

Copyright © 2021 by Daniel Joseph Boe
All Rights Reserved

Abstract

High fidelity, robust numerical schemes for compressible multi-fluid dynamics are critical for advancing numerous active research fields including high-speed reactive turbulence, inertial confinement fusion, supernova explosions, and shock/ultrasound-induced bubble oscillations in medical diagnosis and therapy. Many computational methods have been developed to solve these challenging problems; however, there are few, if any, codes that demonstrate sufficient rigor and efficiency to successfully simulate the multi-dimensional compressible flows relevant in many cutting edge biomedical, sustainable energy, and defense technologies. Within the landscape of numerical methods, high-order, finite-difference schemes are unmatched in their simplicity and demonstrate optimal performance in terms of accuracy per CPU cost, making them a compelling option for these applications.

Hyperbolicity, or the retention of a real valued sound speed, is a required component for a robust computational scheme. Failure to maintain the positivity of physical values such as density and the square of sound speed causes the blow-up of numerical simulations. In the context of two-fluid compressible dynamics, strong shocks and very large interfacial discontinuities are common features that can easily induce positivity related failure in a simulation.

The objective of this work is to enhance a high-order, primitive variable based, weighted essentially non-oscillatory (WENO) finite difference scheme by adding a positivity-preserving algorithm to the method. This is accomplished by incorporating a flux limiting technique that locally adapts high-order fluxes towards first-order fluxes to retain the physical bounds of the system without general loss of high-order convergence. The proposed scheme is equally valid for both high-order linear (upwind or central) reconstruction and for high-order nonlinear reconstruction based on the WENO scheme.

This positivity-preserving scheme has been implemented up through 11th order in one and two dimensions for a two-fluid compressible model, and several test problems have been conducted to validate its performance. The scheme has been found to effectively retain high-order accuracy while allowing for the simulation of several challenging problems that otherwise could not be successfully solved using the base scheme.

For compressible single-fluid flow, the developed scheme is believed to be the first application of the parameterized, flux-limiting positivity approach to any finite difference scheme of accuracy order greater than five. Additionally, the scheme represents the first demonstration of a high-order, flux-limiting, finite difference solver for two-fluid compressible flows. Owing to the inherent efficiency of finite differences and the new robust positivity-preserving quality, the scheme is well suited for modeling the challenging physics associated with problems such as ultrasound activated targeted drug delivery, air/fuel mixing in supersonic flows, and collapse of cavitation bubbles.

Acknowledgements

I wish to thank my advisor Dr. Khosro Shahbazi whose infectious desire for learning and tireless passion for teaching have been of great inspiration to me. I also thank my committee members Dr. Albert Romkes and Dr. Brent Deschamp for their commitment to providing exceptional education. The fundamental engineering/math principles and unique insights I learned from them both have been invaluable. I also thank Mike Marzinske whose energetic presentations of college algebra and calculus first sparked my interest in mathematics; those memories at Inver Hills Community College have stayed with me throughout my academic journey. I also wish to thank my very first educator, my mother, who spent 12 years homeschooling me and taught me perhaps the single most valuable skill I'll ever obtain — how to read.

I am very grateful to my parents for their unending support and selfless commitment to ensuring I had opportunity to pursue an education. I thank my brothers for their encouragement and for leading the way as first generation college graduates giving me the confidence and resolve to do the same. I thank my grandmother, Solveig, whose prayers, letters, and kind words have supplied tremendous encouragement over the years. Finally, I thank Mr. Ardell Knudson whose kindness and generosity helped make my stay at SD Mines a wonderful experience.

This work is funded by the Office of Naval Research, Contract # N000141712965

Table of Contents

| | |
|----------------------------------------------------------|-----------|
| List of Tables | v |
| List of Figures | vi |
| 1 Introduction | 1 |
| 1.1 A Compressible Two-Fluid Model | 1 |
| 1.2 Applications | 5 |
| 1.2.1 Biomedical | 5 |
| 1.2.2 Sustainable Energy | 7 |
| 1.2.3 Propulsion | 7 |
| 1.2.4 Volcanology | 8 |
| 1.3 Numerical Design | 8 |
| 1.4 Outline and Objectives | 12 |
| 2 Theory | 13 |
| 2.1 Discretization | 13 |
| 2.2 Polynomial Reconstruction | 14 |
| 2.3 WENO Weights | 17 |
| 2.4 Mapped WENO Weights | 20 |
| 2.5 Spatial Derivative Calculation | 20 |
| 2.6 Non-Linear Conservation Laws | 23 |
| 2.7 System of 1D Non-linear Conservation Laws | 24 |
| 2.8 Two Dimensional System | 29 |
| 3 Positivity | 30 |
| 3.1 Motivation | 30 |
| 3.2 Flux Limiting Background | 32 |
| 3.3 Implementation for scalar case | 36 |
| 3.3.1 Enforcing the upper limit | 37 |
| 3.3.2 Enforcing the lower limit | 42 |
| 3.3.3 Selecting the Flux Limiters | 44 |
| 3.4 Positivity Implementation for Euler System | 44 |
| 3.5 Positivity of the two-fluid model | 48 |
| 3.6 Extension to Two Dimensions | 49 |
| 4 Verification | 53 |
| 4.1 Implementation | 53 |

| | | |
|----------|---------------------------------------------------------|-----------|
| 4.2 | Scalar Conservation Laws | 54 |
| 4.2.1 | Linear Advection with Continuous Initial Data | 54 |
| 4.2.2 | Inviscid Burger's Equation | 55 |
| 4.3 | Euler System | 56 |
| 4.3.1 | Interacting Blast Waves | 57 |
| 4.3.2 | Euler System — Shock, Interface Problem | 60 |
| 4.4 | Two-Fluid System | 64 |
| 4.4.1 | High Pressure Shock Tube | 64 |
| 4.4.2 | Water and Air Shock, Interface Problem | 68 |
| 4.5 | 2D Two Fluid System | 70 |
| 4.5.1 | Isentropic Vortex | 71 |
| 4.5.2 | Air Shock Helium Bubble Interaction | 74 |
| 4.5.3 | High Mach Shock Air Bubble Interaction | 79 |
| 4.6 | Impact on CPU Time | 84 |
| 5 | Conclusions | 86 |
| 5.1 | Summary | 86 |
| 5.1.1 | First Objective | 86 |
| 5.1.2 | Second Objective | 87 |
| 5.1.3 | Third Objective | 87 |
| 5.2 | Future Work | 88 |
| | Bibliography | 89 |
| | Vita | 97 |

List of Tables

| | | |
|------------|---------------------------------------------------------------------------------------------------------------------------|----|
| Table 1.1. | Performance comparison for first and fifth order schemes . . . | 9 |
| Table 2.1. | Interpolation coefficients $(a_{r,k_s,q})$ up to 6th order | 16 |
| Table 2.2. | Optimal reconstruction coefficients (b_r, k_s) up through 11th order | 18 |
| Table 2.3. | Derivative Coefficients (d_j) up to 12th order | 23 |
| Table 3.1. | Coefficients for conservative flux construction | 35 |
| Table 3.2. | Summary of flux limiters for ρ and Γ_1 in 2D | 51 |
| Table 4.1. | Convergence study for scalar linear advection — CFL=0.05 . . | 55 |
| Table 4.2. | Convergence study for Burger’s Equation — CFL=0.05 | 56 |
| Table 4.3. | L_1 convergence study for stationary isentropic vortex with $\epsilon = 5$ and $t = 0.1$ | 72 |
| Table 4.4. | L_1 convergence study for stationary isentropic vortex with $\epsilon = 9$ and $t = 0.1$ | 73 |
| Table 4.5. | Comparisons of shock/interface speeds [m/s] to numerical and experimental data. | 77 |
| Table 4.6. | Comparison of CPU cost per time step for scheme with limiters and without limiters for $(N, M) = (600, 150)$ | 85 |

List of Figures

| | | |
|--------------|------------------------------------------------------------------------------------------------------------------------------------|----|
| Figure 1.1. | Advected water/air interface for different interface capturing functions | 4 |
| Figure 1.2. | Advected water/air interface for different interface capturing functions - N=400 | 4 |
| Figure 1.3. | Focused shock wave interacting with a cylindrical epoxy sample. Reproduced with permission from Xi and Zhong [55] . . | 6 |
| Figure 1.4. | Comparison of first and fifth order finite difference schemes for advection equation with continuous solution | 9 |
| Figure 1.5. | Demonstration of Gibbs phenomena for 7th order solutions to an advected square pulse | 10 |
| Figure 2.1. | Schematic showing a computational grid | 13 |
| Figure 2.2. | Schematic showing a computational cell | 14 |
| Figure 2.3. | Schematic showing a local polynomial reconstruction | 15 |
| Figure 2.4. | Schematic showing stencil choices for a discontinuous function | 18 |
| Figure 3.1. | Demonstration of a negative pressure value developing in a shock tube with a pressure ratio of 1000 across the diaphragm | 31 |
| Figure 3.2. | Illustration of flux limiters for positive flux case | 39 |
| Figure 3.3. | Illustration of flux limiters for negative flux case. | 40 |
| Figure 4.1. | Pressure solutions for the interacting blast waves problem | 58 |
| Figure 4.2. | Density solutions for the interacting blast waves problem | 59 |
| Figure 4.3. | Comparison of high-order density solutions for interacting blast waves problem | 60 |
| Figure 4.4. | Demonstration of positivity-preserving scheme for strong shock-/interface interaction | 61 |
| Figure 4.5. | 5th order solution for shock air/helium interface problem | 62 |
| Figure 4.6. | 7th order solution for shock air/helium interface problem | 63 |
| Figure 4.7. | 9th order solution for shock air/helium interface problem | 63 |
| Figure 4.8. | Comparison of high and low order solutions for shock interface problem | 64 |
| Figure 4.9. | 3rd order solution to high pressure shock tube problem | 65 |
| Figure 4.10. | 5th order solution to high pressure shock tube problem | 66 |
| Figure 4.11. | 7th order solution to high pressure shock tube problem | 66 |
| Figure 4.12. | 9th order solution to high pressure shock tube problem | 67 |
| Figure 4.13. | 11th order solution to high pressure shock tube problem | 67 |

| | | |
|--------------|-----------------------------------------------------------------------------------------------------------------|----|
| Figure 4.14. | 11th order solution to the shock liquid/air interface problem . | 69 |
| Figure 4.15. | 5th order solution to the shock liquid/air interface problem . | 69 |
| Figure 4.16. | Comparison of high-order solutions for specific heat ratio and Π for liquid/air interface problem | 70 |
| Figure 4.17. | Initial density, pressure, and velocity fields for stationary isentropic vortex | 71 |
| Figure 4.18. | Air/helium shock bubble interaction — 11th order density solutions | 75 |
| Figure 4.19. | Air/helium shock bubble interaction — 11th order pressure solutions | 76 |
| Figure 4.20. | Air/helium shock bubble interaction — numerical Schlieren . | 78 |
| Figure 4.21. | Water/air shock bubble interaction — 7th order density solutions | 80 |
| Figure 4.22. | Water/air shock bubble interaction — 7th order pressure solutions | 82 |
| Figure 4.23. | Water/air shock bubble interaction — numerical Schlieren . . | 83 |
| Figure 4.24. | Air bubble collapse history for water/air shock bubble problem | 84 |

Chapter 1

Introduction

1.1 A Compressible Two-Fluid Model

Conservation laws are fundamental principles that form the basis for modeling numerous physical processes. In the simplest terms, a conservation law enforces the idea that, in a closed system, the overall quantity of some conserved property can only change by adding or removing portions of this property from the system. Mathematically, a conservation law is expressed as

$$\frac{\partial u(x, t)}{\partial t} + \frac{\partial}{\partial x} [f(u(x, t))] = 0, \quad (1.1)$$

where $u(x, t)$ is a function that describes the distribution of some conserved physical property in space and time, and f is a function giving the flux of the conserved property. In the context of gas dynamics, the conservation of mass, momentum, and energy in one spatial dimension form a set of three coupled equations known as the Euler equations of gas dynamics. These three equations together with the ideal gas equation of state ($p = e\rho(\gamma - 1)$) govern the transport of mass, momentum, and total energy for a compressible fluid. The Euler equations are expressed as

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x} [\rho v] = 0 \quad (1.2)$$

$$\frac{\partial}{\partial t} [\rho v] + \frac{\partial}{\partial x} [\rho v^2 + p] = 0 \quad (1.3)$$

$$\frac{\partial}{\partial t} \left[\rho \left(e + \frac{v^2}{2} \right) \right] + \frac{\partial}{\partial x} \left[\left(\rho e + \frac{\rho v^2}{2} + p \right) v \right] = 0, \quad (1.4)$$

where ρ , v , p , and e are density, velocity, pressure, and internal energy, respectively.

Numerical solutions to these equations allow for the simulation of complex, single-fluid gas dynamics with discontinuous features. In order to extend the model to support multi-fluid dynamics, we require some function that describes the distribution of the separate fluids in the flow field at any given time. Restricting ourselves to immiscible flow, this can be accomplished by adding an advection equation that evolves the location of the fluid interfaces [30].

The advection equation, also known as the transport equation, models the distribution of some function being carried by a bulk velocity field. For our application, we seek to model the location of a two-fluid interface described by some interface capturing function, Γ_1 , as it is advected by the local fluid velocity v . Thus, to support two-fluid dynamics, we can append the following equation to the Euler system

$$\frac{\partial \Gamma_1}{\partial t} + v \frac{\partial}{\partial x} [\Gamma_1] = 0. \quad (1.5)$$

The interface capturing function, Γ_1 , can be arbitrarily defined as any function of the material properties (e.g. specific heat ratio) that can sufficiently describe the fluid at any point in the computational domain. For the case of two ideal gasses, the most obvious choice for this function would be the specific heat ratio (γ). As an example, air ($\gamma = 1.4$) and helium ($\gamma = 1.67$) have different values for the specific heat ratio, so γ , by itself, could fully partition the domain into separate regions of air and helium. However, it is well known that the interface capturing function must be selected with care to avoid spurious oscillations near the interface. Specifically, we use the form of (1.6) which has been shown to give non-oscillatory solutions for linear advection [42, 30]

$$\Gamma_1 = \frac{1}{\gamma - 1}. \quad (1.6)$$

Even with the addition of (1.5), the model remains restricted to ideal gasses. To

extend the model to support liquids, we must utilize a different equation of state. The stiffened gas equation of state (1.7) is commonly used as a simple, generalized form of the ideal gas law [42]

$$p = (\gamma - 1)\rho e - \gamma\Pi, \quad (1.7)$$

where for a liquid, γ and Π are empirical parameters calculated by fitting the stiffened gas model to data from a Hugoniot curve [42, 27]. For all ideal gasses, γ corresponds to the specific heat ratio, and $\Pi = 0$. The inclusion of this second fluid parameter necessitates an additional interface capturing function and its linear advection equation

$$\frac{\partial\Gamma_2}{\partial t} + v\frac{\partial}{\partial x}[\Gamma_2] = 0. \quad (1.8)$$

Here, the interface capturing function is defined as

$$\Gamma_2 = \frac{\gamma\Pi}{\gamma - 1}. \quad (1.9)$$

The necessity of the specific forms of (1.6) and (1.9) for the two interface capturing functions can be easily demonstrated numerically by solving for the passive advection of an isolated material interface. To this end, Figures 1.1 and 1.2 show pressure and specific heat ratio solutions for a water/air interface at atmospheric conditions being advected by a constant fluid velocity of 10 m/s. A 5th-order finite difference scheme was used, and the solutions are shown after a single Runge-Kutta time integration step. In each figure, the left panel gives solutions when the equation of state order parameters (γ and Π) were used directly as the interface capturing functions, and the right panel shows the results when (1.6) and (1.9) were utilized.

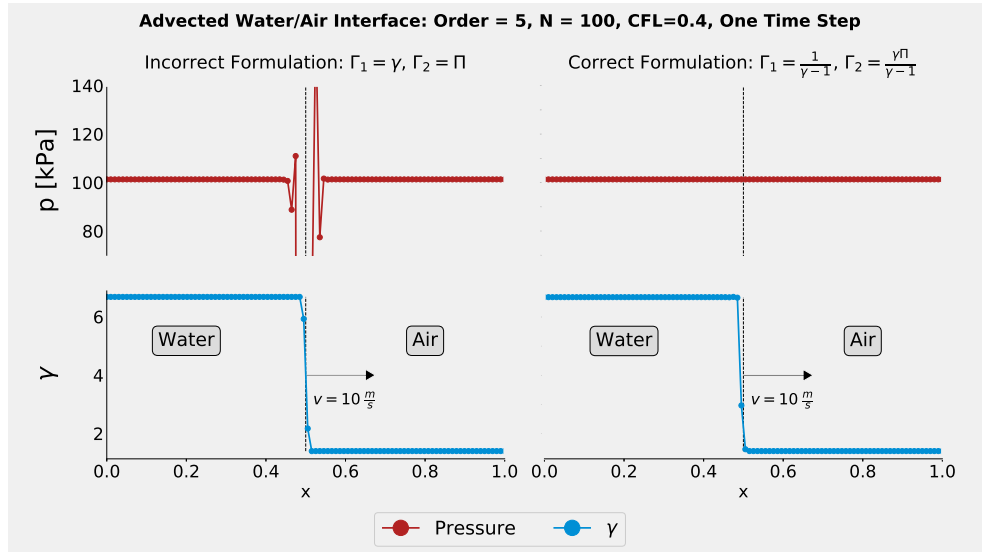


Figure 1.1: Advected water/air interface for different interface capturing functions

On the left panel, corresponding to the case where the order parameters were used directly, large, unphysical jumps are clearly seen in the pressure trace. However, as observed on the right-hand panel, the solutions obtained using the correct interface capturing functions are free of these spurious oscillations. Furthermore, as evidenced by Figure 1.2, these errors persist even with grid refinement.

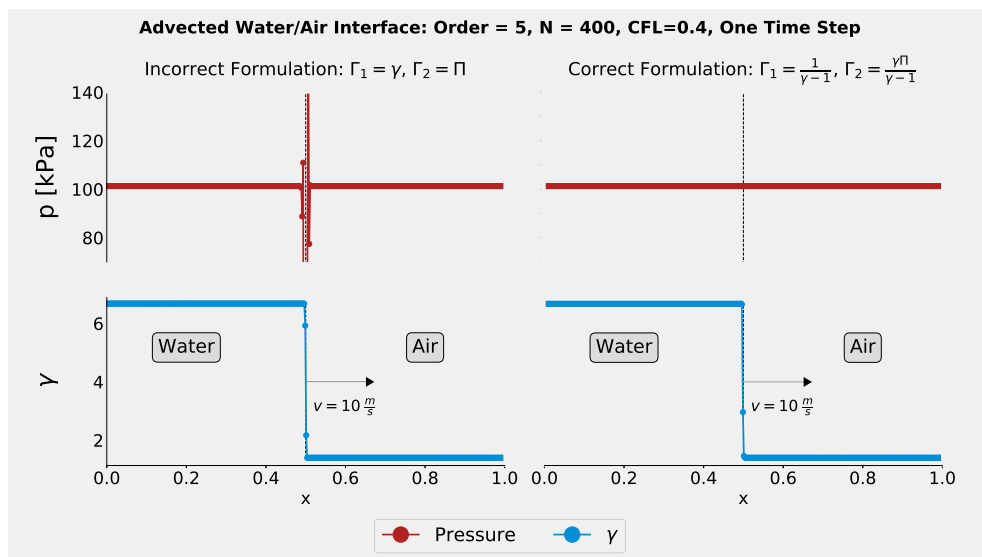


Figure 1.2: Advected water/air interface for different interface capturing functions - N=400

To summarize the model, taking $x_i = [x, y, z]^\top$ as the spatial coordinate, $E = \rho \left(e + \frac{v_i v_i}{2} \delta_{ij} \right)$ as the total energy, and v_i as the velocity vector, the complete set of equations for multi-fluid compressible flow in three spatial dimensions are

$$\frac{\partial \rho}{\partial t} + \frac{\partial}{\partial x_i} [\rho v_i] = 0 \quad (1.10)$$

$$\frac{\partial}{\partial t} [\rho v_i] + \frac{\partial}{\partial x_i} [(\rho v_i v_j) + p \delta_{ij}] = 0 \quad (1.11)$$

$$\frac{\partial E}{\partial t} + \frac{\partial}{\partial x_i} [(E + p)v_i] = 0 \quad (1.12)$$

$$\frac{\partial \Gamma_1}{\partial t} + v_i \frac{\partial \Gamma_1}{\partial x_i} = 0 \quad (1.13)$$

$$\frac{\partial \Gamma_2}{\partial t} + v_i \frac{\partial \Gamma_2}{\partial x_i} = 0. \quad (1.14)$$

1.2 Applications

The applications of compressible multi-phase fluid flow span an extremely diverse range of active research fields and engineering disciplines. Consequently, the development of efficient, robust numerical simulations has potential to impact the furtherance of numerous different technologies and theories.

1.2.1 Biomedical

Equations (1.10) – (1.14) form a model that supports multi-dimensional, multi-species flows comprised of vapor and/or liquid. For instance, with this model, we could simulate a shock wave traveling in liquid water as it interacts with an air bubble suspended in the liquid.

The study of such shock/bubble interactions is highly applicable to the biomedical field. Shock induced lithotripsy has been used since the 1980's as a non-intrusive treatment to break up kidney stones by targeting the calculi with high-energy shock waves generated by lithotripter. Most modern lithotripters utilize an electromagnetic or piezoelectric driver to produce a focused pulse with a maximum pressure jump of

about 40MPa [12]. Figure 1.3 shows photoelastic images from a laboratory experiment carried out utilizing a standard medical lithotripter to target a model stone.

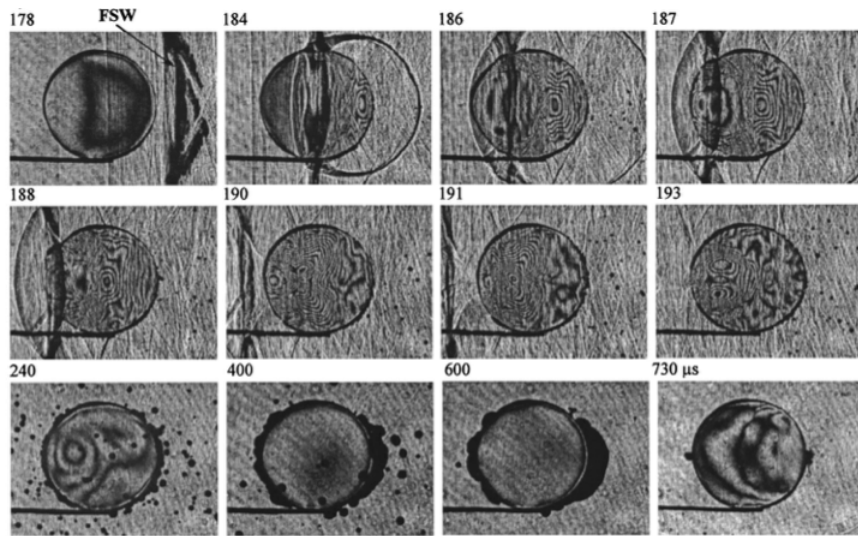


Figure 1.3: Focused shock wave interacting with a cylindrical epoxy sample. Reproduced with permission from Xi and Zhong [55]

The first image in the upper left shows the incident shock wave approaching the model stone from right to left. Subsequent images along the first and second row show the shock wave traveling through the stone and the resulting stress field. In the third row, cavitation bubbles are observed as dark spots collecting on the stone's surface. These cavitation bubbles are generated because the wave forms produced by lithotripters contain a tensile phase in the wake of the shock front. This region of negative pressure (typically around -10MPa) is well known to cause cavitation within urine and tissue [12]. The collapse of these cavitation bubbles generates very high pressure and is believed to be a primary mechanism in breaking apart stones [12, 20]. However, the destructive power of cavitation bubbles may also cause serious collateral damage to tissue and blood vessels. [29, 20].

In addition to these treatments, recent research has identified acoustic droplet vaporization as a potential mechanism for targeted drug delivery [53, 40, 37]. The approach involves transporting a drug encased in a lipid or polymeric droplet and

using ultrasound pulses to rupture the shell, releasing the drug at the targeted cells [48]. A similar technique has been proposed as a means of blocking blood supply to cancerous tumors by acoustically vaporizing perfluorocarbon droplets [35, 14]. Each of these treatments involve the precise usage of acoustic waves interacting with small bubbles immersed in fluid or tissue, and the development of safe, effective procedures can be augmented by the usage of high-fidelity numerical simulations.

1.2.2 Sustainable Energy

The general case of shock/interface interaction also finds application outside the biomedical industry. The so-called Richtmyer-Meshkov instability occurs when a shock wave impinges upon the interface of two fluids of different densities. The result is the impulsive acceleration of the fluids and the development of local turbulence [41, 36]. Accurate modeling of these instabilities is very relevant within the nuclear energy field. Specifically, inertial confinement fusion (ICF) requires the rapid compression and heating of a fuel pellet by way of concentrated laser beams. This process forms shockwaves which generate Richtmyer-Meshkov instabilities inducing undesired mixing within the fusion fuel pellet, significantly reducing the net energy output [8, 52]. Novel techniques for attenuating these instabilities is an active area of research [1] that requires accurate modeling of the small-scale interfacial features associated with shock/bubble interactions.

1.2.3 Propulsion

The scramjet engine was initially theorized in the 1950s as an efficient propulsion system for hypersonic aircraft [15]. It is unique among air breathing engines in that fuel injection and combustion occur in supersonic flow resulting in very short timescales for achieving sufficient fuel-oxidizer mixing [43, 33].

By utilizing various flow control and injection techniques, Richtmyer-Meshkov

instability can be used to promote more complete fuel-oxidizer mixing in scramjets [58, 59]. Designing such systems is very challenging as the relevant physics include high Mach shocks interacting with two or more fluids of differing densities and properties. High-fidelity, rigorous numerical simulations serve as an invaluable tool for building broader understanding of these phenomena and informing engineering design.

1.2.4 Volcanology

Vulcanian eruptions form a specific class of unsteady volcanic activity characterized by the rapid decompression of a vertical magma-filled channel [10, 11]. These eruptions are usually short lived lasting less than a few minutes and are typically preceded by the sudden failure of the containing structure that seals the channel from atmospheric pressure. The dynamics of a Vulcanian eruption consist of a high pressure shock wave propagating from the vent into the atmosphere followed by the expulsion of a gas/magma particle mixture. Additionally, an expansion wave travels down the channel away from the vent triggering a pyroclast producing fragmentation wave. Because the resulting solid particles are very small (about 1mm in diameter), these eruptions can be modeled using the compressible flow equations under the assumption that the pyroclastic particles are accelerated by the bulk velocity of an ideal gas [9, 10]. Such numerical simulations can provide insight into key variables such as plume height, velocity, temperature, and total mass erupted, giving researchers useful tools to assess the scope and potential hazards of volcanic events.

1.3 Numerical Design

When modeling any physical process, it is vital that the solution method retain the physical bounds of the system. For the case of the multi-fluid compressible flow, several phenomena pose specific challenges to numerical methods. The presence of both shock and fluid interface discontinuities is inherent, and any solution method

must be able to accurately capture these features without unphysical distortions.

It is well known that simple first order finite difference or finite volume schemes can capture shock discontinuities without spurious oscillation; however, their large dissipative error, which is inherent in low-order approximations, renders the first-order schemes prohibitively expensive for many applications, especially for problems in multi-space dimensions [26, 45]. A demonstration of this provided in Figure 1.4.

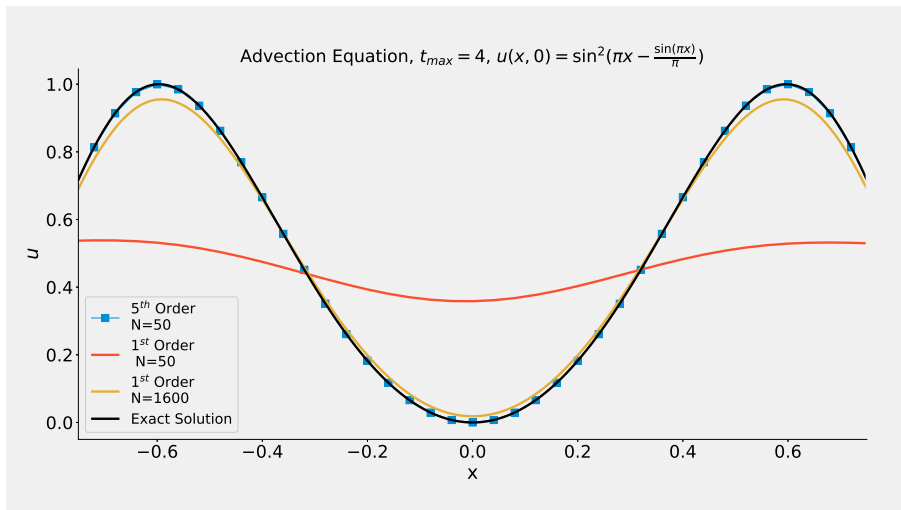


Figure 1.4: Comparison of first and fifth order finite difference schemes for advection equation with continuous solution

This plot shows a fifth order finite difference solution, as well as two 1st order solutions to the linear advection equation ($u_t + u_x = 0$) for a smooth initial condition. Note that the 1st order scheme required 32 times more grid points to achieve accuracy approaching that of the 5th order scheme. Table 1.3 quantifies the superior performance of the high-order scheme in terms of accuracy per CPU time.

Table 1.1: Performance comparison for first and fifth order schemes

| | First Order N=1600 Grid Points | Fifth Order N=50 Grid Points |
|--------------------|-----------------------------------|---------------------------------|
| L_∞ Error % | 4.86% | 0.13% |
| CPU Time [sec] | 0.32 s | 0.20 s |

It should be noted that this was a very simple problem with smooth, low amplitude features; even so, the higher order scheme was significantly more efficient. With more challenging problems containing features such as shocks and high-frequency oscillations, the benefits of high-order methods are even more pronounced [45, 44].

