Computational Fluid Dynamics Internship Report

Department of Aerospace Engineering

Indian Institute of Science, Bangalore

Supervisor: Prof. S.V. Raghurama Rao, Dr. Aravind C.B., Dr. Kedar Wagh

Internship Period: December 20, 2024 – February 7, 2025

Date: February 10, 2025
Location: Bangalore, India

Submitted by:

Aayusha Singh and Amrit Roy

Abstract

This report provides an overview of our internship experience in Computational Fluid Dynamics (CFD), focusing on the application of numerical methods to solve Partial Differential Equations (PDEs). We explored the finite-volume method (FVM) and time-stepping techniques such as Forward Euler and Backward Euler. The internship involved the implementation of test cases including linear advection, Burger's equation, and shallow water equations, with a comparison of numerical solutions against exact results. We also delved into advanced topics like the Riemann problem, shock wave solutions, and Newton-Raphson methods. The results were visualized using the VTK files in Paraview software. The findings highlight the effectiveness and challenges of various numerical methods in solving hyperbolic PDEs, providing valuable insights into the computational aspects of fluid dynamics.

Contents

Chapter 1

Introduction

Computational Fluid Dynamics (CFD) has become an indispensable tool in aerospace engineering, enabling the numerical simulation and analysis of complex fluid flow phenomena. CFD techniques allow researchers and engineers to study aerodynamics, shock waves, turbulence modeling, and multiphase flows, playing a crucial role in the design and optimization of aircraft, spacecraft, and industrial fluid systems.

This report documents the work undertaken during a Computational Fluid Dynamics internship at the Department of Aerospace Engineering, Indian Institute of Science, Bangalore. The primary focus of the internship was solving Partial Differential Equations (PDEs) governing fluid motion, with a specific emphasis on hyperbolic conservation laws and shock wave solutions. The work involved developing and analyzing nonlinear hyperbolic conservation laws, aiming to enhance the understanding of fluid system behaviors and their complex dynamics. The objective was to deepen the understanding of numerical methods and their implementation for solving fundamental fluid dynamics equations, including the Linear Advection Equation, Burger's Equation, and the Shallow Water Equations.

Several numerical techniques were explored, including the Finite Volume Method (FVM), Godunov's scheme, Roe's method, and the Local Lax-Friedrichs (LLF) scheme. Time integration methods such as Forward Euler (explicit) and Backward Euler (implicit) were implemented to assess their stability and accuracy. The numerical solutions were visualized using VTK files in ParaView, allowing for a detailed examination of fluid behavior under different initial and boundary conditions.

Beyond theoretical studies, the internship involved hands-on computational experiments with benchmark test cases, including one-dimensional expansion problems, dam break scenarios, and two-dimensional periodic flow simulations. Special attention was given to key CFD concepts such as the Riemann problem, entropy conditions, and numerical errors, including truncation and round-off errors.

This report is structured as follows:

- **Chapter 2: Theoretical Background** – Classification of PDEs, range of influence, and fundamental concepts such as the method of characteristics, shock wave solutions, and entropy conditions.
- **Chapter 3: Numerical Methods and Implementation** – Finite volume method, numerical flux computation, and time-stepping techniques.
- **Chapter 4: Test Cases and Results** – Implementation of benchmark problems and analysis of numerical solutions.
- **Chapter 5: Visualization and VTK Files** – Post-processing and visualization of simulation results in ParaView.
- **Chapter 6: Conclusion and Future Work** – Summary of key findings and potential improvements in numerical methods.

This internship provided valuable insights into the practical implementation of CFD techniques, bridging the gap between theoretical concepts and computational applications. The experience gained from this work contributes to a deeper understanding of numerical methods used in fluid dynamics research and engineering applications.

Chapter 2

Theoretical Background

Chapter Computational Fluid Dynamics (CFD) is based on the mathematical formulation of fluid motion through Partial Differential Equations (PDEs). Understanding the classification and behavior of these equations is crucial in selecting the appropriate numerical methods. This chapter provides an overview of the theoretical foundations underlying CFD, focusing on PDE classification, the method of characteristics, shock wave solutions, and entropy conditions.

2.1 Linearity and Non-Linearity of Equations

A partial differential equation (PDE) is classified as **linear** if it can be expressed in the form:

$$\sum_{i,j} a_{ij}(x,t)\frac{\partial^2 u}{\partial x_i \partial x_j} + \sum_i b_i(x,t)\frac{\partial u}{\partial x_i} + c(x,t)u = f(x,t), \tag{2.1}$$

where u is the dependent variable, and a_{ij}, b_i, c, f are given functions that do not depend on u or its derivatives in a nonlinear way.
A PDE is classified as **non-linear** if it contains non-linear terms involving u or its derivatives, such as products or other non-linear functions of u and its derivatives. Examples of non-linear PDEs include:

- **Burgers' equation**:

$$u_t + uu_x = \nu u_{xx} \tag{2.2}$$

- **Navier-Stokes equations** (describing fluid motion):

$$\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u}\cdot\nabla\mathbf{u}\right) = -\nabla p + \mu\nabla^2\mathbf{u} \tag{2.3}$$

Nonlinear PDEs often describe complex phenomena such as turbulence, shock waves, solitons, and reaction-diffusion systems.

2.2 Classification of PDEs

Partial Differential Equations (PDEs) governing fluid motion are typically classified based on the nature of their *characteristics.* The three primary classifications are the following:

2.2.1 Parabolic Equations

Parabolic PDEs describe **diffusion-dominated** processes, where the solution evolves smoothly over time. A classic example is the **heat equation**:

$$\frac{\partial u}{\partial t} = \alpha\frac{\partial^2 u}{\partial x^2}, \tag{2.4}$$

where α is the diffusion coefficient. These equations are often associated with irreversible processes and require initial conditions for unique solutions.

2.2.2 Hyperbolic Equations

Hyperbolic PDEs model **wave propagation** and are fundamental in fluid dynamics. A standard example is the **linear advection equation**:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \tag{2.5}$$

where c is the wave speed. These equations describe transport phenomena, acoustic waves, and shocks. They often lead to discontinuities, which require specialized numerical methods, such as finite-volume schemes.

2.2.3 Elliptic Equations

Elliptic PDEs describe **steady-state** solutions where no time dependence exists. A classic example is the **Laplace equation**:

$$\nabla^2 u = 0. \tag{2.6}$$

Elliptic equations commonly arise in potential flow, electrostatics, and steady-state heat conduction. Unlike parabolic and hyperbolic PDEs, they require boundary conditions to determine unique solutions.

