Higher-Order High-Resolution Schemes for Hyperbolic Equations

A thesis submitted
in partial fulfillment for the degree of

Doctor of Philosophy

by

ARUN GOVIND NEELAN A

DEPARTMENT OF AEROSPACE ENGINEERING
INDIAN INSTITUTE OF SPACE SCIENCE AND TECHNOLOGY
THIRUVANANTHAPURAM - 695547, INDIA

JULY 2021

Certificate

This is to certify that the thesis entitled **"Higher-Order High-Resolution Scheme for Hyperbolic Equations"** submitted by **Arun Govind Neelan A**, to the Indian Institute of Space Science and Technology, Thiruvananthapuram, in partial fulfillment for the award of the degree of **Doctor of Philosophy**, is a bonafide record of the project research work carried out by him under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr Manoj T Nair	**Aravind Vaidyanathan**
Associate Professor	Professor
Department of Aerospace Engineering	Department of Aerospace Engineering

Place: Thiruvananthapuram
July 2021.

Abstract

An accurate resolution of shock structures is crucial to the design and load estimation of objects flying beyond sound speed. The aero-thermal loads acting on objects cruising at supersonic speed are very sensitive to the Mach number. Most spacecraft use a very low safety factor. Accurate resolution of shock structures is essential for an optimal and efficient space mission. High-resolution and higher-order schemes are needed to resolve shock structures without much dissipation and dispersion. Higher-order schemes are generally computationally expensive per iteration but are computationally economical for a given error. In addition to that higher-order schemes can able to resolve complex vortex structures relatively better than lower-order schemes.

Some of the novel schemes with high-resolution higher-order properties are presented in this work. These new schemes are computationally cost-effective than other schemes. A new family of Runge-Kutta (RK) method is presented. They are optimized for stability using an evolutionary algorithm and are suitable for shock-related problems. They outperformed the classical, and strong stability Runge-Kutta method in terms of stability and convergence. The procedures for obtaining higher-order schemes in uniform and non-uniform mesh are explored. Conservative discretization leads to lower-order convergence when a non-uniform grid is used. They can give a good convergence rate when a progressivly streched grid is used. This phenomenon is analytically proved using symbolic based code..

A second-order and a third-order limiter with excellent shock-resolution properties are presented. They are ideal for problems with blast waves. In terms of shock-resolving properties, they outperformed other limiters. High-resolution schemes such as weighted essentially non-oscillatory schemes (WENO) and limiter schemes are explored, and novel schemes are presented. The classical WENO schemes are two-level schemes because they stay at either third-order or fifth-order. The current scheme is a three-level scheme that can deliver third-order, fourth-order and fifth-order accuracy. The current scheme has a tuning parameter that can be tuned based on the problem requirement. The current scheme outperformed other WENO schemes considered in this paper based on resolution and computational costs. A hybrid shock capture-fitting algorithm that can solve a shock with three grid points is presented. Some of the standard and non-standard schemes are explored in this work and have observed some significant improvements in terms of computational costs and shock resolution property.

Declaration

I declare that this thesis titled **"Higher-Order High-Resolution Scheme for Hyperbolic Equations"** submitted in partial fulfillment of the Degree of **"Doctor of Philosophy"** is a record of original work carried out by me under the supervision of **Dr Manoj T Nair**, and has not formed the basis for the award of any degree, diploma, associateship, fellowship, or other titles in this or any other Institution or University of higher learning. In keeping with the ethical practice in reporting scientific information, due acknowledgements have been made wherever the findings of others have been cited.

Arun Govind Neelan A
Research Scholar
Department of Aerospace Engineering
Roll No.: SC15D010

Place: Thiruvananthapuram
July 2021.

Acknowledgments

I would like to thank my supervisor **Dr. Manoj T Nair** for providing the freedom to explore various topics, patience, motivation and immense knowledge. Without his support, it would be impossible for me to complete this work. I would like to express my sincere gratitude to **Prof. Vinayak Eswaran** from IIT Hyderabad and **Dr. T Jayachandran** from IIT Madras for supporting my work and motivating me. I would like to say big thank to my doctoral committee members **Prof. Salih A**, **Dr. Vinoth B.R**, **Dr. Satheesh K** and **Dr. Natarajan E** for their patience and helping me to correct my mistakes.

I'm living in a place where most of the peoples think only jobless students do PhD. I thank my parents **Mr. Anbazhagan K N** and **Mrs. Kokilam K** for their patience even after facing criticism from the society and supported me till the end. I thank **Dr. Manu K Vasudevan**, **Dr. Prathap C** and **Mr. Jishnu Chandran R** for helping me to set up the lab. I thank **Prof. Kuruvilla Joseph** and **Dr. Mahesh S** who encouraged me when I was struggling to make publications. I would have quite my PhD if they had not support me.

I thank **Dr. Katate Masatsuka**, **Dr. Hiroaki Nishikawa** from National Institute of Aerospace, Hampton, United States and "I do like CFD" team. Some of their Euler equation codes are modified and extended to a few new schemes in the present work. I also thank **Dr. Manuel A. Diaz** from Université Paris-Sorbonne and **Dr. Jun Peng** from the Chinese Academy of Sciences, Beijing for sharing their WENO codes. The WENO code shared by them is modified and extended to novel WENO schemes in this work.

I like to thank **Dr. Appadu Appanah Rao** from Nelson Mandela University, Port Elizabeth, **Dr. Ritesh Kumar Dubey** from SRM Institute of Science and Technology, Chennai, **Dr. Samala Rathan** from Indian Institute of Petroleum and Energy, Visakhapatnam, **Dr. G Naga Raju** from Visvesvaraya National Institute of Technology, Nagpur and **Prof. Raimund Bürger** from the University of Concepción, Chile helped me to improve the work.

I thank **Mr. Sarath K P**, **Ms. Risha Raju**, **Dr. Shashank Vadlamani**, **Mrs. Anuja Vijayan**, **Dr. Muthukumaran, C.K**, **Mr. Sanjay Yogesh Choudhary**, **Dr. Sathish Kumar P**, **Dr. Solomon Ivan J** and **Mr. Sreekumar** for their support, patience and advice. It is impossible to complete the degree without their support and help. I also thank the **Department of Space**, government of India and **Indian Institute of Space Science and Technology**, Thiruvananthapuram for providing financial support for this work. I like to thank anonymous reviewers and editors of journals for their suggestions to improve the present work. I also thank open-access journals for publishing some of the work available for everybody.

Arun Govind Neelan
Thiruvananthapuram, India.

Table of Contents

List of Tables

List of Figures

Abbreviations

CFD	Computational Fluid Dynamics
CAA	Computational Aero Acoustics
LES	Large Eddy Simulation
DNS	Direct Numerical Simulation
SCS	Shock Capturing Scheme
ENO	Essentially Non-Oscillatory
WENO	Weighted Essentially Non-Oscillatory
BWENO	Bandwidth optimized Weighted Essentially Non-Oscillatory
DRP	Dispersion Relation Preserving
STDRP	Spatial-Temporal Dispersion Relation Preserving
RK	Runge Kutta
HRK	Hyperbolic Runge-Kutta
LRK	Low storage Runge-Kutta
CRK	Classical Runge-Kutta
FVM	Finite Volume Method
FDM	Finite Difference Method
FEM	Finite element Method
N-FVM	Nonuniform Finite Volume Method
DPS	Discontinuity Preserving Scheme
FTBS	Forward in Time and Backward in Space
FTBS2	Forward in Time and second order Backward in Space
FTFS	Forward in Time and Forward in Space

FTCD2	Forward in Time and second order central in Space
CFL	Courant–Friedrichs–Lewy
PBR	Point Based Reconstruction
CAR	Cell Averaged Reconstruction
PPR	Piecewise Parabolic Reconstruction
AV	Artificial Viscosity
CD	Central Difference
N-CD	Non-uniform Central Difference
VNS	von Neumann stability
TVD	Total Variation Diminishing
IC	Initial Condition
BC	Boundary Condition
NS	Navier-Stokes
SVS	Singular Value Decomposition
$\mathbf{A}$	Jacobian Flux
MOL	Method of Line
AP	Arithmetic Progression
PDE	Partial Differential Equation
ODE	Ordinary Differential Equation
MOC	Method of Characteristic
GA	Genetic Algorithm
EA	Evolutionary Algorithm
ABM	Adams-Bashforth method

1

Introduction

Analyzing a system or process is an essential part of our lives. Human evolution is remarkable, perhaps because of their ability to analyze and optimize. First, to analyze the system, we should model the system. Most systems can be modelled using differential equations. Once the model has been identified, we should develop strategies to solve the system. The strategies can be approximate or analytical. Approximate solutions can be either a discrete solution or a series solution. The analytical solution offers the simplest and most compact solution for the model. Unfortunately, due to its complexity and non-linearity, most of the systems do not have an analytical solution. In this case, approximation methods and series solution methods have better applications than analytical solutions. Series solution methods may not be available for most general partial differential equations (PDEs) due to convergence and non-linearity issues. In this case, discrete solutions using numerical methods may be preferred over others.

Most numerical methods such as Finite Difference Methods (FDM), Finite Volume Methods (FVM), Finite Element Methods (FEM) can be integrated into a single-family called Weighted Residual Method. Based on the weighting function that we use, we can derive FDM, FVM, Galerkin-FEM, Ritz-FEM, etc. The weighted residual method reduces to FDM when an impulse or delta weighting function is used and reduces to FVM when a unit weighting function is used. Based on the nature of the weighting function and the shape functions used in the weighted residuals, different FEM families are derived. In this work, we will focus on FDM and FVM because they are ideal for conservative equations. Based on the nature of the characteristics, PDE can be classified as elliptical, parabolic and hyperbolic. In some cases, the equation may have combinations of these; for example, the Navier-Stokes equation has elliptical and hyperbolic characters. Hyperbolic equations are of interest among the different types of PDEs due to their wave-like properties.

Fluid flow is one of the complicated systems that can be modelled using the partial differential equation. Any fluid motion can be modelled using the Boltzmann equation or Boltzmann transport equation which is derived by Ludwig Boltzmann in 1872. It describes the statistical behaviour of the system in the non-equilibrium state. It is given in eq. 1.1.

$$\frac{\partial f}{\partial t} + \frac{\mathbf{p}}{m} \cdot \nabla f + \mathbf{F} \cdot \frac{\partial f}{\partial \mathbf{p}} = \left(\frac{\partial f}{\partial t}\right)_{\text{coll}} \tag{1.1}$$

where, $F(r, t)$ is the force field acting on the particle of mass m. $\mathbf{p}$ is momentum vector. Although this equation looks simpler than the Navier-Stokes (NS) equation, it cannot be solved easily. The collision term of f is difficult to obtain and is generally modelled, e.g. Bhatnagar-Gross-Crook (BGK) model [1]. It is a statistical term that represents a particle collision. In continuum assumption, the Boltzmann equation reduces to the NS equation. We can drop viscosity in the NS equation at high-speed flows if we are not interested in phenomena associated with the boundary layer. On dropping the viscous terms the NS equation reduces to Euler equation. The Euler equation is relatively more difficult to solve using numerical schemes than the NS equation due to lack of physical viscosity. NS equation is computationally expensive to solve because the boundary layer is very small; it is order of microns in supersonic flows. We need a very fine grid to resolve the boundary layer, which significantly increases computing cost. If we use a stretched grid, it will increase the aspect ratio of the grid and reduce the solver's accuracy and robustness. The challenging part of the Euler equation is when two different characteristics interact, leading to shock formation. These shocks are not only observed in the supersonic flow, but also in the pipe flow when the valve is closed instantaneously.

In the aerospace industries, the factor of safety used is about 1.5 or less than that [2]. An accurate estimation of the load is essential to avoid mechanical related failure. Generally, the source of this load is aerodynamic or aero-thermal phenomena. The shocks are responsible for both aerodynamic and aero-thermal loads. The shocks can increase the drag and the temperature which causes severe stress to the structures. If the shocks interact with the boundary layer, it may lead to severe thermal load. Even if the stress is below the yield-stress, it may lead to aero-elastic failures.

1.1 Engineering failures could be because of loads due to shocks

This section will briefly study some engineering failures likely due to a poor estimate of fluid-dynamic or aero-thermal loads.

Soyuz 11 Disaster (6-Jun-1971) Soyuz 11 carried the crew that landed on the world's first space station. This disaster occurred due to the rupture of the breathing ventilation valve leading to the death of three crew members in space. The valve failed due to the force caused by the simultaneous activation of the explosive bolts holding the orbital module and the descent module.

STS-1 (12-Apr-1981) In the solid rocket booster of this orbiter, ignition shock wave increased the pressure four times more than the estimated value. This high pressure bent the struts supporting reaction control system of the fuel tank and the orbiter's flap was pushed up and down by the shock waves.

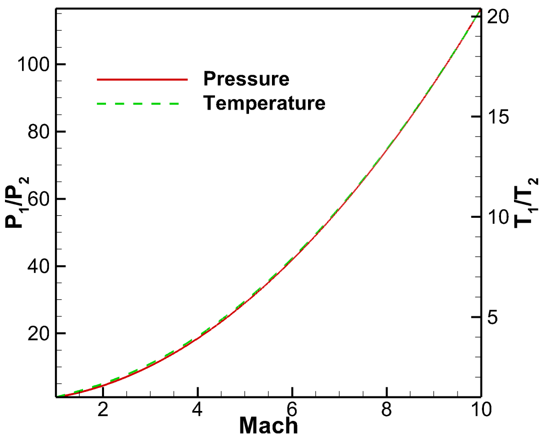


Figure 1.1: Variation of pressure ratio and temperature ratio across normal shock

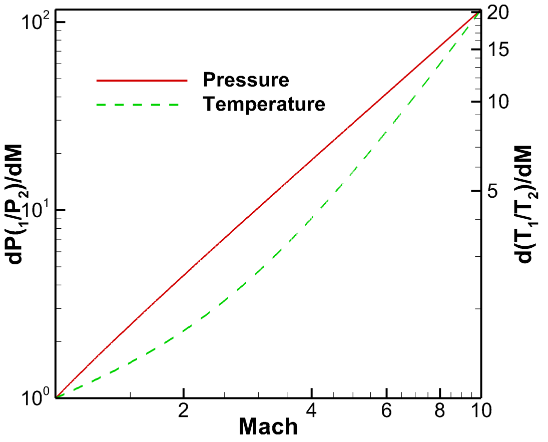


Figure 1.2: Sensitivity of pressure ratio and temperature ratio across normal shock

STS-27 (6-Dec-1988) Ablative insulation material from the solid rocket booster hit the orbiter during take-off. During the re-entry phase, more than 700 tiles were damaged, and one was missing. This shuttle could be one of the heavily damaged shuttles that were safely returned to Earth.

STS-107: Space Shuttle Columbia Disaster (1-Feb-2003) In Columbia Space Shuttle, a piece of foam broke off the shuttle, causing damage to the thermal protection system. This led to the death of seven astronauts, including Kalpana Chawla, the first woman of Indian descent to go to space during the reenty phase.

Some of the space shuttle accidents that have occurred in the past could have been caused by shocks. Failure due to shocks on the hydraulic actuators or the control surface is common in fighter aircraft. An accurate assessment of the shock is essential in the critical manoeuvring of the aircraft at high speed or in the re-entry phase of space shuttles. This causes heavy loads in the form of mechanical and thermal loads on the airframes.