Given the greater efficiency of high-order methods, it is evident that high-order shock capturing schemes are desirable; unfortunately, simple high-order schemes are entirely unable to handle discontinuities. Regardless of grid refinement, they suffer from spurious oscillations near sharp features. The so-called Gibbs phenomena [19, 24] is demonstrated in Figure 1.5, which shows 7th order finite difference solutions for the linear advection of a discontinuous function.

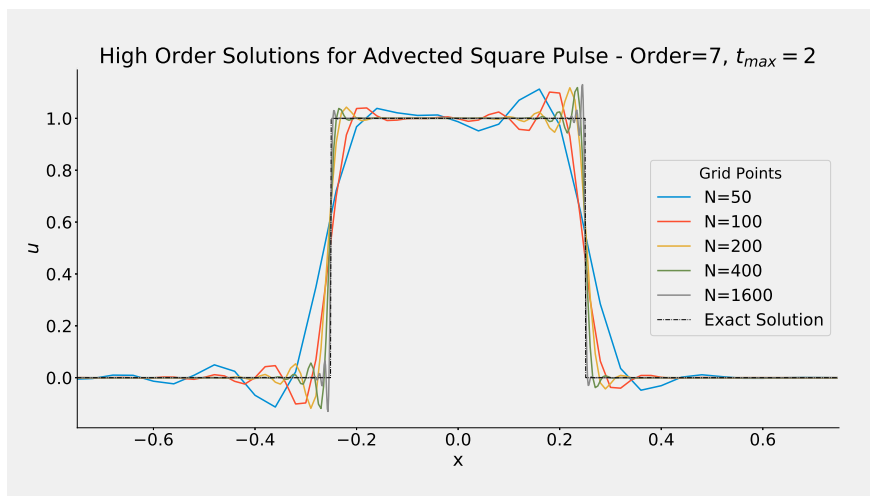


Figure 1.5: Demonstration of Gibbs phenomena for 7th order solutions to an advected square pulse

To overcome this challenge, numerous high-order schemes have been developed. In particular, the WENO (weighted essentially non-oscillatory) schemes first introduced by Liu, Osher, and Chan [34] and later improved by Jiang and Shu [26] have proven very successful at providing high-order convergence in smooth regions while avoiding spurious oscillations at discontinuities. The WENO method is a high-order, nonlinear reconstruction technique that has been applied to various finite difference [45] and finite volume schemes [44]. In this work, we pursue a finite difference method and

build upon the scheme proposed by Shahbazi [45] who demonstrated the superior efficiency and simplicity of finite differences applied to complex two-fluid and two-phase flow models and, owing to their optimal efficiency, noted their potential to be extended to three spatial dimensions.

For an ideal gas, alongside non-oscillatory shock/interface capturing, it is necessary for a numerical method to enforce certain physical limits in the solution, namely the positivity of pressure and density. If either of these variables was to obtain a value below zero, this would imply a complex value for the sound speed ($c = \sqrt{\frac{\gamma p}{\rho}}$) leading to the breakdown of the simulation. For the two-phase model, the physical bounds deviate slightly from the ideal gas case as it can be shown that this model, to some degree, supports negative pressure values [47]. In particular, under the stiffened gas equation of state, the expression for sound speed becomes

$$c = \sqrt{\frac{\gamma(p + \Pi)}{\rho}}. \quad (1.15)$$

Thus, for the model described in (1.10) – (1.14), the required positivity conditions become $\rho > 0$ and $p + \Pi > 0$. The positivity of these variables is guaranteed (under a CFL condition and under a concavity condition for the material parameters γ and Π) only for the first order scheme [47, 46]; it is not guaranteed for high-order solutions — even for *essentially* non-oscillatory methods such as the WENO scheme. In numerous cases, guaranteed positivity preservation is not required of the numerical procedure as the physics do not generate low enough pressure and density values to cause issue. However, with many problems — particularly those involving very large discontinuities — such as liquid/air shock bubble interaction, absolute positivity preservation is essential to prevent the simulation from crashing [44].

1.4 Outline and Objectives

In this thesis, we will build upon the high-order finite difference WENO scheme developed by Shahbazi [45] by incorporating a flux-limiting positivity preservation method based upon the works of Xu [57] and Xiong et al. [56]. In Chapter 2, we provide a review of the finite difference WENO scheme. Chapter 3 discusses the application of the flux-limiting approach to our finite difference scheme and considers the extension of the method to the two-fluid model. Chapter 4 gives verification for the scheme by presenting several convergence studies and demonstrating its ability to solve several challenging test problems. Finally, Chapter 5 will provide closure by summarizing findings and identifying relevant future work.

The specific objectives of this thesis are as follows:

- Implement the finite difference WENO scheme of Shahbazi [45] for the two-fluid model of (1.10) – (1.14) in one and two spatial dimensions up to and including 11th order.
- Apply the flux-limiting positivity-preserving framework from [57, 56] to this finite difference scheme in one and two dimensions.
- Verify the performance of the new positivity-preserving scheme.

Chapter 2

Theory

This chapter provides an overview of high-order WENO methods. The standard WENO framework is reviewed, and the specifics unique to our finite difference method are discussed.

2.1 Discretization

To simplify the description of the numerical scheme, let us begin by focusing upon the one-dimensional conservation law with $u(x, t)$ being the conserved variable and $f(u(x, t))$ being the flux function of the conserved variable. The framework developed here is easily extended to the two-fluid system of (1.10) – (1.14), since we are simply dealing with five coupled conservation laws of the form

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0. \quad (2.1)$$

Our goal is to obtain high-order, non-oscillatory approximations to the spatial derivative term. To begin, we consider the discrete form of (2.1) defined on a grid of N points where $x_i \in \{x_0, x_1, x_2, \dots, x_{N-2}, x_{N-1}\}$, h is the constant step size between each point, and τ is the step size in time. We can denote the solution at a particular point x_i and time t_n as u_i^n .

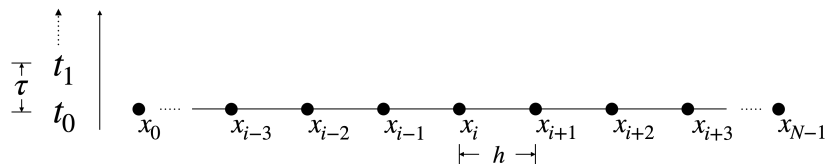


Figure 2.1: Schematic showing a computational grid

The starting point of the numerical scheme is to produce a polynomial reconstruction of $u(x, t)$ at the so-called “cell boundary points”. This notion of a computational cell is central to non-oscillatory finite difference/volume schemes. We can think of each point x_i in the discrete domain as the center of a small volume or “cell” in which the midpoints shared between adjacent x_i ’s define the cell boundaries. In Figure 2.2, we define the cell I_j as consisting of the “cell-center” value x_i , the left cell boundary $x_{i-\frac{1}{2}}$, and the right cell boundary $x_{i+\frac{1}{2}}$. In the most general sense, the WENO procedure involves building a high-order, local approximation to u at each of these cell boundaries by utilizing a carefully weighted combination of multiple different local approximations, each calculated using a different collection of nearby points. What we do with that reconstruction and exactly which functions we reconstruct differs from scheme to scheme, but the basic WENO procedure remains essentially the same. Given a function u known on the sample points x_i , we seek a polynomial approximation to u at the cell boundary points $x_{i\pm\frac{1}{2}}$.

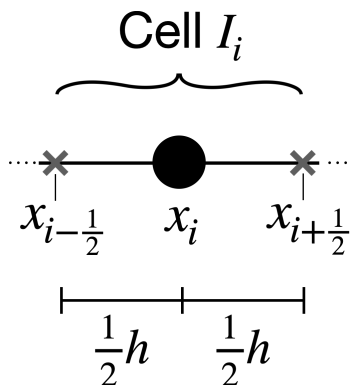


Figure 2.2: Schematic showing a computational cell

2.2 Polynomial Reconstruction

To begin, we consider the r^{th} order accurate local polynomial reconstruction of u near x_i using a stencil (a collection of points) surrounding x_i . For illustrative purposes, we take $r = 4$; hence, we need to select 4 points to comprise the stencil.

Figure 2.3 shows one potential stencil, $\{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$, which defines a set of four points we will use to build the approximation.

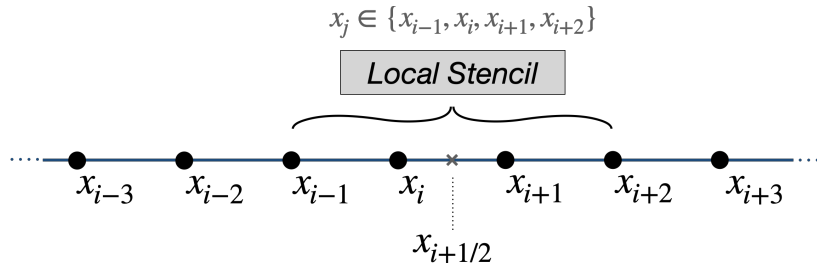


Figure 2.3: Schematic showing a local polynomial reconstruction

Following the Lagrange polynomial approach, we ultimately seek a polynomial that gives $u(x_j)$ when evaluated at x_j and zero everywhere else in the local stencil. Consider the function

$$\phi_j(x) = \frac{\prod_{p=i-1}^{p=i+2} (x - x_p)}{\prod_{p=i-1}^{p=i+2} (x_j - x_p)}, \quad \text{where } j \neq p. \quad (2.2)$$

Utilizing this polynomial, we define a function that is equal to $u(x_j)$ for all j as

$$P_r(x) = \sum_{j=0}^{j=r-1} \phi_j(x)u(x_j). \quad (2.3)$$

Referencing Figure 2.3, it is clear that there are other sets of four adjacent points that could be used to build the polynomial $P_r(x)$. There is no reason why we could not have selected $\{x_i, x_{i+1}, x_{i+2}, x_{i+3}\}$, $\{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$, or $\{x_{i-3}, x_{i-2}, x_{i-1}, x_i\}$ as either set would have provided a 4th order approximation to $u(x_{i+\frac{1}{2}})$ without extrapolation. Specifically, for an r order reconstruction polynomial $P_r(x)$, there are r suitable stencil choices of r points each. Hereafter, we will denote the stencil selection using the index k_s , where $k_s = 0$ represents the stencil consisting of the right-most points and $k_s = r - 1$ contains the left-most points; thus as k_s increases, the stencil is incrementally shifted leftward. As an example, for a second order approximation, we would have $r = 2$ with two possible stencil choices, $k_s = 0$ yielding the stencil

$x_j \in \{x_i, x_{i+1}\}$ and $k_s = 1$ corresponding to stencil $x_j \in \{x_{i-1}, x_i\}$.

These two “sub-stencils” constitute an overall left-biased reconstruction because each sub-stencil contains the point x_i which falls to the *left* of the cell boundary $x_{i+\frac{1}{2}}$. However, the cell boundary value $u(x_{i+\frac{1}{2}})$ could be approximated to the same accuracy by instead including the *rightward* point x_{i+1} in each sub-stencil. The notion of a right or left biased scheme is applicable when solving non-linear conservation laws; the extension from the left-biased is straightforward, so we will only discuss the latter here. Using this convention, we obtain an expression for the left-biased polynomial reconstruction by evaluating (2.3) at $x = x_{i+\frac{1}{2}}$ and solving the resulting system to obtain

$$u(x_i + \frac{1}{2}h) \approx P_{r,k_s,i+\frac{1}{2}} = \sum_{q=0}^{q=r-1} a_{r,k_s,q} u_{i-k_s+q}, \tag{2.4}$$

where $a_{r,k_s,q}$ is constant and given by

$$a_{r,k_s,q} = \frac{\prod_{p=0}^{p=r} (\frac{1}{2} - k_s + p)}{(-1)^q (q!) (r - q)!}, \quad \text{where } q \in [0, r] \quad \text{and } p \neq q. \tag{2.5}$$

Table 2.1 below gives the coefficients for up to 6th order.

Table 2.1: Interpolation coefficients ($a_{r,k_s,q}$) up to 6th order

| r | 2 | | 3 | | 4 | | | 5 | | | | 6 | | | | | | | | |
|-------|---------------|----------------|----------------|----------------|----------------|-----------------|-----------------|-----------------|------------------|------------------|------------------|------------------|------------------|-------------------|--------------------|-------------------|-------------------|-------------------|--------------------|---------------------|
| k_s | 0 | 1 | 0 | 1 | 2 | 0 | 1 | 2 | 3 | 0 | 1 | 2 | 3 | 4 | 0 | 1 | 2 | 3 | 4 | 5 |
| q=0 | $\frac{1}{2}$ | $-\frac{1}{2}$ | $\frac{3}{8}$ | $-\frac{1}{8}$ | $\frac{3}{8}$ | $\frac{5}{16}$ | $-\frac{1}{16}$ | $\frac{1}{16}$ | $-\frac{5}{16}$ | $\frac{35}{128}$ | $-\frac{5}{128}$ | $\frac{3}{128}$ | $-\frac{5}{128}$ | $\frac{35}{128}$ | $\frac{63}{256}$ | $-\frac{7}{256}$ | $\frac{3}{256}$ | $-\frac{3}{256}$ | $\frac{7}{256}$ | $-\frac{63}{256}$ |
| q=1 | $\frac{1}{2}$ | $\frac{3}{2}$ | $\frac{3}{4}$ | $\frac{3}{4}$ | $-\frac{5}{4}$ | $\frac{15}{16}$ | $\frac{9}{16}$ | $-\frac{5}{16}$ | $\frac{21}{16}$ | $\frac{35}{32}$ | $\frac{15}{32}$ | $-\frac{5}{32}$ | $\frac{7}{32}$ | $-\frac{45}{32}$ | $\frac{315}{256}$ | $\frac{105}{256}$ | $-\frac{25}{256}$ | $\frac{21}{256}$ | $-\frac{45}{256}$ | $\frac{385}{256}$ |
| q=2 | | | $-\frac{1}{8}$ | $\frac{3}{8}$ | $\frac{15}{8}$ | $-\frac{5}{16}$ | $\frac{9}{16}$ | $\frac{15}{16}$ | $-\frac{35}{16}$ | $-\frac{35}{64}$ | $\frac{45}{64}$ | $\frac{45}{64}$ | $-\frac{35}{64}$ | $\frac{189}{64}$ | $-\frac{105}{128}$ | $\frac{105}{128}$ | $\frac{75}{128}$ | $-\frac{35}{128}$ | $\frac{63}{128}$ | $-\frac{495}{128}$ |
| q=3 | | | | | | $\frac{1}{16}$ | $-\frac{1}{16}$ | $\frac{5}{16}$ | $\frac{35}{16}$ | $\frac{7}{32}$ | $-\frac{5}{32}$ | $\frac{15}{32}$ | $\frac{35}{32}$ | $-\frac{105}{32}$ | $\frac{63}{128}$ | $-\frac{35}{128}$ | $\frac{75}{128}$ | $\frac{105}{128}$ | $-\frac{105}{128}$ | $\frac{693}{128}$ |
| q=4 | | | | | | | | | | $-\frac{5}{128}$ | $\frac{3}{128}$ | $-\frac{5}{128}$ | $\frac{35}{128}$ | $\frac{315}{128}$ | $-\frac{45}{256}$ | $\frac{21}{256}$ | $-\frac{25}{256}$ | $\frac{105}{256}$ | $\frac{315}{256}$ | $-\frac{1155}{256}$ |
| q=5 | | | | | | | | | | | | | | | $\frac{7}{256}$ | $-\frac{3}{256}$ | $\frac{3}{256}$ | $-\frac{7}{256}$ | $\frac{63}{256}$ | $\frac{693}{256}$ |

2.3 WENO Weights

As previously noted, for any approximation order r , we can form r unique polynomial reconstructions by utilizing different stencils (k_s values). The central idea behind WENO involves forming a weighted sum of each unique approximation ($P_{r,k_s,i+\frac{1}{2}}$) to create an overall approximation $P_{r,i+\frac{1}{2}}^{\text{WENO}*}$. This polynomial takes the form

$$P_{r,i+\frac{1}{2}}^{\text{WENO}*} = \sum_{k_s=0}^{k_s=r-1} b_{r,k_s} P_{r,k_s,i+\frac{1}{2}}. \quad (2.6)$$

If the approximations at each sub-stencil are weighted appropriately such that lower order terms in the Taylor series expansion or $P_{r,k_s,i+\frac{1}{2}}$ cancel, we could use all the available points resulting in an interpolation of higher order. For instance, letting $r = 3$ and $k_s = [0, 1, 2]$, from (2.4), we see that in calculating each of the three polynomials, we utilized the five unique points $\{u_{i-2}, u_{i-1}, u_i, u_{i+1}, u_{i+2}\}$. Hence, if we combine these five points such that the overall coefficients equal those of the polynomial approximation for $r = 5, k_s = 2$, we could obtain the same approximation as if we had utilized $r = 5$ with $k_s = 2$ to begin with. This allows for the definition of a set of optimal weights b_{r,k_s} that give a $2r - 1$ order approximation from the r unique r th order approximations. The equation defining these optimal weights is found by substituting (2.4) into (2.6) and equating the coefficients of each point to obtain

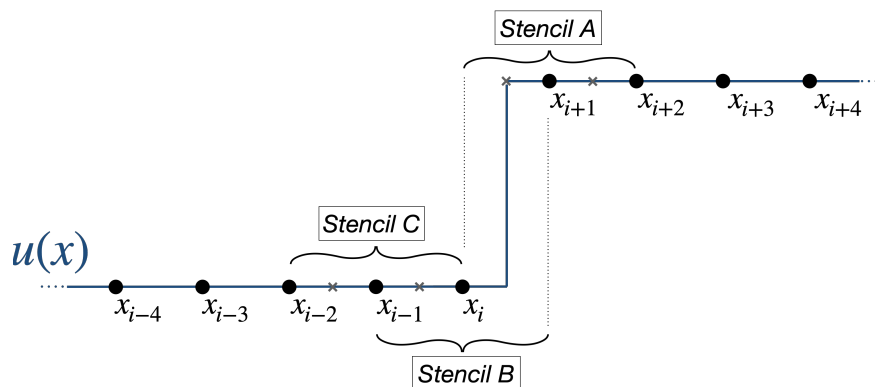
$$\sum_{k_s=0}^{r-1} b_{r,k_s} u_{r,k_s,i+\frac{1}{2}} = \sum_{q=0}^{q=2(r-1)} (a_{2r-1,r-1,q}) u_{i-(r-1)+q}. \quad (2.7)$$

Using (2.7), we can construct a linear system to solve for each b_{r,k_s} . Table 2.2 provides these optimal coefficients up through $r = 6$ (which would yield an 11th order overall approximation).

Table 2.2: Optimal reconstruction coefficients (b_r, k_s) up through 11th order

| k_s | 0 | 1 | 2 | 3 | 4 | 5 |
|-------|-------------------|--------------------|-------------------|-------------------|-------------------|------------------|
| r | | | | | | |
| 2 | $\frac{3}{4}$ | $\frac{1}{4}$ | | | | |
| 3 | $\frac{5}{16}$ | $\frac{5}{8}$ | $\frac{1}{16}$ | | | |
| 4 | $\frac{7}{64}$ | $\frac{35}{64}$ | $\frac{21}{64}$ | $\frac{1}{64}$ | | |
| 5 | $\frac{9}{256}$ | $\frac{21}{64}$ | $\frac{63}{128}$ | $\frac{9}{64}$ | $\frac{1}{256}$ | |
| 6 | $\frac{11}{1024}$ | $\frac{165}{1024}$ | $\frac{231}{512}$ | $\frac{165}{512}$ | $\frac{55}{1024}$ | $\frac{1}{1024}$ |

Equation (2.6) along with Table 2.2 give the best case or optimal combination of the approximations from each sub-stencil for a given r . However, this is only true in smooth regions; consider the case shown in the Figure 2.4, which shows a function that is discontinuous at some point between x_i and x_{i+1} . We seek to approximate the point $x_{i+\frac{1}{2}}$ to high order, but the direct usage of (2.6) with the optimal weights would be impacted by Gibbs phenomena. Stencils “A” and “B” both contain a discontinuity, and if either are used, the reconstruction will be oscillatory.

**Figure 2.4:** Schematic showing stencil choices for a discontinuous function

The WENO scheme seeks to handle this exception by utilizing a smoothness measuring feedback mechanism that locally modifies the optimal weights of Table 2.2 such that sub-stencils containing discontinuities are not used in the overall approximation.

The WENO smoothness indicators were defined by Jiang and Shu [26] as

$$\beta_{r,k_s,i} = \sum_{l=0}^{r-1} h^{2m-1} \int_{-\frac{1}{2}h}^{\frac{1}{2}h} \left[\frac{d^m}{d\xi} P_{r,k_s,i+\frac{1}{2}}(\xi) \right]^2 d(\xi - x_i), \quad (2.8)$$

which is just the sum of the L^2 norms of all derivatives of the interpolating polynomial defined in (2.4). These smoothness indicators are combined with the optimal weights as follows:

$$\alpha_{r,k_s} = \frac{b_{r,k_s}}{(\beta_{r,k_s,i} + 10^{-40})^2} \quad (2.9)$$

$$\omega_{r,k_s} = \frac{\alpha_{r,k_s}}{\sum_{j=0}^R \alpha_{r,j}}. \quad (2.10)$$

Finally, the overall WENO reconstruction polynomial of order $2r - 1$ takes on the form

$$P_{r,i+\frac{1}{2}}^{WENO} = \sum_{k_s=0}^{k_s=R} \omega_{r,k_s,i+\frac{1}{2}} P_{r,k_s,i+\frac{1}{2}}. \quad (2.11)$$

The WENO weights, α_{r,k_s} , are inversely proportional to the smoothness indicators, β , so for a sub-stencil containing a discontinuity, these weights will tend towards zero, effectively eliminating the contribution of the sub-stencil from the overall reconstruction. However, for a smooth stencil where $\beta = 1$, they will tend towards the optimal weights, b_{r,k_s} . Hence, in smooth regions, the WENO reconstruction of (2.11) approaches the optimal $2r - 1$ order reconstruction of (2.6). The results of (2.8) are tabulated for $r \in \{2, 3, 4, 5, 6\}$ in the literature [26, 4] and for higher orders in [18].

In summary, the WENO process involves taking some arbitrary function, w , known on the grid $\{x_0, x_1, \dots, x_{N-1}\}$ and approximating w at the cell boundaries $\{x_{\frac{1}{2}}, x_{\frac{3}{2}}, \dots, x_{N-\frac{1}{2}}\}$ using a non-linear interpolating polynomial. Hereafter, we denote this overall procedure for a left biased reconstruction as $L(w_{x_i}) = \hat{w}_{x_{i+\frac{1}{2}}}^{\text{left}} \approx w_{x_{i+\frac{1}{2}}}$. Similarly, $R(w_{x_i}) = \hat{w}_{x_{i+\frac{1}{2}}}^{\text{right}} \approx w_{x_{i+\frac{1}{2}}}$ represents a right biased WENO reconstruction.

2.4 Mapped WENO Weights

It is well known that the WENO weights of Jiang and Shu [26] may fail to obtain the expected convergence rate for a smooth function in the presence of critical points where the first derivative is zero. This reduction in accuracy arises because at critical points, the smoothness indicators of (2.8) may not exist due to vanishing derivatives. This behavior has been studied extensively in the literature [23, 17, 50, 5]. The widely adopted fix for this issue, introduced by Henrick et al. [23], utilizes a simple mapping function on the non-linear weights to correct (2.8) in the case of vanishing derivatives.

Incorporating this modification into our scheme is trivial. After calculating ω_{r,k_s} using (2.10), we simply evaluate the mapping function g_{r,k_s} and obtain a new set of mapped weights defined by

$$\alpha_{k_s}^* = g_{r,k}(\omega_{r,k_s}) = \frac{\omega_{r,k_s}(b_{r,k_s} + b_{r,k_s}^2 - 3(b_{r,k_s})\omega_{r,k_s} + \omega_{r,k_s}^2)}{b_{r,k_s}^2 + \omega_{r,k_s}(1 - 2b_{r,k_s})}. \quad (2.12)$$

These $\alpha_{k_s}^*$ weights are then substituted back into (2.10) in place of α_{k_s} to obtain the non-linear weights. It has been shown [23] that this simple modification will preserve the $2r - 1$ convergence rates for all $r \geq 3$ in the presence of a critical point. Thus, to avoid reduced accuracy at critical points, we utilize the mapping function (2.12) in our scheme.

2.5 Spatial Derivative Calculation

So far, we have discussed the WENO polynomial reconstruction of a function $u(x_i, t^n)$ in order to obtain an approximation at the cell boundary points $u(x_{i+\frac{1}{2}}, t^n)$. In the context of the conservation law shown in (1.1), we now have a method for a high-order approximation of the conserved variable $u(x, t)$. The next step is to calculate the spatial derivative term $\frac{\partial}{\partial x}[f(u)]$ using the WENO reconstruction; however, before proceeding, it is important to highlight some key differences among WENO

implementations.

Traditionally, the WENO reconstruction is not performed on the conserved variable u ; instead it is performed upon the *flux* function $f(u)$. That is, rather than forming

$$L(u(x, t)) = P_{i+\frac{1}{2}}^{WENO} \approx u_{i+\frac{1}{2}}, \quad (2.13)$$

most WENO implementations [26, 4] have instead been built upon

$$L(f(u(x, t))) = P_{i+\frac{1}{2}}^{WENO} \approx f\left(u_{i+\frac{1}{2}}\right). \quad (2.14)$$

To calculate the spatial derivative, this approach typically considers an unknown “numerical flux function” h defined [49] as

$$f(u(x)) = \frac{1}{\Delta x} \int_{x-\frac{1}{2}\Delta x}^{x+\frac{1}{2}\Delta x} h(u(\xi)) d\xi. \quad (2.15)$$

By its definition, when averaged over a cell, h gives the known cell-center fluxes $f(u(x_i))$. By differentiating both sides of this equation and applying the Leibniz rule, we obtain

$$\left. \frac{\partial}{\partial x} [f(u)] \right|_{x_i} = \frac{h(u_{i+\frac{1}{2}}) - h(u_{i-\frac{1}{2}})}{\Delta x}. \quad (2.16)$$

This equation indicates that if the function $h(u_{i\pm\frac{1}{2}})$ were known, then the derivative of $f(u(x_i))$ could be exactly calculated as above. Hence, to obtain a high-order derivative approximation, the numerical flux function is approximated using the boundary values of the flux $f_{i+\frac{1}{2}}$ which are calculated using the WENO procedure

$$h(u(x_{i+\frac{1}{2}})) \approx L(f(u(x_i, t))) = \hat{f}(u(x_{i+\frac{1}{2}})). \quad (2.17)$$

The traditional approach to a WENO scheme involves directly reconstructing the flux $f(u(x, t))$ and then substituting this result into (2.16) to approximate the spatial

derivative term. For the Euler system, this method would involve finding the WENO reconstructions of the vector of fluxes, $[\rho v, \rho v^2 + p, (\rho e + \frac{\rho v^2}{2} + p)v]$. However, for the case of the multi-fluid model, this approach fails to achieve pressure equilibrium across an interface leading to oscillatory solutions [45, 28]. The remedy is to perform the WENO reconstruction directly upon the primitive variables and then evaluate the fluxes using these approximations [45, 28]. That is, for the Euler system, the WENO reconstructions of ρ , v , and p are first calculated, and then these approximations are utilized to obtain the fluxes $f(x_{i+\frac{1}{2}})$.

Continuing within the framework of a one-dimensional conservation law (2.1), this approach is expressed as

$$f(u(x_{i+\frac{1}{2}})) \approx f(L(u(x_i, t))) = f(\hat{u}(x_{i+\frac{1}{2}})). \quad (2.18)$$

Under this scheme, the calculated cell boundary fluxes $f(\hat{u})$ do not approximate the numerical flux function h , and it can be shown that (2.16) is not valid and will yield a second order approximation regardless of the order of the WENO reconstruction of $u(x, t)$ [46]. In order to obtain high-order derivative calculations, we must utilize additional terms. In particular, we can form a $2p$ order finite difference approximation as

$$h \frac{\partial f}{\partial x} \Big|_{x_i} \approx \sum_{j=1}^{j=p} d_j (f(\hat{u}_{i+j-\frac{1}{2}}) - f(\hat{u}_{i-j+\frac{1}{2}})). \quad (2.19)$$

The d_j weights are calculated by expanding (2.19) and expressing each f term as a Taylor series. Combining the resulting expressions such that lower order Taylor series terms are cancelled yields the values for the d_j weights [45]. Values for up to 11th order are given in Table 2.3.