2.3 Range of Influence and Domain of Dependence

The concept of *characteristics* plays a crucial role in understanding how information propagates in a Partial Differential Equation (PDE). These characteristics help define two fundamental concepts:

- **Domain of Dependence**: For a hyperbolic PDE, the domain of dependence of a point (x, t) is the region in the initial data from which the information influences the solution at (x, t). Determines which initial conditions affect the solution at a given point.
- **Range of Influence**: The range of influence of a point (x, t) is the region in space-time where disturbances originating from (x, t) can propagate and affect the solution.

In hyperbolic PDEs, these concepts are determined by the characteristic curves along which the information travels. Characteristics play a key role in wave propagation, shock formation, and causality in physical systems.

2.4 Method of Characteristics

The *method of characteristics* is a powerful technique for solving first-order hyperbolic Partial Differential Equations (PDEs). It transforms the PDE into a set of ordinary differential equations (ODE) along characteristic curves that describe how information propagates.
For the **linear advection equation**:

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0, \tag{2.7}$$

the **characteristic equations** are obtained by solving:

$$\frac{dx}{dt} = c. \tag{2.8}$$

Solving this equation gives the characteristic curves:

$$x - ct = \text{constant}. \tag{2.9}$$

Along these characteristic lines, the solution u remains unchanged:

$$\frac{du}{dt} = 0 \Rightarrow u = \text{constant along characteristics.} \tag{2.10}$$

This method is particularly useful for analyzing wave propagation and transport phenomena, ensuring that solutions correctly account for causality and information flow.

2.5 Shock Wave Solutions

In **non-linear hyperbolic systems**, discontinuities or *shock waves* can develop even when the initial conditions are smooth. These shocks arise due to the non-linear nature of the system, where wave characteristics intersect, leading to a breakdown of classical solutions.
To determine the speed S at which a shock propagates, we use the **Rankine-Hugoniot condition**, given by:

$$S = \frac{f(u_R) - f(u_L)}{u_R - u_L}. \tag{2.11}$$

Here,

- u_L and u_R are the values of u on the left and right sides of the discontinuity.
- $f(u)$ is the flux function associated with the conservation law.

The Rankine-Hugoniot condition ensures that the shock wave satisfies the conservation law in an integral sense, even though the solution is discontinuous. Unlike linear wave equations, where disturbances propagate smoothly, nonlinear hyperbolic equations often form shocks, requiring specialized numerical techniques for accurate solutions.

2.6 Entropy Condition

Shock waves must satisfy an **entropy condition** to ensure physically admissible solutions. In nonlinear hyperbolic systems, multiple weak solutions may exist, but only those satisfying the entropy condition correspond to physically meaningful shocks.
A common entropy criterion states that the characteristic speeds should decrease across a shock, ensuring that the information does not propagate backward. Mathematically, for a conservation law of the form:

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} = 0, \tag{2.12}$$

a shock satisfying the **Lax entropy condition** must obey:

$$\lambda_L \geq S \geq \lambda_R, \tag{2.13}$$

where:

- S is the shock speed given by the Rankine-Hugoniot condition.
- λ_L and λ_R are the characteristic speeds on the left and right of the shock.

This ensures that characteristics do not emerge from the shock, preventing non-physical solutions such as expansion shocks. Entropy conditions help distinguish shocks from nonphysical discontinuities and play a crucial role in numerical methods for solving hyperbolic PDEs.

2.7 Numerical Considerations

To solve PDEs numerically, discretization techniques such as the Finite Volume Method (FVM) and time-stepping schemes are employed. Important aspects include:

2.8 Numerical Considerations

Solving Partial Differential Equations (PDEs) numerically requires discretization techniques that transform continuous equations into algebraic forms. Two widely used approaches include the **Finite Volume Method (FVM)** and time-stepping schemes. Key considerations for numerical solutions include:

- **Flux Computation:** Numerical flux methods, such as *Godunov's scheme*, *Lax-Friedrichs*, and *Roe's solver*, help capture shock waves and discontinuities while ensuring numerical stability and accuracy.
- **Time Integration:** Stability constraints often determine the choice of time integration schemes:
  - *Explicit methods* (e.g., Forward Euler, Runge-Kutta) are conditionally stable and require small time steps.
  - *Implicit methods* (e.g., Backward Euler, Crank-Nicholson) allow larger time steps but involve solving nonlinear systems.
- **Error Analysis:** Numerical solutions are influenced by:
  - *Truncation errors*, which arise from approximations in discretization.
  - *Round-off errors*, due to finite precision in computations.
  - *Convergence and consistency*, ensuring that numerical solutions approach the exact solution as grid resolution improves.

Choosing an appropriate numerical method depends on the type of PDE, stability constraints (e.g., CFL condition for hyperbolic PDEs), and computational efficiency.

2.9 Summary

This chapter provided an overview of the classification of PDEs, focusing on parabolic, hyperbolic, and elliptic equations. The **method of characteristics** was introduced as a powerful tool for solving the hyperbolic PDEs, highlighting its role in tracking information propagation.
Additionally, we explored fundamental aspects of **hyperbolic conservation laws**, including the formation of shock waves and the importance of the **Rankine-Hugoniot condition** and **entropy criteria** in selecting physically meaningful solutions. Finally, key numerical considerations, such as flux computation, time integration schemes, and error analysis, were discussed as essential components for solving PDEs computationally.
These theoretical foundations provide the basis for implementing robust and accurate numerical schemes, which will be explored in the following chapters.

Chapter 3

Numerical Methods and Implementation

This chapter presents the numerical schemes used to solve hyperbolic conservation laws, particularly in the context of shallow water equations and Euler equations. The focus is on the Local Lax-Friedrichs (LLF) scheme, Roe's approximate Riemann solver, and Godunov's method. These schemes are applied to various test cases to analyze their accuracy, stability, and ability to capture discontinuities.

Numerical methods play a crucial role in solving hyperbolic PDEs, where exact solutions are often unavailable. The choice of a numerical scheme affects the resolution of shock waves, contact discontinuities, and rarefaction waves. To improve solution accuracy, second-order reconstruction was performed in both space and time, reducing numerical diffusion while preserving sharp discontinuities.

To evaluate the performance of these methods, different benchmark test cases, including 1D and 2D Riemann problems, dam-break problems, and periodic flows, are considered. The numerical schemes are implemented with various time integration techniques, ensuring stability under CFL constraints. Additionally, boundary conditions and data visualization using VTK are discussed. These numerical experiments provide insight into the advantages and limitations of each scheme in handling wave propagation, nonlinear effects, and high-resolution shock capturing.