1.2 Waves and hyperbolic equations

Waves provide solutions to some of the most challenging real-world problems. It could be gravitational waves, which are disturbances in the gravitational field, which spread based on general relativity, plasma waves, linking the mechanical and electromagnetic fields. Wave can be defined as a disturbance that travels through a medium, gradually transferring energy from one point to another without transporting matter. Wave may take the form of elastic deformation or variation in pressure, electrical or magnetic intensity, electrical potential or temperature. Waves can be broadly classified as mechanical waves (requiring a medium) and electromagnetic waves (do not require a medium). In this study, we look at mechanical waves. Some popular waves are listed in the table 1.1. Most of the wave equations show hyperbolic characters. The governing equations are hyperbolic if all the eigenvalues of the Jacobian matrix of the system are real. Let us consider a system of s first-order partial differential equations for s unknown functions $\vec{u} = (u_1, \ldots, u_s), \vec{u} = \vec{u}(\vec{x}, t)$, where $\vec{x} \in \mathbb{R}^d$

$$\frac{\partial \vec{u}}{\partial t} + \sum_{j=1}^{d} \frac{\partial}{\partial x_j} \vec{f^j}(\vec{u}) = 0, \tag{1.2}$$

where $\vec{f^j} \in C^1(\mathbb{R}^s, \mathbb{R}^s), j = 1, \ldots, d$. The Jacobian of $\vec{f^j}$ is

$$A^j := \begin{pmatrix} \frac{\partial f_1^j}{\partial u_1} \cdots \frac{\partial f_1^j}{\partial u_s} \\ \vdots ; \ddots \vdots \\ \frac{\partial f_s^j}{\partial u_1} \cdots \frac{\partial f_s^j}{\partial u_s} \end{pmatrix}, \text{ for } j = 1, \ldots, d.$$

Let $\alpha_1, \ldots, \alpha_d$ are the Eigen values of the Jacobian matrix. The eq. 1.2 is hyperbolic if for all $\alpha_1, \ldots, \alpha_d \in \mathbb{R}$

Numerical methods for solving hyperbolic equations introduce numerical dissipation and dispersion in the solution. Numerical phenomena that reduce the magnitude of the signal more than physical dissipation are called **numerical dissipation**. The signal speed is related to the wavenumber. Different wavenumbers may travel at a different speed, leading to **numeric dispersion**. The numerical scheme should minimize those error. The factor of safety used in the aerospace applications is very low. Flow parameters are also highly sensitive to the Mach number (figure 1.2). So, in the high-precision load analysis of the spacecraft, an accurate numerical scheme is unavoidable.

Table 1.1

Popular wave equations

Name	Equation
Electromagnetic wave	$\left(v_{ph}^2\nabla^2 - \frac{\partial^2}{\partial t^2}\right)\mathbf{E} = \mathbf{0}$ $\left(v_{ph}^2\nabla^2 - \frac{\partial^2}{\partial t^2}\right)\mathbf{B} = \mathbf{0}$ $v_{ph} = \frac{1}{\sqrt{\mu\varepsilon}}$
Schrodinger equation	$i\hbar\frac{\partial}{\partial t}\Psi(\mathbf{r},t) = \left[\frac{-\hbar^2}{2m}\nabla^2 + V(\mathbf{r},t)\right]\Psi(\mathbf{r},t)$
Dirac equation	$\left(\beta mc^2 + c\sum_{n=1}^{3}\alpha_n p_n\right)\psi(x,t) = i\hbar\frac{\partial\psi(x,t)}{\partial t}$
Shallow water	$\frac{\partial(\rho\eta)}{\partial t} + \frac{\partial(\rho\eta u)}{\partial x} + \frac{\partial(\rho\eta v)}{\partial y} = 0,$ $\frac{\partial(\rho\eta u)}{\partial t} + \frac{\partial}{\partial x}\left(\rho\eta u^2 + \frac{1}{2}\rho g\eta^2\right) + \frac{\partial(\rho\eta uv)}{\partial y} = 0,$ $\frac{\partial(\rho\eta v)}{\partial t} + \frac{\partial(\rho\eta uv)}{\partial x} + \frac{\partial}{\partial y}\left(\rho\eta v^2 + \frac{1}{2}\rho g\eta^2\right) = 0.$
Navier-Stokes	$\rho\frac{D\mathbf{u}}{Dt} = -\nabla\bar{p} + \mu\,\nabla^2\mathbf{u} + \tfrac{1}{3}\mu\,\nabla(\nabla\cdot\mathbf{u}) + \rho\mathbf{g}$
Frank-Tamm formula	$\frac{d^2E}{dx\,d\omega} = \frac{q^2}{4\pi}\mu(\omega)\omega\left(1 - \frac{c^2}{v^2n^2(\omega)}\right)$
Lee wave	$N = \sqrt{\frac{g}{\theta_0}\frac{d\theta_0}{dz}}$
Non-dispersive Wave Equation	$\nabla^2 A = \frac{1}{v^2}\frac{\partial^2 A}{\partial t^2}$
KdV equation	$\frac{\partial y}{\partial t} + \alpha y\frac{\partial y}{\partial x} + \frac{\partial^3 y}{\partial x^3} = 0$
Benjamin-Bona-Mahony equation	$u_t + u_x + uu_x - u_{xxt} = 0.$
Elastic waves	$\rho\ddot{\mathbf{u}} = \mathbf{f} + (\lambda + 2\mu)\nabla(\nabla\cdot\mathbf{u}) - \mu\nabla\times(\nabla\times\mathbf{u})$
Burgers equation	$u_t + uu_x - au_{xx} = 0$
Fisher equation	$u_t - u_{xx} - u\,(1-u) = 0$
Sine Gordon equation	$u_{tt} = au_{xx} + b\sin(\lambda u)$
Cubic Schrodinger equation	$iu_t + u_{xx} + q\,\lvert u\rvert^2 u = 0$
Boussinesq equation	$u_{tt} - u_{xx} + 3uu_{xx} + \alpha u_{xxxx} = 0$

1.3 Linear and non-linear hyperbolic equations

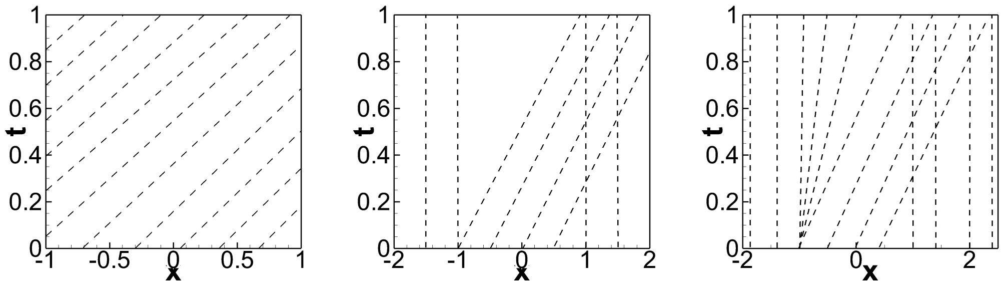
Simplest one-dimensional hyperbolic equation is linear convection equation. The equation is

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{1.3}$$

In simple convection equations, information is propagated at a constant velocity so that the initial condition (IC) moves at a velocity of c. For the initial condition

$$g(x) = \begin{cases} 0 & \text{if } x < 0 \\ k & \text{if } 0 \leq x \leq 1 \\ 0 & \text{if } x > 1 \end{cases} \tag{1.4}$$

where $k > 0$, all the characteristics in x/t planes never interact. So there is no complexity in the analytical solution. The characteristics are given in figure 1.3a.



(a) Convection equation **(b)** Burgers equation **(c)** Burgers equation

Figure 1.3: Solution in x/t plane

The inviscid Burgers equation is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0 \tag{1.5}$$

In the Burgers equation, information is propagated at a speed of u. For the IC given in eq. 1.4, the velocity is high in some regions and low in some regions. The possible solutions are shown in Figure 1.3b and Figure 1.3c. There is a jump in the expansion region in Figure 1.3b (empty space). In addition to that, it did not explain the propagation of signal when two characteristics interact. So this is a wrong and incomplete solution, and it also violates the condition of entropy. Figure 1.3c shows the exact solution for the Burgers equation. It clearly explains the expansion region formed by the left signal moving slower than the right signal. Also, it explains the phenomenon of interaction between the two characteristics leading to the formation of a shock when the left signal speed is more than the right. In such cases, two or more characteristics may merge and form a single wave with discontinuity in flow variables. The degree of discontinuity depends on the speed of the waves.

The solution procedure is relatively different from the linear PDE even for simple non-linear equations. In cases of coupled non-linear equations, it becomes relatively more complex. The numerical scheme should take into account the physics and allow for the solution to be stable. Please refer to [3, 4] for more details on the Burgers equation.

1.4 Challenges in numerical methods

In this section, we shall study some of the challenges in the solution procedures entailed in the numerical algorithm when we solve some simple equations and complicated non-linear equations using the higher-order schemes. Finally, we will investigate the advantages of higher-order high-resolution schemes and well-stable numerical schemes for solving non-linear equations. In this report, we define well-stables schemes are the optimized explicit schemes that allow relatively more time step than classical explicit schemes of same order. In this chapter, some of the basic results are presented for simple test cases. The detailed algorithm to achieve those solutions and the solutions of complicated test cases are presented in the next few chapters.

1.4.1 Linear and non-linear equations

We use linear approximations in the algorithm to solve non-linear equations. We will study one of the basic linearization techniques used to solve the Navier-Stokes(NS) equation in pressure-based solvers. The integral form of the x-momentum equation on a nondeformable control volume is

$$\frac{\Delta x \Delta y}{\Delta t}(u_{i,j}^{n+1} - u_{i,j}^{n}) = \\ -\iint \left[\frac{\partial}{\partial x}\left(u^2 - \frac{1}{Re}\frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial y}\left(uv - \frac{1}{Re}\frac{\partial u}{\partial y}\right)\right] dxdy - \iint pdxdy$$

$$\frac{\Delta x \Delta y}{\Delta t}\left(u_{i,j}^{n+1} - u_{i,j}^{n}\right) + \left(E^1_{i+\frac{1}{2},j} - E^1_{i-\frac{1}{2},j}\right)\Delta y + \left(F^1_{i,j+\frac{1}{2}} - F^1_{i,j-\frac{1}{2}}\right)\Delta x \\ + \left(p^{n+1}_{i+1,j} - p^{n+1}_{i,j}\right)\Delta y = 0$$

where E^1 and F^1 are defined as $E^1 = u^2 - \frac{1}{Re}\frac{\partial u}{\partial x}$ and $F^1 = uv - \frac{1}{\mathrm{Re}}\frac{\partial u}{\partial y}$; E^1 and F^1 are axial and transverse fluxes of x-momentum.

$$E^1_{i+\frac{1}{2},j} = 0.25\left(u_{i,j} + u_{i+1,j}\right)^2 - \frac{1}{Re}\frac{\left(u_{i+1,j} - u_{i,j}\right)}{\Delta x}$$

$$E^1_{i-\frac{1}{2},j} = 0.25\left(u_{i-1,j} + u_{i,j}\right)^2 - \frac{1}{Re}\frac{\left(u_{i,j} - u_{i-1,j}\right)}{\Delta x}$$

$$F^1_{i,j+\frac{1}{2}} = 0.25\left(v_{i,j} + v_{i+1,j}\right)\left(u_{i,j} + u_{i,j+1}\right) - \frac{1}{\mathrm{Re}}\left(\frac{u_{i,j+1} - u_{i,j}}{\Delta y}\right)$$

$$F^1_{i,j-\frac{1}{2}} = 0.25\left(v_{i,j-1} + v_{i+1,j-1}\right)\left(u_{i,j-1} + u_{i,j}\right) - \frac{1}{\mathrm{Re}}\left(\frac{u_{i,j} - u_{i,j-1}}{\Delta y}\right)$$

The linearized form of these equations are

$$E^1_{i+\frac{1}{2},j} = 0.25\left(u^n_{i,j} + u^n_{i+1,j}\right)\left(u^{n+1}_{i,j} + u^{n+1}_{i+1,j}\right) - \frac{1}{Re}\frac{\left(u^{n+1}_{i+1,j} - u^{n+1}_{i,j}\right)}{\Delta x} \tag{1.6a}$$

$$E^1_{i-\frac{1}{2},j} = 0.25\left(u^n_{i-1,j} + u^n_{i,j}\right)\left(u^{n+1}_{i-1,j} + u^{n+1}_{i,j}\right) - \frac{1}{Re}\frac{\left(u^{n+1}_{i,j} - u^{n+1}_{i-1,j}\right)}{\Delta x} \tag{1.6b}$$

$$F^1_{i,j+\frac{1}{2}} = 0.25\left(v^n_{i,j} + v^n_{i+1,j}\right)\left(u^{n+1}_{i,j} + u^{n+1}_{i,j+1}\right) - \frac{1}{Re}\frac{\left(u^{n+1}_{i,j+1} - u^{n+1}_{i,j}\right)}{\Delta y} \tag{1.6c}$$

$$F^1_{i,j-\frac{1}{2}} = 0.25\left(v^n_{i,j-1} + v^n_{i+1,j-1}\right)\left(u^{n+1}_{i,j-1} + u^{n+1}_{i,j}\right) - \frac{1}{Re}\frac{\left(u^{n+1}_{i,j} - u^{n+1}_{i,j-1}\right)}{\Delta y} \tag{1.6d}$$

Eq. 1.6 is the linearized NS equation where i and j points are evaluated at different time level to avoid non-linearity. We can also observe another interesting linearization in the classical Roe-solver which leads to an expansion shock that is unphysical. Using secant-plane approximation, the non-conservative form of the Euler equation can be written as

$$\mathbf{F}(\mathbf{U^R}) = \mathbf{F}(\mathbf{U^L}) + \mathbf{A^{RL}}(\mathbf{U^R} - \mathbf{U^L}) \tag{1.7}$$

where,

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho e_T \end{bmatrix} \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho h_T u \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}u^2 & (3-\gamma)u & \gamma - 1 \\ -uh_T + \frac{1}{2}(\gamma-1)u^3 & h_T - (\gamma-1)u^2 & \gamma u \end{bmatrix}$$

lets take

$$\Delta u = u^R - u^L \qquad \Delta\rho = \rho^R - \rho^L, \qquad \Delta h_T = h_T^R - h_T^L \tag{1.8}$$

Substitute eq. 1.8 in eq. 1.7, we get

$$\begin{matrix} \Delta(\rho u) \\ \begin{bmatrix} \gamma - 1 \\ \frac{\gamma-1}{\gamma}\Delta\left(\rho h_T\right) + \frac{\gamma+1}{2\gamma}\Delta\left(\rho u^2\right) \\ \Delta\left(\rho u h_T\right) \end{bmatrix} \end{matrix}$$

$$= \begin{bmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}(u^{RL})^2 & (3-\gamma)u^{RL} & (\gamma-1) \\ -u^{RL}h^{RL} + \frac{1}{2}(\gamma-1)(u^{RL})^3 & h^{RL_T} - (\gamma-1)(u^{RL})^2 & \gamma u^{RL} \end{bmatrix}$$

$$\times \begin{bmatrix} \Delta\rho \\ \Delta(\rho u) \\ \Big[\frac{1}{\gamma}\Delta\left(\rho h_T\right) + \frac{1}{2\gamma}(\gamma-1)\Delta\left(\rho u^2\right) \end{bmatrix}$$

Eq. 1.8 assumed that the discontinuity in the solution occurs only across the characteristics and this is a valid assumption. But it may create a discontinuity in the rarification fans which is unphysical. Even in the non-linear equations, we make some linearization when solved using numerical methods. In the non-linear equation, when two characteristics from different families intersect and they may lead to the formation of the shocks. The shocks cannot be handled using higher-order schemes without any sophistication because of the high value in the magnitude of the derivatives. Please note that high value in the magnitude of the derivatives is not the only factor that affects the solution but one of the key factors that affects the solution of higher-order schemes. In that case, we need Riemann solvers to make the flows relatively smooth and solve the problem based on the direction of the characteristics. Please refer section 2.5 for more details about the requirement for the different steps in the high-speed solvers that may not be required in the low-speed solvers.