Table 2.3: Derivative Coefficients (d_j) up to 12th order

| | d_0 | d_1 | d_2 | d_3 | d_4 | d_5 |
|---------|-------------------------|-------------------------|------------------------|------------------------|----------------------|----------------------|
| n,order | | | | | | |
| 2 | 1 | - | - | - | - | - |
| 4 | $\frac{9}{8}$ | $-\frac{1}{24}$ | - | - | - | - |
| 6 | $\frac{75}{64}$ | $-\frac{25}{384}$ | $\frac{3}{640}$ | - | - | - |
| 8 | $\frac{1225}{1024}$ | $-\frac{245}{3072}$ | $\frac{49}{5120}$ | $-\frac{5}{7168}$ | - | - |
| 10 | $\frac{2047}{1690}$ | $-\frac{735}{8192}$ | $\frac{567}{40960}$ | $-\frac{86}{48707}$ | $\frac{35}{294912}$ | - |
| 12 | $\frac{160083}{131072}$ | $-\frac{12705}{131072}$ | $\frac{14704}{842749}$ | $-\frac{1013}{341389}$ | $\frac{207}{576593}$ | $-\frac{17}{778110}$ |

2.6 Non-Linear Conservation Laws

The well known advection or transport equation is the simplest case of the conservation law (1.1); here the flux is $f(u(x, t)) = au(x, t)$, where a is the constant advection speed. For such a case, a left biased WENO scheme can be applied directly; however, for a non-linear conservation law, e.g. $f(u(x, t)) = \frac{1}{2}[u(x, t)]^2$, a few additional steps must be employed because the local advection speed is non-constant. Specifically, in order to enforce upwinding and ensure stability, a globally left or right biased finite difference scheme cannot be applied. Rather, we must utilize the Lax-Friedrichs flux-splitting technique to rewrite the flux as

$$f = \frac{f(u) + \alpha u}{2} + \frac{f(u) - \alpha u}{2}, \quad (2.20)$$

where α is the maximum characteristic speed of u , calculated at each time step as

$$\alpha = \max_i \left(\left| \frac{\partial}{\partial u} [f(u_i)] \right| \right). \quad (2.21)$$

In (2.20), the first term is the positive flux and the second term is the negative flux. Upwinding is enforced by utilizing a left biased numerical scheme on the positive flux and a right biased numerical scheme upon the negative flux. Within the context of our

scheme, this is accomplished by obtaining the left and right WENO reconstructions of u and then substituting these approximations into the respective left and right terms of (2.20) to obtain

$$f\left(u_{i+\frac{1}{2}}\right) = \frac{1}{2} \left\{ f\left(\hat{u}_{i+\frac{1}{2}}^{\text{left}}\right) + \alpha \hat{u}_{i+\frac{1}{2}}^{\text{left}} + f\left(\hat{u}_{i+\frac{1}{2}}^{\text{right}}\right) - \alpha \hat{u}_{i+\frac{1}{2}}^{\text{right}} \right\}. \quad (2.22)$$

The right WENO approximation can be readily obtained by just reversing the order of the stored u_i vector and shifting the index to the right before applying the left scheme. The above expression can then be substituted directly into (2.19), and the remaining derivative calculation proceeds just as done for the linear case. In short, the non-linear case requires that we form two separate WENO approximations of u and then assemble them using the LF flux-splitting technique to rewrite f in a form consistent with upwinding.

2.7 System of 1D Non-linear Conservation Laws

Here we consider the extension of the WENO scheme to systems of conservation laws of the form

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial \mathbf{f}}{\partial x} = 0, \quad (2.23)$$

where \mathbf{u} is a vector of coupled conservative variables and \mathbf{f} is a vector of fluxes. Within the context of the general two-fluid system, the conserved variables are $\mathbf{u} = [\rho, \rho v, \rho(e + \frac{v^2}{2}), \Gamma_1, \Gamma_2]^T$. These variables represent the specific forms of mass, momentum, total energy, and the two interface capturing functions, respectively. Taking the internal energy as $e = \frac{p+\Pi}{\rho(\gamma-1)}$, these variables can be further broken down into density, velocity, and pressure along with the interface capturing functions to define the vector of primitive variables $\mathbf{u}^p = [\rho, v, p, \Gamma_1, \Gamma_2]^T$.

In practice, rather than solving (1.10)–(1.14) as given, we instead solve an equivalent system where the two interface capturing advection equations are written in the

following conservative plus source term form:

$$\frac{\partial \Gamma_1}{\partial t} + \frac{\partial}{\partial x} [v \Gamma_1] - \Gamma_1 \frac{\partial v}{\partial x} = 0 \quad (2.24)$$

$$\frac{\partial \Gamma_2}{\partial t} + \frac{\partial}{\partial x} [v \Gamma_2] - \Gamma_1 \frac{\partial v}{\partial x} = 0. \quad (2.25)$$

In (1.5) and (1.8), the velocity terms are outside the spatial derivative, and at each time step, its value is taken as the primitive variable v calculated during the previous time step. However, with the above forms, the velocity terms lie within the derivative allowing us to use the same discretized reconstruction of velocity across all equations to obtain a more consistent scheme [28, 45]. In this form, the vector of fluxes is defined as $\mathbf{f} = [\rho v, \rho v^2 + p, (\rho(e + \frac{\rho v^2}{2}) + p)v, v \Gamma_1, v \Gamma_2]^T$.

The addition of the source terms in (2.24) and (2.25) have minimal impact upon the overall algorithm. The first two terms of each of these equations form conservation laws onto which the WENO reconstruction scheme can be applied directly; the only change is that after calculating the spatial derivative of (2.19), the derivative of the source term must be added in as well. The source terms are discretized as

$$h \Gamma \frac{\partial v}{\partial x} \Big|_{x_i} \approx \Gamma \sum_{j=0}^{j=1} \left(\left(\frac{\hat{v}_{i+j+\frac{1}{2}}^{\text{left}} + \hat{v}_{i+j+\frac{1}{2}}^{\text{right}}}{2} \right) - \left(\frac{\hat{v}_{i-j-\frac{1}{2}}^{\text{left}} + \hat{v}_{i-j-\frac{1}{2}}^{\text{right}}}{2} \right) \right), \quad (2.26)$$

where \hat{v} denote the left or right WENO reconstructions of the velocity. This discretization results in high-order accuracy away from interfaces and second order accuracy near interfaces [45].

Beyond the addition of several conserved variables, the extension to the system of conservation law cases requires a more complicated approach to upwinding. Specifically, the α term (also called the stabilization parameter) of (2.21) is defined for a scalar function $f(u(x_i))$ not for a vector function $f(\mathbf{u}(x_i))$; hence, the definition of

the stabilization parameter must be generalized to

$$\alpha = \max_{i=0}^N \left(\max_{j=1}^{j=m} |\lambda_{i,j}| \right), \quad (2.27)$$

where $\lambda_{i,j}$ are the eigenvalues associated with the Jacobian matrix of $f(\mathbf{u}(x_i))$. To define the eigenvalues associated with the system, we consider the quasi-linear or primitive form:

$$\frac{\partial}{\partial t}[\mathbf{u}_i] + A \frac{\partial}{\partial x}[\mathbf{u}^p_i], \quad (2.28)$$

where for the 1D two-fluid system, A is defined as

$$A = \begin{pmatrix} v_i & \rho_i & 0 & 0 & 0 \\ 0 & v_i & \frac{1}{\rho_i} & 0 & 0 \\ 0 & \rho_i c_i^2 & v_i & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.29)$$

The eigenvalues of this matrix are v_i , $v_i - c_i$, and $v_i + c_i$. Since the speed of sound is always positive, the maximum eigenvalue is $\alpha = \max_{i=0}^N (|v_i| + c_i)$.

Similar to the scalar non-linear case, α is calculated at each time step (and at each stage when using a Runge-Kutta temporal scheme), and the Lax-Friedrichs flux splitting of (2.22) is utilized for each conservative variable to form all three flux values. However, for the coupled system of (2.28), upwinding cannot be applied directly because the advection speeds are unknown in the coupled form. Thus, before calculating the left and right reconstructions, the primitive variables must first be mapped into the characteristic space to approximately decouple the directionality of the system [45]. Considering the form of (2.28), to obtain the characteristic decomposition,

we need to diagonalize A as $A = Q\Lambda Q^{-1}$. This permits the rewriting of (2.28) as

$$\frac{\partial}{\partial t}[Q^{-1}\mathbf{u}_i] = \Lambda \frac{\partial}{\partial x}[Q^{-1}\mathbf{u}_i]. \quad (2.30)$$

Since Λ is a diagonal matrix, this form of the Euler system is a set of advection equations where the advection velocities are the eigenvalues of A . To enforce upwinding, the WENO reconstruction is performed upon $Q^{-1}\mathbf{u}^p$ instead of \mathbf{u}^p and then mapped back into physical space by multiplying the reconstructed characteristic variables by Q before calculating the spatial derivatives. The two transformation matrices are given below:

$$Q = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ -\frac{c}{\rho} & 0 & \frac{c}{\rho} & 0 & 0 \\ c^2 & 0 & c^2 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix} \quad Q^{-1} = \begin{pmatrix} 0 & -\frac{\rho}{2c} & \frac{1}{2c^2} & 0 & 0 \\ 1 & 0 & -\frac{1}{c^2} & 0 & 0 \\ 0 & \frac{\rho}{2c} & \frac{1}{2c^2} & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \quad (2.31)$$

Since the primitive variables $\mathbf{u}^p = [\rho, v, p, \Gamma_1, \Gamma_2]^T$ are defined at each x_i , the transformation matrices Q and Q^{-1} must also be calculated at each x_i as well. However, the output of the WENO reconstruction are the cell boundary primitive variables $\mathbf{u}^p(x_{i+\frac{1}{2}})$, while the transformation matrices were defined using cell-centered values. To this end, we define the transformation matrices using the arithmetic average of the primitive variables adjacent to the cell boundary. That is, for Q and Q^{-1} , we calculate each element using $\rho = \frac{\rho_i + \rho_{i+1}}{2}$, $v = \frac{v_i + v_{i+1}}{2}$, and $p = \frac{p_i + p_{i+1}}{2}$.

After mapping the primitive variables into the characteristic space, the WENO reconstruction is performed directly upon these mapped variables. However, before forming the conservative variables and the LF flux (2.22), the primitive variables have to be mapped back into physical space. So, with L denoting the left WENO

reconstruction operator and R giving the right WENO operator, we have:

$$\hat{\mathbf{u}}_{j,i+\frac{1}{2}}^{\text{left}} = QL(Q^{-1}\mathbf{u}_{j,i}^p) \quad (2.32)$$

$$\hat{\mathbf{u}}_{j,i+\frac{1}{2}}^{\text{right}} = QR(Q^{-1}\mathbf{u}_{j,i}^p) \quad (2.33)$$

In summary, the overall procedure at each time step is as follows:

1. Map the cell centered primitive variables into the characteristic space as $Q^{-1}\mathbf{u}^p$.
2. Form the left and right WENO reconstructions of each transformed primitive variable using (2.11).
3. Transform the left and right WENO reconstructed characteristic variables back into physical space by multiplying by Q .
4. Calculate the cell-boundary conservative variables from the reconstructed cell-boundary primitive variables.
5. Form the Lax-Friedrichs flux split with (2.22) for each conservative variable.
6. Approximate the spatial derivative of each flux using (2.19).
7. Use (2.26) to calculate the source terms. The inputs to this equation are the left and right WENO reconstructions of velocity from Step 3 and the cell-centered order parameters $(\Gamma_{1,i}, \Gamma_{2,i})$ known at the beginning of the time step.
8. Subtract these source terms from the respective flux derivatives calculated in Step 6.
9. Integrate in time to obtain the updated cell centered conservative variables. From these calculate the updated primitive variables.

2.8 Two Dimensional System

Here, the application to the two dimensional case of the two-fluid model is briefly discussed. The general form of the system is

$$\frac{\partial \mathbf{u}}{\partial t} + \frac{\partial}{\partial x}[\mathbf{f}(\mathbf{u})] + \frac{\partial}{\partial y}[\mathbf{g}(\mathbf{u})] + \mathbf{a} \frac{\partial v_x}{\partial x} + \mathbf{a} \frac{\partial v_y}{\partial y} = 0 \quad (2.34)$$

where $\mathbf{u} = [\rho, \rho v_x, \rho v_y, E, \Gamma_1, \Gamma_2]^T$ gives the conserved variables. The vector $\mathbf{f} = [v_x \rho, \rho v_x^2 + p, \rho v_x v_y, (E + p)v_x, v_x \Gamma_1, v_x \Gamma_2]^T$ describes the fluxes in the x direction. The vector of fluxes in the y direction is $\mathbf{g} = [v_y \rho, \rho v_x v_y, \rho v_y^2 + p, (E + p)v_y, v_y \Gamma_1, v_y \Gamma_2]^T$. Finally, the vector $\mathbf{a} = [0, 0, 0, 0, \Gamma_1, \Gamma_2]$ accounts for the source term discretization of the order parameters (2.24). The algorithm follows the same steps as outlined above for the 1D case, except it must be performed twice for each point in the computational domain — once in the x direction and once in the y direction.

Chapter 3

Positivity

In this chapter, a positivity-preserving algorithm is developed for our high-order finite difference scheme. It begins by highlighting the necessity of positivity preservation and briefly reviewing applicable work within this space. The application of the algorithm to our scheme is then detailed for the case of scalar conservation and the 1D Euler system. Finally, the extension to the two-fluid model and two spatial dimensions are considered.

3.1 Motivation

The previously described WENO scheme provides “essentially” non-oscillatory solutions, meaning that some relatively small oscillations that vanish with grid refinement may be present near discontinuities and sharp features after the reconstruction. In many cases, these oscillations have negligible impact on the overall solution; however, for certain scenarios, they can pose problems. For example, simulations that involve very high Mach shock waves can quickly develop negative density and/or pressure values. Figure 3.1 illustrates this for a shock tube containing air with a high pressure ratio across the diaphragm. The initial data for this simulation were $\rho = 1$ and $v = 0$ with $p = 1$ to the right of the diaphragm (located at $x = 0.5$) and $p = 1000$ to the left of the diaphragm. The simulation was run to one time step (CFL=0.4) with the 9th order scheme over a domain $[0, 1]$ with $N = 100$ grid points.

Figure 3.1 shows that within one time step, the simulation obtained a negative pressure value implying a complex sound speed which would cause the simulation to

crash if subsequent time steps were attempted.

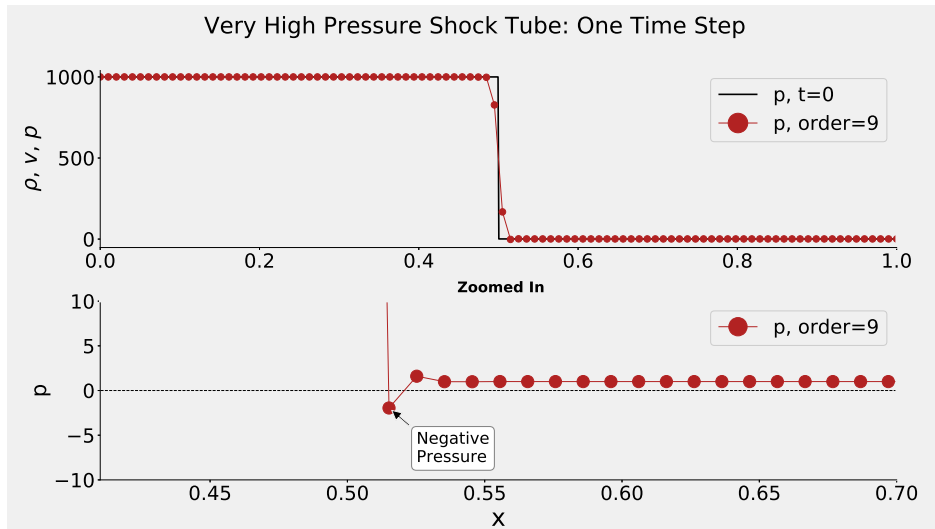


Figure 3.1: Demonstration of a negative pressure value developing in a shock tube with a pressure ratio of 1000 across the diaphragm

In the context of the Euler equations or the two-fluid model, we need to enforce some lower limit on the primitive variables of density and pressure that guarantees a real sound speed. For the scalar conservation law, the analogous, albeit more restrictive, behavior is typically referred to as “maximum-principle preserving,” meaning that all solutions stay within the upper and lower bounds of the initial data.

Several techniques have been developed to counter this behavior and achieve rigorous positivity-preserving schemes. In particular, Zhang and Shu introduced a limiter [62] for scalar conservation laws that is high-order accurate and valid in multi-dimensions. They later extended their method to finite volume and finite difference schemes for the Euler equations [63, 61]. This method utilizes a limiting process upon the polynomial reconstructions of fluxes rendering it unamendable to our scheme, which is based upon the reconstruction of primitive variables. Furthermore, this technique imposes an undesirable CFL restriction of $\frac{1}{12}$. An alternate flux limiting approach for 1D scalar conservation laws was introduced by Xu [57] and subsequently extended to 2D [32].

This method was designed to be high-order and independent of flux reconstruction strategy. Later, Hu et al. [25] introduced a positivity-preserving scheme for the Euler equations based upon the flux limiter approach; however, this scheme suffered from reduced accuracy unless very small CFL numbers were used. Finally, the technique was successfully extended to the 1D and 2D Euler system by Xiong et al. [56] who demonstrated positivity preservation for high-order schemes with very minimal CFL restrictions. Because of these desirable qualities, coupled with the fact that their technique is independent of flux reconstruction, we will adopt their approach for devising a high-order positivity scheme to our two-fluid compressible model.

3.2 Flux Limiting Background

In the most basic sense, the positivity scheme of [57, 32, 56] works by exploiting the maximum principle preserving (MPP) property of the first order Lax-Friedrichs method. It is well known that this first order scheme will always be bounded by the maximum and minimum values of the initial data, so the goal is to locally modify the calculated, high-order flux reconstructions towards this first order solution whenever the high-order reconstructions yield unphysical solutions.

The starting point of the scheme is to express the conservation law (1.1) in the general conservative form

$$\frac{\partial}{\partial t}[u_i] = -\frac{1}{h}F(u_i), \quad (3.1)$$

where

$$F(u_i) = H_{i+\frac{1}{2}} - H_{i-\frac{1}{2}}. \quad (3.2)$$

Here $F(u_i)$ is some arbitrary numerical scheme that calculates the high-order cell boundary fluxes $H_{i+\frac{1}{2}}$ and $H_{i-\frac{1}{2}}$. Using some time integration method, the updated

solution at each time step is expressed as

$$u_i^{n+1} = u_i^n - \frac{\tau}{h} \left(H_{i+\frac{1}{2}}^* - H_{i-\frac{1}{2}}^* \right), \quad (3.3)$$

where $H_{i\pm\frac{1}{2}}^*$ represent an overall high-order flux obtained at the end of a multi-stage time integration. It is important to note that, as pointed out in [57, 56], positivity is not enforced during the intermediary time integration stages; it is only enforced after the final time integration stage has been completed but before the updated solution is calculated. For the classic 4th order Runge-Kutta time integration scheme, we obtain the following expression at the end of all four RK stages

$$u_i^{n+1} = u_i^n + \tau \left(\frac{1}{6}k_1 + \frac{1}{3}k_2 + \frac{1}{3}k_3 + \frac{1}{6}k_4 \right), \quad (3.4)$$

where b_q and α_{qj} are tabulated constants and k_q is

$$k_q = F \left(u_i^n + \tau \sum_{j=1}^{q-1} \alpha_{qj} k_j \right). \quad (3.5)$$

Since each k term is just an evaluation of F , we recast (3.4) as

$$u_i^{n+1} = u_i^n + \frac{\tau}{h} \left(\frac{1}{6}F(w_1) + \frac{1}{3}F(w_2) + \frac{1}{3}F(w_3) + \frac{1}{6}F(w_4) \right), \quad (3.6)$$

where w_j represents u_i at the j^{th} RK stage. Utilizing the definition of F (3.2), we can expand this equation as

$$u_i^{n+1} = u_i^n - \frac{\tau}{h} \left[\frac{1}{6}H_{i+\frac{1}{2}}^1 + \frac{1}{3}H_{i+\frac{1}{2}}^2 + \frac{1}{3}H_{i+\frac{1}{2}}^3 + \frac{1}{6}H_{i+\frac{1}{2}}^4 - \left(\frac{1}{6}H_{i-\frac{1}{2}}^1 + \frac{1}{3}H_{i-\frac{1}{2}}^2 + \frac{1}{3}H_{i-\frac{1}{2}}^3 + \frac{1}{6}H_{i-\frac{1}{2}}^4 \right) \right], \quad (3.7)$$

where the superscripts denote the reconstructed fluxes from a specific RK stage. After collecting all positive and negative terms in (3.7), the updated solution after the final

RK stage can be expressed in the required form of (3.3) as

$$u_i^{n+1} = u_i^n - \frac{\tau}{h} \left(H_{i+\frac{1}{2}}^{RK} - H_{i-\frac{1}{2}}^{RK} \right). \quad (3.8)$$

The results of (3.8) indicate that the left and right fluxes from each RK stage must be separately summed to form the overall H_{\pm}^{RK} fluxes; however, care must be taken in how these terms are defined within our primitive variable based reconstruction. In the present work, the right hand side of (3.1) is calculated to $2r$ order accuracy using (2.19) to obtain

$$h \frac{\partial f}{\partial x} \Big|_{x_i} \approx (H_{i+\frac{1}{2}} - H_{i-\frac{1}{2}}) = \sum_{j=1}^{j=r} d_j (\hat{f}_{i+j-\frac{1}{2}} - \hat{f}_{i-j+\frac{1}{2}}), \quad (3.9)$$

where, for brevity, we denote the $2r - 1$ order cell boundary LF fluxes, $f(\hat{u}_{i\pm\frac{1}{2}})$, as $\hat{f}_{i\pm\frac{1}{2}}$. Expanding the summation term gives

$$\begin{aligned} h \frac{\partial f}{\partial x} \Big|_{x_i} \approx & \left(d_1 \hat{f}_{i+\frac{1}{2}} + d_2 \hat{f}_{i+\frac{3}{2}} + d_3 \hat{f}_{i+\frac{5}{2}} + \cdots + d_r \hat{f}_{i+\frac{2r-1}{2}} \right) \\ & - \left(d_1 \hat{f}_{i-\frac{1}{2}} + d_2 \hat{f}_{i-\frac{3}{2}} + d_3 \hat{f}_{i-\frac{5}{2}} + \cdots + d_r \hat{f}_{i-\frac{2r-1}{2}} \right). \end{aligned} \quad (3.10)$$

Comparing (3.10) with (3.2), it may seem reasonable to utilize the two terms in parentheses as the fluxes taking $H_{i+\frac{1}{2}} = d_1 \hat{f}_{i+\frac{1}{2}} + d_2 \hat{f}_{i+\frac{3}{2}} + \cdots + d_r \hat{f}_{i+\frac{r}{2}}$ and $H_{i-\frac{1}{2}} = d_1 \hat{f}_{i-\frac{1}{2}} + d_2 \hat{f}_{i-\frac{3}{2}} \cdots + d_r \hat{f}_{i-\frac{r-1}{2}}$ as the right and left fluxes at cell I_i . However, this is incorrect as it would violate conservation. To demonstrate this point, we substitute $i = i - 1$ into $H_{i+\frac{1}{2}} = d_1 \hat{f}_{i+\frac{1}{2}} + d_2 \hat{f}_{i+\frac{3}{2}} + \dots$ to obtain the *right* flux at cell I_{i-1} . Doing so yields $H_{i-\frac{1}{2}} = d_1 \hat{f}_{i-\frac{1}{2}} + d_2 \hat{f}_{i+\frac{1}{2}} + \dots$, which clearly contradicts the proposed expression for $H_{i-\frac{1}{2}}$ based upon the negative term in (3.10).

In order to have a conservative scheme, the left flux of cell I_i must be equal to the right flux of cell I_{i-1} , or in general, the flux at any cell boundary must be the same

regardless whether it is approached from the left or right. Hence, we must define the $H_{i\pm\frac{1}{2}}$ flux terms such that $H_{i+\frac{1}{2}}^+ = H_{i+\frac{1}{2}}^-$. Similar to Del Zanna et al. [60], we can construct a conservative flux as

$$H_{i+\frac{1}{2}} = c_0 L_{i+\frac{1}{2}} + c_1(L_{i+\frac{3}{2}} + L_{i-\frac{1}{2}}) + c_2(L_{i+\frac{5}{2}} + L_{i-\frac{3}{2}}) + \dots \quad (3.11)$$

such that

$$h \frac{\partial f}{\partial x} \Big|_{x_i} \approx (H_{i+\frac{1}{2}} - H_{i-\frac{1}{2}}) = \sum_{q=1}^{q=r} d_q (\hat{f}_{i+q-\frac{1}{2}} - \hat{f}_{i-q+\frac{1}{2}}) \quad (3.12)$$

is satisfied. To obtain the c constants, the difference $(H_{i+\frac{1}{2}} - H_{i-\frac{1}{2}})$ can be formed using (3.11) and substituted into (3.12) to obtain a linear system. The results are tabulated for $r \in \{2, 3, 4, 5, 6\}$ in Table 3.1.

Table 3.1: Coefficients for conservative flux construction

| | c_0 | c_1 | c_2 | c_3 | c_4 | c_5 |
|---|--------------------------|-------------------------|------------------------|-----------------------|----------------------|----------------------|
| r | | | | | | |
| 2 | $\frac{13}{12}$ | $-\frac{1}{24}$ | - | - | - | - |
| 3 | $\frac{1067}{960}$ | $-\frac{29}{480}$ | $\frac{3}{640}$ | - | - | - |
| 4 | $\frac{30251}{26880}$ | $-\frac{7621}{107520}$ | $\frac{159}{17920}$ | $-\frac{5}{7168}$ | - | - |
| 5 | $\frac{1013909}{894323}$ | $-\frac{55509}{716006}$ | $\frac{11303}{926795}$ | $-\frac{425}{258048}$ | $\frac{35}{294912}$ | - |
| 6 | $\frac{799692}{701963}$ | $-\frac{61116}{744283}$ | $\frac{7639}{515538}$ | $-\frac{684}{260063}$ | $\frac{251}{744459}$ | $-\frac{17}{778110}$ |

With the left and right high-order fluxes defined in (3.11), these results can be substituted into (3.3) for each RK stage to obtain the overall flux approximations $H_{i+\frac{1}{2}}^{RK}$ and $H_{i-\frac{1}{2}}^{RK}$ for the final time stage as required by (3.8). It is worth noting that the above procedure is not limited to a scalar case; indeed, $H_{\pm\frac{1}{2}}^{RK}$ can represent a vector of fluxes $\mathbf{H}_{\pm\frac{1}{2}}^{RK}$, where each $\hat{f}_{i\pm\frac{1}{2}}$ is a Lax-Friedrich flux (2.22) as in the Euler system or the two-fluid model.

3.3 Implementation for scalar case

In general, we seek to enforce the following maximum principle preserving constraint upon our updated solution

$$u_m \leq u_i^{n+1} \leq u_M. \quad (3.13)$$

After substituting (3.8) for u_i^{n+1} , we can write this inequality as

$$u_m \leq u_i^n - \frac{\tau}{h}(H_{i,+}^{RK} - H_{i,-}^{RK}) \leq u_M \quad (3.14)$$

where, for conciseness, we have dropped $\pm\frac{1}{2}$ from the subscripts of the flux terms. This inequality cannot be guaranteed based upon the existing high-order flux values, so in order to ensure the positivity property, we introduce modified fluxes to replace the $H_{i,\pm}^{RK}$ terms in the preceding equation. These modified fluxes are defined as

$$\begin{cases} \tilde{H}_{i,+}^{RK} = h_{i,+} + \theta_{i,+}(H_{i,+}^{RK} - h_{i,+}) \\ \tilde{H}_{i,-}^{RK} = h_{i,-} + \theta_{i,-}(H_{i,-}^{RK} - h_{i,-}) \end{cases} \quad (3.15)$$

where the $h_{i,\pm}$ terms represent the right and left, first-order, Lax-Friedrichs fluxes at cell I_i . Since the first order scheme always satisfies (3.14), if $\theta_{i,\pm} = 0$ in (3.15), then the validity of (3.14) can be guaranteed at the expense of reducing the modified fluxes to first order. Hence, the goal of this scheme is to identify the largest value between 0 and 1 that the flux limiter, $\theta_{i,\pm}$, can take on such that the inequality of (3.14) is true; if $\theta_{i,\pm} = 1$, then the original high-order flux is retained, but if $\theta_{i,\pm}$ go to zero, then the scheme locally approaches first order accuracy. To implement this scheme, we need to methodically calculate the maximum values of $\theta_{i,\pm}$ at each cell, I_i . These maximum values are denoted $\left\{ \Lambda_{-\frac{1}{2},i}^M, \Lambda_{+\frac{1}{2},i}^M, \Lambda_{-\frac{1}{2},i}^m, \Lambda_{+\frac{1}{2},i}^m \right\}$ with $\Lambda_{\pm\frac{1}{2}}^M$ corresponding to

the restrictions associated with the maximum bound and $\Lambda_{\pm\frac{1}{2}}^m$ being related to the lower bound. The process involves four logical statements for each part (eight in total) that fully define all four Λ at each cell.

To describe the process, we begin with the inequality of (3.14) that we wish to preserve using our modified flux values

$$u_m \leq u_i^n - \frac{\tau}{h} \left(\tilde{H}_{i,+}^{RK} - \tilde{H}_{i,-}^{RK} \right) \leq u_M. \quad (3.16)$$

Substituting the definitions for the modified fluxes (3.15), we obtain

$$u_m \leq u_i^n - \frac{\tau}{h} \left[h_{i,+} + \theta_{i,+} (H_{i,+}^{RK} - h_{i,+}) - h_{i,-} + \theta_{i,-} (H_{i,-}^{RK} - h_{i,-}) \right] \leq u_M. \quad (3.17)$$

In an effort to simplify the expression, we introduce the notations

$$F_+ = H_{i,+}^{RK} - h_{i,+} \quad (3.18)$$

$$F_- = H_{i,-}^{RK} - h_{i,-} \quad (3.19)$$

to obtain

$$u_m \leq u_i^n - \lambda (h_{i,+} - h_{i,-}) - (\lambda \theta_{i,+} F_+) + (\lambda \theta_{i,-} F_-) \leq u_M, \quad (3.20)$$

where $\lambda = \frac{\tau}{h}$. Next, we break the inequality of (3.20) into the upper and lower parts and enforce these limits separately. We begin by considering the upper limit.