3.1 Overview of Numerical Schemes

Solving hyperbolic conservation laws numerically requires robust and stable schemes capable of capturing shock waves, rarefaction waves, and contact discontinuities. This section presents three widely used numerical schemes: the Local Lax-Friedrichs (LLF) scheme, Roe's approximate Riemann solver, and Godunov's method. Each of these methods is applied to both the Euler equations and the shallow water equations (SWE) with first- and second-order accuracy in space and time.

3.1.1 Local Lax-Friedrichs (LLF) Scheme

The LLF scheme, is a widely used Riemann solver based on a diffusive flux formulation. It is formulated as:

$$F_{i+1/2} = \frac{1}{2}\left[F(U_i) + F(U_{i+1})\right] - \frac{\alpha}{2}(U_{i+1} - U_i), \tag{3.1}$$

where α is the maximum wave speed in the domain, computed as:

$$\alpha = \max(|\lambda_i|, |\lambda_{i+1}|) \tag{3.2}$$

with λ representing the characteristic wave speeds.

This method is simple, robust, and easy to implement but introduces numerical diffusion, which can smear sharp features like shock waves. To improve accuracy, second-order spatial reconstruction is performed using a high-order interpolation technique such as MUSCL (Monotonic

Upstream-centered Scheme for Conservation Laws), coupled with a Strong Stability Preserving Runge-Kutta (SSPRK2) method for time integration.

3.1.2 Roe's Approximate Riemann Solver

Roe's scheme is an approximate Riemann solver that provides better resolution of discontinuities compared to LLF. The numerical flux is computed as:

$$F_{i+1/2} = \frac{1}{2}\left[F(U_i) + F(U_{i+1})\right] - \frac{1}{2}\sum_p |\lambda_p| \tilde{r}_p \tilde{\alpha}_p, \tag{3.3}$$

where:

- λ_p are the eigenvalues of the Jacobian matrix $\frac{\partial F}{\partial U}$,
- $\tilde{r}_p$ are the right eigenvectors, and
- $\tilde{\alpha}_p$ are the wave strengths.

Roe's solver improves shock resolution compared to LLF but may generate non-physical expansion shocks. To address this, an entropy fix is applied, ensuring physically admissible solutions. Like LLF, Roe's method is extended to second-order accuracy through a higher-order reconstruction scheme.

3.1.3 Godunov's Method

Godunov's method is a first-order upwind scheme that solves the exact Riemann problem at each cell interface. It provides sharper shock capturing capabilities compared to approximate solvers like LLF and Roe's method. The update formula is:

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}\left(F_{i+1/2} - F_{i-1/2}\right), \tag{3.4}$$

where $F_{i+1/2}$ is the numerical flux obtained from the exact Riemann solver.

While Godunov's method accurately captures shock waves and contact discontinuities, it is computationally expensive due to the need to solve a Riemann problem at each cell interface. For practical applications, approximate Riemann solvers such as Roe's method are often preferred.

3.1.4 Second-Order Reconstruction in Space and Time

To enhance the accuracy of these schemes, second-order reconstruction is employed:

- Spatial Accuracy: Higher-order interpolation methods such as MUSCL reconstruction are used to reduce numerical diffusion.
- Time Accuracy: Time integration is performed using the SSPRK2 scheme, which enhances stability while maintaining accuracy.

This ensures that numerical solutions remain accurate while minimizing artificial diffusion, making them well-suited for problems involving shallow water equations (SWE) and Euler equations.

Chapter 4

Test Cases and Results

This chapter presents the implementation of various test cases using different numerical schemes. These test cases are designed to evaluate the accuracy, stability, and shock-capturing ability of the Local Lax-Friedrichs (LLF) method, Roe's method, and Godunov's method in both first- and second-order formulations.

4.1 1D Test Cases from Culbert B. Laney's Computational Gasdynamics

The following test cases are taken from Computational Gasdynamics by Culbert B. Laney (1998). The numerical solutions are obtained using first- and second-order LLF, Roe, and Godunov schemes.

4.1.1 Test Case 1: Linear Advection Equation with LLF

The governing equation is the 1D linear advection equation:

$$\frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0. \tag{4.1}$$

The initial condition is given by:

$$u(x, 0) = -\sin(\pi x), \tag{4.2}$$

with periodic boundary conditions. The test is performed using the LLF scheme with first- and second-order accuracy in space and time. The image of the solution for Linear Advection Equation using the LLF method is given in Figure 4.1.

4.1.2 Test Case 4: Burgers' Equation with LLF

The governing equation is the 1D inviscid Burgers' equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0. \tag{4.3}$$

The initial condition is given by:

$$u(x, 0) = \begin{cases} 1, & \text{for } |x| < \frac{1}{3}, \\ 0, & \text{for } \frac{1}{3} < |x| \leq 1. \end{cases} \tag{4.4}$$

with periodic boundary conditions. The test is performed using the LLF scheme with first- and second-order accuracy in space and time.

The image of the solution for Burgers' equation using the LLF method is given in Figure 4.2.

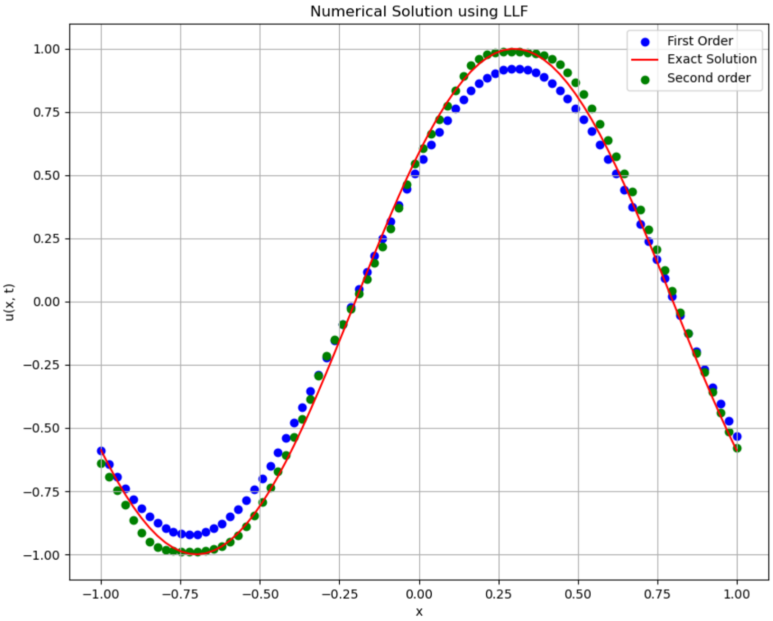


Figure 4.1: Solution of the linear advection equation using the LLF scheme.