1.4.2 Advantages of high-resolution scheme and higher-order schemes

High-resolution methods are the numerical methods that can resolve the discontinuity without significant oscillations and dissipation. The oscillations present in the non-smooth data using higher-order methods can be explained by Taylors series. The Taylors series is:

$$f(x+h) = f(x) + hf^{(1)}(x) + \frac{h^2}{2!}f^{(2)}(x) + \frac{h^3}{3!}f^{(3)}(x) + ...\mathcal{O}(\Delta x^4) \tag{1.9}$$

Eq. 1.9 converges when $h < 1$; $h >> |f^{(1)}(x)|$; $\frac{h^2}{2!} >> |f^{(2)}(x)|$ and $|hf^{(1)}(x) + \frac{h^2}{2!}f^{(2)}(x)| >> |\frac{h^3}{3!}f^{(3)}(x) + ...|$. We can observe that the magnitude of the derivative and grid size plays an important role in the Taylors series convergence and accuracy. The smoothness is measured by the magnitudes of the derivatives of a given function. If the magnitude of the derivatives is high, it leads to divergence. The problems having shocks and discontinuity are likely to have high gradients so it may cause convergence issues in the Taylors series. We need sophisticated algorithms to handle this and, they are called as high-resolution schemes. Generally, high-resolution schemes are reconstruction schemes which are monotone so they do not create oscillations in the reconstruction step.

Godunov's order barrier theorem stated that: Any linear schemes higher than the first-order is non-monotone. Non-monotonicity of a scheme mainly depends on the nature of the solution, i.e. the magnitude of the derivatives, the order of the differential equation, variation in the magnitudes of its eigenvalue, grid size and time step used. The safest monotone region is the region defined by the positivity condition. The condition given by the positivity analysis may be an under-estimated value for some smooth and moderate-smooth solution.

Figure 1.4 shows the solution of linear convection equation using *forward in time first-order backward in space* scheme (FTBS) and *forward in time and second-order backward in space* scheme (FTBS2) using $CFL = 0.25$ on the domain [-5, 5]. The initial condition used is $u(x,t) = \exp(-5x^2)$ and the solution is obtained using 101 grid points. The solution at 1 s is shown in this figure. It is clear from this figure that the higher-order scheme produced oscillations in the solution. We should be wary of using the higher-order scheme when dealing with the non-smooth region, especially the discontinuities. High-resolution schemes are designed to solve shocks without dispersion. The scheme reduces to lower order and maintains monotonicity in the presence of shocks.

Higher-order methods may show a high convergence rate for problems with a smooth solution. If the numerical methods show a higher convergence rate, they will converge relatively faster to the exact solution. The convergence rate depends on the size of the grid, the magnitude of the derivatives of the solution, and the discretization used. It is not necessarily the case that higher-order methods always give a more accurate result than the lower-order methods.

This section contains a brief case study of higher-order methods that show higher error than lower-order methods in some grids. The test function is $f(x) = \exp(-32(x-5)^2)$ and the derivative is obtained from the domain [-10 10]. The domain is discretized using an arithmetic sequence grid of 50, 500, 5000 and 50000 grid points. The root mean square (RMS) error of the derivative approximation is calculated and plotted against the number of grid points in the log-log plot shown in figure 1.5.

The derivative is calculated using the first, second and third-order schemes. The initial error of the first and second-order scheme is lower than that of the third-order scheme. The third-order scheme is more accurate than the other methods considered here only when the grid is refined. This is because the convergence rate of the third-order scheme is higher than that of the first and second-order methods. Higher-order methods may not show a theoretical convergence rate when resolving non-smooth data.

So far, we have studied the limitations of higher-order on non-smooth data. Now, we

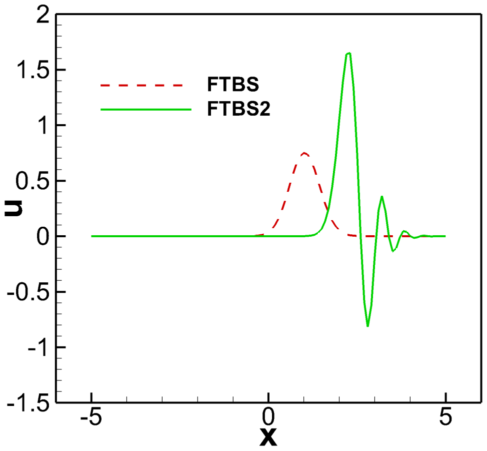


Figure 1.4: FTBS and FTBS2 for linear convection equation

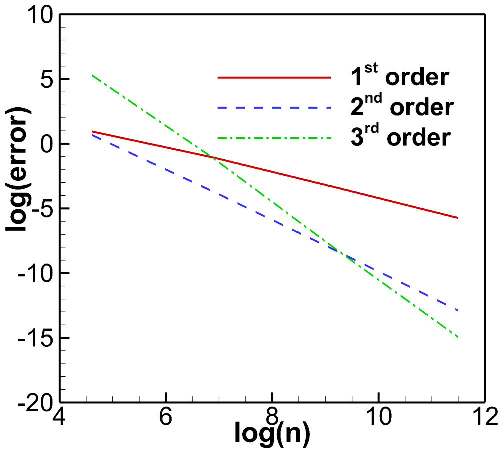


Figure 1.5: Convergence curve of different methods on derivative calculation

are going to study the advantage of higher-order schemes in a shock problem. The problem with shocks likely to have very high gradients. Theoretically, shocks are the location that has an infinite slope. Since we cannot handle such an infinite slope in differential models, we should spread shocks over several grid points. Shocks in the Navier-Stokes (NS) equation always have a finite thickness order of molecular distance due to the viscous term present in the equation. The Euler equation has no viscous term and admits a solution that has a step-like discontinuity. To stabilize the solver, we always add artificial dissipation. Non-physical viscosity may be introduced by including directional-based solvers or using Riemann solvers or adding a scalar dissipation term to the governing equation. Figure 1.6, shows the viscous Burgers equation solution with viscosity (ν) 0.1 and 0.01. The viscous burgers equation is

$$\frac{\partial u}{\partial t} + 0.5\frac{\partial u^2}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{1.10}$$

The figure shows that when viscosity is increased, the shock spreads over several grid points and spreads in magnitude. When solving the problem using a numerical scheme, the magnitude of the derivatives is reduced due to added dissipation. In this case, higher-order schemes may be appropriate. We will study the performance of higher and lower order schemes on the Euler equation using some standard initial conditions. The test problems considered here are Sod shock tube problem [5], right expansion and left strong

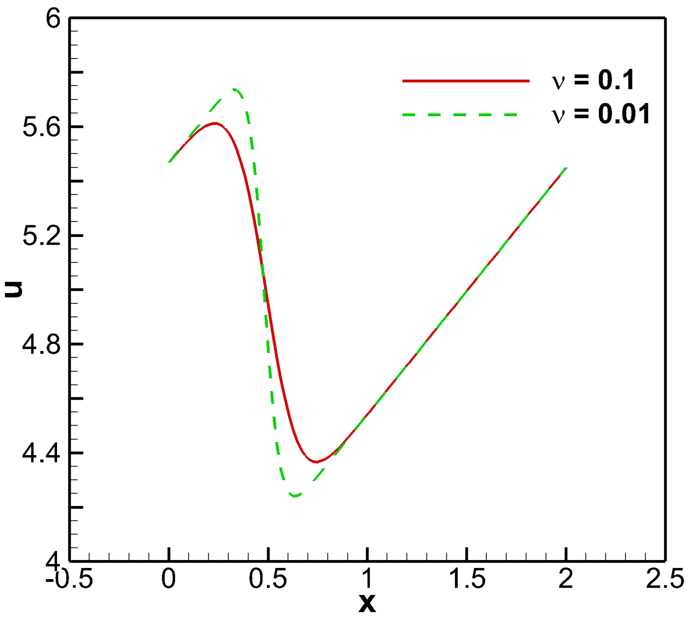


Figure 1.6: Solution of viscous Burgers equation with viscosity 0.1 and 0.01 at 0.1 s

shock (RELS) and Mach 3 test case [6]. Here, we intend to demonstrate the advantage of higher-order methods over lower-order methods. Numerical schemes, mathematics and solutions are illustrated in the following chapters.

Comparison of the results of the first-order scheme, the second-order scheme using minmod limiter, and WENO-JS scheme [7] evaluated on the Euler equation with three initial conditions. For a given root mean square (RMS) error, the time and grid points required to achieve a given error is tabulated in table 1.2. In this table, $\eta = \frac{Time_{scheme}}{Time_{WENO-JS}}$. That shows WENO-JS is approximately 7.2 times computationally economical than the second-order scheme for Sod test case. For a given RMS error, WENO-JS is at least seven times computationally more economical than second-order minmod limiter for the cases considered here. WENO-JS is 290 times more accurate than the first-order limiter for Mach 3 test case. In all the test case considerer here, higher-order schemes are much more accurate than the lower-order method. WENO-JS can theoretically give at least third-order accuracy in the problem involving shocks. However, it cannot show the third-order convergence when solving numerically in problems involving discontinuities, such as shocks, where Taylor series approximations are not valid.

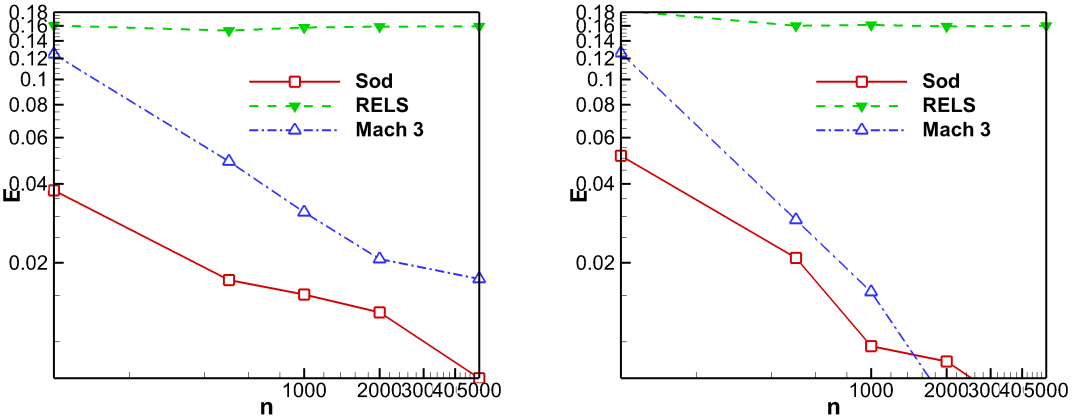
Table 1.2
Comparison of different order of accuracy schemes on different benchmark problems

Test case	Scheme	RMS error	Grid points	Time	η
Sod	WENO-JS	0.014266	400	3.21828	1
	2^{nd} order scheme	0.014705	900	23.2415	7.2217
	1^{st} order scheme	0.014543	2800	120.1137	37.3223
RELS	WENO-JS	0.157944	400	6.190975	1
	2^{nd} order scheme	0.156201	900	49.86687	8.0548
	1^{st} order scheme	0.156764	1800	82.72014	13.3614
Mach 3	WENO-JS	0.036704	390	6.97253	1
	2^{nd} order scheme	0.033434	900	71.91051	10.3134
	1^{st} order scheme	0.039399	8000	2021.416	289.9113

From the table 1.2, it is clear that the higher-order schemes are relatively more accurate than the lower-order schemes. The higher-order schemes generally have a higher rate of convergence than the lower-order scheme. Unfortunately, this may not be the reason for this extraordinary performance of higher-order schemes over the lower-order schemes. This is illustrated in figure 1.7. This figure shows the rate of convergence of the second-order scheme and WENO-JS for different test cases considered. As expected, they did not show the expected rate of convergence for the problem involving shocks. In most of the cases, they did not show linear convergence in the log-log plot of error versus the number of grid points.

The average line slope in figure 1.7 is tabulated in table 1.3. It is clear from the table that the convergence rate of the second-order scheme and the WENO-JS is less than one. Log-log computational time plot of the second-order method and WENO-JS vs grid point number is shown in Figure 1.8. In the log-log plot, the computational time increases linearly with the number of grid points, and the slope of the line is approximately two in all the test cases considered. From this relationship, we can approximately estimate the run-time of a particular problem at different grid points. In table 1.3, r_e is the average slope of the line in figure 1.7 and r_t is average slope of the line in figure 1.8.

Figure 1.9 shows the log-log plot of the number of iteration, time versus the number of grid points in a supersonic flow past a 10° wedge with an inflow Mach number 1.5. This also shows linearity in the log-log curve and the slope of the computational time curve is 1.3121. The computational time is directly proportional to the square of the number of



(a) Second-order scheme

(b) WENO-JS scheme

Figure 1.7: The log-log plot of RMS error *vs* number of grid points.

Table 1.3

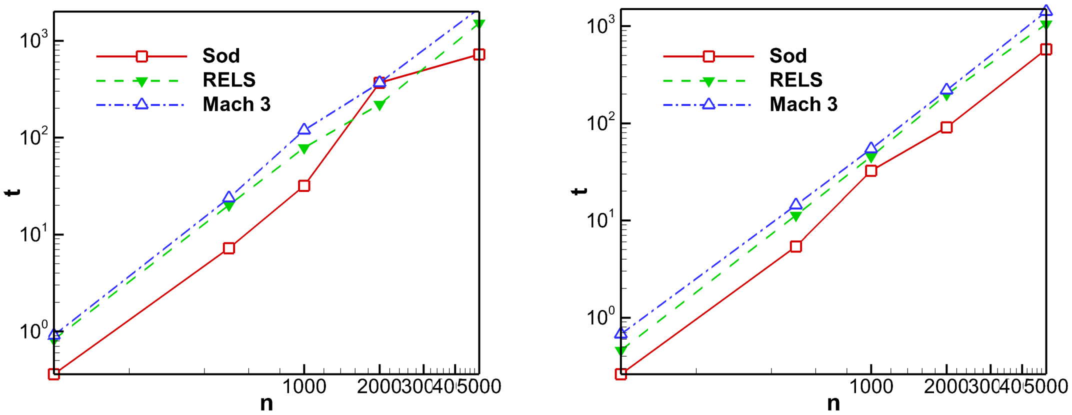
Average slope of the log-log plot in figure 1.7 and 1.8

Schemes	Test cases	r_t	r_e
2^{nd} order scheme	Sod	2.0668	-0.3822
	RELS	1.8861	0.0064
	Mach 3	1.9768	-0.5025
WENO-JS	Sod	1.9934	-0.6277
	RELS	1.9856	-0.021
	Mach 3	1.9667	-0.9184

grid points for shock tube problem and 1.3121 for supersonic flow past a wedge.

1.4.3 The advantage of well-stable explicit schemes

The Current-Friedrichs-Lewy number (CFL) is defined as $CFL = c\frac{\Delta t}{\Delta x}$. It is a non-dimensional number that defines the stability limits of the numerical scheme of the differential equation. Δx and Δt are grid size and time step; c is the characteristic velocity of the hyperbolic equation. CFL is not the only parameter that affects the stability limit; other parameters, such as the initial condition and boundary condition, also play a role in determining the stability of the numerical method. If we push the discretization stability limit for a given equation, it will significantly reduce the computational cost by allowing a larger time step. This will reduce the computational cost, but it will also increase the truncation error.