3.3.1 Enforcing the upper limit

Rearranging (3.20) and only considering the upper limit gives

$$-u_M + [u_i^n - \lambda (h_{i,+} - h_{i,-})] - (\lambda \theta_{i,+} F_+) + (\lambda \theta_{i,-} F_-) \leq 0 \quad (3.21)$$

Focusing on the term in the brackets, we see that this is just the first order approximation to u_i at the next time step, which we know is maximum principle preserving and must be less than u_M . Thus, the inequality

$$u_i^{n+1} = u_i^n - \lambda(h_{i,+} - h_{i,-}) \leq u_M \quad (3.22)$$

is always true. Rearranging the above equation as

$$\Gamma_i^M = u_M - u_i^n + \lambda(h_{i,+} - h_{i,-}) \geq 0 \quad (3.23)$$

and substituting this back into (3.21), the inequality that must be preserved is found to be

$$(\lambda\theta_{i,-}F_-) - \lambda\theta_{i,+}F_+ - \Gamma_i^M \leq 0. \quad (3.24)$$

Since $\Gamma_i^M \geq 0$, the validity of (3.24) is entirely dependent upon the signs of F_+ and F_- . To identify the maximum $\theta_{i,\pm}$ values, all four combinations for the signs of F_+ and F_- are considered.

1. $F_+ \geq 0$ and $F_- \leq 0$

Since both the first and second terms in (3.24) are less than zero, the inequality is inherently preserved, and the fluxes do not need to be limited. Thus, we take

$$\Lambda_{+,i}^M = 1$$

$$\Lambda_{-,i}^M = 1.$$

2. $F_+ < 0$ and $F_- \leq 0$

The first term in (3.24) is less than zero, but the second is not. Hence, to preserve positivity, we need to solve for the maximum $\theta_{i,+}$ that guarantees the inequality. In general, we seek a set $(\theta_{i,-}, \theta_{i,+}) \in [0, \Lambda_{-,i}] \times [0, \Lambda_{+,i}]$ that defines

a rectangular region on the $(\theta_{i,-}, \theta_{i,+})$ plane. The vertices of this region are $[(0, 0), (\Lambda_{-,i}, 0), (0, \Lambda_{+,i}), (\Lambda_{-,i}, \Lambda_{+,i})]$. Figure 3.2 provides a visualization; the green rectangle defines the set of acceptable θ values.

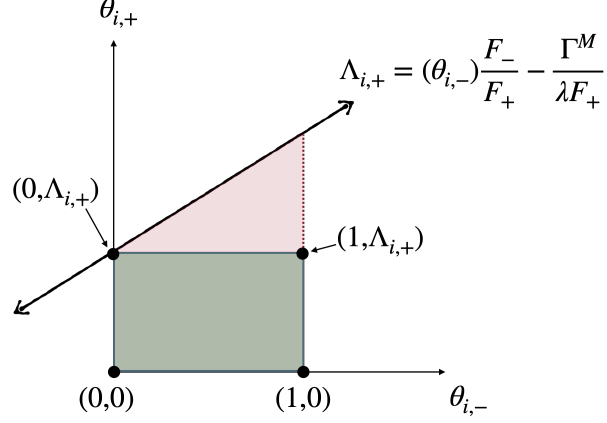


Figure 3.2: Illustration of flux limiters for positive flux case

For the case we are currently considering, (3.24) forces no conditions on the maximum value of $\theta_{i,-}$; therefore $\Lambda_{-,i} = 1$. Owing to this fact, the lower right vertex is fixed as $\Lambda_{i,-} = 1$. From (3.24), it is clear that $\Lambda_{i,+}$ depends linearly on the value we select for $\theta_{i,-}$; this is illustrated in Figure 3.2 as the diagonal line. Although we don't yet know what value $\theta_{i,-}$ will eventually take on, we do know that it must be between 0 and 1. In order to define a $\Lambda_{+,i}$ that guarantees positivity for any $\theta_{i,-}$, we need to take the minimum value of this linear function over the interval $\theta_{i,-} \in [0, 1]$ which occurs at $\theta_{i,-} = 0$. Hence, we define the limiter using (3.24) as

$$\Lambda_{+,i} = \frac{-\Gamma_i^M}{\lambda F_+}. \quad (3.25)$$

To avoid the case of division by zero, we introduce a small factor $\delta = 10^{-16}$ into the denominator. Because the denominator must be negative, this term must be negative as well. Thus, for this case, we take

$$\Lambda_{-,i}^M = 1$$

$$\Lambda_{+,i}^M = \begin{cases} 1 & \text{if } \frac{-\Gamma_i^M}{\lambda F_+ - \delta} > 1 \\ \frac{-\Gamma_i^M}{\lambda F_+ - \delta} & \text{otherwise.} \end{cases}$$

3. $F_+ \geq 0$ and $F_- > 0$

Here, the second term in (3.24) is less than zero, while the first term is greater than zero. Thus, in order to preserve the inequality, we need to limit the F_- flux term by solving for $\theta_{i,-}$. We take the exact same approach as in the previous case. Solving for $\theta_{i,-}$ in (3.24) gives

$$(\lambda\theta_{i,-}F_-) + (-\lambda\theta_{i,+}F_+) - \Gamma_i^M \leq 0$$

$$\theta_{i,-} \leq \frac{\Gamma_i^M + (\lambda\theta_{i,+}F_+)}{\lambda F_-}.$$

Figure 3.3 illustrates the limiting procedure for this case. On the $(\theta_{i,-}, \theta_{i,+})$ plane, the acceptable region of limiter values is the rectangle bounded by $[(0, 0), (\Lambda_{-,i}, 0), (0, 1), (\Lambda_{-,i}, 1)]$.

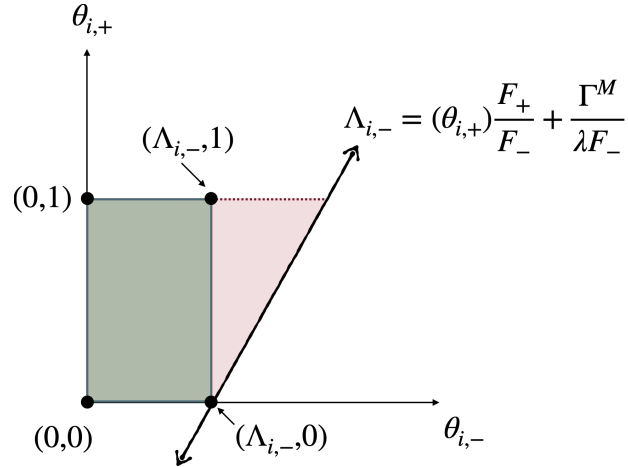


Figure 3.3: Illustration of flux limiters for negative flux case.

Analogous to the previous case, we need to identify the minimum value of $\theta_{i,-}$

that guarantees (3.24) for any $\theta_{i,+}$. Hence, we assign the limiters at this cell as

$$\Lambda_{-,i}^M = \begin{cases} 1 & \text{if } \frac{\Gamma_i^M}{\lambda F_- + \delta} > 1 \\ \frac{\Gamma_i^M}{\lambda F_- + \delta} & \text{otherwise} \end{cases}$$

$$\Lambda_{+,i}^M = 1.$$

4. $F_- > 0$ and $F_+ < 0$

For this case, both the first and second terms of the (3.24) are positive. To establish this inequality, we need to limit both flux terms. This is accomplished by taking $\theta_{i,-} = \theta_{i,+} = \theta_i$ and then solving for the maximum θ value using (3.24) to obtain

$$\theta_i \leq \frac{\Gamma_i^M}{(\lambda F_-) + (-\lambda F_+)}.$$

If the above expression gives a value greater than 1, then there is no need to limit the fluxes and we can take $\theta_i = 1$. Thus, the limiters are defined by

$$\Lambda_{-,i}^M = \begin{cases} 1 & \text{if } \frac{\Gamma_i^M}{-\lambda F_+ + \lambda F_-} \leq 1 \\ \frac{\Gamma_i^M}{-\lambda F_+ + \lambda F_- + \delta} & \text{otherwise} \end{cases}$$

$$\Lambda_{+,i}^M = \begin{cases} 1 & \text{if } \frac{\Gamma_i^M}{-\lambda F_+ + \lambda F_-} \leq 1 \\ \frac{\Gamma_i^M}{-\lambda F_+ + \lambda F_- + \delta} & \text{otherwise.} \end{cases}$$

The four cases described above outline how the limiters for the maximum part can be calculated at each cell. The integration of this process into the finite difference scheme is straightforward and involves nothing more than looping through each point in the domain and applying the above logic accordingly.

3.3.2 Enforcing the lower limit

Next, the lower limit of the inequality (3.20) is considered. Specifically, we seek to enforce

$$-u_m + [u_i^n - \lambda(h_{i,+} - h_{i,-})] - (\lambda\theta_{i,+}F_+) + (\lambda\theta_{i,-}F_-) \geq 0. \quad (3.26)$$

Once again, the term in brackets is just the first order solution for u_i^{n+1} which is guaranteed to be greater than u_m . Hence, we know that the sum of the first two terms must be greater than zero, that is

$$\Gamma_i^m = u_m - u_i^n + \lambda[h_{i,+} - h_{i,-}] \leq 0. \quad (3.27)$$

Substituting this definition back into (3.26) gives

$$(-\lambda\theta_{i,+}F_+) + (\lambda\theta_{i,-}F_-) - \Gamma_i^m \geq 0. \quad (3.28)$$

Just as for the maximum part, noting that the third term $-\Gamma_i^m$ is always greater than zero, we can consider the four possible cases for the other two terms and identify the required flux limiter values.

1. $F_+ \leq 0$ and $F_- \geq 0$

Here both the first and second terms in (3.28) are greater than zero, so no limiting is required. Thus, we can assign the limiters as

$$\Lambda_{+,i}^m = 1$$

$$\Lambda_{-,i}^m = 1.$$

2. $F_+ > 0$ and $F_- \geq 0$

The first term in (3.28) is less than zero while the second is not. Hence, we must solve for the $\theta_{i,+}$ value that guarantees the inequality for any $\theta_{i,-}$. Thus, for this case, we can assign the limiters as

$$\Lambda_{-,i}^m = 1$$

$$\Lambda_{+,i}^m = \begin{cases} 1 & \text{if } \frac{\Gamma_i^m}{-\lambda F_+ - \delta} > 1 \\ \frac{\Gamma_i^m}{-\lambda F_+ - \delta} & \text{otherwise.} \end{cases}$$

3. $F_- < 0$ and $F_+ \leq 0$

The first term in (3.28) is greater than zero while the second is less than zero. Solving the corresponding limiters gives

$$\Lambda_{-,i}^m = \begin{cases} 1 & \text{if } \frac{\Gamma_i^m}{\lambda F_- - \delta} > 0 \\ \frac{\Gamma_i^m}{\lambda F_- - \delta} & \text{otherwise} \end{cases}$$

$$\Lambda_{+,i}^m = 1.$$

4. $F_- < 0$ and $F_+ > 0$

Finally, we have the case where both the first and second term in (3.28) are less than zero. Just as with the maximum part, we can take $\theta_{i,+} = \theta_{i,-} = \theta_i$ and then solve directly for θ_i to obtain the following limiters

$$\Lambda_{-,i}^m = \begin{cases} 1 & \text{if } \frac{\Gamma_i^m}{-\lambda F_+ + \lambda F_-} \geq 1 \\ \frac{\Gamma_i^m}{-\lambda F_+ + \lambda F_- - \delta} & \text{otherwise} \end{cases}$$

$$\Lambda_{+,i}^m = \begin{cases} 1 & \text{if } \frac{\Gamma_i^m}{-\lambda F_+ + \lambda F_-} \geq 1 \\ \frac{\Gamma_i^m}{-\lambda F_+ + \lambda F_- - \delta} & \text{otherwise.} \end{cases}$$

3.3.3 Selecting the Flux Limiters

After going through the process of calculating the $\Lambda_{i,\pm}$ values at each cell for both the upper and lower bound, we select the conservative local flux limiters of (3.15) as

$$\begin{cases} \theta_{i,+} = \min \{ \Lambda_{-,i+1}^M, \Lambda_{+,i}^M, \Lambda_{-,i+1}^m, \Lambda_{+,i}^m \} \\ \theta_{i,-} = \min \{ \Lambda_{-,i}^M, \Lambda_{+,i-1}^M, \Lambda_{-,i}^m, \Lambda_{+,i-1}^m \}. \end{cases} \quad (3.29)$$

This selection is designed to preserve conservation by enforcing the modified fluxes of (3.15) to be equivalent at the shared cell boundary of adjacent cells.

3.4 Positivity Implementation for Euler System

The extension of the positivity scheme to the Euler system of equations adds some additional complexities to the process. For this case, we now seek to enforce the physical limits of positive density and pressure

$$\begin{cases} \rho_i^n \geq 0 \\ p_i^n \geq 0 \end{cases} \quad \forall i, n. \quad (3.30)$$

Within the framework of a general non-linear system, governing equation enforced upon a discrete grid takes the form

$$\left. \frac{\partial \mathbf{u}}{\partial t} \right|_{x_i} + \left. \frac{\partial f(\mathbf{u})}{\partial x} \right|_{x_i} = 0, \quad (3.31)$$

where $\mathbf{u}_i = [\rho_i, \rho_i v_i, \rho_i (e_i + \frac{v_i^2}{2})]^T$. Just as in the scalar case, we need to express the updated solution in time as the difference of two fluxes

$$\mathbf{u}_i^{n+1} = \mathbf{u}_i^n - \lambda \left(\mathbf{H}_{i+\frac{1}{2}}^{RK} - \mathbf{H}_{i-\frac{1}{2}}^{RK} \right). \quad (3.32)$$

Once again, the $\mathbf{H}_{i\pm\frac{1}{2}}^{RK}$ flux vectors are the combination of all respective fluxes obtained at each Runge-Kutta stage, so their actual composition will be as shown in (3.7). It is not necessary to enforce positivity at each RK stage as we are only interested in the overall solution at the final stage. This approach leads to some ambiguity for the Euler system because at each RK stage, we must identify the maximum characteristic speed to perform Lax-Friedrichs flux splitting. This requires calculating the local sound speed $\sqrt{\frac{\gamma p}{\rho}}$, which will become complex if either density or pressure are negative. Following Xiong et al. [56], we can avoid this issue by simply taking the absolute value of pressure and density when calculating the sound speed.

Our task is to identify flux limiters $\theta_{\pm,i}$ at each cell boundary that modify the high-order fluxes towards the first order, monotone fluxes whenever needed. Note that the fluxes of each conserved variable are all modified by the same factor $\theta_{\pm,i}$ as

$$\tilde{\mathbf{H}}_{i\pm\frac{1}{2}}^{RK} = \mathbf{h}_{i\pm\frac{1}{2}} + \theta_{\pm,i} \left(\mathbf{H}_{i\pm\frac{1}{2}}^{RK} - \mathbf{h}_{i\pm\frac{1}{2}} \right). \quad (3.33)$$

Enforcing positivity upon density is straightforward; the process is the same as for the scalar case, except only the lower limit is enforced. Additionally, instead of using 0 for the lower bound, we define a small number ϵ_ρ that acts as the lower limit. In particular, ϵ_ρ is found at each time step as $\min_i \{\rho_i^{n+1}, \epsilon^*\}$ where ρ_i^{n+1} is just the first order density solution at the $n + 1$ time step and ϵ^* is some small constant. In [56], $\epsilon^* = 10^{-13}$ was used in all cases; however, we found that for very high orders (i.e. greater than 7), slightly larger values (e.g., 10^{-9}) gave better performance for some problems. This is believed to be the result of precision loss due to the very large constants associated with the very high-order reconstruction process.

The output of the positivity algorithm applied to density is a pair of maximum flux limiters $\{\Lambda_{-,i}^\rho, \Lambda_{+,i}^\rho\}$ for each cell in the domain. These limiters are identical to the $\Lambda_{\pm,i}^m$ values of (3.29) and guarantee the positivity of density for any $(\theta_{i,-}, \theta_{i,+}) \in$

$$[0, \Lambda_{-,i}^\rho] \times [0, \Lambda_{+,i}^\rho].$$

Enforcing positivity upon pressure poses a different challenge because pressure is non-linearly dependent upon both density and velocity. Specifically,

$$p_i^{n+1} = \left\{ E^{n+1} - \frac{1}{\rho^{n+1}} \frac{[(\rho v)^{n+1}]^2}{2} \right\} \frac{1}{\Gamma_1^{n+1}} - \frac{\Gamma_2^{n+1}}{\Gamma_1^{n+1}}. \quad (3.34)$$

Whereas ρ is both a primitive variable and a conserved variable, pressure must be derived from the conserved total energy using the equation of state. Thus, it is not possible to express the updated pressure in time as the difference between positive and negative fluxes in the form of (3.14). A different approach is required.

The goal of the pressure positivity algorithm remains the same as for the density; we need to identify the maximum limiters for (3.33) that guarantee $p \geq 0$. However, the methodology is quite different. For the scalar case, we just substituted the modified fluxes directly into $u_m \leq u_i^{n+1} \leq u_M$ and enforced each side of the inequality separately. For pressure, we utilize an iterative procedure that employs (3.32) with the modified fluxes of (3.33) to solve for E_i^{n+1} , and consequently, pressure at the next time step p_i^{n+1} . If $p_i^{n+1} \geq 0$, then pressure positivity is satisfied at x_i , but if $p_i^{n+1} < 0$, then we need to identify which pair of limiters cause p_i^{n+1} to be positive. This is realized using a numerical root finding technique.

To facilitate the discussion, we define a function $Y(\mathbf{u}_i)$ that accepts the conserved variable vector and outputs the pressure. Additionally, we define an operator M that accepts a pair flux limiters and calculates the updated conserved variables by substituting the modified fluxes (3.33) into (3.32); that is, $\mathbf{u}_i^{n+1} = M(\theta_{+,i}, \theta_{-,i})$ where \mathbf{u} is the vector of conserved variables at time step $n + 1$ calculated from the modified fluxes. Thus, the updated pressure can be written as

$$p_i^{n+1} = Y(M(\theta_{+,i}, \theta_{-,i})). \quad (3.35)$$

The condition that must be satisfied for positivity is

$$p_i^{n+1} = Y(M(\theta_{+,i}, \theta_{-,i})) \geq \epsilon_p \quad (3.36)$$

where ϵ_p is defined at each time step as $\epsilon_p = \min_i \{\bar{p}_i^{n+1}, \epsilon^*\}$ where \bar{p}_i^{n+1} is the first order solution for pressure at the $n+1$ time step. The algorithm for positivity preservation is as follows:

1. Begin with the maximum limiters obtained from the density procedure $[\Lambda_{-,i}^\rho, \Lambda_{+,i}^\rho]$.

From these, define three new sets of limiters:

- $A_i^1 = [0, \Lambda_{+,i}^\rho]$,
- $A_i^2 = [\Lambda_{-,i}^\rho, 0]$,
- $A_i^3 = [\Lambda_{-,i}^\rho, \Lambda_{+,i}^\rho]$.

2. For each A_i^j , find the pressure at the next time step, $p_i^{n+1} = Y(M(A_i^j))$. If $p_i^{n+1} \geq \epsilon_p$, then let $B_i^j = A_i^j$, otherwise, we need to find r such that $Y(M(rA_i^j)) \geq \epsilon_p$. This is realized by employing a numerical root finder to identify the zero of the function $F(r_i) = Y(M(r_i A_i^j)) - \epsilon_p$ within $r \in [0, 1]$. The existence of a single root is guaranteed because of the convexity of the pressure, and the usage of $[\Lambda_{-,i}^\rho, \Lambda_{+,i}^\rho]$ ensures the positivity of density [56]. We then take $B_i^j = r_i A_i^j$. Note that this evaluation and comparison are done on a cell by cell basis; that is, r_i is only defined locally for A^j at cell I_i .
3. After applying Step 2 for each A^j , the output is three fully defined B^j vectors that represent the limited forms of A^j . The set of limiters that guarantee positivity for both pressure and density are defined as

$$[\Lambda_{-,i}^p, \Lambda_{+,i}^p] = [\min \{B_0^2, B_0^3\}, \min \{B_1^1, B_1^3\}],$$

where the subscripts denote the zeroth or first element of each B^j .

4. Finally, the overall limiting parameters are defined in a conservative fashion as

$$\begin{cases} \theta_{i,+} = \min \{ \Lambda_{-,i+1}^p, \Lambda_{+,i}^p \} \\ \theta_{i,-} = \min \{ \Lambda_{-,i}^p, \Lambda_{+,i-1}^p \}. \end{cases} \quad (3.37)$$

For Step 2, we found that Brent's method [7] is well suited for this task as it provides a fast, bounded method with guaranteed convergence. However, it should be noted that care must be taken because of the tolerances associated with the root. Because of the convexity of pressure, we know that over $r_i \in [0, 1]$, $F(r)$ must have exactly one root; furthermore, we know $F(r)$ must be an increasing function of r . This can be informally established by considering that $r = 1$ corresponds to the high-order solution and $r = 0$ gives the first order solution. This observation coupled with the monotonicity of F guarantees that if r_0 is the calculated root of F , then for all $r < r_0$, $F < 0$ and positivity is preserved. Likewise, for all $r > r_0$, $F > 0$ and positivity is not preserved. If the root finding algorithm gives r_0 within some tolerance $\pm\delta$, then to guarantee that positivity of pressure, we need to take the root as $\bar{r}_0 = r_0 - \frac{1}{2}\delta$. We found that not incorporating this restriction led to the failure of the positivity scheme in some cases.

3.5 Positivity of the two-fluid model

The application of the above scheme to the two-fluid model is relatively straightforward. Because this model is based upon the stiffened gas equation of state, the

conditions of positivity are slightly different

$$\begin{cases} \Gamma_1 > 0 \\ \rho_i^n \geq 0 \\ p_i^n + \Pi_i^n \geq 0 \end{cases} \quad \forall \quad i, n. \quad (3.38)$$

These three conditions are sufficient to enforce hyperbolicity (real sound speed). The third term arises from the expression for sound speed under the two-fluid model, $c = \sqrt{\frac{\gamma(p+\Pi)}{\rho}}$ [47]. Hereafter, we refer to $(p_i^n + \Pi_i^n)$ as the “modified pressure”.

The implementations of positivity for Γ_1 and ρ are identical to the procedure for ρ in the Euler system case. We treat each case independently and obtain a set of limiters for the order parameter $[\Lambda_{-,i}^\Gamma, \Lambda_{+,i}^\Gamma]$ and density $[\Lambda_{-,i}^\rho, \Lambda_{+,i}^\rho]$. These limiters are then combined as

$$[\Lambda_{-,i}, \Lambda_{+,i}] = [\min \{ \Lambda_{-,i}^\Gamma, \Lambda_{-,i}^\rho \}, \min \{ \Lambda_{+,i}^\Gamma, \Lambda_{+,i}^\rho \}]. \quad (3.39)$$

It should be noted that when forming the overall high-order fluxes at the end of the fourth RK stages $(\mathbf{H}_{i+\frac{1}{2}}^{RK})$, the flux terms for the order parameters (Γ_1 and Γ_2) include the discretization of the source terms (2.26) at each stage. The results of (3.39) are passed into modified pressure positivity algorithm in the same way the density limiters are used in the Euler case. Positivity of modified pressure proceeds in the same manner as laid out for the Euler case; we simply need to substitute the expression $(p_i + \Pi)$ in place of p_i .

3.6 Extension to Two Dimensions

In two spatial dimensions, the addition of a second spatial derivative term requires an extra set of fluxes to limit. Namely, at the end of the final RK stage at each cell

$I = (i, j)$, we have

$$\mathbf{u}_I^{n+1} = \mathbf{u}_I^n - \lambda_x(\mathbf{H}_{i+\frac{1}{2},j}^{RK} - \mathbf{H}_{i-\frac{1}{2},j}^{RK}) - \lambda_y(\mathbf{G}_{i,j+\frac{1}{2}}^{RK} - \mathbf{G}_{i,j-\frac{1}{2}}^{RK}). \quad (3.40)$$

The modified fluxes in each direction are defined as

$$\tilde{\mathbf{H}}_{i\pm\frac{1}{2},j}^{RK} = \mathbf{h}_{i\pm\frac{1}{2},j} + \theta_{\pm,I}^x(\mathbf{H}_{i\pm\frac{1}{2},j}^{RK} - \mathbf{h}_{i\pm\frac{1}{2},j}) \quad (3.41)$$

$$\tilde{\mathbf{G}}_{i,j\pm\frac{1}{2}}^{RK} = \mathbf{g}_{i,j\pm\frac{1}{2}} + \theta_{\pm,I}^y(\mathbf{G}_{i,j\pm\frac{1}{2}}^{RK} - \mathbf{g}_{i,j\pm\frac{1}{2}}). \quad (3.42)$$

For ρ and Γ_1 , we seek to enforce $u^{n+1} \geq \epsilon$ (where we take u as either ρ or Γ_1). Substituting the above definitions for modified fluxes into (3.40), we obtain the following inequality:

$$-\lambda_x\theta_E F_E + \lambda_x\theta_W F_W - \lambda_y\theta_N F_N + \lambda_y\theta_S F_S - \Gamma \geq \epsilon, \quad (3.43)$$

where $\Gamma = -u^n + \lambda_x(h_E - h_W) + \lambda_y(g_N - g_S) \leq \epsilon$, and for brevity, $F_E = H_E^{RK} - h_E$ is denoted as the ‘‘east’’ $(i + \frac{1}{2}, j)$ boundary of the cell $I = (i, j)$, and $F_N = H_N^{RK} - g_N$ is the ‘‘north’’ boundary $(i, j + \frac{1}{2})$, etc.

The decoupling procedure for ρ and Γ_1 is the same as in 1D except we now have sixteen different cases to consider instead of four. Our goal is to identify a vector of four limiters $[\Lambda_E, \Lambda_W, \Lambda_N, \Lambda_S]$ at each cell that gives the maximum permissible θ values that guarantee $u^{n+1} \geq \epsilon$. As a simplification, we adopt the following notation: the tuple $K = (k_1, k_2, k_3, k_4)$ maps to the truth of the following four conditions ($F_E > 0, F_W \geq 0, F_N > 0, F_S \geq 0$), where $k_i \in \{0, 1\}$. Table 3.2 summarizes the decoupling procedure for all 16 combinations.

Table 3.2: Summary of flux limiters for ρ and Γ_1 in 2D

| K | Λ_E | Λ_W | Λ_N | Λ_S | Result |
|-----------|-------------|-------------|-------------|-------------|--------------------------------------------------------------------------------------------|
| (0,0,0,0) | | 1 | | 1 | $\frac{\Gamma}{\lambda_x F_W + \lambda_y F_S - \epsilon}$ |
| (0,0,0,1) | | 1 | | | $\frac{\Gamma}{\lambda_x F_W - \epsilon}$ |
| (0,0,1,0) | | 1 | 1 | 1 | $\frac{\Gamma}{\lambda_x F_W - \lambda_y F_N + \lambda_y F_S - \epsilon}$ |
| (0,0,1,1) | | 1 | 1 | | $\frac{\Gamma}{\lambda_x F_W - \lambda_y F_N - \epsilon}$ |
| (0,1,0,0) | | | | 1 | $\frac{\Gamma}{\lambda_y F_S - \epsilon}$ |
| (0,1,0,1) | | | | | |
| (0,1,1,0) | | | 1 | 1 | $\frac{\Gamma}{-\lambda_y F_N + \lambda_y F_S - \epsilon}$ |
| (0,1,1,1) | | | 1 | | $\frac{\Gamma}{-\lambda_y F_N - \epsilon}$ |
| (1,0,0,0) | 1 | 1 | | 1 | $\frac{\Gamma}{-\lambda_x F_E + \lambda_x F_W + \lambda_y F_S - \epsilon}$ |
| (1,0,0,1) | 1 | 1 | | | $\frac{\Gamma}{-\lambda_x F_E + \lambda_x F_W - \epsilon}$ |
| (1,0,1,0) | 1 | 1 | 1 | 1 | $\frac{\Gamma}{-\lambda_x F_E + \lambda_x F_W - \lambda_y F_N + \lambda_y F_S - \epsilon}$ |
| (1,0,1,1) | 1 | | | 1 | $\frac{\Gamma}{-\lambda_x F_E + \lambda_x F_W - \lambda_y F_N - \epsilon}$ |
| (1,1,0,0) | 1 | | | 1 | $\frac{\Gamma}{-\lambda_x F_E + \lambda_y F_S - \epsilon}$ |
| (1,1,0,1) | 1 | | | | $\frac{\Gamma}{-\lambda_x F_E - \epsilon}$ |
| (1,1,1,0) | 1 | | 1 | 1 | $\frac{\Gamma}{-\lambda_x F_E - \lambda_y F_N + \lambda_y F_S - \epsilon}$ |
| (1,1,1,1) | 1 | 1 | | | $\frac{\Gamma}{-\lambda_x F_E - \lambda_y F_N - \epsilon}$ |

For each K case, the noted Λ values are set to the value in the “Result” column. Just as in the 1D case, all Λ^ρ and Λ^{Γ_1} values are found for ρ and Γ_1 , and then the minimum between them is taken at each cell boundary. Thus, at each cell, taking

$$\{(\theta_E, \theta_W, \theta_N, \theta_S) \mid 0 \leq \theta_E \leq \Lambda_E, 0 \leq \theta_W \leq \Lambda_W, 0 \leq \theta_N \leq \Lambda_N, 0 \leq \theta_S \leq \Lambda_S\} \quad (3.44)$$

guarantees the positivity of both ρ and Γ_1 .