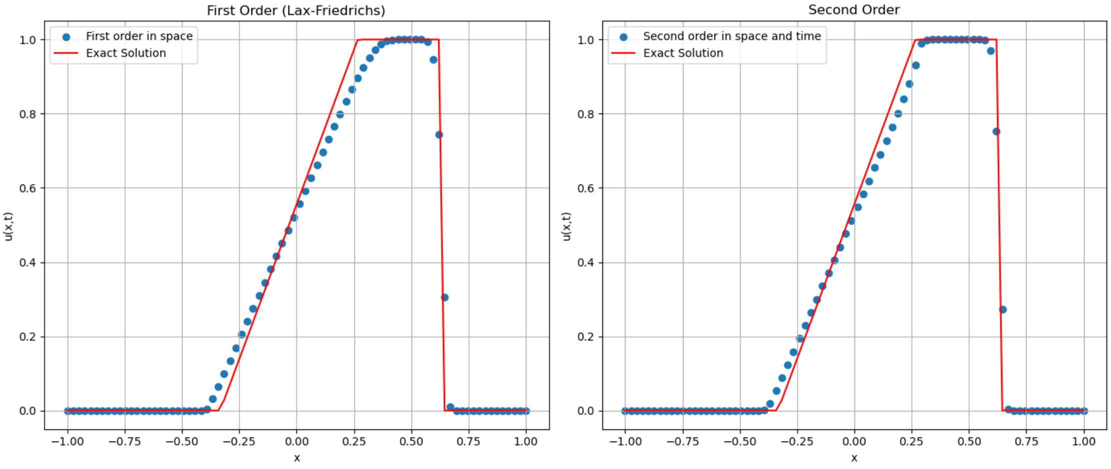


Figure 4.2: Solution of Burgers' equation using the LLF scheme.

4.1.3 Test Case 5: Burgers' Equation with LLF (Modified Initial Condition)

The governing equation is the 1D inviscid Burgers' equation:

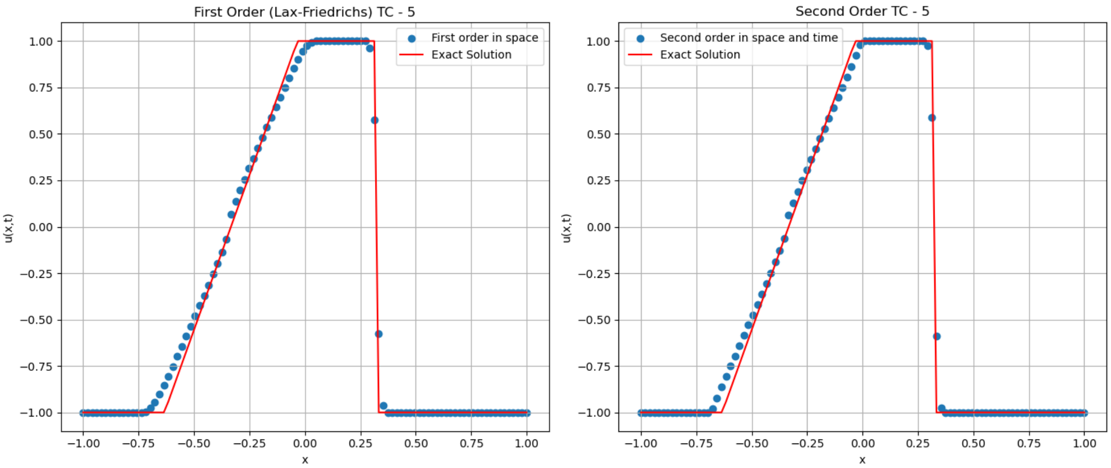
$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0. \tag{4.5}$$

The initial condition is given by:

$$u(x,0) = \begin{cases} 1, & \text{for } |x| < \frac{1}{3}, \\ -1, & \text{for } \frac{1}{3} < |x| \le 1. \end{cases} \tag{4.6}$$

with periodic boundary conditions. The test is performed using the LLF scheme with first- and second-order accuracy in space and time.

The image of the solution for Burgers' equation using the LLF method is given in Figure 4.3.



First Order (Lax-Friedrichs) TC - 5

Second Order TC - 5

Figure 4.3: Solution of Burgers' equation using the LLF scheme with modified initial conditions.

4.1.4 Test Case 4: Burgers' Equation with Godunov's Method

The governing equation is the 1D inviscid Burgers' equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0. \tag{4.7}$$

The initial condition is given by:

$$u(x,0) = \begin{cases} 1, & \text{for } |x| < \frac{1}{3}, \\ 0, & \text{for } \frac{1}{3} < |x| \le 1. \end{cases} \tag{4.8}$$

with periodic boundary conditions. The test is performed using Godunov's method with first- and second-order accuracy in space and time.

The image of the solution for Burgers' equation using Godunov's method is given in Figure 4.4.

4.1.5 Test Case 5: Burgers' Equation with Godunov's Method (Modified Initial Condition)

The governing equation is the 1D inviscid Burgers' equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0. \tag{4.9}$$

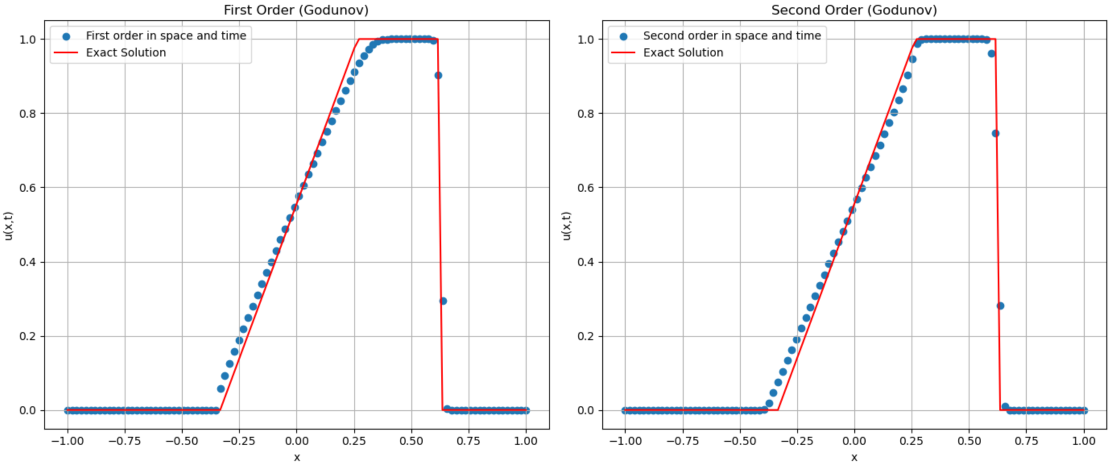


Figure 4.4: Solution of Burgers' equation using Godunov's method.

