercritical-pressure round jets," *AIAA Journal*, vol. 48, no. 9, pp. 2133–2144, 2010.

[21] J. D. Anderson, *Hypersonic and High Temperature Gas Dynamics*. New York: McGraw-Hill, 1989.

[22] S. Gordon and B. J. McBride, "Computer program for calculation of complex chemical equilibrium compositions and applications I. analysis," Reference Publication 1311, NASA, 1994.

[23] B. J. McBride and S. Gordon, "Computer program for calculation of complex chemical equilibrium compositions and applications II. users manual and program description," Reference Publication 1311, NASA, 1996.

[24] S. Mathur, P. K. Tondon, and S. C. Saxena, "Thermal conductivity of binary, ternary and quaternary mixtures of rare gases," *Molecular Physics*, vol. 12, no. 6, pp. 569–579, 1967.

[25] H. Lomax, T. H. Pulliam, and D. W. Zingg, *Fundamentals of Computational Fluid Dynamics*. Scientific Computation, Springer-Verlag Berlin Heidelberg, 1 ed., 2001.

[26] C. Hirsch, *Numerical Computation of Internal and External Flows: The Fundamentals of Computational Fluid Dynamics*. Elsevier, 2007.

[27] B. Riviere, *Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation*. SIAM, 2008.

[28] P. R. Woodward and P. Colella, "The numerical simulation of two-dimensional fluid flow with strong shocks," *J. Comput. Phys.*, vol. 54, no. 1, pp. 115–173, 1984.

[29] P. Colella, "Multidimensional upwind methods for hyperbolic conservation laws," *J. Comput. Phys.*, vol. 87, no. 1, pp. 171–200, 1990.

[30] J. D. Lambert, *Numerical Methods for Ordinary Differential Systems*. New York: John Wiley & Sons, 1991.

[31] X. Gao, S. M. J. Guzik, and P. Colella, "A fourth-order boundary treatment for viscous fluxes on cartesian grid finite-volume methods," AIAA 2014-1277, 52nd AIAA Aerospace Sciences Meeting, 2014.

[32] R. J. Kee, R. M. Ruply, E. Meeks, and J. A. Miller, "Chemkin-III: A FORTRAN chemical kinetics package for the analysis of gas-phase chemical and plasma kinetics," Technical Report SAND96-8216, Sandia National Laboratories, Livermore, 1996.

[33] R. P. Brent, "An algorithm with guaranteed convergence for finding a zero of a function," *The Computer Journal*, vol. 14, pp. 422–425, January 1971.

[34] L. D. Owen, S. M. Guzik, and X. Gao, "A fourth-order finite-volume algorithm for compressible flow with chemical reactions on mapped grids," in *23rd AIAA Computational Fluid Dynamics Conference*, no. AIAA 2017-4498, AIAA AVIATION Forum, 2017.

[35] L. D. Owen, S. M. Guzik, and X. Gao, "A high-order adaptive algorithm for multispecies gaseous flows on mapped domains," *Comput. Fluids*, vol. 170, pp. 249–260, 2018.

[36] A. A. I. Peer, A. Gopaul, M. Z. Dauhoo, and M. Bhuruth, "A new fourth-order non-oscillatory central scheme for hyperbolic conservation laws," *Appl. Numer. Math.*, vol. 58, no. 5, pp. 674–688, 2008.

[37] L. F. Figueria da Silva and B. Deshaies, "Stabilization of an oblique detonation wave by a wedge: a parametric numerical study," *Combust. and Flame*, vol. 121, no. 1, pp. 152–166, 2000.

[38] G. Billet, "Improvement of convective concentration fluxes in a one step reactive flow solver," *J. Comput. Phys.*, vol. 204, no. 1, pp. 319–352, 2005.

[39] X. Gao and C. P. T. Groth, "Parallel adaptive mesh refinement scheme for three-dimensional turbulent non-premixed combustion," AIAA 2008-1017, 46th AIAA Aerospace Sciences Meeting, 2008.

[40] U. Ghia, K. N. Ghia, and C. T. Shin, "High-Re solutions for incompressible flow using the navier-stokes equations and a multigrid method," *J. Comput. Phys.*, vol. 48, no. 3, pp. 387–411, 1982.

[41] N. Attal, P. Ramaprabhu, J. Hossain, V. Karkhanis, M. Uddin, J. R. Gord, and S. Roy, "Development and validation of a chemical reaction solver coupled to the FLASH code for combustion applications," *Comput. Fluids*, vol. 107, pp. 59–76, 2015.

[42] C. R. Wilke, "A viscosity equation for gas mixtures," *J. Chem. Phys.*, vol. 18, pp. 517–519, 1950.

[43] G. Billet, V. Giovangigli, and G. de Gassowski, "Impact of volume viscosity on a shock–hydrogen-bubble interaction," *Combust. Theory Model.*, vol. 12, no. 2, pp. 221–248, 2008.

[44] R. D. Richtmyer, "Taylor instability in shock acceleration of compressible fluids," *Comm. Pure and Appl. Math.*, vol. 13, no. 2, pp. 297–319, 1960.

[45] E. E. Meshkov, "Instability of the interface of two gases accelerated by a shock wave," *Fluid Dynamics*, vol. 4, no. 5, pp. 101–104, 1969.

[46] G. Ben-Dor and I. I. Glass, "Domains and boundaries of non-stationary oblique shock-waves reflections. 2. monatomic gas," *J. Fluid Mech.*, vol. 94, pp. 735–756, 1980.

[47] X. Gao, L. D. Owen, and S. M. J. Guzik, "A parallel adaptive numerical method with generalized curvilinear coordinate transformation for compressible Navier-Stokes equations," *Int. J. Numer. Meth. Fluids*, vol. 82, pp. 664–688, 2016.

[48] J. Tryner, Q. An, J. Mohr, A. Steinberg, A. P. Yalin, and A. J. Marchese, "High-speed OH* chemiluminescence, OH PLIF, and acetone PLIF imaging of an inverse non-premixed cross-flow flame," in *Proceedings of the Western States Section of the Combustion Institute Spring 2016 Meeting*, WSSCI, 2016.

[49] C. Li, K. Kailasanath, and E. S. Oran, "Detonation structures behind oblique shocks," *Physics of Fluids*, vol. 6, no. 4, pp. 1600–1611, 1994.

[50] A. Thaker and H. K. Chelliah, "Numerical prediction of oblique detonation wave structures using detailed and reduced reaction mechanisms," *Combust. Theory Model.*, vol. 1, no. 4, pp. 347–376, 1997.

[51] T. Wang, Y. Zhang, H. Teng, Z. Jiang, and H. D. Ng, "Numerical study of oblique detonation wave initiation in a stoichiometric hydrogen-air mixture," *Physics of Fluids*, vol. 27, no. 9, p. 096101, 2015.

[52] P. R. Woodward, *Piecewise-Parabolic Methods for Astrophysical Fluid Dynamics*, pp. 245–326. Dordrecht: Springer Netherlands, 1986.