(a) Second-order scheme

(b) WENO-JS scheme

Figure 1.8: The log-log plot of computational time *vs* number of grid points.

Table 1.4
Variation of error and computational time for different CFL number for Sod test case using 2^{nd} order scheme.

CFL	Time	Error
0.1	31.78174	0.020098113
0.2	14.42127	0.02196497
0.4	6.326084	0.024517284
0.6	4.054642	0.025206254

Table 1.5
Variation of error and computational time for different CFL number for Sod test case using WENO-JS.

CFL	Time	Error
0.1	19.19856	0.014380815
0.2	11.98532	0.014371849
0.4	4.992381	0.014324188
0.8	1.951299	0.01445243
1	1.520889	0.024856074
1.1	1.344956	0.024839349

Table 1.4 shows the variation of RMS error at 0.1 s on Sod shock tube problem for different CFL number using 2^{nd} order scheme. From the table, we can see that increasing the CFL number increases the error and reduces the computational cost. The reduction in computational time is higher than the error, so it is beneficial to push the CFL problem limit. Similarly, table 1.5 shows the variation of RMS error at 0.1 s on Sod shock tube problem for different CFL numbers using WENO-JS. Here, too, the trend is similar to the 2^{nd} order scheme. The slope of the second-order scheme and the WENO-JS in figure 1.10 is -1.1419 and -1.1410 respectively.

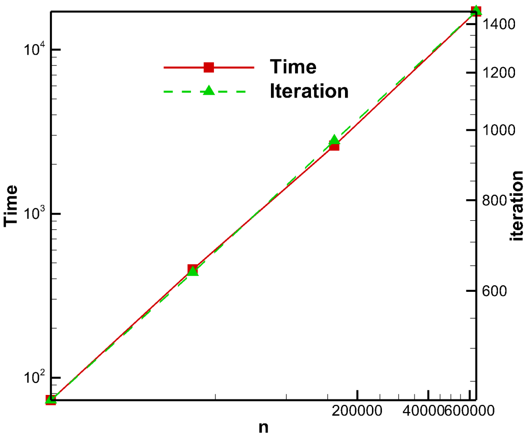


Figure 1.9: Log-log plot of time, iteration *vs* number of grid points for supersonic flow past a wedge

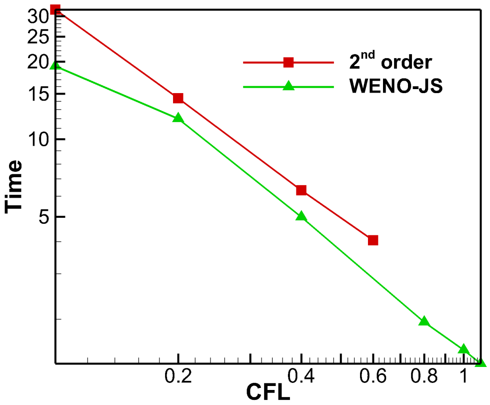


Figure 1.10: Log-log plot of CFL number versus computational time for Sod test case.

1.5 Some of the objectives of the schemes developed in this work are:

- The scheme should be higher-order
- It should be high-resolution.
- It should allow relativity large time-step and CFL number.
- It should be computationally economical.

2

Governing Equation and Discretization

Most of the physical phenomena are modelled using ordinary differential equations (ODEs) or partial differential equations (PDE). PDEs can be categorized as elliptical, parabolic or hyperbolic, or hybrid (mix) of these classes. In this PDE classification, hyperbolic PDEs have a wave-like solution. In this work, most simulations are performed for convection-dominated problems. The differential equation used in this work are the linear convection equation, convection-diffusion equation, inviscid Burgers equation, viscous Burgers equation and the Euler equation.

At high Mach numbers, the Euler equations are used as the preliminary method to obtain a solution to an aerodynamics problem. This is generally used in conditions where viscous effects are negligible and can be ignored. Although the computational cost of the Euler equation is much lower than that of the Navier-Stokes (NS) equation, it is comparatively more difficult to stabilize the Euler equation when using numerical methods. This is due to the lack of viscosity in the Euler equation. The NS calculations are computationally expensive, as boundary layer resolution requires smaller time steps.

2.1 Linear convection equation

Linear convection equation is the simplest hyperbolic equation which represents the transport of a property. The convection equation is given as

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

This equation propagates the initial condition with a given speed (c) without changing its shape.

For an initial condition

$$u(x, 0) = u_0(x)$$

The solution is

$$u(x, t) = u_0(x - ct)$$

This equation is highly sensitive to CFL number, grid size, initial condition and discretization. This equation is widely used examine a new method for improving the dissipation and dispersion error for numerical discretization.

2.2 Linear convection diffusion equation

The convection-diffusion equation is the simplest equation that describes both convection and a diffusion phenomenon. In general, the governing equation of several flows has both convection and diffusion terms. This equation is used to analyse the numerical methods used to model the two phenomena together. When a drop of dye is placed in a river, the movement of the ink in the direction of the movement of the water is modeled by convection term. Spreading or diffusion of dye in water is modelled using diffusion term. The linear convection-diffusion equation is has both the term in its so it is a ideal equation to understand simple transport phenomenon with diffusion. The standard linear convection diffusion equation is

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} - D\frac{\partial^2 u}{\partial x^2} = 0$$

2.3 Burgers equation

Burgers equation is a quasi-linear equation that was first introduced by Harry Bateman in 1915 [8]. The inviscid form of the Burgers equation has only the convection term; the viscous form has both convection and diffusion terms. The general Burgers equation is:

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = \nu\frac{\partial^2 u}{\partial x^2} \tag{2.1}$$

Inviscid Burgers equation can be solved using the Method of Characteristic (MOC) in some situations. If the characteristics of the equation intersect, shocks are produced, and classical equation procedures cannot be applied. Chandrasekhar [9] provides an explicit solution to the Burgers equation for linear initial conditions (IC). Viscous Burgers equation can be converted into a linear equation by using Cole-Hopf transformation [10]. The solution to this linear equation can be obtained and an inverse Cole-Hopf transformation can be done to obtain the solution for the original equation.

2.4 Euler equation

One-dimensional Euler equation in conservative differential form is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{2.2}$$

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix} \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho E + p) \end{bmatrix}$$

where ρ is density, u is velocity, p is pressure, e_t is specific total energy. $E = \frac{1}{\gamma-1}\frac{p}{\rho} + \frac{1}{2}u^2$ for calorically perfect gas. Solving a problem using conservative discretization is more accurate compared to non-conservative discretization in terms of obtaining exact shock speed based on Rankine-Hugoniot jump conditions.

2.4.1 Non-conservative form

Eq. 2.2 can be written as:

$$\frac{\partial \mathbf{U}}{\partial t} + \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)\frac{\partial \mathbf{U}}{\partial x} = 0 \tag{2.3}$$

The Jacobian matrix $\mathbf{A} = \left(\frac{\partial \mathbf{F}}{\partial \mathbf{U}}\right)$ is:

$$\mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 & 0 \\ (\gamma-3)\frac{q^2}{2} & (3-\gamma)u & (\gamma-1) \\ \left(\frac{\gamma-1}{2}u^2 - H\right)u & H + (1-\gamma)u^2 & \gamma u \end{bmatrix}$$

where $H = E + \frac{p}{\rho}$. Eq. 2.3 can be written as

$$\frac{\partial \mathbf{U}}{\partial t} + \mathbf{A}\frac{\partial \mathbf{U}}{\partial x} = 0 \tag{2.4}$$

Eq. 2.4 is not equivalent to linear convection equations because $\mathbf{A}$ is not a diagonal matrix. So we cannot assume that the conserved variable U is convected with the velocity $\mathbf{A}$. A can be diagonalized using Singular Value Decomposition (SVD). The left and right eigen vectors of A can be found using symbolic packages such as Matlab, Mathematica, etc.

SVD of Jacobian flux is $\mathbf{A} = \mathbf{R}\Lambda\mathbf{L}$. where,

$$\Lambda = \begin{bmatrix} u-c & 0 & 0 \\ 0 & u & 0 \\ 0 & 0 & u+c \end{bmatrix}$$

$$\mathbf{R} = \begin{bmatrix} 1 & 1 & 1 \\ u-c & u & u+c \\ H-uc & \frac{u^2}{2} & H+uc \end{bmatrix}$$

$$\mathbf{L} = \begin{bmatrix} \frac{1}{2}\left(\frac{\gamma-1}{2c^2}u^2 + \frac{u}{c}\right) & -\frac{1}{2}\left(\frac{\gamma-1}{c^2}u + \frac{1}{c}\right) & \frac{\gamma-1}{2c^2} \\ 1 - \frac{\gamma-1}{2c^2}u^2 & \frac{\gamma-1}{c^2}u & -\frac{\gamma-1}{c^2} \\ \frac{1}{2}\left(\frac{\gamma-1}{2c^2}u^2 - \frac{u}{c}\right) & -\frac{1}{2}\left(\frac{\gamma-1}{c^2}u - \frac{1}{c}\right) & \frac{\gamma-1}{2c^2} \end{bmatrix}$$

The solution method usually followed in high-speed flows (density-based algorithm) when implemented at low speeds, has convergence issues as one or more eigenvalues approaches zero. This problem can be overcome by preconditioning the Euler equation. Since this form of the Euler equation specifically has its eigenvalue and its eigenvector in the formulation, it can be modified. Also, matrix-based dissipation schemes use this formulation to change the scheme for a problem by modifying these values.

2.4.2 Primitive form

U matrix cannot explicitly provide primitive variable, so we need extra steps to calculate the primitive variables from **U**. Euler equation in the primitive variable is

$$\frac{\partial \mathbf{W}}{\partial t} + \mathbf{A}^w \frac{\partial \mathbf{W}}{\partial x} = 0$$

where,

$$\mathbf{W} = \begin{bmatrix} \rho \\ u \\ p \end{bmatrix} \mathbf{A}^w = \begin{bmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \rho c^2 & u \end{bmatrix}$$

$\mathbf{A}^w$ is co-efficient of primitive variable form of the Euler equation. If we diagonalize $\mathbf{A}^w$ we get, $\mathbf{A}^w = \mathbf{R}^w \Lambda \mathbf{L}^w$, where,

$$\Lambda = \begin{bmatrix} u-c & 0 & 0 \\ 0 & u & 0 \\ 0 & 0 & u+c \end{bmatrix}$$

$$\mathbf{R}^w = \begin{bmatrix} -\frac{\rho}{2c} & 1 & \frac{\rho}{2c} \\ \frac{1}{2} & 0 & \frac{1}{2} \\ -\frac{\rho c}{2} & 0 & \frac{\rho c}{2} \end{bmatrix} \mathbf{L}^w = \begin{bmatrix} 0 & 1 & -\frac{1}{\rho c} \\ 1 & 0 & -\frac{1}{c^2} \\ 0 & 1 & \frac{1}{\rho c} \end{bmatrix}$$

This form is used generally in the analysis of the Euler equation and Riemann solvers.

2.4.3 Characteristic form

It is reported that interpolation of the characteristic variable is relatively less oscillatory compared to primitive and conservative variables [11]. To convert primitive variable form of the Euler equation to characteristic form, we should multiply that by $\mathbf{L}^w$

$$\mathbf{L}^w \frac{\partial \mathbf{W}}{\partial t} + \mathbf{L}^w \mathbf{A}^w \frac{\partial \mathbf{W}}{\partial x} = 0$$

simplifying yields,

$$\frac{\partial \mathbf{W}^c}{\partial t} + \mathbf{\Lambda} \frac{\partial \mathbf{W}^c}{\partial x} = 0 \tag{2.5}$$

where,

$$\partial \mathbf{W}^c = \begin{bmatrix} \partial u - \frac{1}{\rho c} \partial p \\ \partial \rho - \frac{1}{c^2} \partial p \\ \partial u + \frac{1}{\rho c} \partial p \end{bmatrix}$$

Eq. 2.5 is the characteristic form of the Euler equation where all equations are decoupled; this is equivalent to three nonlinear advection equations. For adiabatic flows, the characteristic variables can be integrated and are simplified to the form

$$\partial \mathbf{W}^c = \begin{bmatrix} \partial u - \frac{1}{\rho c} \partial p \\ \partial \rho - \frac{1}{c^2} \partial p \\ \partial u + \frac{1}{\rho c} \partial p \end{bmatrix} = \begin{bmatrix} \partial \left(u - \frac{2}{\gamma - 1} c \right) \\ \partial s \\ \partial \left(u + \frac{2}{\gamma - 1} c \right) \end{bmatrix}$$

Integral quantities are Riemann invariants. They are useful in solving the Euler equation analytically for some special cases. The characteristic form of Euler equation is

$$\frac{\partial}{\partial t} \left(u - \frac{2}{\gamma - 1} c \right) + (u - c) \frac{\partial}{\partial x} \left(u - \frac{2}{\gamma - 1} c \right) = 0 \tag{2.6}$$

$$\frac{\partial s}{\partial t} + u \frac{\partial s}{\partial t} = 0$$

$$\frac{\partial}{\partial t} \left(u + \frac{2}{\gamma - 1} c \right) + (u + c) \frac{\partial}{\partial x} \left(u + \frac{2}{\gamma - 1} c \right) = 0$$

These equations are also used to implement characteristic boundary condition.

2.4.4 Parametric form

Most of the discretization is carried out in the conservative form. However, the conservative form does not explicitly provide primitive variables. The parametric form is useful for the conversion of variables and the linearization of the Euler equation. The parameter is **Z**

$$\mathbf{Z} = \sqrt{\rho}\begin{bmatrix} 1 \\ u \\ H \end{bmatrix} = \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix}$$

Conservative variable and flux are given as

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix} = \begin{bmatrix} z_1^2 \\ z_1 z_2 \\ \frac{z_1 z_3}{\gamma} + \frac{\gamma-1}{\gamma}\frac{z_2^2}{2} \end{bmatrix}; \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho E + p) \end{bmatrix} = \begin{bmatrix} z_1 z_2 \\ z_2^2 + p \\ z_2 z_3 \end{bmatrix}$$

and the closure equation for a calorically perfect gas is $p = \frac{\gamma-1}{\gamma}\left(z_1 z_3 - \frac{z_2^2}{2}\right)$

We have studied various differential forms of the Euler equation. In the following chapters, we will examine different discretization procedures to solve the Euler equation.

2.5 Why high-speed solvers are unique and differ from the low speed solvers?

The general form of conservative equation is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial \mathbf{x}} = 0 \tag{2.7}$$

On discretizing the above using the conservative discretization, it becomes

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \frac{\mathbf{F_{i+0.5}}(\mathbf{U}) - \mathbf{F_{i-0.5}}(\mathbf{U})}{\Delta x} = 0 \tag{2.8}$$

Eq. 2.7 is a partial differential equation and can be converted to an ordinary differential equation using method of line (MOL). Then the eq. 2.8 becomes

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \frac{\mathbf{F_{i+0.5}}(\mathbf{U}) - \mathbf{F_{i-0.5}}(\mathbf{U})}{\Delta x} = 0 \tag{2.9}$$

Since the Euler equation is difficult to analyze and study, we could use simple equations such as the Burgers equation or the simple hyperbolic linear convection equation. We may gain some insight from simple convection equations or Burgers equations, but *we should bear in mind that all the aspects we infer from the simple model might not be valid for the Euler equation.*

We need $\mathbf{F}(\mathbf{U})$ at $(i + 0.5)$ and $(i - 0.5)$ to solve 2.9. This value of $\mathbf{F}(\mathbf{U})$ at $(i + 0.5)$ and $(i - 0.5)$ cell-interface value is obtained by interpolating cell-average value. Higher-order schemes are accomplished depending on the procedure in which the interpolation is carried out. Since high-speed flow problems involve steep gradients in its solution, the numerical schemes are prone to the Runge phenomenon. Oscillations in the solution can therefore occur. We need some special tools to deal with this issue. This problem is not common in low-speed solvers because the solution is smooth in most cases.