The initial condition is given by:

$$u(x,0) = \begin{cases} 1, & \text{for } |x| < \frac{1}{3}, \\ -1, & \text{for } \frac{1}{3} < |x| \leq 1. \end{cases} \tag{4.10}$$

with periodic boundary conditions. The test is performed using Godunov's method with first- and second-order accuracy in space and time.

The image of the solution for Burgers' equation using Godunov's method is given in Figure 4.5.

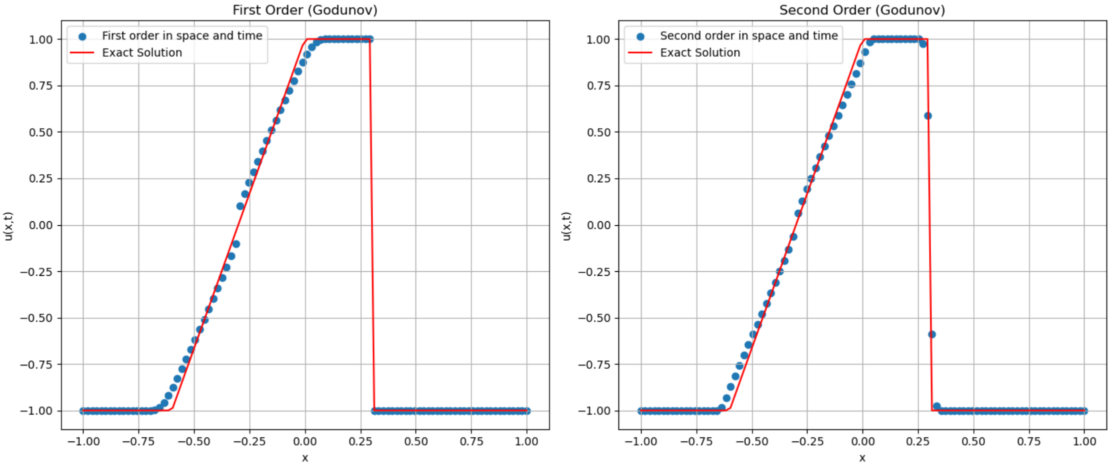


Figure 4.5: Solution of Burgers' equation using Godunov's method with modified initial conditions.

4.1.6 Test Case 4: Burgers' Equation with Roe's Method

The governing equation is the 1D inviscid Burgers' equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0. \tag{4.11}$$

The initial condition is given by:

$$u(x,0) = \begin{cases} 1, & \text{for } |x| < \frac{1}{3}, \\ 0, & \text{for } \frac{1}{3} < |x| \leq 1. \end{cases} \tag{4.12}$$

with periodic boundary conditions. The test is performed using Roe's method with first- and second-order accuracy in space and time.

The image of the solution for Burgers' equation using Roe's method is given in Figure 4.6.

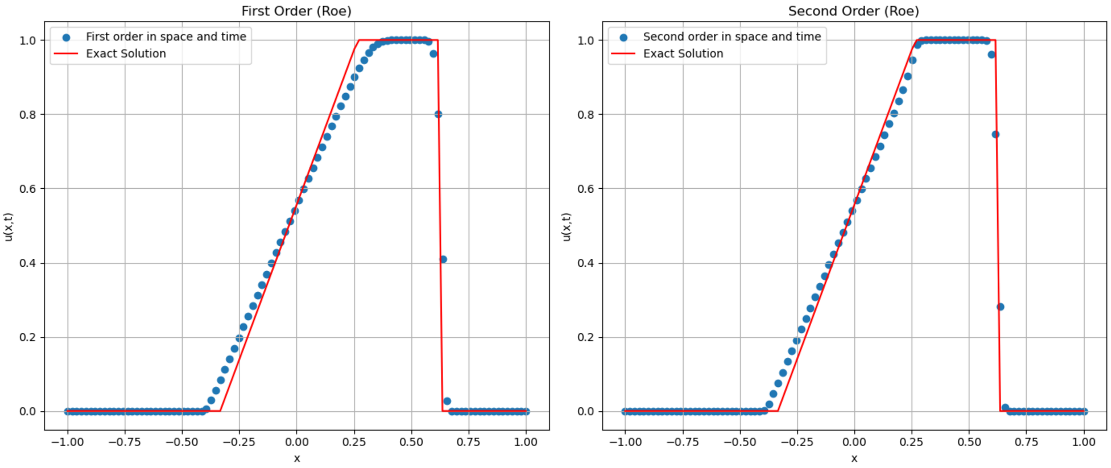


Figure 4.6: Solution of Burgers' equation using Roe's method.

4.1.7 Test Case 5: Burgers' Equation with Roe's Method (Modified Initial Condition)

The governing equation is the 1D inviscid Burgers' equation:

$$\frac{\partial u}{\partial t} + \frac{\partial}{\partial x}\left(\frac{1}{2}u^2\right) = 0. \tag{4.13}$$

The initial condition is given by:

$$u(x,0) = \begin{cases} 1, & \text{for } |x| < \frac{1}{3}, \\ -1, & \text{for } \frac{1}{3} < |x| \leq 1. \end{cases} \tag{4.14}$$

with periodic boundary conditions. The test is performed using Roe's method with first- and second-order accuracy in space and time.

The image of the solution for Burgers' equation using Roe's method is given in Figure 4.7.

4.2 Test Cases from "Entropy Conserving/Stable Schemes for a Vector-Kinetic Model of Hyperbolic Systems"

These test cases are taken from the paper Entropy Conserving/Stable Schemes for a Vector-Kinetic Model of Hyperbolic Systems. They evaluate the numerical schemes' ability to resolve expansion and shock waves in first- and second-order accuracy in space and time.