These limiters are then passed into an algorithm that modifies these limiters such that $p + \Pi$ positivity is guaranteed by $[\Lambda_E^p, \Lambda_W^p, \Lambda_N^p, \Lambda_S^p]$. The process closely follows the 1D case; we let $p^{n+1} + \Pi^{n+1} = Y(M(\theta_E, \theta_W, \theta_N, \theta_S))$ and iterate through each combination of limiters checking whether $p^{n+1} + \Pi^{n+1} \geq \epsilon$ is satisfied. However, we

now have fifteen different combinations to check. To this end, we define

$$A^{(k_1, k_2, k_3, k_4)} = (k_1 \Lambda_E, k_2 \Lambda_W, k_3 \Lambda_N, k_4 \Lambda_S) \quad (3.45)$$

where $k_i \in \{0, 1\}$ and $(k_1, k_2, k_3, k_4) \neq (0, 0, 0, 0)$. Just as for the 1D case, for each A , if $Y(M(A^{(k_1, k_2, k_3, k_4)})) \geq \epsilon$, we take $B^{(k_1, k_2, k_3, k_4)} = A^{(k_1, k_2, k_3, k_4)}$, else we solve for r such that $Y(M(rA^{(k_1, k_2, k_3, k_4)})) \geq \epsilon$ and take $B^{(k_1, k_2, k_3, k_4)} = rA^{(k_1, k_2, k_3, k_4)}$.

After solving for each of the fifteen $B^{(k_1, k_2, k_3, k_4)}$ at every cell in the grid, the pressure limiters are assigned as

$$\Lambda_E^p = \min \left\{ B_1^{(1,0,0,0)}, B_1^{(1,0,0,1)}, B_1^{(1,0,1,0)}, B_1^{(1,0,1,1)}, B_1^{(1,1,0,0)}, B_1^{(1,1,0,1)}, B_1^{(1,1,1,1)} \right\} \quad (3.46)$$

$$\Lambda_W^p = \min \left\{ B_2^{(0,1,0,0)}, B_2^{(0,1,0,1)}, B_2^{(0,1,1,0)}, B_2^{(0,1,1,1)}, B_2^{(1,1,0,0)}, B_2^{(1,1,0,1)}, B_2^{(1,1,1,1)} \right\} \quad (3.47)$$

$$\Lambda_N^p = \min \left\{ B_3^{(0,0,1,0)}, B_3^{(0,0,1,1)}, B_3^{(0,1,1,0)}, B_3^{(0,1,1,1)}, B_3^{(1,0,1,0)}, B_3^{(1,0,1,1)}, B_3^{(1,1,1,1)} \right\} \quad (3.48)$$

$$\Lambda_S^p = \min \left\{ B_4^{(0,0,0,1)}, B_4^{(0,0,1,1)}, B_4^{(0,1,0,1)}, B_4^{(0,1,1,1)}, B_4^{(1,0,0,1)}, B_4^{(1,0,1,1)}, B_4^{(1,1,1,1)} \right\} \quad (3.49)$$

Finally, the overall, conservative limiters are locally taken to be

$$\theta_{i+\frac{1}{2},j} = \min \left\{ \Lambda_{E,I}^p, \Lambda_{W,I+1}^p \right\} \quad (3.50)$$

$$\theta_{i-\frac{1}{2},j} = \min \left\{ \Lambda_{E,I-1}^p, \Lambda_{W,I}^p \right\} \quad (3.51)$$

$$\theta_{i,j+\frac{1}{2}} = \min \left\{ \Lambda_{N,I}^p, \Lambda_{S,I+1}^p \right\} \quad (3.52)$$

$$\theta_{i,j-\frac{1}{2}} = \min \left\{ \Lambda_{N,I-1}^p, \Lambda_{S,I}^p \right\}. \quad (3.53)$$

Chapter 4

Verification

This chapter is concerned with numerically verifying the methods previously developed. Convergence studies for two cases of scalar advection (both linear and non-linear) are considered first. These examples are followed by several 1D numerical problems designed to test specific features of the positivity-preserving method. Finally, two applications of 2D shock induced bubble collapse are presented.

4.1 Implementation

The finite difference WENO scheme along with the positivity algorithm were implemented in the Python language for one and two dimensions. To enhance performance, the code utilized the Numpy package, which greatly extends the computational capabilities of Python by introducing N -dimensional arrays and vectorized operations [22]. Throughout the code, Numpy arrays were used exclusively, as the native Python list object was found to provide detrimentally slow performance.

Several test problems were considered to rigorously verify accuracy and performance of our proposed high-order positivity-preserving WENO finite difference scheme. For all test problems, a standard 4th order Runge-Kutta time integration was utilized. Additionally, the global Lax-Friedrichs flux splitting technique was employed for all problems. In most cases, the CFL number was set to 0.4; however, we did find that some problems required a lower CFL (about 0.1) to run the 9th and 11th order schemes. These instances are noted.

4.2 Scalar Conservation Laws

We begin by verifying the high-order convergence and maximum principle preserving quality of the scheme for scalar conservation laws.

4.2.1 Linear Advection with Continuous Initial Data

We first consider the linear advection equation

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0 \quad (4.1)$$

with a smooth initial solution $u(x, 0) = \sin^2(\pi x - \sin(\pi x)/\pi)$ over the domain $x \in [-1, 1]$. Periodic boundary conditions are enforced, and the maximum time is set to $t_{\max} = 20$ corresponding to 10 full advections of the property u through the computational domain. The exact solution is

$$u(x, t) = \sin^2 \left(\pi(x - t) - \frac{\sin(\pi(x - t))}{\pi} \right). \quad (4.2)$$

Note that, in an effort to avoid polluting the spatial error with temporal error, we use a very small CFL number (0.05) for this example. The results are tabulated in Table 4.1, where the L_∞ error norms for two cases, with and without the positivity limiters, are given along with the global minimum and maximum values obtained throughout the numerical solution.

The effectiveness and accuracy of the positivity scheme are clearly demonstrated in Table 4.1. For all simulations, the upper and lower bounds are preserved with no observable impact upon accuracy. In other words optimal high-order accuracy and maximum principle property are both preserved using our proposed scheme.

Table 4.1: Convergence study for scalar linear advection — CFL=0.05

| Order | N | Without Limiters | | | | With Limiters | | | |
|-------|-----|------------------|------------|------------|-------|---------------|------------|------------|-------|
| | | Error | min(u) | max(u) | Rate | Error | min(u) | max(u) | Rate |
| 5 | 20 | 8.152e-02 | -6.084e-03 | 1.001 | | 9.317e-02 | 8.800e-17 | 1.000 | |
| | 40 | 2.211e-03 | -9.728e-05 | 1.000 | 5.204 | 2.552e-03 | 9.926e-17 | 1.000 | 5.190 |
| | 80 | 6.361e-05 | -2.966e-06 | 1.000 | 5.119 | 7.466e-05 | 9.992e-17 | 1.000 | 5.095 |
| | 160 | 1.998e-06 | -9.281e-08 | 1.000 | 4.992 | 2.345e-06 | 9.999e-17 | 1.000 | 4.993 |
| | 320 | 6.262e-08 | -2.901e-09 | 1.000 | 4.996 | 7.347e-08 | 1.000e-16 | 1.000 | 4.997 |
| | 640 | 1.958e-09 | -9.065e-11 | 1.000 | 4.999 | 2.297e-09 | 1.000e-16 | 1.000 | 4.999 |
| 7 | 20 | 2.772e-02 | -3.329e-04 | 1.001 | | 2.947e-02 | 9.610e-17 | 1.000 | |
| | 40 | 2.402e-04 | -1.377e-06 | 1.000 | 6.851 | 3.251e-04 | 9.987e-17 | 1.000 | 6.503 |
| | 80 | 1.479e-06 | -3.349e-09 | 1.000 | 7.343 | 1.492e-06 | 9.995e-17 | 1.000 | 7.768 |
| | 160 | 9.797e-09 | -2.290e-11 | 1.000 | 7.238 | 9.883e-09 | 9.999e-17 | 1.000 | 7.238 |
| | 320 | 7.327e-11 | -1.725e-13 | 1.000 | 7.063 | 7.392e-11 | 1.000e-16 | 1.000 | 7.063 |
| | 640 | 2.297e-13 | -1.065e-15 | 1.000 | 7.000 | 2.297e-13 | 1.000e-16 | 1.000 | 7.000 |
| 9 | 20 | 8.687e-03 | -5.520e-05 | 1.003 | | 1.126e-02 | 1.033e-05 | 1.000 | |
| | 40 | 1.224e-05 | 5.464e-09 | 1.000 | 9.471 | 1.224e-05 | 5.464e-09 | 1.000 | 9.846 |
| | 80 | 2.845e-08 | 5.543e-12 | 1.000 | 8.749 | 2.845e-08 | 5.543e-12 | 1.000 | 8.749 |
| | 160 | 9.901e-11 | 5.346e-15 | 1.000 | 8.166 | 9.901e-11 | 5.346e-15 | 1.000 | 8.166 |
| 11 | 20 | 3.903e-03 | -7.674e-05 | 1.002 | | 6.955e-03 | 3.860e-05 | 1.000 | |
| | 40 | 1.465e-06 | 2.018e-10 | 1.000 | 11.4 | 1.465e-06 | 2.018e-10 | 1.000 | 12.2 |
| | 80 | 1.650e-09 | 2.776e-14 | 1.000 | 9.794 | 1.650e-09 | 2.776e-14 | 1.000 | 9.794 |

4.2.2 Inviscid Burger's Equation

Next, we consider the non-linear inviscid Burger's equation

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) = 0$$

on the domain $x \in [-1, 1]$ with $t_{\max} = 0.2$ and the initial condition $u(x, 0) = \frac{1+\sin(\pi x)}{2}$. Once again, we use very small CFL number to avoid contaminating the convergence rate with temporal error. Table 4.2 presents the L_∞ error norms along with the minimum and maximums values. Note that for fine meshes ($N \geq 320$), the higher order schemes give error values of the same order as the error associated with double machine precision; these values are meaningless and are not reported.

Table 4.2: Convergence study for Burger’s Equation — CFL=0.05

| Order | N | Without Limiters | | | | With Limiters | | | |
|-------|-----|------------------|------------|------------|------|---------------|------------|------------|------|
| | | Error | min(u) | max(u) | Rate | Error | min(u) | max(u) | Rate |
| 5 | 40 | 1.62e-05 | -1.39e-06 | 1.00 | | 1.72e-05 | 9.97e-17 | 1.00 | |
| | 80 | 5.41e-07 | -4.61e-08 | 1.00 | 4.90 | 5.63e-07 | 6.61e-10 | 1.00 | 4.93 |
| | 160 | 1.70e-08 | -1.37e-09 | 1.00 | 4.99 | 2.11e-08 | 1.06e-11 | 1.00 | 4.74 |
| | 320 | 5.34e-10 | -4.05e-11 | 1.00 | 4.99 | 9.36e-10 | 1.67e-13 | 1.00 | 4.49 |
| | 640 | 2.67e-11 | -1.23e-12 | 1.00 | 4.33 | 5.19e-11 | 2.62e-15 | 1.00 | 4.17 |
| 7 | 40 | 2.37e-06 | -1.90e-08 | 1.00 | | 2.38e-06 | 9.98e-17 | 1.00 | |
| | 80 | 2.24e-08 | -1.28e-10 | 1.00 | 6.72 | 2.25e-08 | 9.99e-17 | 1.00 | 6.73 |
| | 160 | 1.82e-10 | -9.70e-13 | 1.00 | 6.95 | 1.84e-10 | 8.28e-17 | 1.00 | 6.93 |
| | 320 | 1.49e-12 | -7.51e-15 | 1.00 | 6.94 | 1.67e-12 | 8.10e-19 | 1.00 | 6.78 |
| 9 | 40 | 5.55e-07 | -1.36e-10 | 1.00 | | 5.55e-07 | 9.99e-17 | 1.00 | |
| | 80 | 1.77e-09 | -2.34e-13 | 1.00 | 8.30 | 1.77e-09 | 5.85e-18 | 1.00 | 8.30 |
| | 160 | 3.97e-12 | -4.34e-16 | 1.00 | 8.80 | 4.13e-12 | 1.75e-19 | 1.00 | 8.74 |
| 11 | 20 | 3.90e-03 | -7.67e-05 | 1.00 | | 6.95e-03 | 3.86e-05 | 1.00 | |
| | 40 | 1.47e-06 | 2.02e-10 | 1.00 | 11.4 | 1.47e-06 | 2.02e-10 | 1.00 | 12.2 |
| | 80 | 1.65e-09 | 2.78e-14 | 1.00 | 9.79 | 1.65e-09 | 2.78e-14 | 1.00 | 9.79 |

These results confirm the high-order accuracy and maximum principle preserving quality of the scheme for the non-linear case. For all orders and grid sizes, the scheme without limiters obtained values below zero while the scheme with limiters always remained positive.

4.3 Euler System

For the next set of test problems, we consider two challenging single fluid flow problems that contain very large density and pressure ratios across sharp features. For each of these problems, we found the positivity scheme a requirement to run the simulation; without the flux limiting algorithm, the simulations crashed.

4.3.1 Interacting Blast Waves

Here we simulate the collision of a Mach 199 shock wave with a Mach 63 shock wave traveling the other direction. This is a modified form of the interacting blast wave problem discussed in [54, 18]. Specifically, we have moved the two initial blast waves much closer together to reduce the effects of numerical diffusion and preserve the sharpness of the shock discontinuities at the collision. Owing to the large pressure ratios across the shocks and the very low minimum pressure values, this problem is designed to verify the preservation of pressure positivity. The domain is taken as $[0, 1]$ and the specific heat ratio is selected to correspond to air ($\gamma = 1.4$). The initial data are

$$(\rho, v, p) = \begin{cases} (1, 0, 1000) & 0 \leq x < 0.6 \\ (1, 0, \frac{1}{100}) & 0.6 \leq x < 0.7 \\ (1, 0, 100) & 0.7 \leq x \leq 1. \end{cases} \quad (4.3)$$

Figure 4.1 shows 11th order pressure solutions at various time steps. The first panel shows the initial setup with the very large pressure discontinuities. Panel (b) shows the two shock waves approaching each other and refraction waves propagating in the opposite direction, and in panel (c), the two shock waves have collided to generate a very high pressure. Note how few grid points are contained within this jump; for the 11th order solution, this feature is accurately captured by about five grid points, demonstrating the high-order schemes' ability to resolve extremely fine features. In contrast, the first order scheme required about 10 times the grid resolution to approach the sharpness of the high-order solutions (this is further explored in Figure 4.3). Panel (d) shows the leftward and rightward traveling shocks propagating after the collision into the air previously compressed by the wave traveling the opposite direction. Finally, panel (e) shows the rightward traveling transmitted wave overtaking the initial rightward traveling expansion wave.

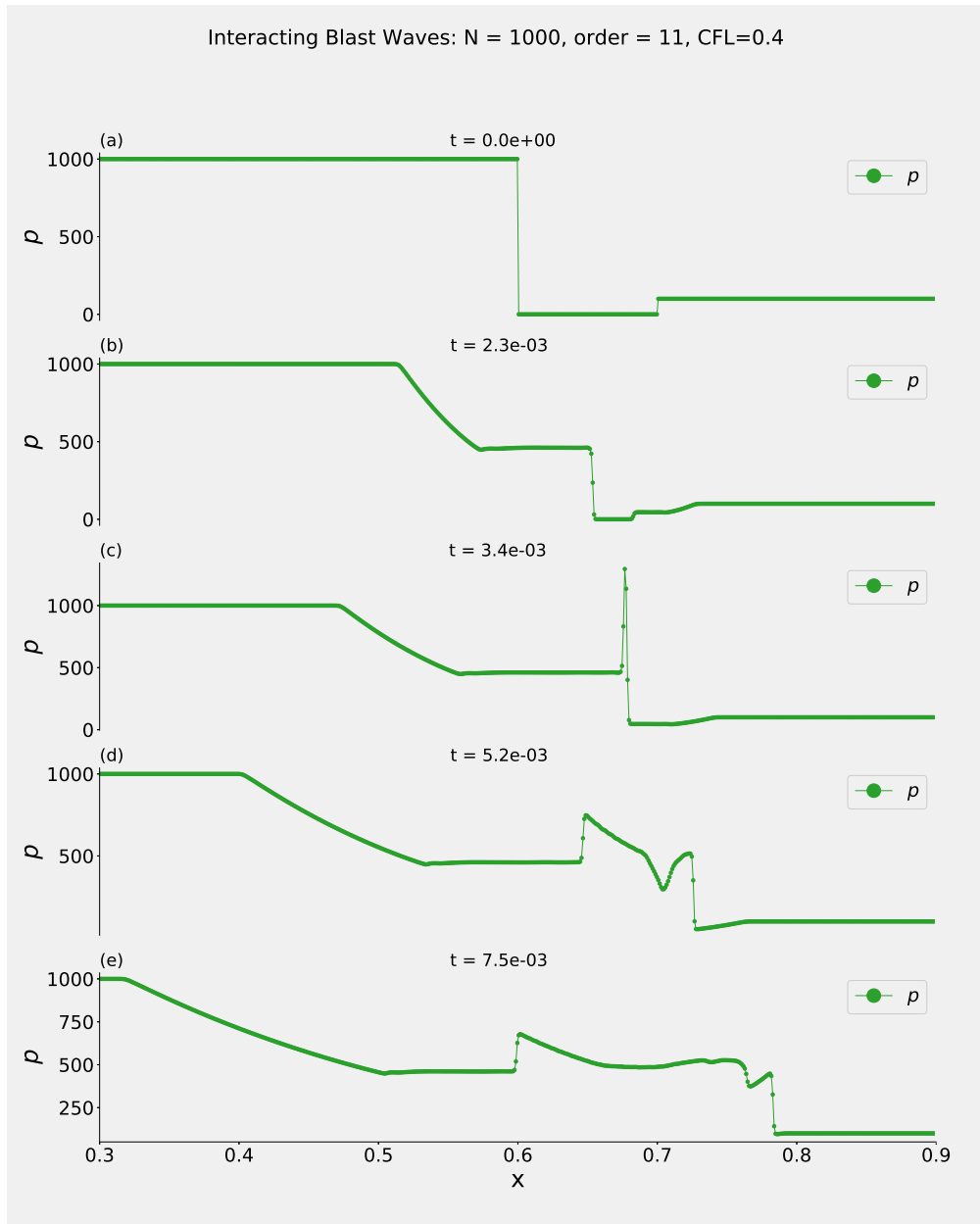


Figure 4.1: Pressure solutions for the interacting blast waves problem

The 7th order density solutions are shown in Figure 4.2 for the same time steps as in Figure 4.1. The second panel shows the two shock waves traveling towards each other, and the third shows the sudden density jump immediately following their collision.

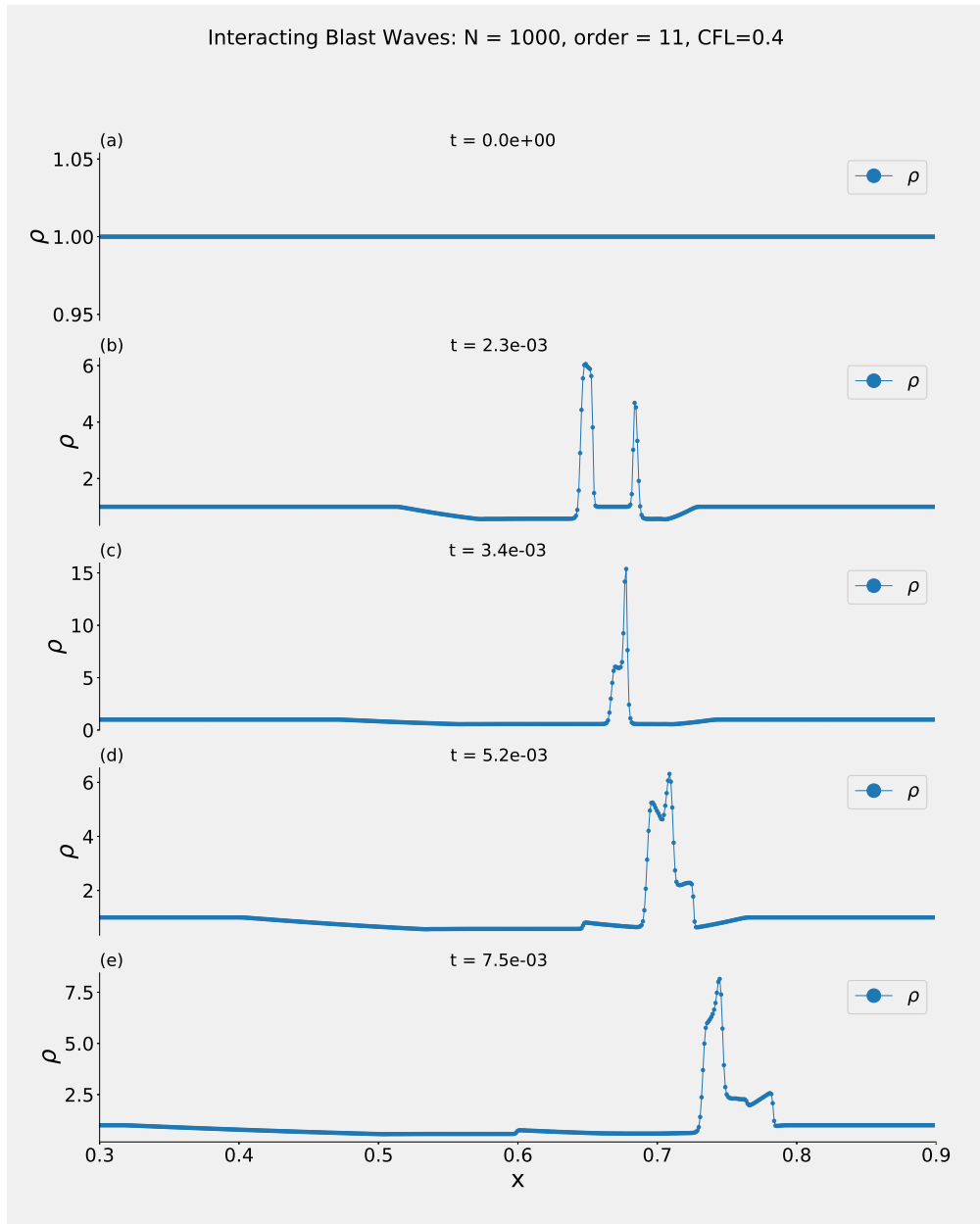


Figure 4.2: Density solutions for the interacting blast waves problem

The fourth and fifth panels show the continued evolution of the complex post-shock density profile. Note the sharp features present throughout the simulation; as evidenced in Figure 4.3, the high-order schemes are able to effectively capture these profiles while the lower order schemes require significant grid refinement to approach comparable sharpness.

A comparison of the 3rd, 5th, 7th, 9th, and 11th order density solutions at $t = 5.25 \times 10^{-3}$ is given in Figure 4.3. The solid black line is the “exact” converged solution calculated using the 7th order scheme with 10^4 grid points. Clearly, the 3rd order solution fails to capture any of the sharp features. The 5th order solution fares better, but it is noticeably less sharp than the higher orders, which are nearly indistinguishable at this scaling. However, near the peak, it is clear that the 11th order solution is indeed the sharpest.

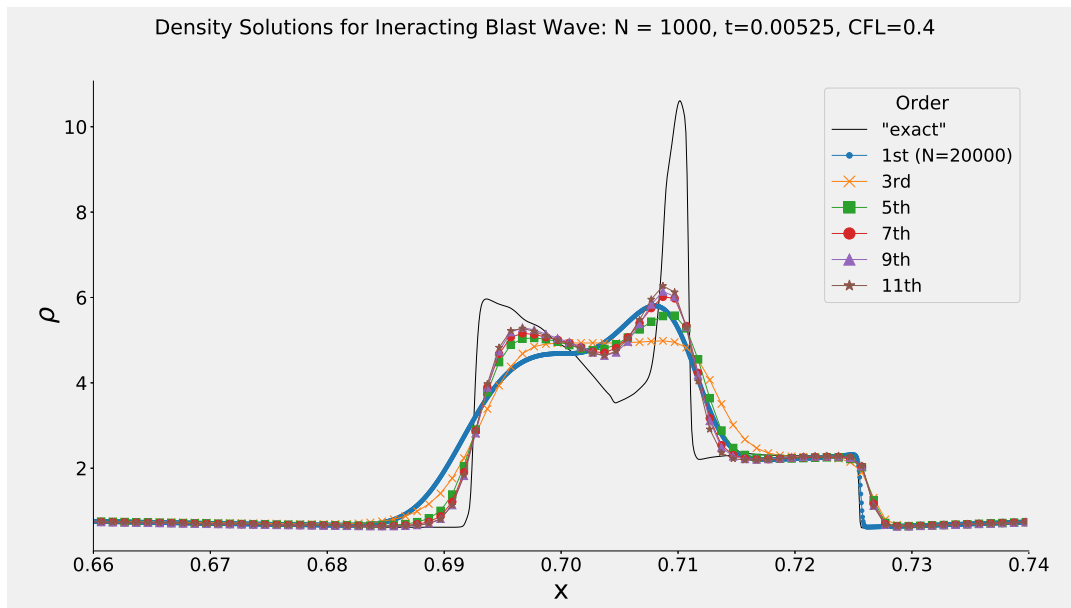


Figure 4.3: Comparison of high-order density solutions for interacting blast waves problem

This problem demonstrates the positivity-reserving quality of the scheme in the presence of very large pressure jumps (of magnitude 10^5) from low pressure regions ($p_{min} = 0.01$). Furthermore, it was found that for all high-order schemes, the positivity limiters were required to prevent the simulation from crashing due to negative pressure thus demonstrating the necessity of the positivity algorithm.

4.3.2 Euler System — Shock, Interface Problem

Next, we consider the problem of a strong shock wave ($M = 8.96$) interacting with a density interface. This problem is designed to test the ability of the algorithm

to preserve the positivity of density in the extreme case of a very large density jump. The shock is initially traveling rightward through helium ($\gamma = 1.67$) and is placed two grid points away from the density interface. This is done to make the problem more challenging by forcing the shock/interface interaction to occur before the effects of numerical diffusion reduce the sharpness of the discontinuities. The initial conditions are

$$(\rho, v, p) = \begin{cases} (0.384, 27.086, 100.176) & 0 \leq x < (0.5 - 2h) \\ (0.1, 0, 1) & (0.5 - 2h) \leq x \leq 0.5 \\ (100, 0, 1) & 0.5 < x \leq 1 \end{cases} \quad (4.4)$$

Figure 4.4 demonstrates the necessity for positivity limiters for this problem. This figure shows a zoomed in view of the density interface after one time step for the 1st-order solution along with two 5th-order solutions — one using the positivity limiters and one without. The scheme with limiters maintains positive density, while the standard scheme develops a negative density value prohibiting any further time steps.

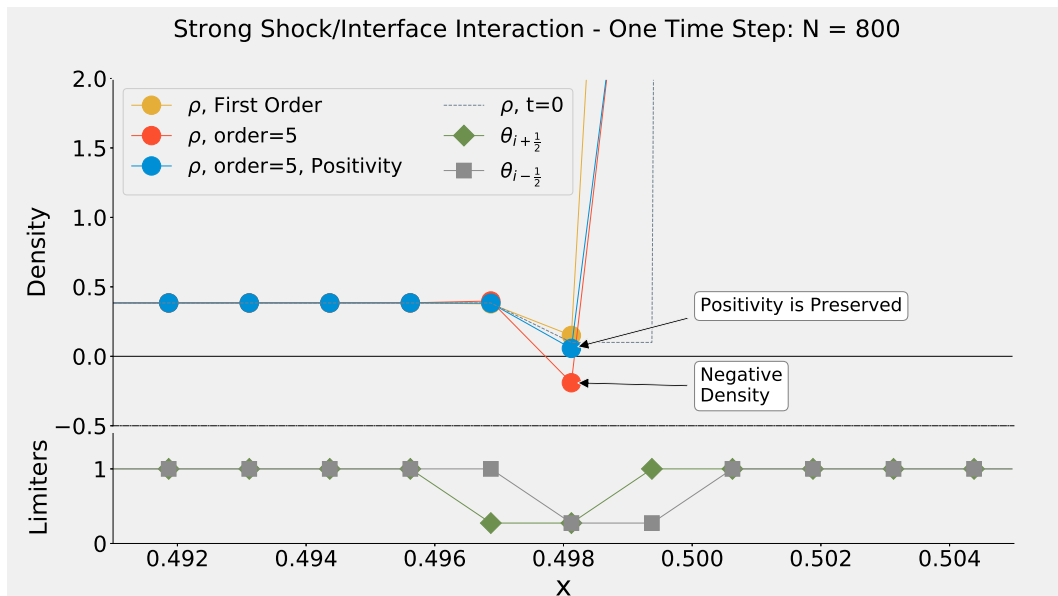


Figure 4.4: Demonstration of positivity-preserving scheme for strong shock/interface interaction

Using the positivity scheme, this problem has been solved for all orders. We found that for 9th order and higher, a lower CFL number ($\text{CFL} \approx 0.1$) was required; however, both the 5th and 7th order schemes had no issue with the standard $\text{CFL}=0.4$. Additionally, the 9th and 11th order solution suffered from some slight oscillations at coarser grids. Solutions at $t = 0.01$ are presented in Figures 4.5, 4.6, and 4.7; the black lines indicate the grid converged “exact” solution calculated using the 7th order scheme with 10^4 points, and the dashed lines give the initial conditions.