In order to calculate face value which is defined in between two cell centres, we need one interpolation function. The second order interpolation function is

$$\mathbf{U}_{i+0.5} = \mathbf{U}_i + \frac{1}{2}\frac{d\mathbf{U}}{dx} \qquad \mathcal{O}(\Delta x^2) \tag{2.10}$$

The slope $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}$ can be calculated in different ways.

$$\frac{\mathrm{d}\mathbf{U^c}}{\mathrm{d}x} = \frac{\mathbf{U_{i+1}} - \mathbf{U_{i-1}}}{2\Delta x} \qquad \mathcal{O}(\Delta x^2) \tag{2.11a}$$

$$\frac{\mathrm{d}\mathbf{U^p}}{\mathrm{d}x} = \frac{\mathbf{U_{i+1}} - \mathbf{U_i}}{\Delta x} \qquad \mathcal{O}(\Delta x) \tag{2.11b}$$

$$\frac{\mathrm{d}\mathbf{U^m}}{\mathrm{d}x} = \frac{\mathbf{U_i} - \mathbf{U_{i-1}}}{\Delta x} \qquad \mathcal{O}(\Delta x) \tag{2.11c}$$

Taylor series may not perform well on the non-smooth solutions, so we will evaluate the performance of the interpolation function on a non-smooth solution. Figure 2.1a shows the interpolation of the unit step function using different slope estimates from eq. 2.11. Of the various slope estimators, atleast one slope estimator will work well in the entire region. This principle is important for high-speed flow solvers, and this is achieved by non-linear switches such as limiter or WENO. More about limiters and WENO schemes can be found in chapter 5 and chapter 6. Once we get $(i + 0.5)$ and $(i - 0.5)$ states, the derivative cannot be determined directly as oscillations may occur.

In Figure 2.1a, the solution $y^m_{i+0.5}$ shows oscillations because the slope of the step function at the upper arm is positive, but the slope suddenly becomes zero in the analytical solution when $x > 0$. Since $y^m_{i+0.5}$ is calculated using i and $i - 1$, it has a positive slope and the first value after $x > 0$ is overestimated. Therefore, we observe an overshoot in the figure 2.1a. Since $y^p_{i+0.5}$ is calculated from i and $i + 1$, zero slope is calculated exactly after $x > 0$, and there are no oscillations. Slope calculated using $y^m_{i+0.5}$ generates a problem and can be managed by manipulating the slope. If the smallest magnitude is used between $\frac{\mathbf{U_i - U_{i-1}}}{\Delta x}$ and $\frac{\mathbf{U_{i+1} - U_i}}{\Delta x}$, the solution will not oscillate. This principle is used in the minmod limiter, which fits well for most of the flow problems. It is note worthy that central scheme ($y^c_{i+0.5}$) produced oscillations at all the sharp corners. Another way to eliminate this oscillation is by the use of the ENO/WENO scheme. We can study the concept behind it by using a step function test case.

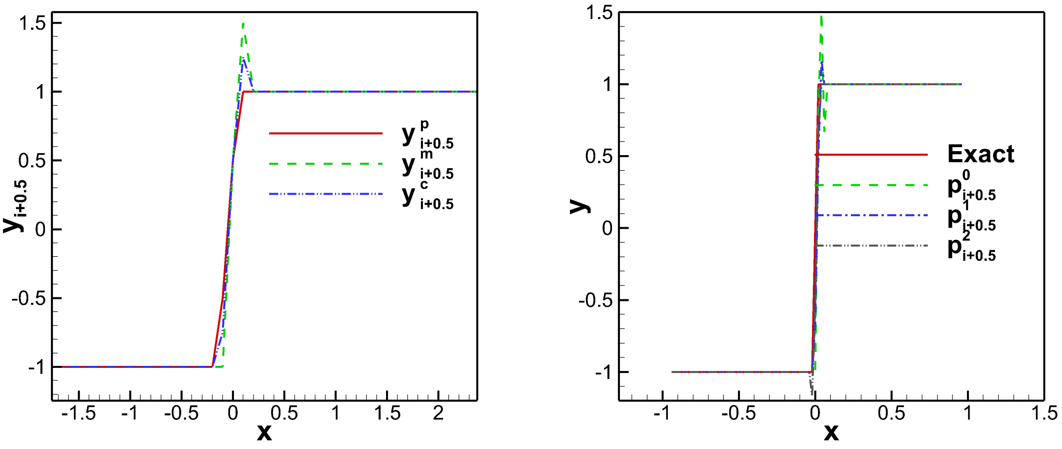
Figure 2.1b demonstrates the interpolation of the step function using third-order polynomials. These polynomials are

$$p^0_{i+0.5} = \frac{1}{6}(-\bar{u}_{i-1} + 5\bar{u}_i + 2\bar{u}_{i+1})$$

$$p^1_{i+0.5} = \frac{1}{6}(2\bar{u}_i + 5\bar{u}_{i+1} - \bar{u}_{i+2})$$

$$p^2_{i+0.5} = \frac{1}{6}(2\bar{u}_{i+3} - 7\bar{u}_{i+2} + 11\bar{u}_{i+1})$$

In Figure 2.1b, the left side of $p^0_{i+0.5}$ does not oscillate and the right side of $p^2_{i+0.5}$ does not oscillate. This concept is used in the Essentially Non-Oscillatory Scheme (ENO) to obtain a monotone scheme. Although not all of the polynomials used in the ENO scheme



(a) Using different slopes

(b) Using third order scheme

Figure 2.1: Interpolation on a step function

are monotone, the resulting scheme is monotone. This observation contradicts the order barrier theorem and the monotone region in the Sweby diagram. Order barrier theorem states that all schemes higher than the first order are non-monotone. Third-order schemes developed using the ENO scheme will not fall under the Sweby diagram.

Forward in time and central in space (FTCD2) scheme is given as

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \frac{\mathbf{F}(\mathbf{U})_{i+1} - \mathbf{F}(\mathbf{U})_{i-1}}{\Delta x} = 0 \tag{2.13}$$

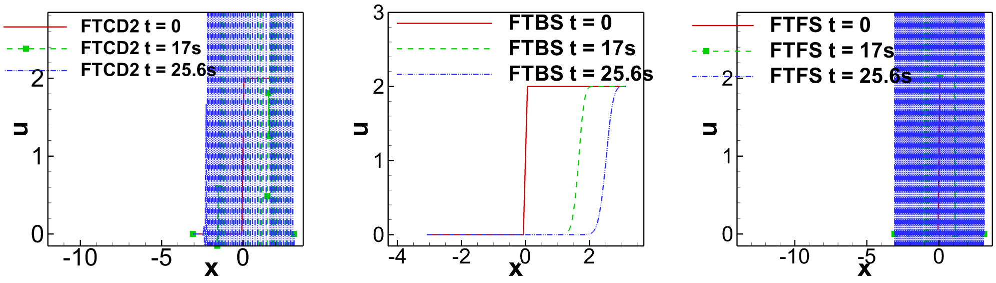
Forward in time and backward in space (FTBS) scheme is given as

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \frac{\mathbf{F}(\mathbf{U})_{i} - \mathbf{F}(\mathbf{U})_{i-1}}{\Delta x} = 0 \tag{2.14}$$

Forward in time and forward in space (FTFS) scheme is given as

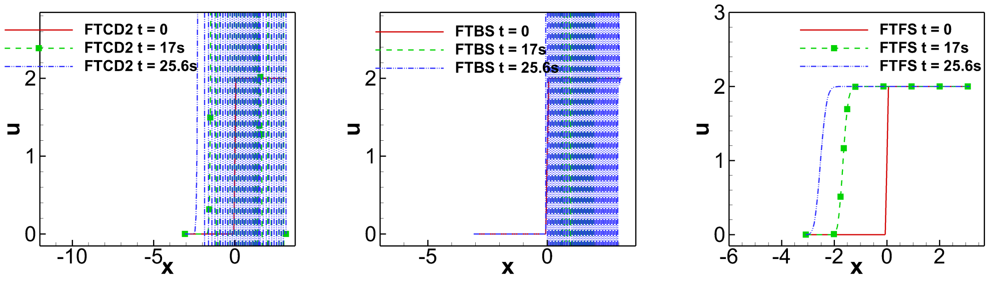
$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \frac{\mathbf{F}(\mathbf{U})_{i+1} - \mathbf{F}(\mathbf{U})_{i}}{\Delta x} = 0 \tag{2.15}$$

Figure 2.2 and figure 2.3 show the solution of the advection equation moving from left to right and from right to left, respectively. The advection equation is retained by $\mathbf{F}(\mathbf{U}) = cu$ in eq. 2.9, where the advection velocity is c. The initial condition used is $u(x, 0) = \tanh(10000x) + 1$ with $c = \pm 0.1$. In Figure 2.2, FTBS is able to give a



(a) Using FTCD2 scheme **(b)** Using FTBS scheme **(c)** Using FTFS scheme

Figure 2.2: The solution of advection equation when moving left to right



(a) Using FTCD2 scheme **(b)** Using FTBS scheme **(c)** Using FTFS scheme

Figure 2.3: The solution of advection equation when moving right to left

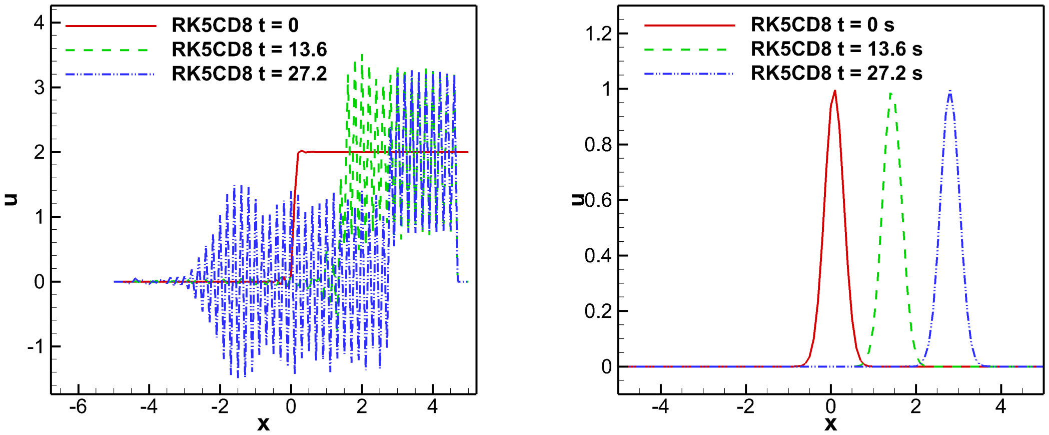
reasonable result and other schemes are unstable, but in Figure 2.3, FTFS has produced a good result. The conclusion that we can get from this is when the wave moves from left to right FTBS should be used, and when the wave moves from right to left, FTFS should be used. This is the basis of the Godunov scheme. In the case of the advection equation, the directional characteristics are simple. In the case of non-linear coupled problems, the directional characteristics are not straight forward, and the Singular Value Decomposition can be used to find directional characteristics. In Lax Friedrichs scheme, it is taken care by the following expression

$$\mathbf{F}(\mathbf{U}) = \mathbf{F}^{+}(\mathbf{U}) + \mathbf{F}^{-}(\mathbf{U})$$

$$\frac{\mathrm{d}\mathbf{F}^{+}(\mathbf{U})}{\mathrm{d}\mathbf{U}} \geq 0 \qquad \frac{\mathrm{d}\mathbf{F}^{-}(\mathbf{U})}{\mathrm{d}\mathbf{U}} \leq 0$$

Then the Lax-Friedrich splitting is

$$\mathbf{F}^{\pm} = \frac{1}{2}\left[\mathbf{F}(\mathbf{U}) \pm \alpha\mathbf{U}\right] \tag{2.17a}$$



(a) RK5CD8 with step-like IC

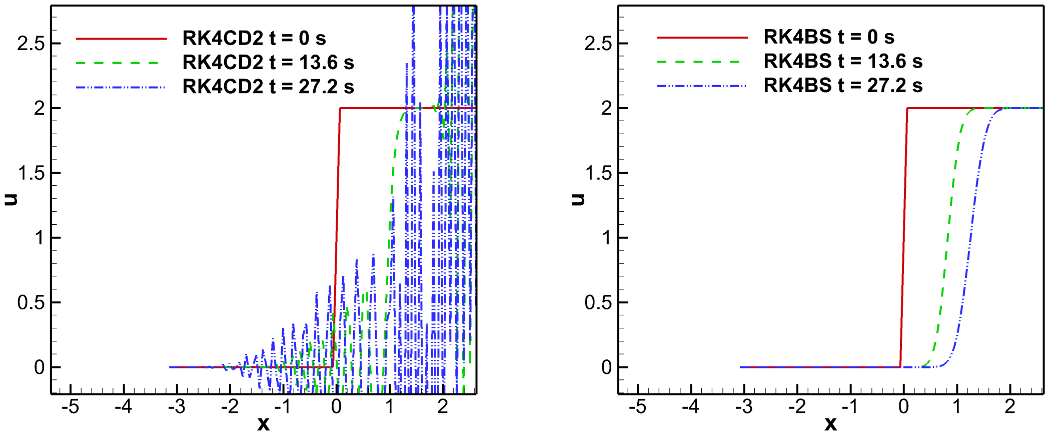
(b) RK5CD8 with Gaussian IC

Figure 2.4: The solution of advection equation using RK5CD8

$$\alpha = \max_{\mathbf{\Lambda}} |\mathbf{F}'(\mathbf{U})| \tag{2.17b}$$

Different Riemann solvers use a different procedure to calculate wave speed and provide bias as described in 2.6.2. Let us try to solve the linear convection equation by using a higher-order time integration. The most popular time integration scheme is the RK method. Figure 2.4 shows the solution of the linear convection equation using RK5CD8. We have already noted that the directional solvers are better than the central difference scheme. In this case, the central difference scheme is oscillating but not blowing up while using RK5CD8 (figure 2.4b). RK5CD8 does not show oscillations when using Gaussian-like IC shown in Figure 2.4b. Explanations such as “FTBS are stable because the flow is from left to right and the scheme obeys physics as it takes information from left to right” may not be the exact explanation for this. There could be some physics beyond this which can be explained by space-time approximate dispersion relation preserving analysis.

We shall study the performance of RK schemes based on directional discretization and non-directional discretization. Figure 2.5 shows the result of a linear convection scheme with directional based discretion (RK4BS) and non-directional based discretion (RK4CD2). Directional based discretization has produced a better result than non-directional based discretization. Generally, directional based discretization works and is relatively more stable. The direction-based solution procedure is a sufficient condition, but not always a necessary condition for the solution of the hyperbolic equation.



(a) RK4CD2 with step-like IC

(b) RK4BS with step-like IC

Figure 2.5: The solution of advection equation using RK4

We have already observed that the central difference scheme gives oscillations to the step function in Figure 2.1a, so we will explore a few measures to address this.