4.2.1 Test Case: 1D Expansion Problem

The governing equation for this test case follows a vector-kinetic model of hyperbolic systems. The initial condition is given by:

$$\rho(x_1, 0) = 1, \tag{4.15}$$

$$u_1(x_1, 0) = \begin{cases} 4, & \text{for } x_1 < 0, \\ -4, & \text{for } x_1 \geq 0. \end{cases} \tag{4.16}$$

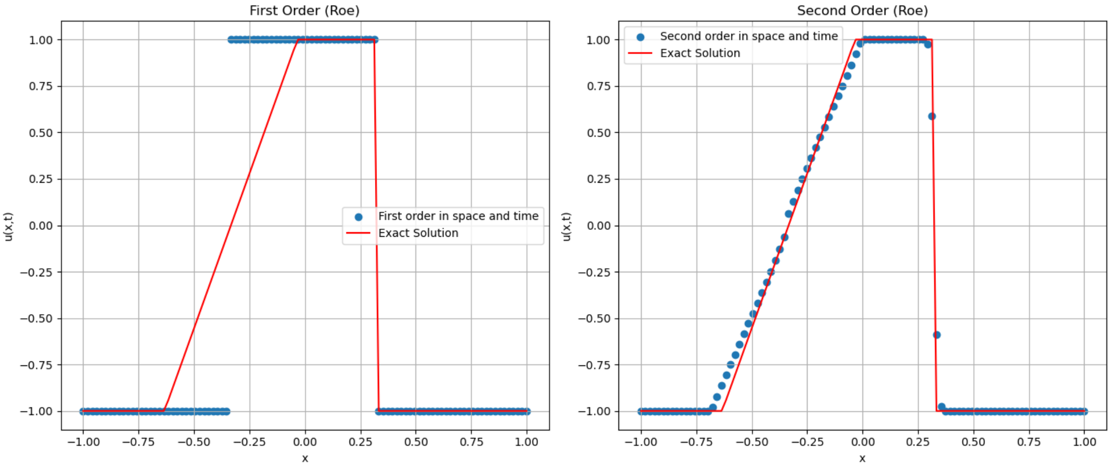


Figure 4.7: Solution of Burgers' equation using Roe's method with modified initial conditions.

The test is performed using first- and second-order schemes in space and time. Fixed boundary conditions are applied.

The image of the numerical solution for the 1D expansion problem is given in Figure 4.8.

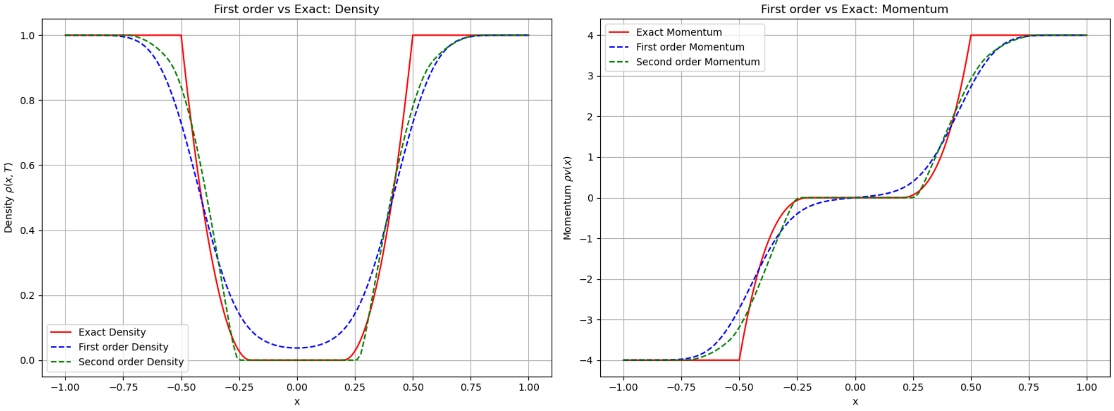


Figure 4.8: Solution of the 1D expansion problem.

4.2.2 Test Case: 1D Dam Break Problem

The governing equation for this test case follows a hyperbolic system model. The initial condition is given by:

$$\rho(x_1, 0) = \begin{cases} 15, & \text{for } x_1 < 0, \\ 1, & \text{for } x_1 \geq 0. \end{cases} \tag{4.17}$$

$$u_1(x_1, 0) = 0. \tag{4.18}$$

The test is performed using first- and second-order numerical schemes in space and time, along with the exact solution for comparison. Fixed boundary conditions are applied.

The numerical and exact solutions for the 1D dam break problem are given in Figure 4.9.

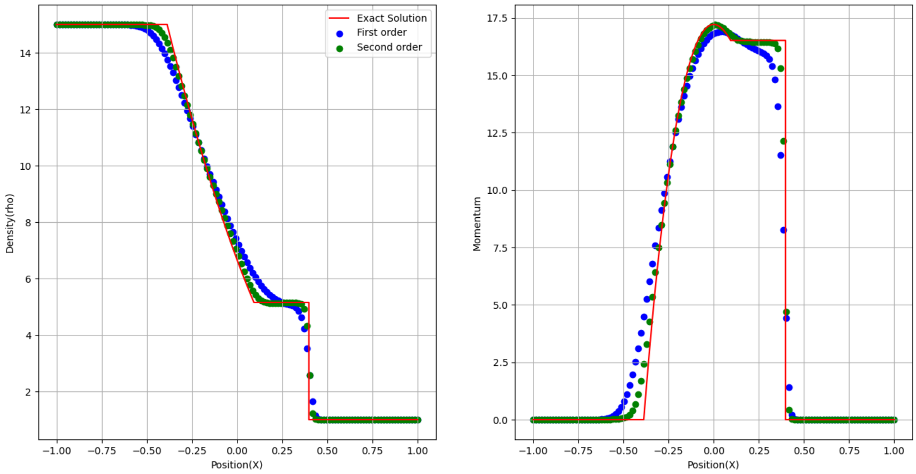


Figure 4.9: Solution of the 1D dam break problem: numerical vs. exact solution.

4.3 2D Test Cases

To evaluate numerical schemes in multidimensional settings, the following 2D test cases are considered. These test cases assess the ability of the numerical schemes to capture shock waves, rarefaction waves, and other complex flow features accurately.

4.3.1 2D Periodic Flow – Density and Momentum

This test case examines the performance of numerical schemes in a periodic flow setting, evaluating their ability to maintain periodic structures over time.

The initial conditions are given by:

$$\rho(x_1, x_2, 0) = 1 + \sin^2(2\pi(x_1 + x_2)), \tag{4.19}$$

$$u_1(x_1, x_2, 0) = u_2(x_1, x_2, 0) = \sin(2\pi(x_1 - x_2)). \tag{4.20}$$

Periodic boundary conditions are applied in both spatial directions.

The results of the numerical simulation for density and momentum are presented in Figure 4.10.

4.3.2 2D Cylindrical Dam Break Problem – Density

This test case evaluates the ability of numerical schemes to capture the evolution of a cylindrical dam break, where an initially high-density region collapses under its own pressure.