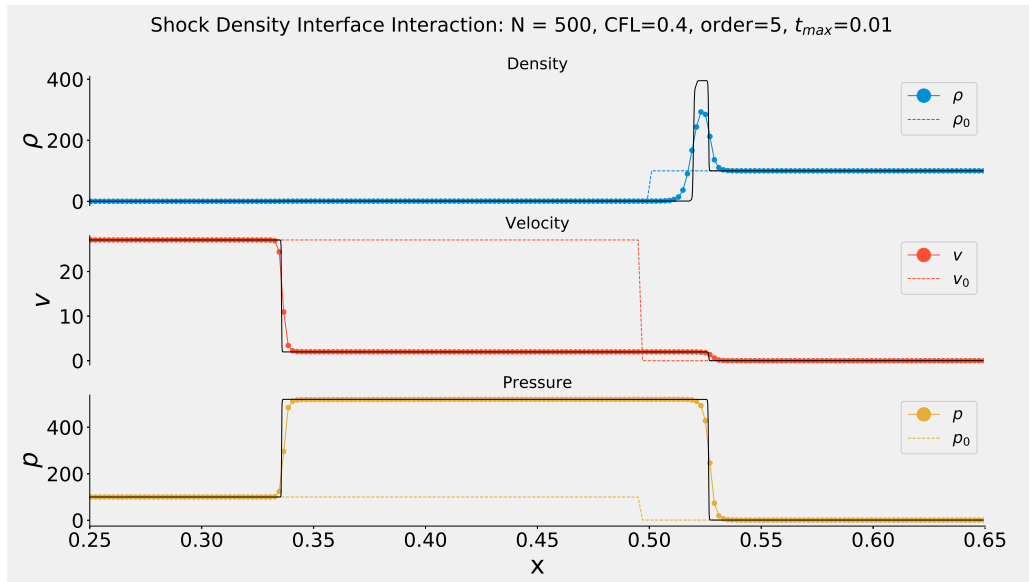


Figure 4.5: 5th order solution for shock air/helium interface problem

In the pressure trace, we observe the incident shock wave traveling rightward through the high density region near $x \approx 0.55$. Additionally, a reflected wave is propagating leftward into the initial high pressure region near $x \approx 0.35$. The density jump observed immediately trailing the rightward traveling shock is the result of the shock wave leading the advection of the initial density/temperature interface. Note the increasing sharpness of this feature with approximation order.

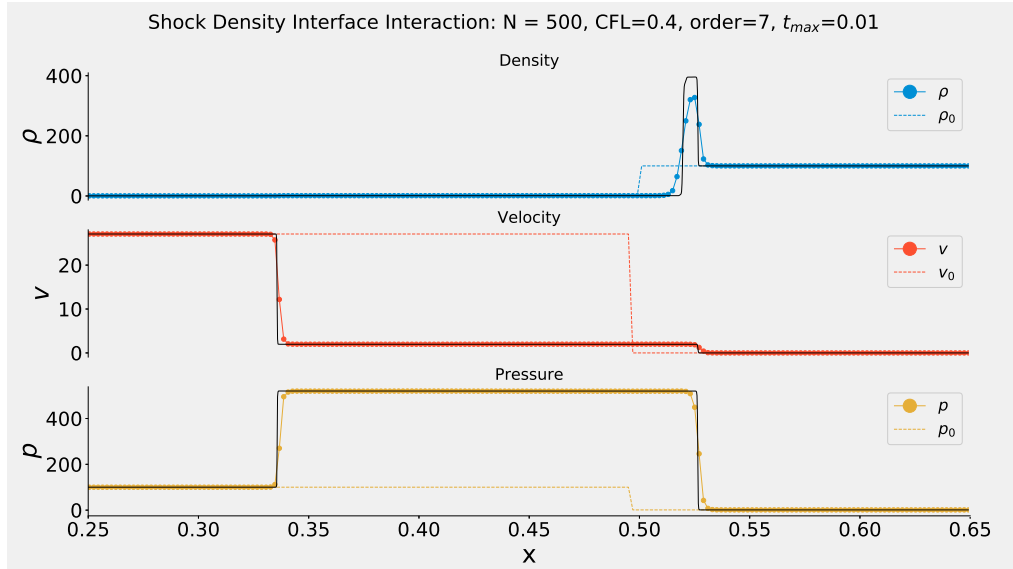


Figure 4.6: 7th order solution for shock air/helium interface problem

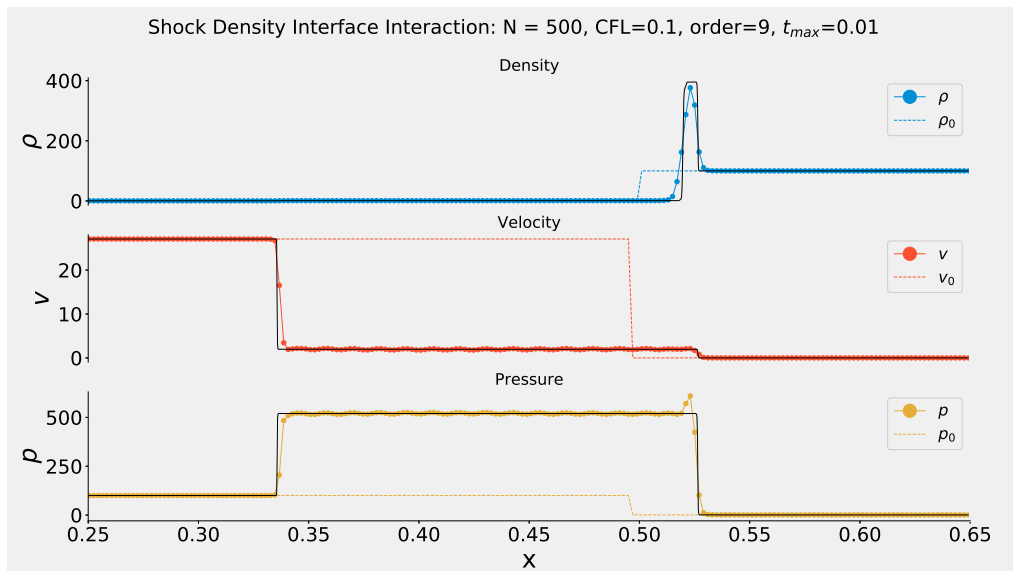


Figure 4.7: 9th order solution for shock air/helium interface problem

To confirm the benefits of high-order schemes, Figure 4.8 compares high-order solutions to a first order solution of comparable accuracy. The number of grid points and CPU time for each scheme are shown in the legend. Clearly, the 5th order accurate scheme outperforms the 1st-order scheme as it delivers a similar level of accuracy to the first-order scheme at an approximately 34-fold reduction in CPU time.

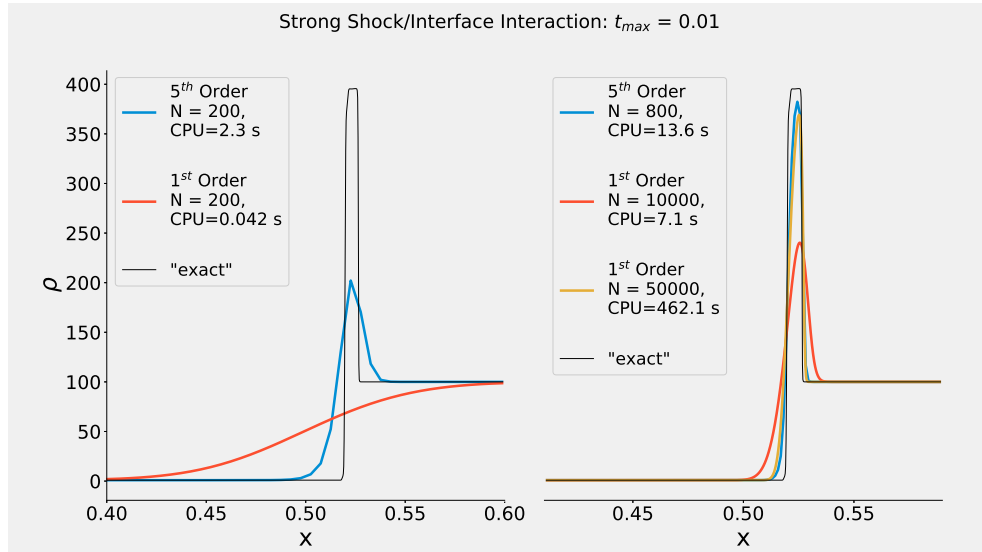


Figure 4.8: Comparison of high and low order solutions for shock interface problem

This test problem demonstrates the ability of the algorithm to successfully maintain positivity of density under very extreme conditions. Additionally, the significantly greater efficiency of high-order schemes in the presence of sharp features is demonstrated. As in the last test problem, the limiters were required for all orders to solve this problem.

4.4 Two-Fluid System

The next two problems test the performance of the positivity scheme applied to the two-fluid model.

4.4.1 High Pressure Shock Tube

To verify the positivity scheme in the presence of an interface between two vapors, we solve a shock tube or Riemann problem consisting of very high pressure air separated from a volume of low pressure helium with both gasses initially at rest. The

initial data are

$$(\rho, v, p, \Gamma, \Pi) = \begin{cases} (1, 0, 10^4, 1.4, 0) & 0 \leq x < 0.5 \\ (1, 0, \frac{1}{100}, 1.67, 0) & 0.5 \leq x \leq 1 \end{cases}. \quad (4.5)$$

This problem has been solved for all orders with no observed CFL restriction. Once again, it was found that the positivity scheme was required for all orders. All attempts to solve the problem without the positivity algorithm resulted in negative pressure values after the first time step. The solutions are shown in Figs. 4.10—4.13 with the “exact” solution calculated using a 7th order scheme with $N = 5000$ grid points. The initial data are plotted as dashed lines.

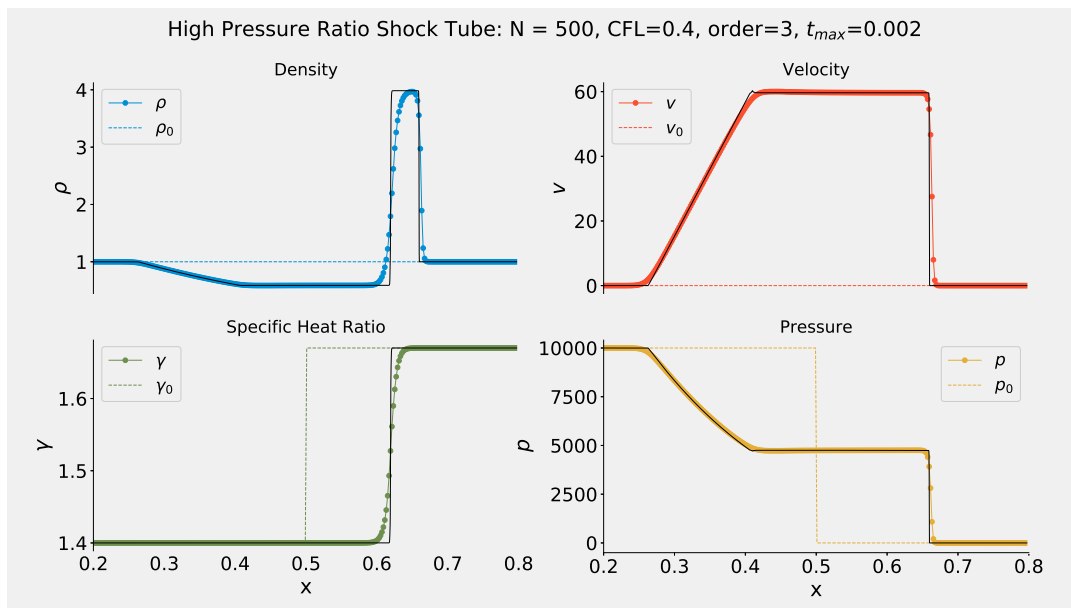


Figure 4.9: 3rd order solution to high pressure shock tube problem

The dynamics of the shock tube problem consist of a strong shock wave propagating rightward into the low pressure helium and a series of expansion waves traveling leftward into the high pressure air. Both features are clearly seen in the pressure trace. Another key component of the solutions is the high density, compressed helium region between the shock front and the material interface defined by the specific

heat ratio. As in the previous test problem, the increased sharpness of the higher order schemes is most clearly observed near this feature.

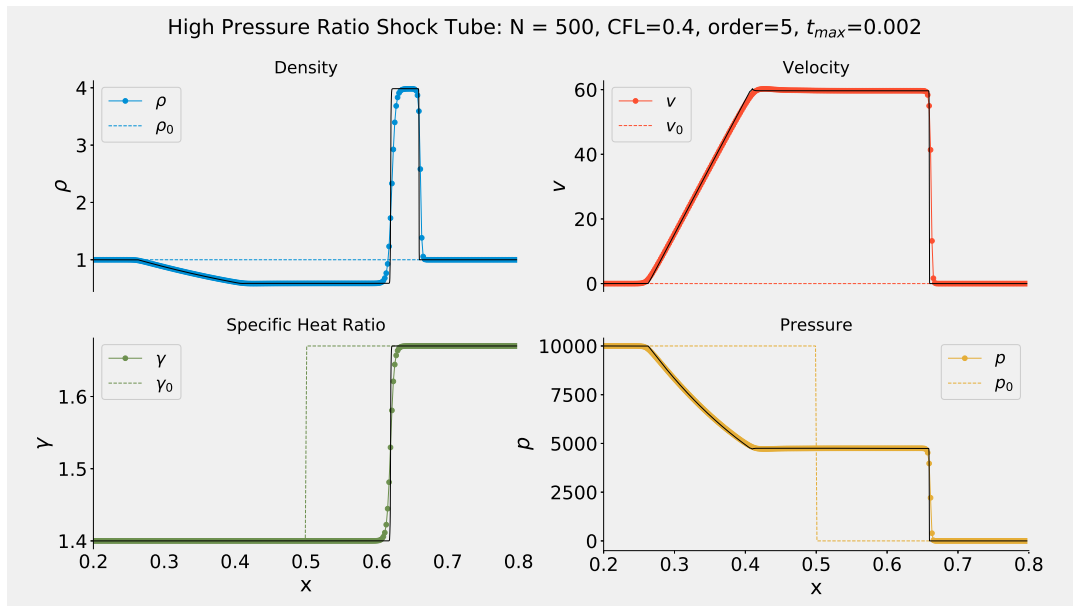


Figure 4.10: 5th order solution to high pressure shock tube problem

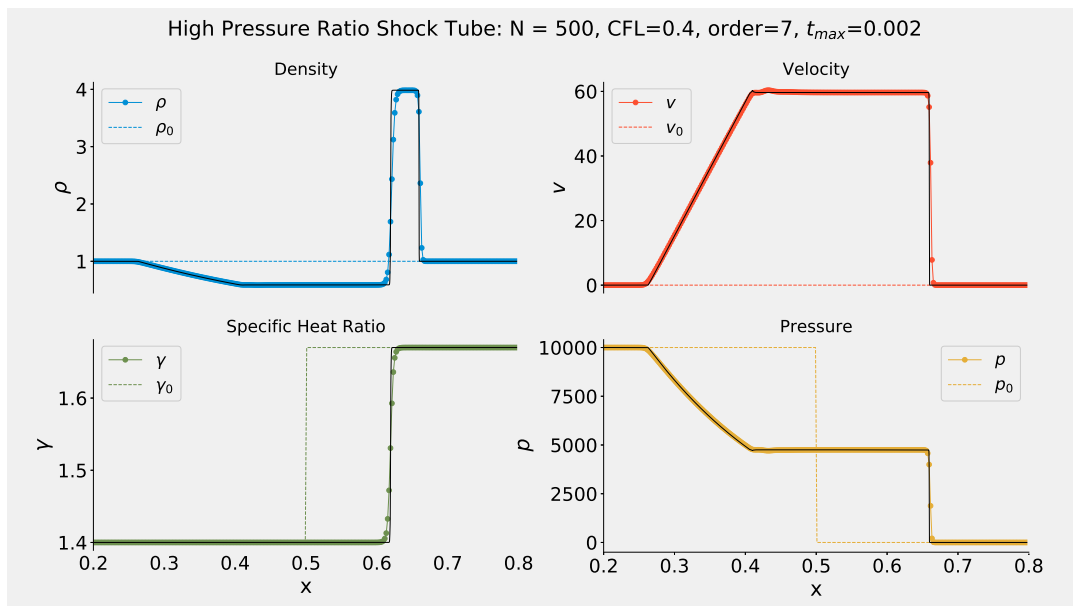


Figure 4.11: 7th order solution to high pressure shock tube problem

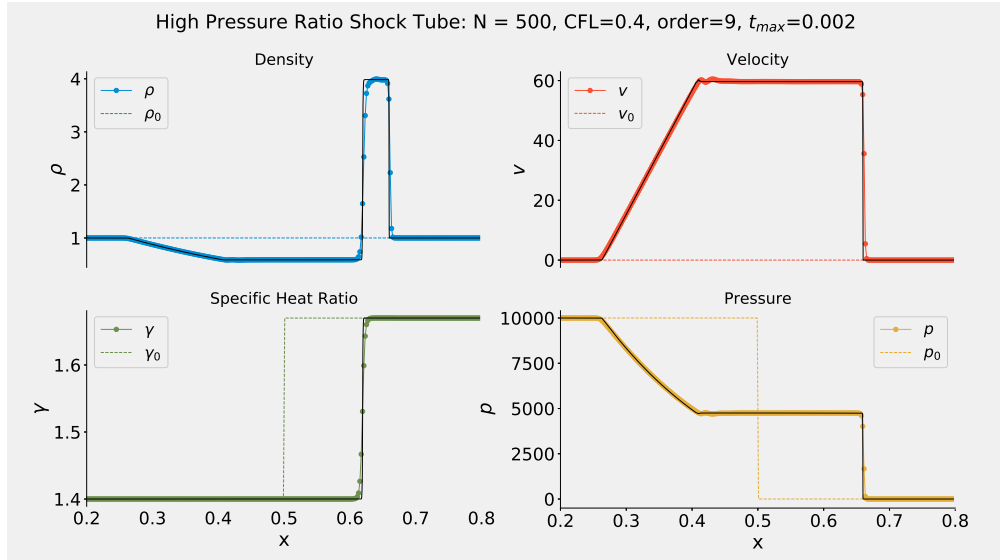


Figure 4.12: 9th order solution to high pressure shock tube problem

For the 9th and 11th order solutions, some oscillation can be observed near the density peak trailing the shock wave and at other sharp features. These oscillations diminish with grid refinement and do not impact the overall convergence of the solution. Similar behavior was observed by Gerolymos et al. [18] for their very high-order accurate finite volume methods.

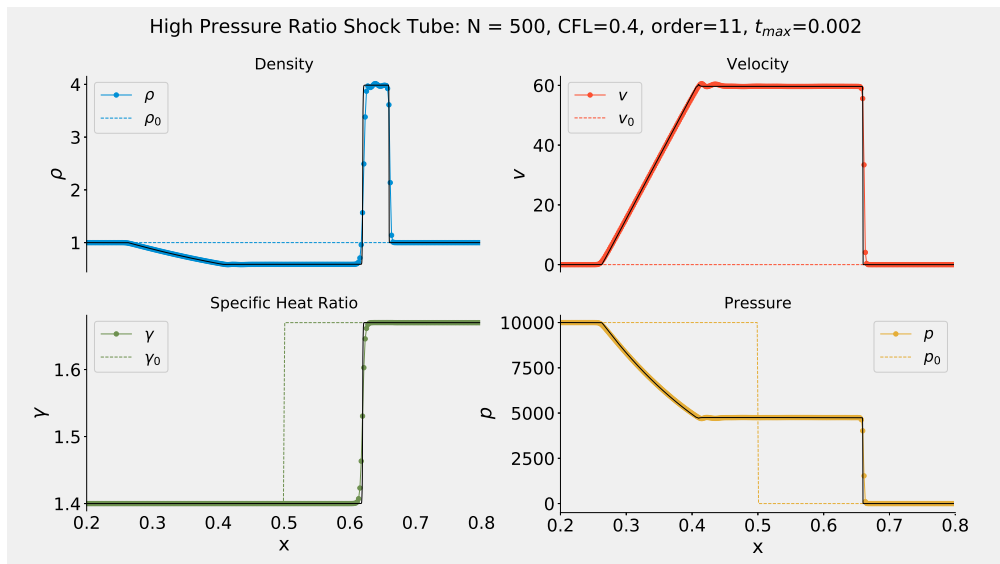


Figure 4.13: 11th order solution to high pressure shock tube problem

This problem confirms the performance of the positivity method for a two-vapor

interface separated by an inordinately large pressure ratio (10^6). No CFL restrictions were encountered, as the problem was solved up through 11th order using the standard $CFL = 0.4$.

4.4.2 Water and Air Shock, Interface Problem

Next, we verify the positivity-preserving performance for a two-phase problem. The initial data consist of a region of quiescent air in equilibrium with water at atmospheric conditions. A Mach 2.5 shock is propagating through the water and impinges upon the interface. The water and air properties are chosen as

$$(\rho, v, p, \Gamma, \Pi) = \begin{cases} (1449 \frac{kg}{m^3}, 1281 \frac{m}{s}, 5.26 \times 10^9 Pa, 4.40, 6.15 \times 10^8 Pa) & \text{Post Shock} \\ (998 \frac{kg}{m^3}, 0, 1.01 \times 10^5 Pa, 4.40, 6.15 \times 10^8 Pa) & \text{Pre Shock} \\ (1.22 \frac{kg}{m^3}, 0, 1.01 \times 10^5 Pa, 1.4, 0) & \text{Air.} \end{cases} \quad (4.6)$$

The post shock conditions have been calculated using the Rankine-Hugoniot relations for the stiffened gas equation of state derived in [27]. The computational domain is $[-1m, 1m]$, and the initial locations of the shock front and liquid/air interface are selected as $x = -0.1m$ and $x = 0$ respectively. The equation of state parameters for water (γ and Π) have been chosen to coincide with the values presented in [42] for the given Mach number regime.

In solving this problem, we found that some reduction in CFL was required for the 7th and higher order schemes. In particular, we used $CFL=0.3$ for 7th and 9th order and $CFL=0.21$ for 11th order. Additionally, we found that the positivity limiters were only required for the 7th and higher order schemes; both the 3rd and 5th order schemes were able to provide solutions with $CFL=0.4$ without the limiters. The 11th and 5th order solutions for density, pressure and the two material properties are

presented in Figures 4.14 and 4.15.

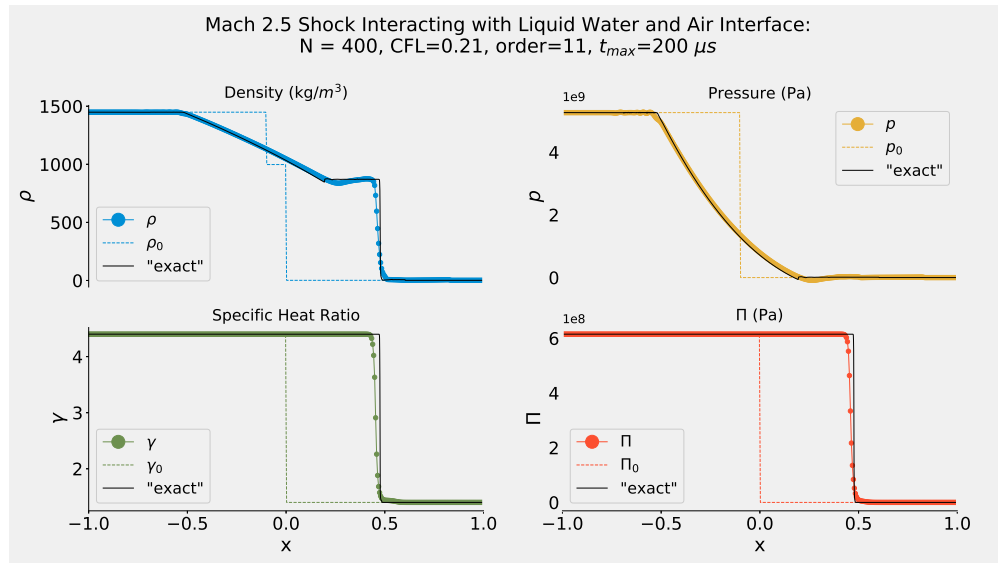


Figure 4.14: 11th order solution to the shock liquid/air interface problem

The visible distinguishing features are the interface near $x = 0.5$ and the refraction wave propagating leftward into the water. This expansion wave is the result of the initial shock impinging upon the liquid/gas interface. Because of the scaling of the problem, the actual shock front is not discernable; although a zoomed in view reveals the shock propagating within the gas ahead of the interface near $x = 0.8$ m.

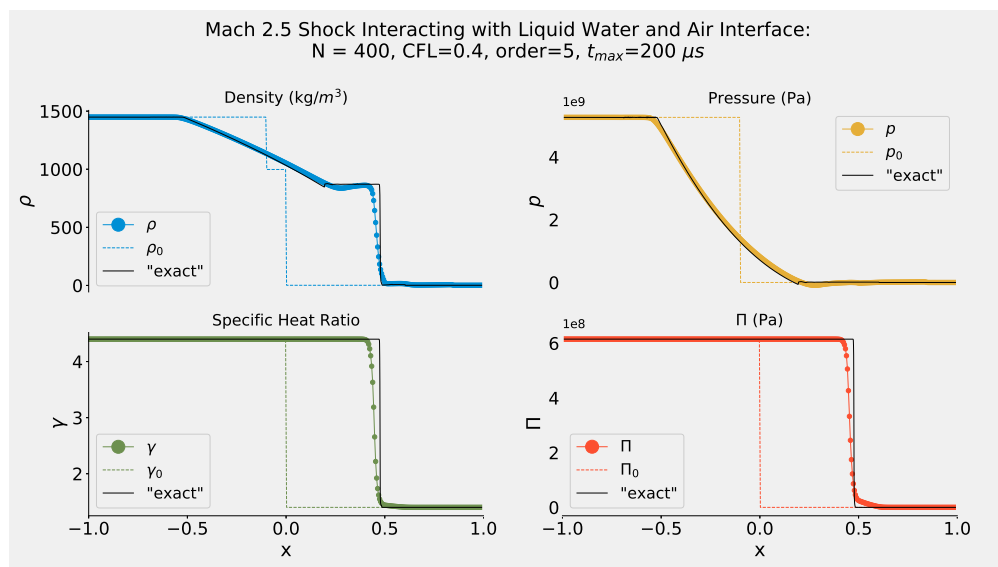


Figure 4.15: 5th order solution to the shock liquid/air interface problem

Figure 4.16 depicts the material properties at the water/air interface for several different solution orders. Clearly, the higher order methods capture the interface more sharply; this is especially prominent for Π in which the lower order solutions are notably less sharp on the right side of the discontinuity.

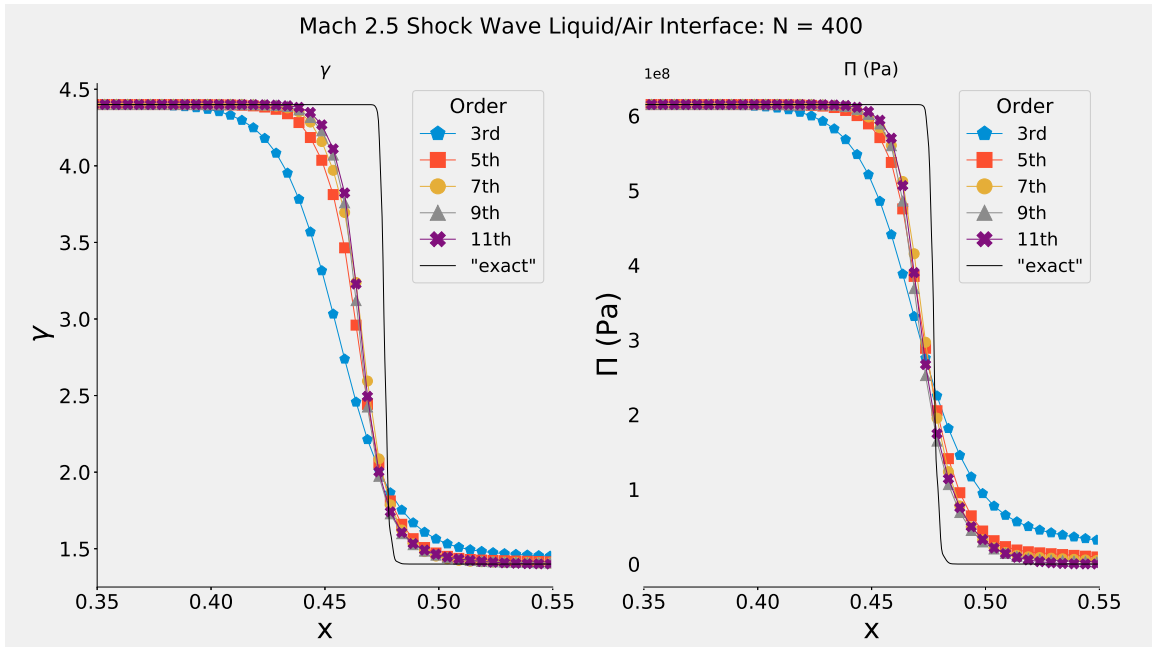


Figure 4.16: Comparison of high-order solutions for specific heat ratio and Π for liquid/air interface problem

These results demonstrate the positivity-preserving qualities of the scheme for flows involving a liquid/vapor interface. This is an important test, as it forms the basis for the 2D shock/bubble interaction problems involving gaseous bubbles immersed in liquid. Using our scheme, we were able to solve this challenging problem to 7th, 9th, and 11th order — which was not possible using the base scheme without limiters.

4.5 2D Two Fluid System

Having verified the positivity-preserving capabilities of our scheme for several single and two-component flows in 1D, we now consider three problems using the two-dimensional extension of the scheme.