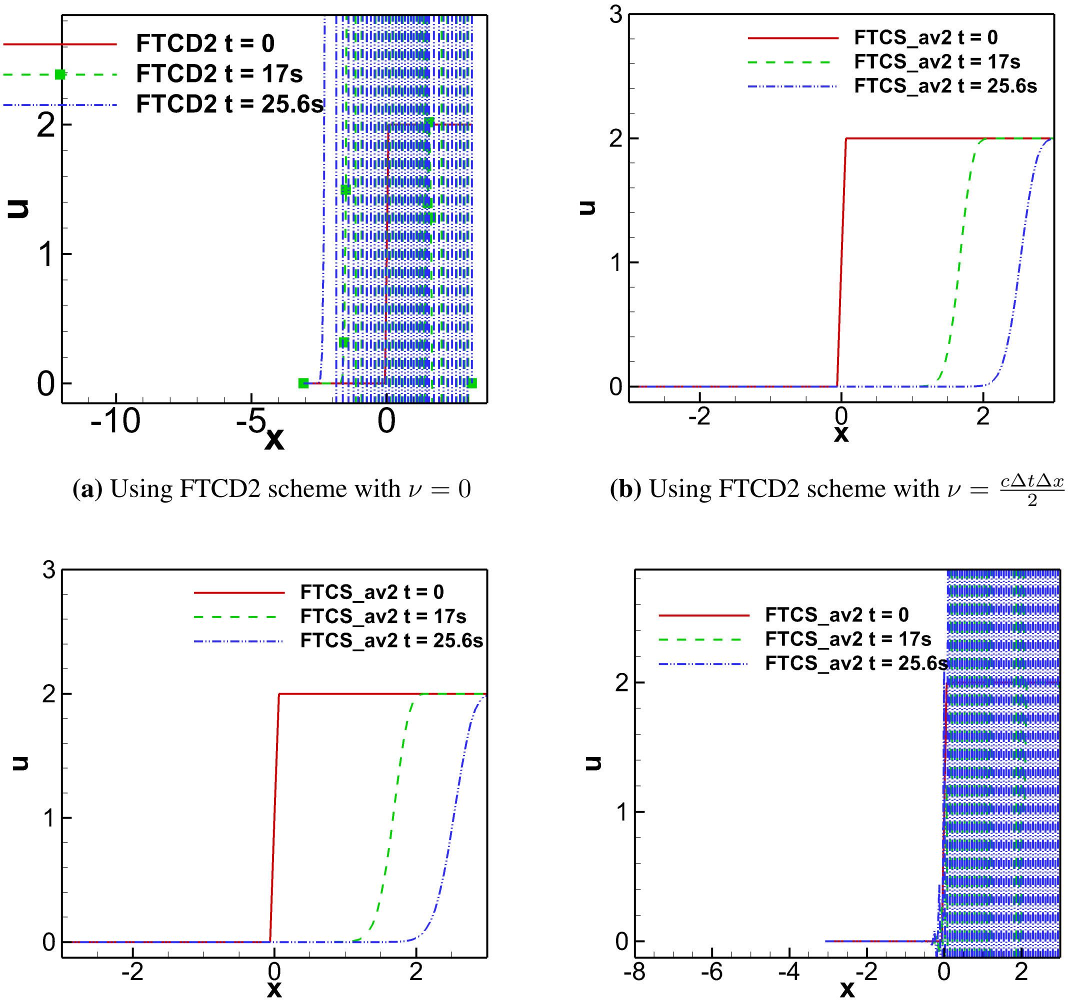
FTCD2 scheme with second-order artificial viscosity can be written as

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} + \frac{\mathbf{F}(\mathbf{U})_{i+1} - \mathbf{F}(\mathbf{U})_{i-1}}{\Delta x} - \nu \frac{\mathbf{F}(\mathbf{U})_{i+1} - 2\mathbf{F}(\mathbf{U})_{i} + \mathbf{F}(\mathbf{U})_{i-1}}{\Delta x^2} = 0 \qquad (2.18)$$

Eq. 2.18 becomes FTBS when $\nu = \frac{c\Delta t\Delta x}{2}$ for advection equation. Figure 2.6 shows the solution of the linear advection equation using FTCD2 with added artificial viscosity. This scheme is only consistent, if $\nu = \frac{c\Delta t\Delta x}{2}$, otherwise it is not even first-order accurate. FTCD2 with artificial viscosity $\nu = 1.1\frac{c\Delta t\Delta x}{2}$ can provide a reasonable solution, even if it is inconsistent. From this we can observe that adding viscosity can stabilize the solver, but only to a threshold value. Adding artificial viscosity beyond this threshold value may lead to instability. The instability in Figure 2.6d can be explained empirically by the *positivity* principle. We refer to [12] for more details. The condition for stability is:

$$CFL^2 \leq \frac{2\nu}{\Delta x^2} \leq 1 \qquad (2.19)$$

Adding artificial viscosity is a powerful and computationally effective way to stabilize the Euler equation. The most popular artificial viscosity based solver (scalar viscosity) is the JST scheme. This section focuses on the various reconstruction algorithms, the



(**a**) Using FTCD2 scheme with $\nu = 0$

(**b**) Using FTCD2 scheme with $\nu = \frac{c\Delta t\Delta x}{2}$

(**c**) Using FTCD2 scheme with $\nu = 1.1\frac{c\Delta t\Delta x}{2}$

(**d**) Using FTCD2 scheme with $\nu = 1.5\frac{c\Delta t\Delta x}{2}$

Figure 2.6: The solution of advection equation when moving right to left using FTCS2 with artificial viscosity

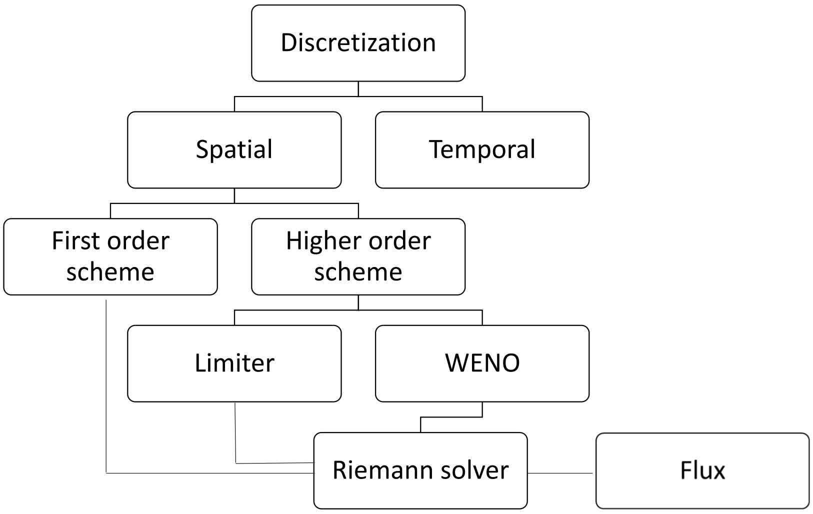


Figure 2.7: Discretization steps in high speed flows

Riemann solver, and the scalar dissipation used in the Euler equation, using a simple test case based on the advection equation. We can draw some inference from these simple equations, but we should not expect all the inferences we obtain from the linear equation to apply to the Euler equation.

2.6 Discretization of the Euler equation

Conservative discretization can accurately predict shock speed, so it is used in this work. Discretization procedures for structured and unstructured mesh are presented here. The discretization step has two main components – time discretization and space discretization.

2.6.1 One-dimensional discretization of Euler equation

Formulating the integral form of the Euler equation by volume averaging in the cell $I_i \equiv [i - 0.5, i + 0.5]$ and using the method of line leads to,

$$\frac{\mathrm{d}\bar{\mathbf{U}}_i}{\mathrm{d}t} = -\frac{1}{\Delta x}(\mathbf{F}_{i+0.5} - \mathbf{F}_{i-0.5}) \tag{2.20}$$

where $\mathbf{F}_{i+0.5}$ is numerical flux at the cell face computed from $\mathbf{F}_{i+0.5} = \mathbf{F}(\mathbf{U}^R_{i+0.5}, \mathbf{U}^L_{i+0.5})$ and $\mathbf{U} \in \mathbb{R}$. The second-order numerical flux can be achieved by evaluating $\mathbf{U}^R_{i+0.5}$, $\mathbf{U}^L_{i+0.5}$ given in eq. 2.21

$$\mathbf{U}^L_{i+0.5} = \mathbf{U}_i + \frac{\Delta x}{2}\mathrm{d}\mathbf{U}(\mathbf{U}^+, \mathbf{U}^-) \qquad \mathbf{U}^R_{i+0.5} = \mathbf{U}_{i+1} - \frac{\Delta x}{2}\mathrm{d}\mathbf{U}(\mathbf{U}^+, \mathbf{U}^-) \tag{2.21}$$

Unlike incompressible flow, we cannot directly calculate $\mathrm{d}\mathbf{F}$ or $\mathrm{d}\mathbf{U}$ due to the presence of shocks in the compressible flow. Limiting the value of $\mathrm{d}\mathbf{U}$ using a slope-limiter should be carried out, otherwise non-physical oscillations would appear in the solution. Non-linear reconstruction procedures such as Essentially Non-Oscillatory Scheme (ENO) or Weighted Essentially Non-Oscillatory Scheme (WENO) may also prevent oscillations. More details on WENO and limiters are presented in chapter 5 and chapter 6.

The standard conservative framework is based on the central discretization, so it may not always be suitable for convection-dominated problems. The Euler equation is purely convection-dominated, requiring a characteristic direction-dependent numerical scheme or artificial viscosity for the stability. Once the left and the right states are calculated, the values are transferred to the Riemann solver. The Riemann solver adds the necessary artificial dissipation and makes the solution relatively smooth so that the flux gradient can be calculated. In this work, the majority of numerical results are obtained using a Riemann solver. Using artificial viscosity, we can obtain a solution without the use of a Riemann solver. Jameson-Schmidt-Turkel Scheme [13] is one of the schemes based on artificial viscosity.

2.6.2 Riemann solvers

Riemann solvers are generally used to find the solution of the discontinuous initial condition. The exact Riemann solver is quite expensive so we have used approximate Riemann solver. Some of the approximate Riemann solvers used here are Roe scheme, Lax-Friedrichs scheme, Rusanov Flux, HLLC scheme and AUSM scheme. Though the integral form is used, we still calculate numerical derivatives across the shocks where the derivative does not exist. In some of the integral forms, there are no mathematical derivative terms but the final form has the numerical derivative term.

One-dimensional general form of the governing equation is

$$\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{F} = 0$$

Formulating integral form of the equation by volume averaging leads to

$$\int_v \left(\frac{\partial \mathbf{U}}{\partial t} + \boldsymbol{\nabla} \cdot \mathbf{F} \right) dv = 0$$

Using Green-Gauss theorem

$$\int_v \frac{\partial \mathbf{U}}{\partial t} dv + \oint_s \mathbf{F}.nds = 0$$

For one-dimensional case, with cell interfaces at $i - 0.5$ and $i + 0.5$ and length Δx

$$\Delta x \frac{\mathrm{d}\bar{\mathbf{U}}}{\mathrm{d}t} + (\mathbf{F}_{i+0.5} - \mathbf{F}_{i-0.5}) = 0 \tag{2.22}$$

Eq. 2.22 has no spatial derivative term. It seems like a good formulation because we do not need to worry about the spatial term's differentiability.

On rearranging, one gets

$$\frac{\mathrm{d}\bar{\mathbf{U}}_i}{\mathrm{d}t} = -\frac{1}{\Delta x}(\mathbf{F}_{i+0.5} - \mathbf{F}_{i-0.5}) \tag{2.23}$$

The right-hand side of eq. 2.23 is similar to second-order central difference term. Although eq. 2.23 looks like a second-order scheme, based on the way we reconstruct $F_{i+0.5}$ and $F_{i-0.5}$, we can achieve higher-order discretization. In eq. 2.22, we eliminated mathematical derivative term, but in eq. 2.23, we ended up in numerical derivative term. Note that the numerical derivative is vulnerable to numeric dissipation and dispersion errors.

Here, we are trying to calculate the numerical derivative across the shock, so we need smooth data and/or a fine grid. If the solution is not smooth, the Taylor series cannot be used. If the Taylor series cannot give a good approximation, most of the higher-order numerical schemes may fail. To provide smoothness to the solution, we may add dissipation. This dissipation can be added manually or by discretizing schemes based on directional information or by both ways.

Some of the Riemann solvers add dissipation in both ways so that they can provide a better solution than scalar dissipation for a given grid size. Please note that scalar dissipation schemes are relatively simple and are computationally economical for a given grid than for matrix-based dissipation schemes. As this work is dedicated to studying higher-order high-resolution schemes, only a basic overview of Riemann solvers is presented, and more importance is given to non-linear switches. The main objective of the Riemann solver is to add dissipation based on the magnitude of discontinuity. The objective of reconstruction schemes such as limiter, WENO etc. is to provide non-oscillatory input to the Riemann solver. Here, we shall study the role of Riemann solvers using Sod shock tube problem.

1-D Euler equation is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{2.24}$$

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho e_t \end{bmatrix} \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho e_t + p) \end{bmatrix}$$

Initial condition for Sod shock tube problem is:

$$(\rho, u, p) = \begin{cases} (1, 0, 1) & x \leq 0 \\ (0.125, 0, 0.1) & x > 0 \end{cases}$$

FVM discretization gives

$$\frac{\mathrm{d}\bar{\mathbf{U}}_i}{\mathrm{d}t} = -\frac{1}{\Delta x}(\mathbf{F}_{i+0.5} - \mathbf{F}_{i-0.5}) \tag{2.25}$$

We used the WENO scheme to calculate $\mathbf{F}_{i+0.5}$ and $\mathbf{F}_{i-0.5}$. The WENO algorithm is presented in section 6.

The initial velocity condition is zero, so the first and third rows of the flux matrix are zero in the entire domain. Because of the zero value in the first and third rows of the flux matrix, the derivative of the flux ($\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}x}$) is also zero in the first and third rows. So the first and the third rows will not change in the second time step, since the flux gradient at that location is zero. The flow is driven by a disturbance caused by the pressure difference of p in the second row of the flux vector. We know that these disturbances propagate at least at the speed of sound.

We can observe that the information propagation velocity in the numerical scheme is not matching with the physics. Over the time, the numerical scheme may attain the information propagating velocity of the actual equation but the initial transition should be improved. When we do small analysis, we can note that time derivative matrix $\mathbf{U}$ is not zero. We can make a relationship between $\mathbf{U}$ and $\mathbf{F}$ by linearization that is $\mathbf{F} = \Lambda\mathbf{U}$, where Λ is a characteristic speed. There is no exact and straightforward relation for Λ relating to $\mathbf{U}$ and $\mathbf{F}$. This could be one reason why there are many formulations for estimating the sound speed in different Riemann solvers, which is one of the most challenging tasks.