The initial conditions are given by:

$$\rho(x_1, x_2, 0) = \begin{cases} 2, & \text{if } \sqrt{x_1^2 + x_2^2} < 0.5, \\ 1, & \text{otherwise.} \end{cases} \tag{4.21}$$

$$u_1(x_1, x_2, 0) = u_2(x_1, x_2, 0) = 0. \tag{4.22}$$

Periodic boundary conditions are applied in both spatial directions.

The evolution of the density field due to the dam break is shown in Figure 4.11.

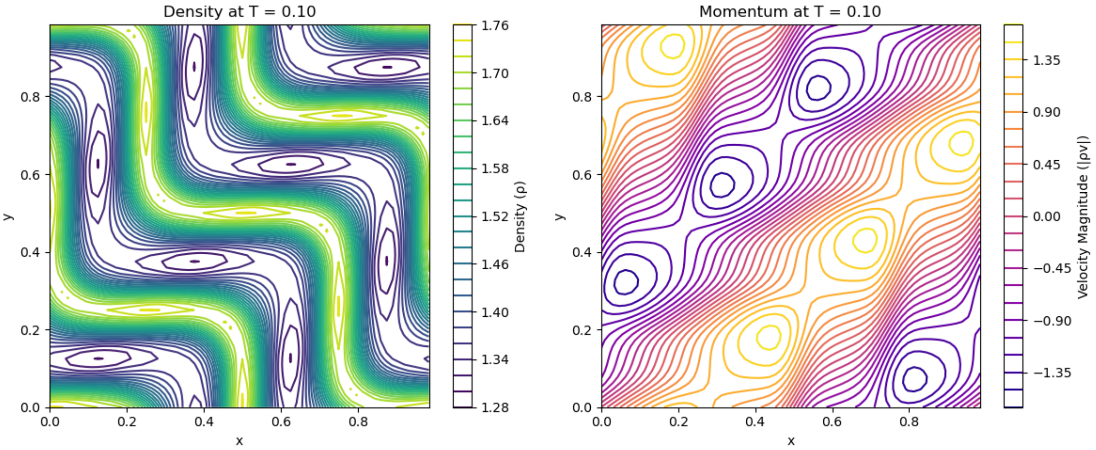


Figure 4.10: Solution of the 2D periodic flow problem showing the evolution of density and momentum.

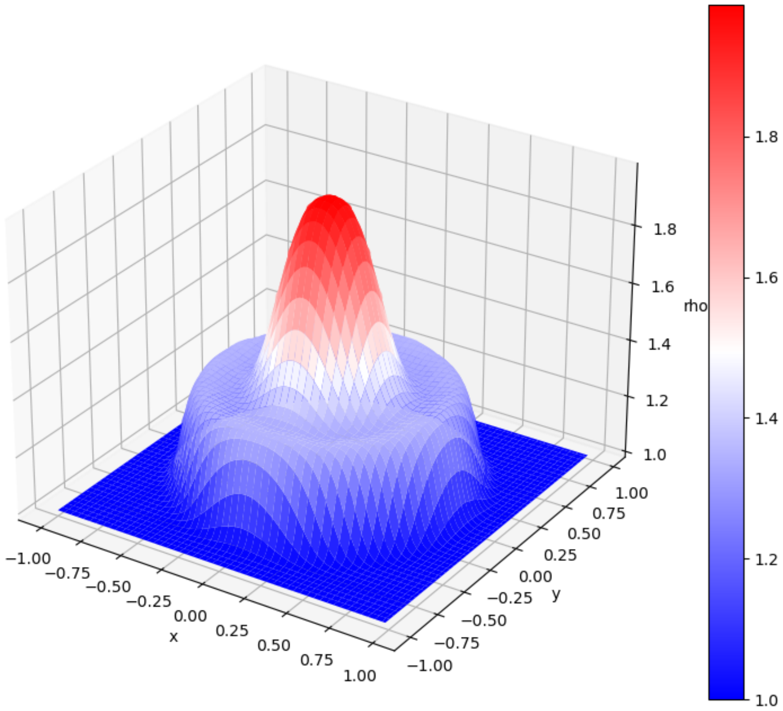


Figure 4.11: Solution of the 2D cylindrical dam break problem showing the evolution of density.

4.3.3 Multigrid Solution for Hyperbolic Conservation Laws

This test case is based on Stefan Spekreijse's (1987) work on solving hyperbolic conservation laws using monotone second-order discretizations. The goal is to evaluate the efficiency of the multigrid method in solving hyperbolic PDEs.

The governing equation for linear convection is given by:

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} + b\frac{\partial u}{\partial y} = 0, \tag{4.23}$$

where the wave propagation direction is defined as:

$$a = \cos\phi, \quad b = \sin\phi, \quad \phi \in \left(0, \frac{\pi}{2}\right). \tag{4.24}$$

The initial conditions are:

$$u(x, y, 0) = 0. \tag{4.25}$$

The boundary conditions are specified as:

$$u(0, y) = 1, \quad 0 < y < 1, \tag{4.26}$$

$$u(x, 0) = 1, \quad 0 < x < 1. \tag{4.27}$$

The numerical solution is obtained using a multigrid method with a monotone second-order discretization scheme.

The evolution of the solution using the multigrid method is illustrated in Figure 4.12.