4.5.1 Isentropic Vortex

In order to verify the convergence of our two-fluid scheme, we begin by solving for a stationary isentropic vortex where the field variables are defined on the domain $[-10, 10] \times [-10, 10]$ as

$$v_x = -\frac{\epsilon}{2\pi} \exp\left(\frac{1-r^2}{2}\right) y \quad (4.7)$$

$$v_y = \frac{\epsilon}{2\pi} \exp\left(\frac{1-r^2}{2}\right) x \quad (4.8)$$

$$\rho = \left[1 - \frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} \exp(1-r^2)\right]^{\frac{1}{\gamma-1}} \quad (4.9)$$

$$p = \rho^\gamma - \Pi \quad (4.10)$$

where $r^2 = x^2 + y^2$, and ϵ is the vortex strength parameter. The $\epsilon = 5$ with $\Pi = 0$ case is commonly used in the literature [45, 13] to verify performance of high-order schemes. Since taking $\epsilon = 9$ generates very small pressure ($p_{min} = 6 \times 10^{-3}$) and density ($\rho_{min} = 2 \times 10^{-2}$) values, this case has been used to confirm positivity preservation [44, 47, 56]. The fluid parameters are chosen as ($\gamma = 1.4$, $\Pi = 0.1$) to coincide with [44]. Figure 4.17 provides a visualization of the field variables.

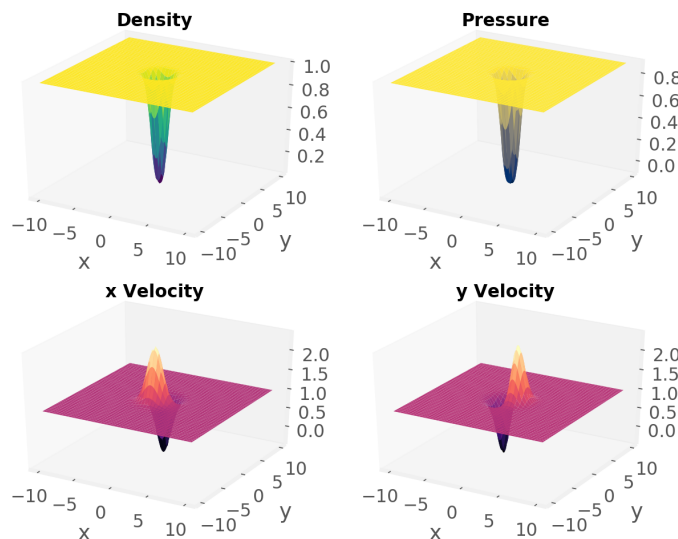


Figure 4.17: Initial density, pressure, and velocity fields for stationary isentropic vortex

The results from the $\epsilon = 5$ case are presented in Table 4.3. Here the L_1 error for density, pressure, and the sum of the conserved variables (density, momentum, and total energy) are shown confirming the high-order convergence of our scheme. It is important to note that in order to obtain the expected convergence rates, we had to calculate all errors with respect to the finest grid resolution. That is, after obtaining the solution on a course grid, we utilized repeated applications of the high-order interpolation scheme outlined in Section 2.2 to obtain the solution on the finest grid. Failure to do this yielded artificially low convergence rates.

Table 4.3: L_1 convergence study for stationary isentropic vortex with $\epsilon = 5$ and $t = 0.1$

| Order | Resolution | Density | | Pressure | | Total | |
|-------|------------------|----------|------|----------|------|----------|------|
| | | Error | Rate | Error | Rate | Error | Rate |
| 5 | 40×40 | 3.84e-05 | | 5.01e-05 | | 3.67e-04 | |
| | 80×80 | 1.49e-06 | 4.69 | 2.16e-06 | 4.54 | 1.34e-05 | 4.78 |
| | 160×160 | 4.75e-08 | 4.97 | 7.32e-08 | 4.88 | 3.40e-07 | 5.07 |
| | 320×320 | 1.37e-09 | 5.12 | 2.13e-09 | 5.11 | 1.05e-08 | 5.26 |
| 7 | 40×40 | 1.75e-05 | | 2.57e-05 | | 1.86e-04 | |
| | 80×80 | 2.92e-07 | 5.90 | 5.04e-07 | 5.67 | 3.00e-06 | 5.95 |
| | 160×160 | 2.95e-09 | 6.63 | 5.27e-09 | 6.58 | 2.58e-08 | 6.86 |
| | 320×320 | 2.25e-11 | 7.04 | 4.02e-11 | 7.04 | 1.66e-10 | 7.28 |
| 9 | 40×40 | 1.01e-05 | | 1.74e-05 | | 1.26e-04 | |
| | 80×80 | 9.32e-08 | 6.76 | 1.66e-07 | 6.71 | 1.05e-06 | 6.92 |
| | 160×160 | 2.92e-10 | 8.32 | 5.45e-10 | 8.26 | 2.73e-09 | 8.58 |
| | 320×320 | 5.93e-13 | 8.94 | 1.12e-12 | 8.93 | 4.44e-12 | 9.26 |
| 11 | 40×40 | 7.00e-06 | | 1.41e-05 | | 9.71e-05 | |
| | 80×80 | 3.71e-08 | 7.56 | 6.83e-08 | 7.69 | 4.53e-07 | 7.75 |
| | 160×160 | 3.75e-11 | 9.95 | 7.36e-11 | 9.86 | 3.90e-10 | 10.2 |
| | 320×320 | 2.05e-14 | 10.8 | 4.30e-14 | 10.7 | 1.69e-13 | 11.2 |

Table 4.4 shows the convergence results for $\epsilon = 9$. In contrast to the $\epsilon = 5$ case, the positivity limiters were required to solve this problem for all orders. Additionally, we observed that the flux-limiting algorithm for pressure outlined in Ref.

[56] failed to maintain positivity beyond a few time steps. On the other hand, our scheme with detailed steps in (3.46) — (3.49) performs flawlessly. In our algorithm, we assign the limiter at each cell boundary using each of the seven $B^{(k_1, k_2, k_3, k_4)}$ values that depend upon that cell boundary flux. In Ref. [56], only three $B^{(k_1, k_2, k_3, k_4)}$ values for each limiter are considered. Using our method, we were able to run the isentropic vortex problem with $\epsilon = 10$ for longer than $t = 14$ without issue; however, using the scheme in [56], the simulation obtained negative pressure values within a few time steps, even with grid refinement and CFL reduction.

Table 4.4: L_1 convergence study for stationary isentropic vortex with $\epsilon = 9$ and $t = 0.1$

| Order | Resolution | Density | | Pressure | | Total | |
|-------|------------|----------|------|----------|------|----------|------|
| | | Error | Rate | Error | Rate | Error | Rate |
| 5 | 40 × 40 | 1.47e-04 | | 2.14e-04 | | 1.54e-03 | |
| | 80 × 80 | 9.35e-06 | 3.98 | 1.33e-05 | 4.01 | 7.99e-05 | 4.27 |
| | 160 × 160 | 3.41e-07 | 4.78 | 5.22e-07 | 4.68 | 2.76e-06 | 4.86 |
| | 320 × 320 | 1.03e-08 | 5.04 | 1.61e-08 | 5.02 | 7.74e-08 | 5.16 |
| 7 | 40 × 40 | 1.05e-04 | | 1.47e-04 | | 1.13e-03 | |
| | 80 × 80 | 2.59e-06 | 5.35 | 4.25e-06 | 5.11 | 2.57e-05 | 5.47 |
| | 160 × 160 | 2.96e-08 | 6.45 | 5.77e-08 | 6.20 | 2.83e-07 | 6.50 |
| | 320 × 320 | 2.46e-10 | 6.91 | 4.82e-10 | 6.90 | 2.06e-09 | 7.11 |
| 9 | 40 × 40 | 8.39e-05 | | 1.15e-04 | | 9.68e-04 | |
| | 80 × 80 | 9.29e-07 | 6.50 | 1.92e-06 | 5.90 | 1.12e-05 | 6.43 |
| | 160 × 160 | 4.45e-09 | 7.71 | 9.26e-09 | 7.70 | 4.49e-08 | 7.97 |
| | 320 × 320 | 1.10e-11 | 8.67 | 2.17e-11 | 8.74 | 8.91e-11 | 8.98 |
| 11 | 40 × 40 | 7.29e-05 | | 9.66e-05 | | 8.77e-04 | |
| | 80 × 80 | 4.44e-07 | 7.36 | 1.02e-06 | 6.56 | 6.16e-06 | 7.15 |
| | 160 × 160 | 1.08e-09 | 8.68 | 1.91e-09 | 9.06 | 9.66e-09 | 9.32 |
| | 320 × 320 | 8.62e-13 | 10.3 | 1.36e-12 | 10.5 | 5.78e-12 | 10.7 |

4.5.2 Air Shock Helium Bubble Interaction

The next test problem involves a Mach 1.22 shock wave impinging upon a cylindrical bubble comprised of 78% helium and 22% air by mass suspended in a volume of air. This particular problem has been experimentally studied [21] and thus provides an excellent case to quantitatively evaluate the scheme's performance for multi-component flows. The initial data are

$$(\rho, v_x, v_y, p, \Gamma, \Pi) = \begin{cases} (1.024 \frac{\text{kg}}{\text{m}^3}, 0, 0, 1.01 \times 10^5 \text{ Pa}, 1.4, 0) & \text{Pre-shock} \\ (1.658 \frac{\text{kg}}{\text{m}^3}, 114.49 \frac{\text{m}}{\text{s}}, 0, 1.59 \times 10^5 \times 10^5 \text{ Pa}, 1.4, 0) & \text{Post-shock} \\ (0.22 \frac{\text{kg}}{\text{m}^3}, 0, 0, 1.01 \times 10^5 \times 10^5 \text{ Pa}, 1.64, 0) & \text{Bubble.} \end{cases} \quad (4.11)$$

with a computational domain of $[-222.5\text{mm}, 222.5\text{mm}] \times [-445\text{mm}, 445\text{mm}]$. The initial shock front was placed at $x = -0.01\text{m}$, and the bubble center was located at $(-50\text{mm}, 0)$ with a radius of 25mm. The problem was solved for all orders with the standard CFL=0.4, and in no cases were the flux limiters required.

Contour plots of the simulation are shown for density and pressure in Figures 4.18 and 4.19 respectively. The first panel visualizes the initial setup with the shock front and bubble location being clearly seen in the density plot. The second panel shows the bubble shortly after the shock first impinges upon the bubble. The density contour reveals slight deformation in the upstream face of the bubble, and the refracted wave can be faintly seen running through the bubble. In the pressure contour, the incident wave front is clearly defined above and below the bubble, the refracted shock propagating through the bubble is seen running ahead within the bubble, and a reflected wave is seen traveling leftward.

Density Solutions for Air/Helium Shock Bubble Interaction Problem:
 $N = 1500$, $M = 300$, Order = 11, CFL=0.4

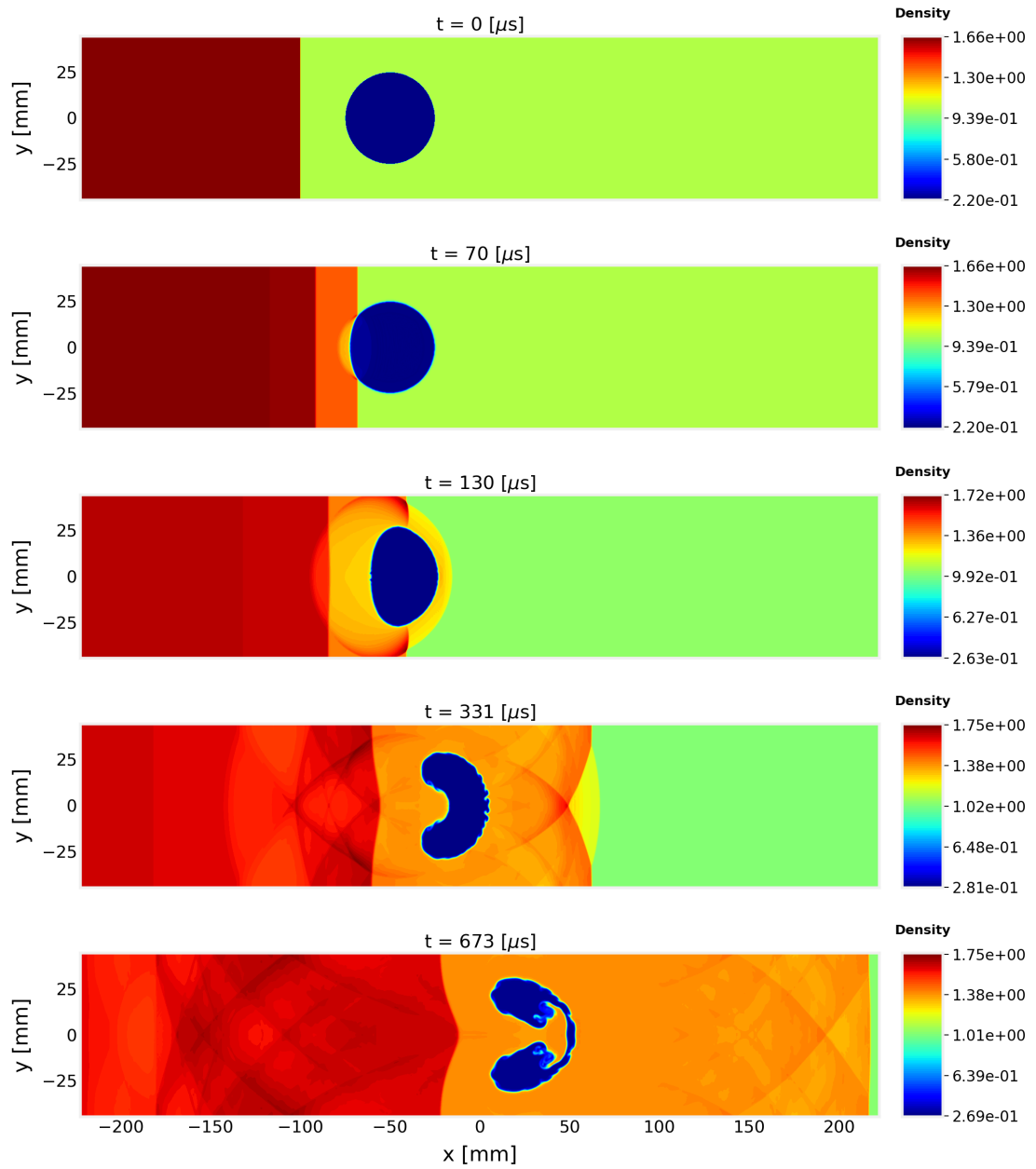


Figure 4.18: Air/helium shock bubble interaction — 11th order density solutions

To assess the accuracy of the scheme, the velocities of several prominent features were calculated and compared to experimental [21] and numerical [45] data. The velocities of the incident, refracted, and transmitted shocks (u_s , u_r , and u_t respectively)

were calculated by tracking the average distance traveled per time step of the shock front along the center line ($y = 0$) over specific time intervals.

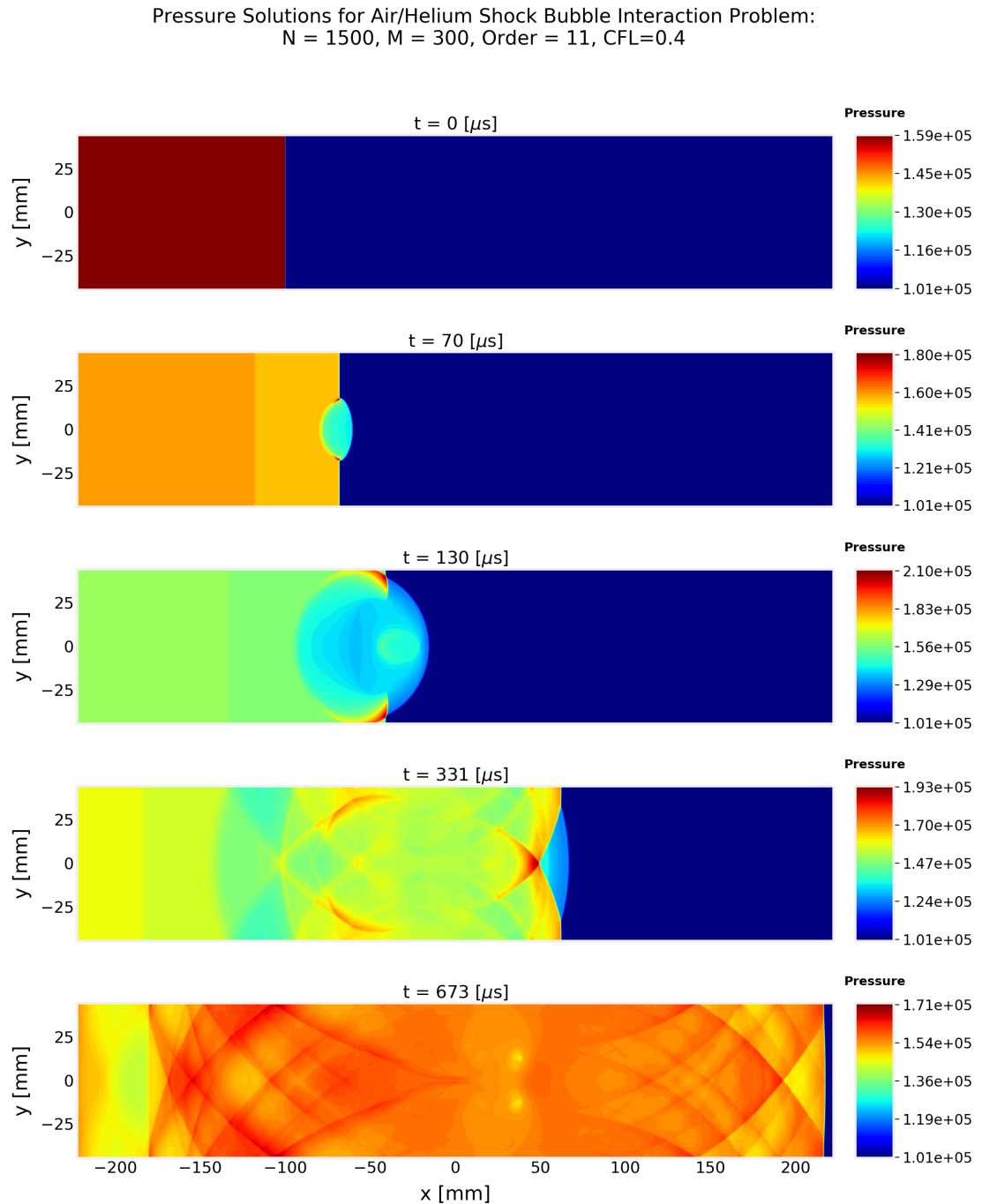


Figure 4.19: Air/helium shock bubble interaction — 11th order pressure solutions

The velocities of the upstream interface, downstream interface, and air jet (u_{ui} ,

u_{di} , and u_j , respectively) were calculated by tracking the left and right interface along the center line. The results are tabulated in Table 4.5.

Table 4.5: Comparisons of shock/interface speeds [m/s] to numerical and experimental data.

| | u_s | u_r | u_t | u_{ui} | u_{di} | u_j |
|---------------------------------------------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
| Time (μ s) | 0-60 | 70-110 | 120-300 | 70-110 | 200-400 | 200-400 |
| 11th order — Present Work ($\Delta x = 742 \mu\text{m}$) | 441 | 958 | 413 | 179 | 147 | 221 |
| 9th order FD simulation ($\Delta x = 278 \mu\text{m}$) [45] | 420 | 960 | 371 | 178 | 150 | 223 |
| 5th order FV simulation ($\Delta x = 50 \mu\text{m}$) [13] | 420 | 945 | 379 | 173 | 145 | 230 |
| Experiment [21] | 410 ± 41 | 900 ± 90 | 393 ± 39 | 170 ± 17 | 145 ± 15 | 230 ± 23 |

To illustrate the various features measured in Table 4.5, Figure 4.20 shows numerical Schlieren plots that provide clear visualization of wave fronts. To develop these images, the process outlined by [39] is used. Specifically, the shading function is defined as

$$\phi = \exp\left(-k \frac{|\nabla\rho|}{|\nabla\rho|_{\max}}\right), \quad (4.12)$$

where $k=75$ for helium and $k=25$ for air. To calculate the gradient terms, the high-order finite difference scheme was used.

In Figure 4.20, the first panel shows the initial setup with the shock front and bubble interface sharply defined. The second panel shows the shock shortly after impinging upon the bubble. The shock front is seen as the thick vertical line near $x = -0.05\text{m}$. Immediately to the right of the incident shock, the refracted shock front is observed as the convex shape running ahead through the bubble cavity at velocity $u_r \approx 959\text{ms}^{-1}$, which is over twice the speed of the incident shock. Directly to the left of the shock front, the reflected shock is seen traveling back into the pressurized air. Also visible is the expansion wave traveling leftward behind the initial shock location. The third panel shows the transmitted shock that has fully traversed the bubble volume and continues to propagate into the air ahead of the incident shock; additionally, the deformation of the upstream bubble interface is

evident. The fourth and fifth panels mark the starting and stopping times used in calculating the down stream interface and jet velocities. Note that the numerous intersecting waves propagating from the top and bottom of the domain are the result of the weak reflected waves re-entering the domain by way of the periodic top and bottom boundaries.

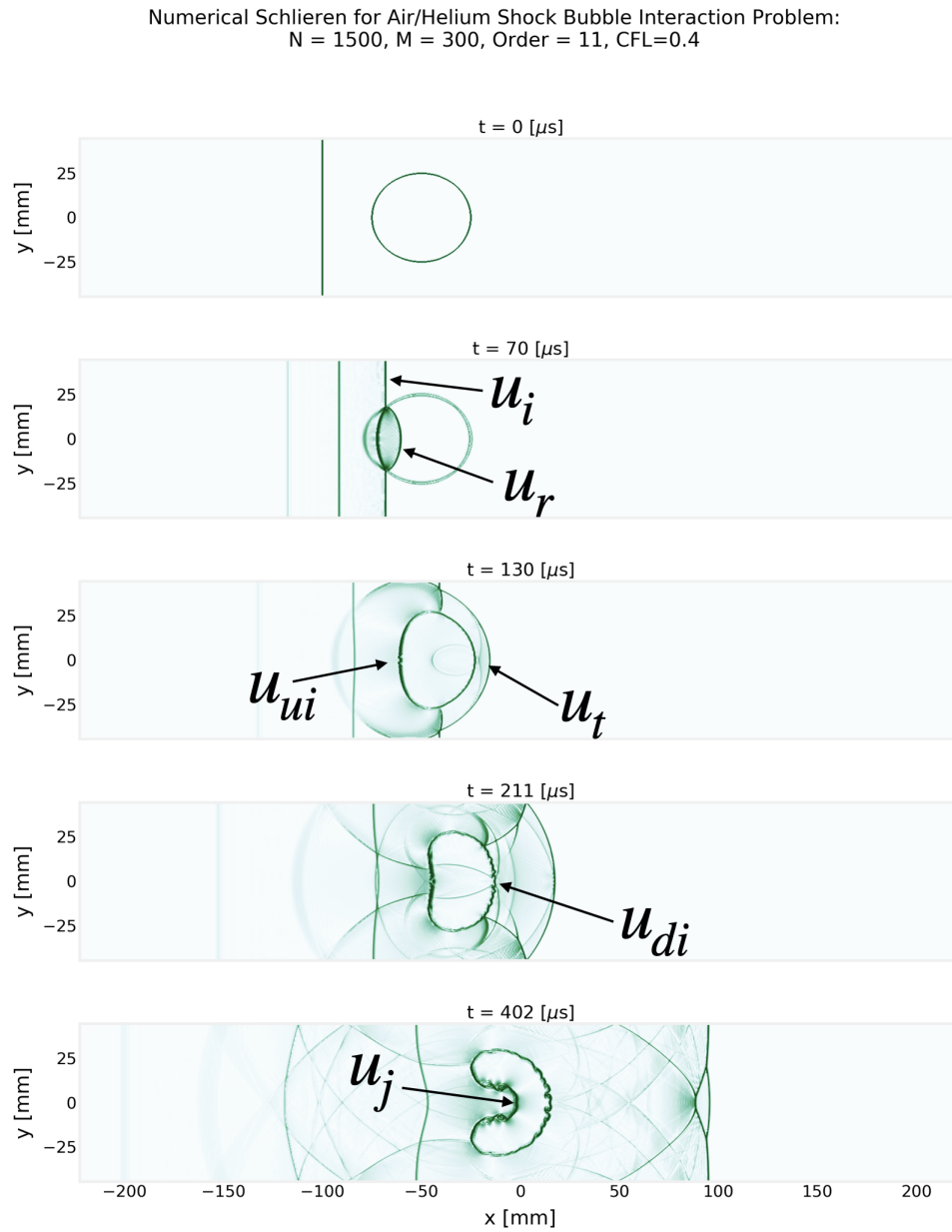


Figure 4.20: Air/helium shock bubble interaction — numerical Schlieren

4.5.3 High Mach Shock Air Bubble Interaction

The final test problem we solve is designed to verify the positivity-preserving nature of the scheme for 2D, two-fluid compressible flow with both liquid and vapor phases. Specifically, this simulation investigates the shock induced collapse of a cylindrical air bubble immersed in liquid water. The initial data represent a Mach 1.9 shock wave propagating into a body of water containing a single air bubble at atmospheric conditions. The initial field variables are defined as

$$(\rho, v_x, v_y, p, \Gamma, \Pi) = \begin{cases} (998 \frac{kg}{m^3}, 0, 0, 1.01 \times 10^5 Pa, 6.68, 4050 \times 10^5 Pa) & \text{Pre} \\ (1231 \frac{kg}{m^3}, 600.3 \frac{m}{s}, 0, 1.9 \times 10^9 Pa, 6.68, 4050 \times 10^5 Pa) & \text{Post} \\ (1.22 \frac{kg}{m^3}, 0, 1.01 \times 10^5 Pa, 1.4, 0) & \text{Air.} \end{cases} \quad (4.13)$$

The computational domain $[-20\text{mm}, 20\text{mm}] \times [-20\text{mm}, 20\text{mm}]$ is defined to be large enough such that the incident/reflected waves never interact with the boundaries. The bubble is centered at $(3\text{mm}, 0)$ and radius is taken as $R = 6\text{mm}$ to coincide with the experimental and computational data of [6, 3, 51]. The incident shock front is initially placed 1mm away from the left bubble interface at $x = -1\text{mm}$.

Figures 4.21, 4.22, and 4.23 show the 7th order density, pressure, and numerical Schlieren fields at several different times to illustrate the flow evolution. Note that all times are reported with $t = 0$ corresponding to the time the shock first encounters the interface. This problem is unique in that it results in the development of negative pressures (see 4th panel in Figure 4.22) thus testing the ability of the scheme to maintain positivity of the so called modified pressure $P + \Pi$. The positivity limiters were required to run this simulation; for all orders, the non-limiting scheme failed regardless of the CFL and grid resolution used. Additionally, we found the very high-order accurate (greater than 7th-order) schemes required interface smoothing of the initial data to avoid oscillations at the interface, which caused positivity issues even

with the limiters. Interestingly, of all the test problems we studied, the very high-order accurate simulations of this problem were the only cases where the positivity of the interface capturing function Γ_1 was actually required.

Density Solutions for Water/Air High Mach Shock Bubble Interaction Problem:
 $N = 450, M = 450, \text{Order} = 7, \text{CFL} = 0.4$

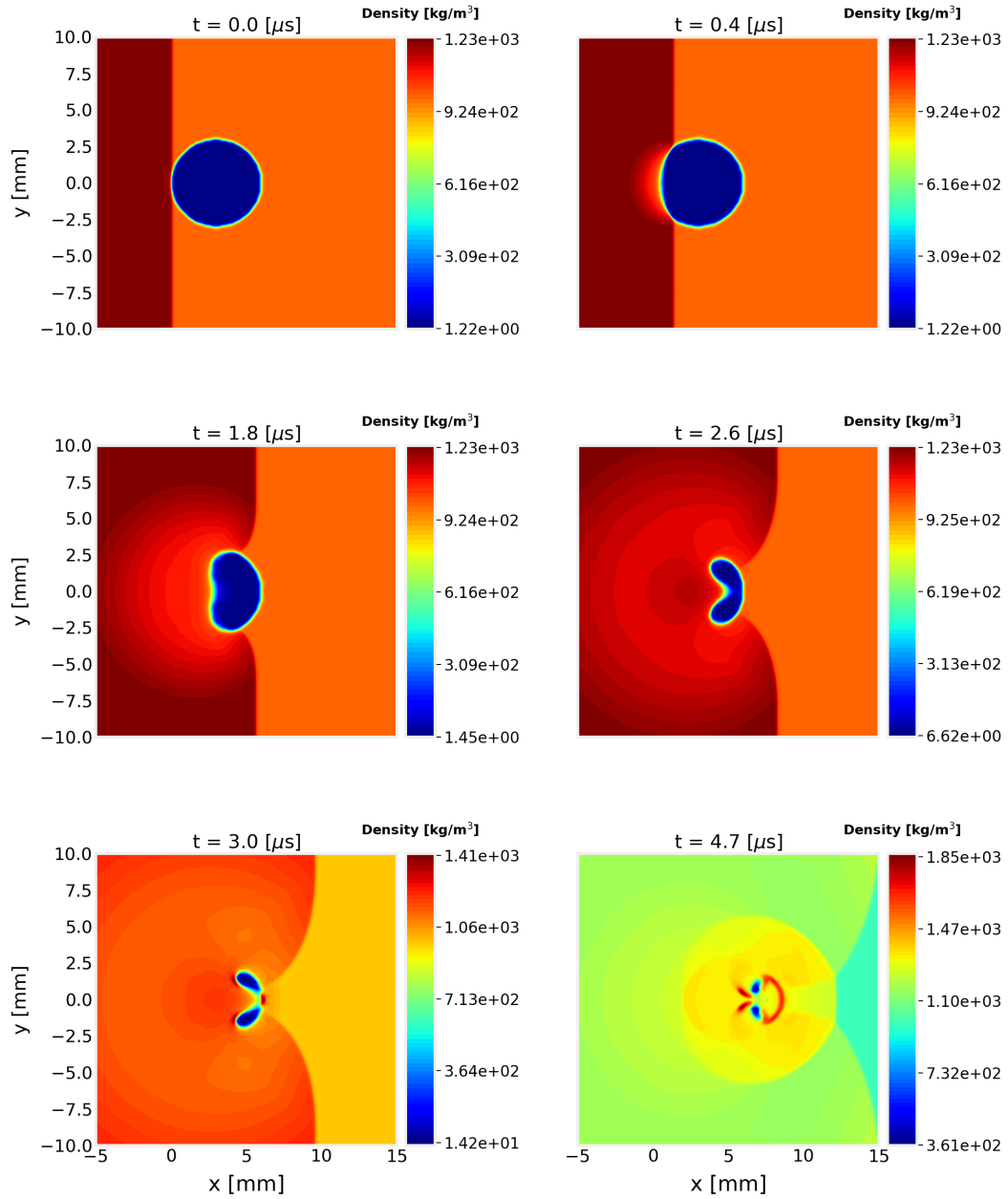


Figure 4.21: Water/air shock bubble interaction — 7th order density solutions

Just as in the previous problem (shock induced collapse of a helium bubble in air), the initial interaction between the shock and bubble interface results in three unique wave features. In the unperturbed regions above and below the bubble, the shock front remains unchanged and continues propagating at the initial shock speed. Within the bubble, a weak shock is transmitted through the air; this can be seen most clearly in the 2nd and 3rd panels of Figure 4.21. Behind the bubble, an expansion wave is reflected off the bubble surface and begins radially propagating upstream; this wave front is clearly observed in the pressure contours (panels 1–5 of Figure 4.22). However, in contrast to the previous problem where the sound speed within the bubble was less than in the surrounding fluid, this problem represents the opposite case. In the 3rd panel of Figure 4.23, we observe that the incident shock leads the transmitted shock. This coupled with the effects of the reflected expansion wave cause the concave “bending” of the shock front.