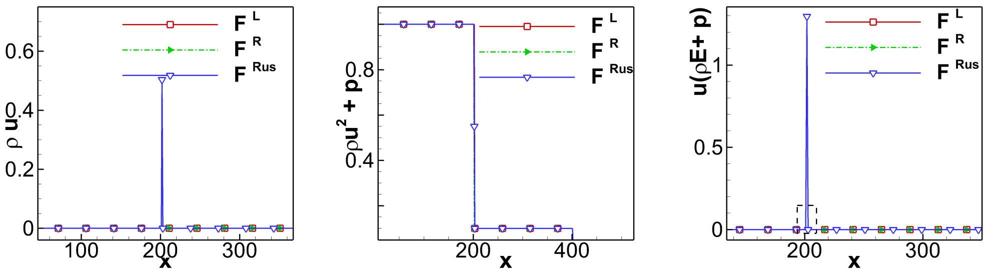
Since $\frac{\mathrm{d}\mathbf{F}}{\mathrm{d}x}$ is determined from a physical flux that has a mismatch in the propagation velocity, it needs to be corrected. We observed that $\mathbf{F} = \Lambda\mathbf{U}$ does not always have zero in the first and the third rows based on the value of Λ. For the better prediction of the information propagation velocity, this term ($\Lambda\mathbf{U}$) can be included in the physical flux. The Lax-Friedrich splitting is

$$\mathbf{F}^{\pm} = \frac{1}{2}\left[\mathbf{F}(\mathbf{U}) \pm \alpha\mathbf{U}\right] \tag{2.26a}$$

$$\alpha = \max_{\mathbf{\Lambda}} |\mathbf{F}'(\mathbf{U})| \tag{2.26b}$$

Because $\Lambda\mathbf{U}$ is present in the Riemann flux, the first and third rows of the Riemann flux are not zero. The additional term added to the physical flux corresponds to the physical flux's dimension (MLT-mass, length, time). Changing $\Lambda\mathbf{U}$ is not the only way to fix this issue, we can make many Riemann solvers with other variables related to flux.

The Riemann solvers diffuse the physical flux. In some conditions, the magnitude



(a) First row of flux (ρu) **(b)** Second row of flux ($\rho u^2 + p$) **(c)** Third-row ($u(\rho e_t + p)$)

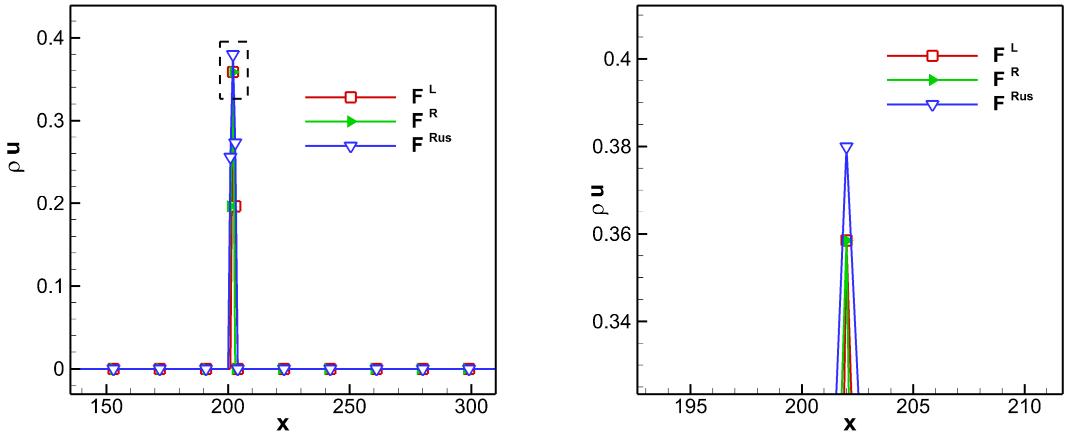
Figure 2.8: Flux of Euler equation using Sod shock tube problem at approximately $T = 0.0017$ s

of the physical flux increases. This is required because $-\frac{1}{\Delta x}(\mathbf{F}_{i+0.5} - \mathbf{F}_{i-0.5})$ is generally the first-order (theoretically) when the Riemann flux is different from the physical flux. In non-smooth regions, Riemann flux is different from physical flux when higher-order terms lose their importance in parabolic reconstruction procedures. The term $-\frac{1}{\Delta x}(\mathbf{F}_{i+0.5} - \mathbf{F}_{i-0.5})$ is first-order accurate even on the moderate non-smooth regions. First-order schemes are diffusive; we can improve the result by adding anti-diffusion terms. Some of the Riemann solvers increase the magnitude of the physical flux for better resolution in some regions.

Figure 2.8 displays the Euler equation flux after the first time-step in the Sod shock tube problem at flow time $t = 0.0017$s. It is obvious from the figure 2.8a and the figure 2.8c that the Riemann solver flux calculated using Rusanov's method is different from the physical flux. We have already studied that the spike is caused by the term $\Lambda \mathrm{U}$, which makes the speed of information propagation more sophisticated. Figure 2.9 shows the flux of the Euler equation with the initial condition of the Sod shock tube problem. It is clear from Figure 2.9b that Riemann solver may not always reduce the amplitude of the signal. So we can conclude that all Riemann solvers have diffusion and/or anti-diffusion terms. Good Riemann solver should avoid oscillations at any cost of the solution. Although F^R is oscillatory in Figure 2.10, the output of the Riemann solver is not oscillatory.

Roe scheme

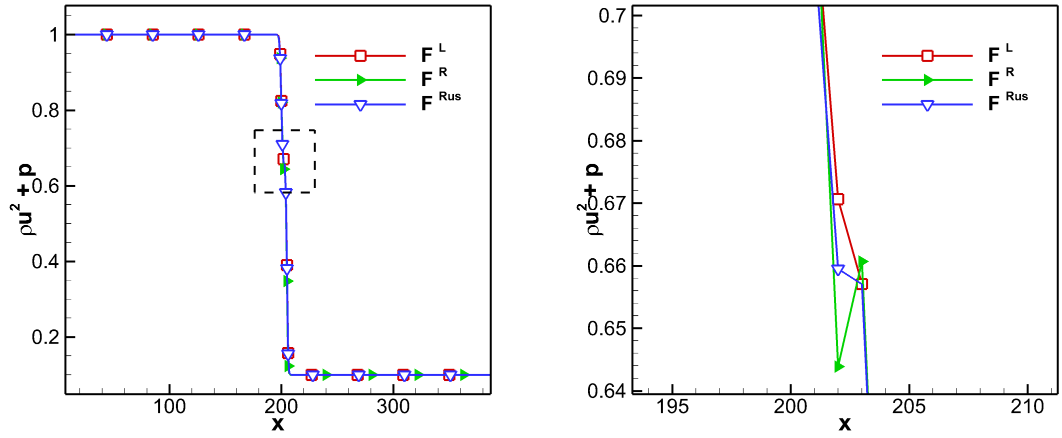
Roe scheme [14] is based on the Godunov scheme. It gives an approximate flux at the interface. Classical Roe does not satisfy entropy condition so Harten entropy fix [15]



(a) Second row of the flux $(\rho u^2 + p)$

(b) Second row of the flux $(\rho u^2 + p)$

Figure 2.9: Flux of Euler equation with Sod IC



(a) Second row of the flux $(\rho u^2 + p)$

(b) Second row of the flux $(\rho u^2 + p)$

Figure 2.10: Flux of Euler equation with Sod IC

based on jump present in the conserved variable is employed.

The general form of Riemann solver is,

$$\mathbf{F_q} = \frac{1}{2}\left[\mathbf{F}(\mathbf{U}^R) + \mathbf{F}(\mathbf{U}^L) + \tilde{A}_n\left(\mathbf{U}^R - \mathbf{U}^L\right)\right] \tag{2.27}$$

where $\tilde{A}_n$ is Jacobian of Roe-averaged quantities. The Roe-averaged primitive variables are

$$\tilde{\rho} = \sqrt{\rho^L\rho^R}$$

$$\tilde{u} = \frac{\sqrt{h^L}u^L + \sqrt{h^R}u^R}{\sqrt{h^R} + \sqrt{h^L}}$$

$$\tilde{v} = \frac{\sqrt{h^L}v^L + \sqrt{h^R}v^R}{\sqrt{h^R} + \sqrt{h^L}}$$

$$\tilde{c} = \sqrt{(\gamma - 1)\left(H - \frac{q^2}{2}\right)}$$

Superscript ^{L} and ^{R} are the left and the right state of the primitive variable. Roe-averaged wave speed is

$$\lambda_1 = \tilde{u}$$

$$\lambda_2 = \tilde{u} + \tilde{c}$$

$$\lambda_3 = \tilde{u} - \tilde{c}$$

The wave speed is obtained as

$$\Delta v_1 = \Delta\rho - \frac{\Delta p}{\tilde{c}^2}$$

$$\Delta v_2 = \Delta u + \frac{\Delta p}{\tilde{c}\tilde{\rho}}$$

$$\Delta v_3 = \Delta u - \frac{\Delta p}{\tilde{c}\tilde{\rho}}$$

where Δ is the difference between the left and the right state.

The right characteristics are given by

$$\mathbf{R} = [\mathbf{r_1} \quad \mathbf{r_2} \quad \mathbf{r_3}] = \begin{pmatrix} 1 & \frac{\tilde{\rho}}{2\tilde{c}} & -\frac{\tilde{\rho}}{2\tilde{c}} \\ \tilde{u} & \frac{\tilde{\rho}(\tilde{u}+\tilde{c})}{2\tilde{c}} & -\frac{\tilde{\rho}(\tilde{u}-\tilde{c})}{2\tilde{c}} \\ \frac{\tilde{u}^2}{2} & \frac{\tilde{\rho}(\tilde{h}+\tilde{c}\tilde{u})}{2\tilde{c}} & -\frac{\tilde{\rho}(\tilde{h}-\tilde{c}\tilde{u})}{2\tilde{c}} \end{pmatrix}$$

In Roe scheme, all the jumps happen only across the characteristics so we can connect the one state to other state using simple formulations. Flux at $x = 0$ when $\frac{x}{t} = 0$ is

$$\mathbf{FU(0)} \approx \tilde{\mathbf{A}}\mathbf{U(0)} = \mathbf{F(U^L)} + \sum_{i=1}^{3} \mathbf{r_i} min(0, \lambda_i)\Delta v_i \tag{2.31a}$$

$$\mathbf{FU(0)} \approx \tilde{\mathbf{A}}\mathbf{U(0)} = \mathbf{F(U^R)} - \sum_{i=1}^{3} \mathbf{r_i} min(0, \lambda_i)\Delta v_i \tag{2.31b}$$

$$\mathbf{FU(0)} \approx \tilde{\mathbf{A}}\mathbf{U(0)} = \frac{\mathbf{F(U^R)} + \mathbf{F(U^L)}}{2} - \frac{1}{2}\sum_{i=1}^{3} \mathbf{r_i} |\lambda_i|\Delta v_i \tag{2.31c}$$

From eq. 2.31c, it is clear that Roe-averaging adds dissipation based on the wave strength. We refer [16] for more details and multi-dimensional extension of the scheme.

Lax-Friedrichs scheme

Lax-Friedrichs scheme is the simplest Riemann solver. It adds dissipation based on the maximum eigenvalue. Lax-Friedrichs flux is

$$\mathbf{F}^{LF}(\mathbf{U}^L_{i+0.5}, \mathbf{U}^R_{i+0.5}) = \frac{1}{2}\left[\mathbf{F}(\mathbf{U}^R_{i+0.5}) + \mathbf{F}(\mathbf{U}^L_{i-0.5}) - \alpha(\mathbf{U}^R_{i+0.5} - \mathbf{U}^L_{i+0.5})\right]$$

The finite difference formulation can be applied through flux-splitting procedure so the final flux is given by

$$\mathbf{F(U)} = \mathbf{F^+(U)} + \mathbf{F^-(U)}$$

$$\frac{\mathrm{d}\mathbf{F^+(U)}}{\mathrm{d}\mathbf{U}} \geq 0 \qquad \frac{\mathrm{d}\mathbf{F^-(U)}}{\mathrm{d}\mathbf{U}} \leq 0$$

Then the Lax-Friedrich splitting is

$$\mathbf{F}^{\pm} = \frac{1}{2}\left[\mathbf{F}(\mathbf{U}) \pm \alpha\mathbf{U}\right] \tag{2.33a}$$

$$\alpha = \max_{\mathbf{\Lambda}} |\mathbf{F}'(\mathbf{U})| \tag{2.33b}$$

In eq 2.33a, dissipation is added based on the characteristic direction and the magnitude of the eigenvalues.

Rusanov scheme

Rusanov scheme is an extension of the Lax-Friedrichs scheme, where the speed of sound is based on Roe-averaged quantity. The maximum signal speed may differ from the Lax-Friedrichs scheme.

AUSM scheme

Advection Upstream Splitting Method (AUSM) is based on Gudunov scheme. It is proposed in [17], where upwinding is based on Mach number.

$$\mathbf{F}(\mathbf{U}) = \begin{bmatrix} \rho u \\ \rho u^2 \\ u(\rho E + p) \end{bmatrix} + \begin{bmatrix} 0 \\ p \\ 0 \end{bmatrix} = \mathbf{F}^c + \mathbf{F}^p$$

$$\mathbf{F}^c = M \begin{bmatrix} \rho a \\ \rho a u \\ \rho a H \end{bmatrix}$$

The cell-interface Mach number is given by the splitting

$$M_{i+0.5} = M_i^+ + M_{i+1}^-$$

and pressure is given by

$$p_{i+0.5} = p_i^+ + p_{i+1}^-$$

Mach number splitting is given by

$$M^{\pm} = \begin{cases} \pm\frac{1}{4}(M \pm 1)^2 & if \quad |M| \quad \leq 1 \\ \frac{1}{2}(M \pm |M|) & if \quad |M| \quad > 1 \end{cases}$$

Pressure splitting is given by

$$p^{\pm} = \begin{cases} \frac{1}{2}p(M \pm 1) & if \quad |M| \quad \leq 1 \\ \frac{1}{2}p\frac{(M \pm |M|)}{M} & if \quad |M| \quad > 1 \end{cases}$$

or

$$p^{\pm} = \begin{cases} \frac{1}{4}p(M \pm 1)^2(2 \mp M) & if \quad |M| \quad \leq 1 \\ \frac{1}{2}p\frac{(M \pm |M|)}{M} & if \quad |M| \quad > 1 \end{cases}$$

A lot of variants of AUSM scheme are available. Some of them are AUSM+ [18], AUSM+up [19], Simple Low-dissipation AUSM [20], AUSMDV [21], AUSMPW [22].

HLLC scheme

The HLLC scheme is a modified HLL scheme. HLLC is based on three characteristic formulations but HLL is based on two characteristic formulations. It is a wave-based model and has three jumps across characteristics.

Pressure estimate is given by

$$P^* = max(0, p_{pvrs})$$

where

$$p_{pvrs} = \frac{1}{2}(p^L + p^R) - \frac{1}{2}(u^R - u^L)\bar{\rho}\bar{a}$$

$$\bar{\rho} = \frac{1}{2}(\rho^L + \rho^R)$$

$$\bar{a} = \frac{1}{2}(a^L + a^R)$$

Once the pressure estimation is over, wave speed is calculated based on p^*.

$$S^L = u^L - a^L q^L$$

$$S^R = u^R + a^R q^R$$

$$q^k = \begin{cases} 1 & if \quad p^* \quad \leq p^k \\ \left[1 + \frac{\gamma+1}{2\gamma}\left(\frac{p*}{p^k} - 1\right)\right]^{0.5} & if \quad p^* \quad > p^k \end{cases}$$

intermediate speed, S^* is given by

$$S^* = \frac{p^R - p^L + \rho^L u^L(S^L - u^L) - \rho^R u^R(S^R - u^R)}{\rho^L(S^L - u^L) - \rho^R(S^R - u^R)}$$

The HLLC flux is:

$$\mathbf{F}^{hllc}_{i+0.5} = \begin{cases} \mathbf{F}^L, & \text{if } 0 \leq S^L \\ \mathbf{F}^{*L}, & \text{if } S^L \leq 0 \leq S^* \\ \mathbf{F}^{*R}, & \text{if } S^* \leq 0 \leq S^R \\ \mathbf{F}^R, & \text{if } 0 \geq S^R \end{cases}$$

$$\mathbf{F}^{*k} = \mathbf{F}^k + S^k(\mathbf{U^{*k}} - \mathbf{U^k})$$

$$\mathbf{U}^{*k} = \rho^k \left(\frac{S^k - u^k}{S^k - S^*}\right) \begin{bmatrix} 1 \\ S^* \\ v^k \\ \frac{E^k}{\rho^k} + (S^* - u^k)\left[S^* + \frac{p^k}{\rho^k(S^k - u^k)}\right] \end{bmatrix}$$

$$\mathbf{F}^{*k} = \frac{S^*(S^k\mathbf{U}^k - \mathbf{F}^k) + S^k(p^k + \rho^L(S^k - u^k)(S^* - u^k))\mathbf{D}^*}{S^k - S^*}$$

$$\mathbf{D}^* = [0 \quad 1 \quad 0 \quad S^*]^T$$

Once the cell-interface flux is calculated, time marching is carried using a time integration scheme.