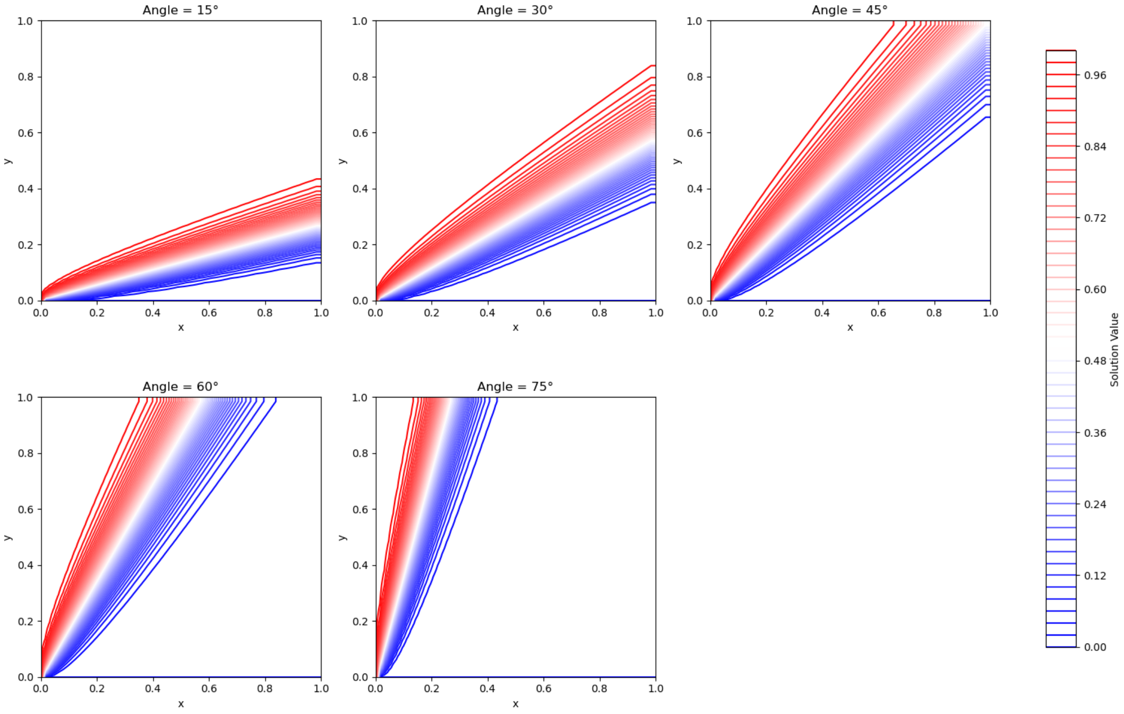


Figure 4.12: Numerical solution of the linear convection equation using a multigrid method.

Chapter 5

Visualization and VTK Files

This chapter discusses various visualization techniques and the use of VTK (Visualization Toolkit) files for post-processing numerical results in ParaView. Proper visualization is essential for analyzing the accuracy and stability of numerical schemes, particularly in solving hyperbolic conservation laws.

5.1 Visualization Techniques

Visualization plays a crucial role in interpreting numerical solutions. The following techniques are commonly used:

- **Contour Plots:** Used to display scalar fields, such as density or pressure distributions, in 2D and 3D simulations.
- **Vector Field Plots:** Represent flow characteristics using velocity vectors, which help in understanding directional properties.
- **Surface Plots:** Provide a 3D representation of solution variables over computational grids.
- **Iso-Surfaces:** Used for visualizing 3D fields by extracting surfaces of constant value.
- **Streamlines and Pathlines:** Illustrate the flow of particles along the velocity field, aiding in fluid dynamics analysis.

5.2 VTK File Format and Usage

The VTK (Visualization Toolkit) file format is widely used for storing and visualizing computational results. VTK files can be generated from numerical simulations and imported into visualization software such as ParaView.

5.2.1 VTK File Structure

A typical VTK file consists of the following sections:

- **# vtk DataFile Version 4.2** – Header specifying the VTK file version.
- **ASCII / BINARY** – Data format specification (ASCII or binary).
- **Dataset Type** – Specifies the structure (e.g., structured grid, unstructured grid, polydata).
- **Points** – Defines the coordinate system and grid points.
- **Cells** – Describes the connectivity of the elements.
- **Point Data / Cell Data** – Contains scalar or vector field values.

5.2.2 Generating VTK Files from Simulation Data

Numerical simulation results can be exported as VTK files using Python or C++ utilities. An example of a simple ASCII VTK file for a structured grid is:

```
# vtk DataFile Version 4.2
VTK Output Example
ASCII
DATASET STRUCTURED_POINTS
DIMENSIONS 10 10 1
ORIGIN 0 0 0
SPACING 1 1 1
POINT_DATA 100
SCALARS density float
LOOKUP_TABLE default
0.5 0.6 0.7 ... (values for 100 points)
```

5.2.3 Post-Processing in ParaView

Once VTK files are generated, they can be loaded into ParaView for detailed visualization and analysis. The steps include:

1. Open ParaView and import the generated VTK file.
2. Use filters such as **Contour**, **Slice**, and **Stream Tracer** for enhanced visualization.
3. Apply colormap settings to improve clarity of scalar fields.
4. Animate time-dependent data to observe wave propagation and shock formation.

5.3 Conclusion

VTK-based visualization provides powerful tools for analyzing numerical simulations. By utilizing ParaView, researchers can effectively interpret results, identify numerical artifacts, and validate computational methods.

Chapter 6

Conclusion and Future Work

6.1 Conclusion

This work has explored various numerical schemes for solving hyperbolic conservation laws, focusing on their accuracy, stability, and computational efficiency. Through a series of test cases, including the 1D and 2D dam break problems, periodic flow scenarios, and multigrid solutions, we have demonstrated the effectiveness of different discretization techniques.

The key findings of this study are as follows:

- **High-Resolution Schemes:** Second-order accurate methods improve solution quality while maintaining numerical stability.
- **Multigrid Acceleration:** The use of multigrid techniques significantly reduces computational cost while preserving accuracy.
- **Periodic Boundary Conditions:** These conditions were effectively implemented to ensure smooth solutions in periodic flow simulations.
- **Shock Capturing:** The numerical schemes successfully resolved shock structures and rarefaction waves in dam break problems.

Overall, the numerical experiments validate the robustness of the chosen discretization methods and highlight the trade-offs between accuracy and computational efficiency.

6.2 Future Work

Although this study provides valuable insights into numerical methods for hyperbolic conservation laws, several avenues for future research remain:

- **Higher-Order Schemes:** Implementing and analyzing third- and fourth-order schemes to improve solution accuracy further.
- **Adaptive Mesh Refinement (AMR):** Investigating AMR techniques to dynamically refine the computational grid in regions of high gradients.
- **GPU Acceleration:** Exploring parallel computing strategies, including GPU-based solvers, to enhance computational performance.
- **Alternative Time Integration Methods:** Studying implicit and semi-implicit time-stepping approaches to handle stiff problems efficiently.
- **Validation with Experimental Data:** Comparing numerical results with experimental or real-world data to assess practical applicability.

By addressing these areas, future research can further refine numerical models and improve their application to real-world fluid dynamics problems.

Chapter 7

References

1. Culbert B. Laney, *Computational Gas-Dynamics*, Cambridge University Press, 1998.
2. Megala Anandan, "Entropy Conserving/Stable Schemes for a Vector-Kinetic Model of Hyperbolic Systems."
3. Stefan Spekreijse, "Multigrid Solution of Monotone Second-Order Discretizations of Hyperbolic Conservation Laws," 1987.
4. Eleuterio F. Toro, *Shock Capturing Methods for Free Surface Shallow Flows*, Wiley, 2001.