In the 3rd panel of Figure 4.21, we can begin to see the early formation of a water jet developing along the $y = 0$ line. At this point in the collapse, the jet is manifested as a slight indentation at the center of the bubble’s deformed left face. This jet rapidly cuts through the center of the air bubble in the 4th panel of Figures 4.21 and 4.23 and in the 5th panels, the water jet has completely severed the air bubble in half. A shock wave resulting from this event is faintly seen propagating downstream in panel 5 of the numerical Schlieren contours.

In the literature, this event has been used to characterize the bubble collapse time. Specifically, [6] defines the collapse time, τ , as the duration between the initial shock-interface interaction and the bubble splitting. In their experiments using high-speed Schlieren photography, they found this time be about $3.5\mu s$. The numerical studies of [3, 51] measured this collapse time at $3.1\mu s$ and $3.0\mu s$, respectively, matching well with our result of $3.0\mu s$.

Pressure Solutions for Water/Air High Mach Shock Bubble Interaction Problem:
 $N = 450$, $M = 450$, Order = 7, CFL=0.4

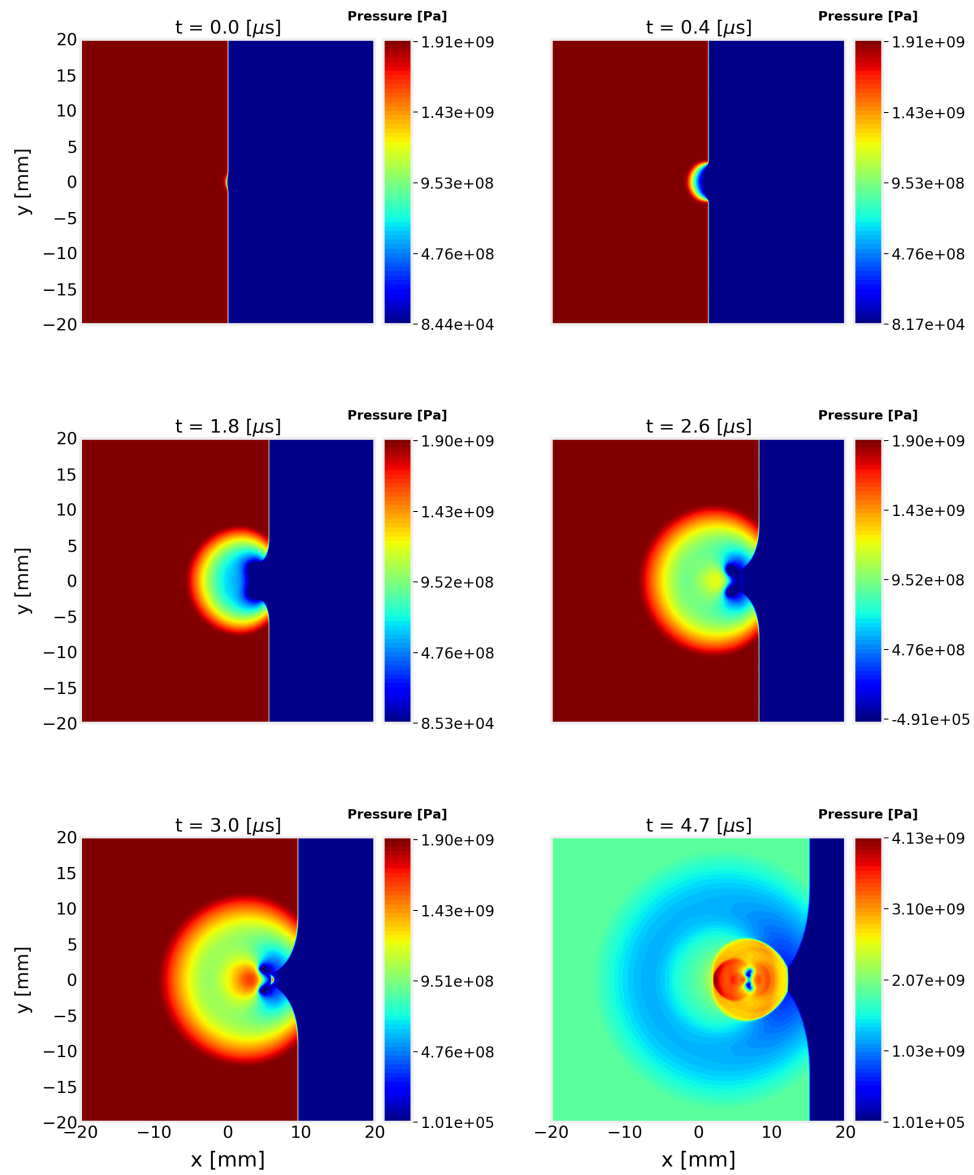


Figure 4.22: Water/air shock bubble interaction — 7th order pressure solutions

Numerical Schlieren for Water/Air High Mach Shock Bubble Interaction Problem:
 $N = 450$, $M = 450$, Order = 7, CFL=0.4

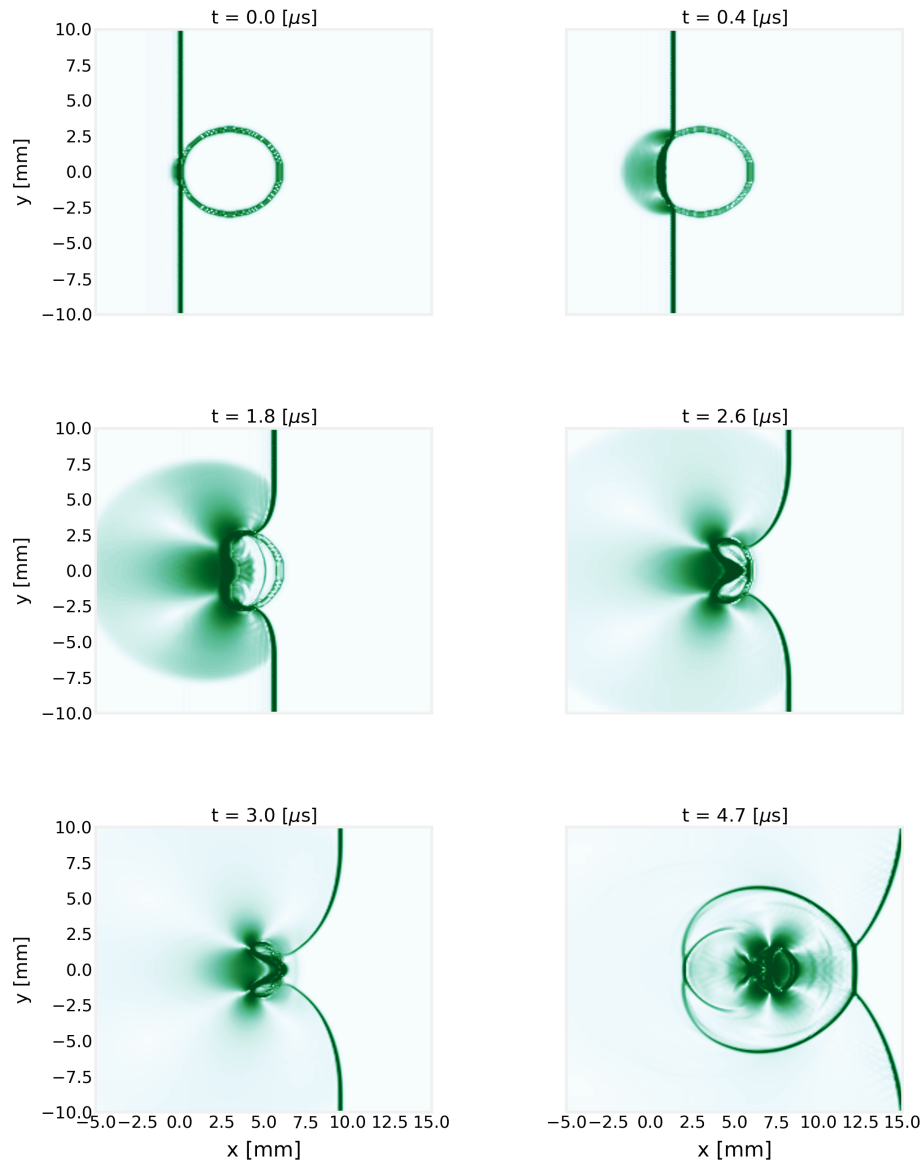


Figure 4.23: Water/air shock bubble interaction — numerical Schlieren

The evolution of the normalized bubble volume is presented in Figure 4.24. At each time, the volume is calculated simply by summing the total number of cells bounded by the bubble. The key feature to highlight in this plot is the linearity of the bubble collapse. Both [3] and [6] observed that for a cylindrical bubble, volume

ratio decreases linearly and that the collapse time, τ , marks the transition to a non-linear regime. This trend is clearly observed in Figure 4.24.

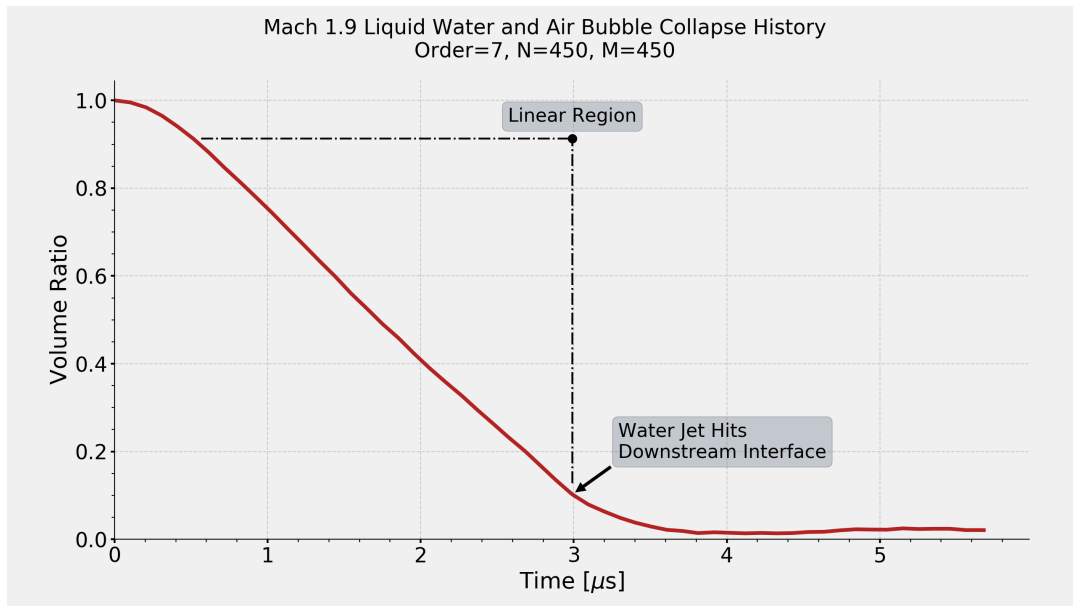


Figure 4.24: Air bubble collapse history for water/air shock bubble problem

4.6 Impact on CPU Time

The impacts of the flux limiting process upon the overall CPU time of the scheme are investigated. Table 4.6 shows the average CPU time per time step for our scheme with and without the positivity algorithm. Results are given for both the helium/air shock bubble problem (Section 4.5.2) and the liquid shock air bubble interaction (Section 4.5.3). For the latter problem, negative density and modified pressure are obtained after the first time step unless the limiters are used; thus, all reported CPU times are based on the first time step. For all cases, the percentage difference was less than 10%, so the overall impact of the positivity-preserving scheme for these problems was minimal. For both orders, the difference between the two methods is larger for the liquid shock air bubble interaction. This was expected as only the liquid/air problem required the positivity limiters, which necessitate the additional step of solving for the pressure limiters. However, it is important to note the overall number of time

steps that demanded this additional computational expense was insignificant.

Table 4.6: Comparison of CPU cost per time step for scheme with limiters and without limiters for $(N, M) = (600, 150)$

| Order | | Helium/Air | | Water/Air | |
|-------|------------------|--------------|--------------|--------------|--------------|
| | | CPU Time [s] | % difference | CPU Time [s] | % difference |
| 5 | With Limiters | 6.04 | | 6.06 | |
| | Without Limiters | 5.52 | 9.35% | 5.53 | 9.71% |
| 7 | With Limiters | 8.19 | | 8.23 | |
| | Without Limiters | 7.67 | 6.91% | 7.66 | 7.37% |

For the 5th order simulation, only the very first time step resulted in the fluxes being limited. This number increased to three during the 7th order simulation. Both simulations ran for at least 800 time steps, so the number of steps that actually demanded positivity preservation represents a very small percentage.

Chapter 5

Conclusions

To summarize and conclude this thesis, we revisit the stated objectives and outline the realization and contribution of each goal within this work.

5.1 Summary

The objectives for this thesis were enumerated in the introduction as follows:

- Implement the finite difference WENO scheme of Shahbazi [45] for the two fluid model of (1.10) – (1.14) in one and two spatial dimensions up to 11th order.
- Apply the flux limiting positivity-preserving framework from [57, 56] to this finite difference scheme in one and two dimensions.
- Verify the performance of the new positivity-preserving scheme.

The subsequent sections describe the realization of each stated objective.

5.1.1 First Objective

The finite difference WENO scheme of Shahbazi [45] was implemented to 11th order for both one and two spatial dimensions using the Python programming language. This code represents the first implementation of the scheme at 11th order accuracy. Additionally, general relations were derived for the schemes' interpolation coefficients, optimal weights, and finite difference weights. A Python module for calculating these parameters to arbitrarily high-order was developed, allowing for relatively simple extension to even higher orders.

5.1.2 Second Objective

A high-order flux-limiting, positivity-preserving technique was developed and applied to the finite difference method. The positivity-preserving algorithm is based upon the methods of [57, 56] who worked exclusively within the framework of a 5th-order flux reconstruction based WENO scheme. Since our finite difference scheme uses the WENO reconstruction of primitive variables, their methods are not directly applicable to our work. The application of their flux-limiting technique to our primitive variable based WENO scheme necessitated the reformulation of our high-order flux terms into a compatible form thus representing a major contribution of this thesis. Additionally, we found the truncated pressure limiter assignment procedure for the 2D case as described in [56] failed for certain problems. The complete set of limiters we present in (3.46)–(3.49) were found to guarantee positivity for all test problems. The scheme developed herein is regarded as the first flux limiting scheme for orders higher than five for single fluid compressible flow computations. For two-fluid compressible systems, the developed solver is the first genuinely high-order positivity-preserving finite difference scheme.

5.1.3 Third Objective

To confirm the high-order accuracy of the scheme, we conducted convergence studies for scalar linear advection, scalar non-linear advection, and two cases of the 2D isentropic vortex problem. In all cases, we found the algorithm successfully maintained positivity (or maximum principle preserving for the scalar cases) without impacting high-order convergence. To assess the effectiveness of the positivity algorithm, we considered several challenging test scenarios with very large pressure and density ratios coupled with large variance in equation of state parameters. For all of these cases, the base high-order scheme without flux limiters crashed, while our new scheme with limiters succeeded. These simulations verify the new scheme’s ability to preserve

hyperbolicity with only a slight increase in computational cost (less than 10%). Furthermore, positivity-preservation of the developed scheme is demonstrated for spatial approximation orders as high as eleven.

5.2 Future Work

The rigor of the positivity limiters coupled with the intrinsic accuracy, simplicity, and efficiency of the base finite difference scheme [45] lead to a high performing robust method that lends itself well to further extension. Immediate next steps would involve developing an efficient parallel implementation in the C programming language to minimize run-time for computationally intensive simulations; this would prove an essential step in applying the scheme to three spatial dimensions.

Other relevant future work includes the application of our scheme to mixture theory models. Specifically, the general seven equation model described by [2] can be reduced to a four equation Homogeneous Relaxation Model (HRM) [16] which assumes constant temperature, pressure, and velocity across constituents. Of particular interest is the development of a positivity-preserving solver for this model. Such a scheme would form the foundation for the extension to more complicated models that include additional physics such as phase change, heat transfer, and viscous effects [38, 31]. Finally, the scheme can also be applied to Runge-Kutta schemes of orders higher than four, which may yield higher efficiency when coupled with very high spatial orders (e.g., 11th-order).

Bibliography

- [1] Roy C Allen IV. *Development of an experiment for investigating the magnetohydrodynamic richtmyer-meshkov instability*. PhD thesis, University of Missouri–Columbia, 2019.
- [2] MR Baer and JW Nunziato. A two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials. *International journal of multiphase flow*, 12(6):861–889, 1986.
- [3] GJ Ball, BP Howell, TG Leighton, and MJ Schofield. Shock-induced collapse of a cylindrical air cavity in water: a free-lagrange simulation. *Shock Waves*, 10(4):265–276, 2000.
- [4] Dinshaw S. Balsara and Chi-Wang Shu. Monotonicity preserving weighted essentially non-oscillatory schemes increasingly high order of accuracy. *Journal of Computational Physics*, 160:405–452, 2000.
- [5] Rafael Borges, Monique Carmona, Bruno Costa, and Wai Sun Don. An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. *Journal of Computational Physics*, 227:3191–3211, 2008.
- [6] NK Bourne and JE Field. Shock-induced collapse of single cavities in liquids. *Journal of Fluid Mechanics*, 244:225–240, 1992.
- [7] R. P. Brent. An algorithm with guaranteed convergence for finding the a zero of a function. *Computer Journal*, 14:422–425, 1971.
- [8] Martin Brouillette. The richtmyer-meshkov instability. *Annual Review of Fluid Mechanics*, 34(1):445–468, 2002.

- [9] Kirsten Chojnicki, AB Clarke, and JC Phillips. A shock-tube investigation of the dynamics of gas-particle mixtures: Implications for explosive volcanic eruptions. *Geophysical research letters*, 33(15), 2006.
- [10] Amanda B Clarke. Unsteady explosive activity: Vulcanian eruptions. *Fagents SA, Tracy KP G, Rosaly MC L (eds) Modeling Volcanic Processes: The Physics and Mathematics of Volcanism. Cambridge University Press, England*, pages 129–152, 2013.
- [11] Amanda Bachtell Clarke, Tomaso Esposti Ongaro, and Alexander Belousov. Vulcanian eruptions. In *The encyclopedia of volcanoes*, pages 505–518. Elsevier, 2015.
- [12] Robin O Cleveland and James A McAteer. The physics of shock wave lithotripsy. *Smith's textbook on endourology*, 1:529–558, 2007.
- [13] Vedran Coralic and Tim Colonius. Finite-volume weno scheme for viscous compressible multicomponent flows. *Journal of computational physics*, 274:95–121, 2014.
- [14] Constantin C. Coussios and Ronald A. Roy. Applications of acoustics and cavitation to noninvasive therapy and drug delivery. *Annual Review of Fluid Mechanics*, 40(1):395–420, 2008.
- [15] Edward T Curran. Scramjet engines: the first forty years. *Journal of Propulsion and Power*, 17(6):1138–1148, 2001.
- [16] P Downar-Zapolski, Z Bilicki, Léon Bolle, and J Franco. The non-equilibrium relaxation model for one-dimensional flashing liquid flow. *International journal of multiphase flow*, 22(3):473–483, 1996.

- [17] Hui Feng, Cong Huang, and Rong Wang. An improved mapped weighted essentially non-oscillatory scheme. *Applied Mathematics and Computation*, 232:453–468, 2014.
- [18] G.A. Gerolymos, D. Sénéchal, and I. Vallet. Very-high-order weno schemes. *Journal of Computational Physics*, 228:8481–8524, 2009.
- [19] David Gottlieb and Chi-Wang Shu. On the gibbs phenomenon and its resolution. *SIAM Review*, 39(4):644–668, 1997.
- [20] Gail ter Haar. Ultrasonic imaging: safety considerations. *Interface Focus*, 1(4):686–697, 2011.
- [21] Jean-Francois Haas and B. Sturtevant. Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities. *Journal of Fluid Mechanics*, 181:41–76, 09 1987.
- [22] Charles R. Harris, K. Jarrod Millman, St’efan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fern’andez del R’io, Mark Wiebe, Pearu Peterson, Pierre G’erard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.
- [23] Andrew K. Henrick, Tariq D. Aslam, and Joseph M. Powers. Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points. *Journal of Computational Physics*, 207:542–567, 2005.
- [24] J. Hesthaven, S. M. Kaber, and L. Lurati. Padé-legendre interpolants for gibbs reconstruction. *Journal of Scientific Computing*, 28:337–359, 2006.

- [25] Xiangyu Y. Hu, Nikolaus A. Adams, and Chi-Wang Shu. Positivity-preserving method for high-order conservative schemes solving compressible euler equations. *Journal of Computational Physics*, 242:169–180, 2013.
- [26] Guang-Shan Jiang and Chi-Wang Shu. Efficient implementation of weighted eno schemes. *Journal of Computational Physics*, 126:202–228, 1996.
- [27] Eric Johnsen. *Numerical simulations of non-spherical bubble collapse with applications to shockwave lithotripsy*. PhD thesis, California Institute of Technology, 2008.
- [28] Eric Johnsen and Tim Colonius. Implementation of weno schemes in compressible multicomponent flow problems. *Journal of Computational Physics*, 219(2):715–732, 2006.
- [29] Eric Johnsen and Tim Colonius. Shock-induced collapse of a gas bubble in shock-wave lithotripsy. *The Journal of the Acoustical Society of America*, 124(4):2011–2020, 2008.
- [30] Smadar Karni. Multicomponent flow calculations by a consistent primitive algorithm. *Journal of Computational Physics*, 112(1):31–43, 1994.
- [31] G La Spina, M de’Michieli Vitturi, and E Romenski. A compressible single-temperature conservative two-phase model with phase transitions. *International Journal for Numerical Methods in Fluids*, 76(5):282–311, 2014.
- [32] Chao Liang and Zhengfu Xu. Parametrized maximum principle preserving flux limiters for high order schemes solving multi-dimensional scalar hyperbolic conservation laws. *Journal of Scientific Computing*, 58:41–60, 2014.

- [33] Martin Forrester Lindsey. *Assessing the Potential for Improved Scramjet Performance Through Application of Electromagnetic Flow Control*. PhD thesis, Air Force Institute of Technology, 2006.
- [34] Xu-Dong Liu, Stanley Osher, and Tony Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200–212, 1994.
- [35] Ksenia Loskutova, Dmitry Grishenkov, and Morteza Ghorbani. Review on acoustic droplet vaporization in ultrasound diagnostics and therapeutics. *BioMed research international*, 2019, 2019.
- [36] EE Meshkov. Instability of the interface of two gases accelerated by a shock wave. *Fluid Dynamics*, 4(5):101–104, 1969.
- [37] Siew-Wan Ohl, Evert Klaseboer, and Boo Cheong Khoo. Bubbles with shock waves and ultrasound: a review. *Interface Focus*, 5(5):20150019, 2015.
- [38] Marica Pelanti and Keh-Ming Shyue. A mixture-energy-consistent six-equation two-phase numerical model for fluids with interfaces, cavitation and evaporation waves. *Journal of Computational Physics*, 259:331–357, 2014.
- [39] James J Quirk and Smadar Karni. On the dynamics of a shock–bubble interaction. *Journal of Fluid Mechanics*, 318:129–163, 1996.
- [40] Natalya Rapoport, Zhonggao Gao, and Anne Kennedy. Multifunctional nanoparticles for combining ultrasonic tumor imaging and targeted chemotherapy. *Journal of the National Cancer Institute*, 99(14):1095–1106, 2007.
- [41] Robert D Richtmyer. Taylor instability in shock acceleration of compressible fluids. Technical report, Los Alamos Scientific Lab., N. Mex., 1954.
- [42] Richard Saurel and Rémi Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computation*, 21(3):1115–1145, 1999.

- [43] C. Segal. *The Scramjet Engine: Processes and Characteristics*. Cambridge Aerospace Series. Cambridge University Press, 2009.
- [44] Khosro Shahbazi. Robust second-order scheme for multi-phase flow computations. *Journal of Computational Physics*, 339:163–178, 2017.
- [45] Khosro Shahbazi. High-order finite difference scheme for compressible multi-component flow computations. *Computers and Fluids*, 190:425–439, 2019.
- [46] Khosro Shahbazi. *Computational Transport Phenomena*. Unpublished, 2020.
- [47] Khosro Shahbazi. Positivity-preserving of a first-order scheme for a quasi-conservative compressible two-fluid model. *Accepted, SIAM Journal on Scientific Computing*, May, 2021.
- [48] Oleksandr Shpak, Martin Verweij, Hendrik J. Vos, Nico de Jong, Detlef Lohse, and Michel Versluis. Acoustic droplet vaporization is initiated by superharmonic focusing. *Proceedings of the National Academy of Sciences*, 111(5):1697–1702, 2014.
- [49] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77:439–479, 1988.
- [50] U. S. Vevek, B. Zang, and T.H. New. Adaptive mapping for high order weno methods. *Journal of Computational Physics*, 381:162–188, 2019.
- [51] Chunwu Wang and Chi-Wang Shu. An interface treating technique for compressible multi-medium flow with runge-kutta discontinuous galerkin method. *Journal of Computational Physics*, 229(23):8823–8843, 2010.

- [52] V. Wheatley, R. Samtaney, D. I. Pullin, and R. M. Gehre. The transverse field richtmyer-meshkov instability in magnetohydrodynamics. *Physics of Fluids*, 26(1):016102, 2014.
- [53] Zheng Zheng Wong, Oliver D Kripfgans, Adnan Qamar, J Brian Fowlkes, and Joseph L Bull. Bubble evolution in acoustic droplet vaporization at physiological temperature via ultra-high speed imaging. *Soft Matter*, 7(8):4009–4016, 2011.
- [54] Paul Woodward and Phillip Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54(1):115–173, 1984.
- [55] Xufeng Xi and Pei Zhong. Dynamic photoelastic study of the transient stress field in solids during shock wave lithotripsy. *The Journal of the Acoustical Society of America*, 109(3):1226–1239, 2001.
- [56] Tao Xiong, Jing-Mei Qiu, and Zhengfu Xu. Parametrized positivity preserving flux limiters for high order finite difference weno scheme solving compressible euler equations. *J Sci Comput*, 67:1066–1088, 2016.
- [57] Zhengfu Xu. Parametrized maximum principle preserving flux limiters for high order schemes solving hyperbolic conservation laws: One-dimensional scalar problem. *Mathematics of Computation*, 83:22213–2238, 2014.
- [58] Joseph Yang, Toshi Kubota, and Edward E. Zukoski. Applications of shock-induced mixing to supersonic combustion. *AIAA Journal*, 31(5):854–862, 1993.
- [59] Qingchun Yang, Juntao Chang, and Wen Bao. Richtmyer-meshkov instability induced mixing enhancement in the scramjet combustor with a central strut. *Advances in Mechanical Engineering*, 6:614189, 2014.

- [60] L. Del Zanna, O. Zanotti, N. Bucciantini, and P. Londrillo. Echo: a eulerian conservative high-order scheme for general relativistic megnetohydrodynamics and magetodynamics. *Astronomy and Physics*, 473(1):11–30, 2007.
- [61] Xiangxion Zhang and Chi-Wang Shu. Positivity-preserving high order finite difference weno schemes for compressible euler equations. *Journal of Computational Physics*, 231:2245–2258, 2012.
- [62] Xiangxiong Zhang and Chi-Wang Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *Journal of Computational Physics*, 229(9):3091–3120, 2010.
- [63] Xiangxiong Zhang and Chi-Wang Shu. On positivity-preserving high order discontinuous galerkin schemes for compressible euler equations on rectangular meshes. *Journal of Computational Physics*, 229(23):8918–8934, 2010.

Vita

Daniel is from Cottage Grove, Minnesota. He was home educated his entire life from primary school through high school. As a junior, he began attending Inver Hills Community College full time through the PSEO program ultimately graduating from high school in 2016 having completed over 60 college credits. Daniel began attending South Dakota Mines in fall of 2016 and graduated in spring 2019 with a B.S. in Mechanical Engineering.

Following three summer internships, Daniel began full time employment in spring 2020 with a research group associated with Intellectual Ventures Laboratory of Bellevue, WA. The group focuses on the development of various global health solutions for usage in underdeveloped regions. He has remained with the group (now part of Global Health Labs) to date.