2.6.3 Discretization of the Euler equation on unstructured mesh

For a uniform mesh, one-dimensional discretization of the Euler equation can be extended to multi-dimensional. In this section, the discretization of the Euler equation for unstructured mesh is presented.

The integral form of the Euler equation on the control volume Ω with the boundary Γ is,

$$\frac{\partial}{\partial t}\iint_{\Omega} \mathbf{U} d\Omega + \oint_{\Gamma} (\mathbf{E}.\vec{n}) d\Gamma = 0 \tag{2.36}$$

where $\vec{n}$ is unit normal vector at edge of control volume. For 2-D it can be written as [23]

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}t} = -\frac{1}{|\Omega_p|}\sum_i (\mathbf{F_i}.\vec{n_x} + \mathbf{G_i}.\vec{n_y})\delta\Gamma_i \tag{2.37}$$

where Σ_p is a the cell, $\Sigma_p \subset \Sigma$; Γ_i is the edge and $\Gamma_i \subset \Gamma$. The Jacobian of flux is

$$\mathbf{A_n} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & \mathbf{n_t} & 0 \\ (\gamma - 1)\frac{q^2}{2}\mathbf{n} - q_n\mathbf{v} & \mathbf{v} \otimes \mathbf{n} + (1-\gamma)\mathbf{n} \otimes \mathbf{v} + q_n I & (\gamma - 1)\mathbf{n} \\ \left(\frac{\gamma-1}{2}q^2 - H\right) q_n & H\mathbf{n_t} - (1-\gamma)\mathbf{v^T}q_n & q_n \end{bmatrix}$$

where $\mathbf{q_n} = un_x + vn_y$, $\mathbf{n_t} = [1 \quad 0]^T$, $\mathbf{v} = [u \quad v]$. I is identity matrix of 2×2.

Using Single Value Decomposition, the Jacobian matrix $\mathbf{A}$ is decomposed and the decoupled characteristic equations are obtained.

$$\mathbf{A}_n = \mathbf{R}_n \Lambda_n \mathbf{L}_n \tag{2.38}$$

$$\Lambda_n = \begin{bmatrix} q_n - c & 0 & 0 & 0 \\ 0 & q_n & 0 & 0 \\ 0 & 0 & q_n + c & 0 \\ 0 & 0 & 0 & q_n \end{bmatrix}$$

where c is the velocity of sound.

All characteristics run towards the right side for the supersonic flow, so there is no need for an explicit boundary condition. A supersonic outflow boundary condition [24] with physical variables extrapolated from within the domain can be used. All the Riemann solvers use the characteristic based splitting.

$$\mathbf{R}_n = \begin{bmatrix} 1 & 1 & 1 & 0 \\ u - cn_x & u & u + cn_x & l_x \\ v - cn_y & v & v + cn_y & l_y \\ H - q_n c & \frac{q^2}{2} & H + q_n c & q_l \end{bmatrix}$$

$[l_x \quad l_y]$ are tangent vector of edge.

$$\mathbf{L}_n = \begin{bmatrix} \frac{1}{2}\left(\frac{\gamma-1}{2c^2}q^2 + \frac{q_n}{c}\right) & -\frac{1}{2}\left(\frac{\gamma-1}{c^2}u + \frac{n_x}{c}\right) & -\frac{1}{2}\left(\frac{\gamma-1}{c^2}v + \frac{n_y}{c}\right) & \frac{\gamma-1}{2c^2} \\ 1 - \frac{\gamma-1}{2c^2}q^2 & \frac{\gamma-1}{c^2}u & \frac{\gamma-1}{c^2}v & -\frac{\gamma-1}{c^2} \\ \frac{1}{2}\left(\frac{\gamma-1}{2c^2}q^2 - \frac{q_n}{c}\right) & -\frac{1}{2}\left(\frac{\gamma-1}{c^2}u - \frac{n_x}{c}\right) & -\frac{1}{2}\left(\frac{\gamma-1}{c^2}v - \frac{n_y}{c}\right) & \frac{\gamma-1}{2c^2} \\ -q_l & l_x & l_y & 0 \end{bmatrix}$$

where $q_l = ul_x + vl_y$. The different forms of the Euler equation and relation between characteristic, physical and conservative variabel can be found in [23].

In most of the test cases, the dual volume finite volume method [25, 26, 4] is used. On reconstruction, the variables are limited using a limiter. The slope values are obtained using the moving least square algorithm. The flux is determined using the Riemann solver, and a time integration scheme is used to perform time marching. The different reconstruction procedures used in this work are explained in chapter 5 and chapter 6.

2.7 Summary

In this chapter, we studied the different forms of the Euler equation and their applications. Following that, we have studied the basic difference between solving procedures in compressible flows and incompressible flows. We also studied how direction based approaches and artificial viscosity based approaches can help to increase the stability of the advection equation solution procedures. Different Riemann solvers are discussed and Riemann solver add diffusion when required is shown. We have also shown, Riemann solver not only add diffusion to the solution but also anti-diffusion when required.

3

Time Discretization Schemes

3.1 Introduction

The time discretization methods used in this thesis are presented in this chapter. For any unsteady problem, accurate discretization of the time derivative term is essential. Techniques of time discretization can be categorized into techniques that are implicit, explicit, or semi-implicit. Depending on the number of steps taken to achieve the value of the unknown at the next stage, time integration methods are often categorized into single-step methods and multi-step methods.

New hyperbolic Runge-Kutta (RK) methods that are designed for better stability are proposed in this study. In hyperbolic RK techniques, both real and imaginary limits are optimized instead of optimizing only real stability limits (as in the parabolic RK method). Optimization is achieved using the Evolutionary Algorithm (EA) in the current work. The time integration schemes used in this work are the forward Euler scheme, the backward Euler scheme, the Strong Stability Runge-Kutta Method (SSPRK), the Classical Runge-Kutta Method (CRK), the Low Storage Runge-Kutta Method (LRK) and the new class of hyperbolic Runge-Kutta Method (HRK).

Forward Euler is a first-order method. The discretized equation of the forward Euler method for general conservative equation in the conservative framework is:

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} = -\frac{\bar{\mathbf{F}}_{i+0.5}^n - \bar{\mathbf{F}}_{i-0.5}^n}{\Delta x} \qquad \mathcal{O}(\Delta x^2, \Delta t)$$

Forward Euler is an explicit single-step method, and it is conditionally stable. The implicit equivalent of the forward Euler method is the backward Euler method which is unconditionally stable for most of the equation but for some equations and boundary conditions it may give diverge solutions. It is a first-order method. For the general conservative equation, the backward Euler method is:

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} = -\frac{\bar{\mathbf{F}}_{i+0.5}^{n+1} - \bar{\mathbf{F}}_{i-0.5}^{n+1}}{\Delta x} \qquad \mathcal{O}(\Delta x^2, \Delta t)$$

Implicit methods for time accurate calculation are computationally expensive and comparatively more dissipative compared to explicit methods [27]. Because of this, explicit approaches are widely used in the current thesis.

Runge-Kutta methods are considered as one of the most reliable time discretization methods in time integration schemes [28]. This approach was introduced by Runge [29] and further developed by Kutta. A background for Runge-Kutta methods can be found in [30, 28]. To reduce the computational costs, the Embedded RK methods [31, 32] are proposed which use the same function evaluation at different levels.

Some adaptive time step refinement RK methods that can decide time-step based on smoothness or truncation error can be found in [33, 34, 35]. This class of Runge-Kutta method is used in Matlab® *ODE* function. An implicit version of RK methods suitable for stiff differential equations is presented in [36]. Semi-implicit RK methods [37, 38] are another class of RK methods where some of the functional evaluation are implicit, and some are explicit. They are relatively more stable than the explicit method for the convection-diffusion equation [39]. Conservation laws with stiff relaxation time have stability issues. More details of RK methods for such equations can be found in [40, 41].

Williamson [42] introduced a low storage RK method. He discovered that all second-order and some third-order RK methods could be written in the 2N-storage form. 2N-storage RK method for third and fourth-order is implemented in [43]. One of the simplest ways to integrate 2N-storage RK methods is given in [44]. The 3N-storage form of the Runge-Kutta method with low dissipation and dispersion is given in [45]. A fifth-order Improved Runge-Kutta Method (IRK) can be found in [46]. In this approach, the functions are evaluated at the left and the right points (time steps) instead of focusing only at the right points. Verwer [47] increased the stability of RK methods by releasing the stability equation coefficients using Chebyshev-polynomial; however, the procedure is time-consuming for higher-order methods. Runge-Kutta methods for improving stability by applying positivity condition along with Total Variation Bound (TVB) can be found in [48, 49]. This method is later renamed Strong Stability Runge-Kutta (SSPRK) method. It has a better stability property than the classical one and has gained prominence in aeroacoustic problems due to its positivity and TVD properties. A low storage version of the SSPRK system can be found at [50].

Most of the schemes are prone to error in dispersion and diffusion. This error can be minimized by optimizing the scheme to minimize dispersion and dissipation. Optimization of the RK-method by minimizing dispersion and diffusion error in the spectral plane is carried out in [45, 51, 52]. There have been attempts to reduce the dispersion and diffusion by optimizing the spectral property of RK methods by optimizing only time discretization part [45, 51, 52, 53]. Improving only the temporal discretion for dispersion and diffusion is not optimal [53] as the spectral property depends on the combined spatial and temporal discretion. The optimization of the spectral properties of the combined spatial and temporal parts of the hyperbolic equation is studied in [54, 55]. Framing an RK procedure that resolves a set of wavenumbers for a specific CFL number without substantial numerical dispersion and diffusion is still a major challenge.

3.2 RK methods formulation

For an initial value problem (IVP) ,

$$u' = F(u) \qquad u(0) = u_0$$

s stage RK method takes the form

$$u^{n+1} = u^n + \sum_{i=1}^{s} W_i k_i$$

$$k_i = \Delta t F\left(u^n + \sum_{j=1}^{i-1} a_{ij} k_j\right)$$

Where W_i and a_{ij} are determined from matching Taylor series expansion of $u(t^{n+1})$ [28].

In the case of a four-stage, fourth-order method, the following order condition must be satisfied [55]

$$\sum_{i=1}^{4} W_i = 1$$

$$W_2 a_{21} + W_3(a_{31} + a_{32}) + W_4(a_{41} + a_{42} + a_{43}) = \frac{1}{2}$$

$$W_2 a_{21}^2 + W_3(a_{31} + a_{32})^2 + W_4(a_{41} + a_{42} + a_{43})^2 = \frac{1}{3}$$

$$W_3 a_{32} a_{21} + W_4[a_{21} a_{42} + a_{43}(a_{31} + a_{32})] = \frac{1}{6}$$

$$W_2 a_{21}^3 + W_3(a_{31} + a_{32})^3 + W_4(a_{41} + a_{42} + a_{43})^3 = \frac{1}{4}$$

$$\begin{aligned} W_3[a_{32} a_{21}^2 + 2a_{21} a_{32}(a_{31} + a_{32})] + W_4[a_{42} a_{21}^2 + a_{43}(a31 + a32)^2 \\ +2(a_{41} + a_{42} + a_{43})\{(a_{31} + a_{32})a_{43} + a_{42} a_{21}\}] = \frac{1}{3} \end{aligned}$$

$$W_4 a_{43} a_{32} a_{21} = \frac{1}{24}$$

Low storage RK methods are of the form [45, 52, 56].

$$u^{n+1} = u^s \tag{3.3a}$$

$$u^i = u^n + \alpha_i \Delta t F(u^{i-1}) \quad for \quad i = 1, 2, ..., s. \tag{3.3b}$$

$$\alpha_i = \frac{\beta_s}{\beta_{s-1}} \quad and \quad \beta_j = \frac{1}{j!} \quad for \quad j = 1, 2, .., s. \tag{3.3c}$$

where Δt is the time step used, α_i is to be determined using Taylors series and stability criteria of the RK methods. This can give the same order of accuracy in time without losing accuracy, but the number of condition required to satisfy Taylor series in this method is lower than the classical RK method.

For low storage RK methods, the number of conditions and complexity of the conditions in eq. 3.2 reduce to

$$W_4 = 1 \tag{3.4a}$$

$$W_4\alpha_3 = \frac{1}{2} \tag{3.4b}$$

$$W_4\alpha_2\alpha_3 = \frac{1}{6} \tag{3.4c}$$

$$W_4\alpha_1\alpha_2\alpha_3 = \frac{1}{24} \tag{3.4d}$$

Using a linear test case

$$u_t = Mu, \quad 0 \leq t \leq T, \quad u(0) = u_0 \tag{3.5}$$

We can examine the stability by adding small perturbation in the scheme [47], then the RK method becomes

$$\tilde{Y}_0 = \tilde{u}^n$$

$$\tilde{Y}_j = \tilde{u}_n + \Delta t \sum_{l=0}^{j-1} a_{ij} \tilde{F}_l + \tilde{r}_j; \quad \tilde{F}_l = F(t_n + c_l \Delta t, Y_l)$$

$$\tilde{u}^{n+1} = \tilde{Y}_s$$

where $\tilde{u}$ is the perturbation of u and $\tilde{r_j}$ at stage j. Let us define $e_n = \tilde{u^n} - u^n$ and $d_j = \tilde{Y_j} - Y_j$ as the errors introduced by these perturbations. For a linear system the error in explicit RK method can be written as

$$e^{n+1} = P_s(M\Delta t)e^n + \sum_{j=1}^{s} Q_{sj}(M\Delta t)\tilde{r_j}$$

where P_s is the absolute stability polynomial and Q_{sj} are internal stability polynomials. Then error can be written as

$$\|e^{n+1}\| \leq \|P_s(M\Delta t)\|\|e^n\| + \sum_{s}^{j=1}\|Q_{sj}(M\Delta t)\|\|\tilde{r_j}\|$$

$$\|e^{n+1}\| = \sigma\left(\|P_s(M\Delta t)\|\|e^n\| + \sum_{s}^{j=1}\|Q_{sj}(M\Delta t)\|\right)\|\tilde{r_j}\|$$

$$\|e^{n+1}\| = \max_{z=\Delta t\lambda}\left(\|P_s(M\Delta t)\|\|e^n\| + \sum_{s}^{j=1}\|Q_{sj}(M\Delta t)\|\right)\|\tilde{r_j}\|$$

where $\|.\|$ is Euclidean norm $\in \mathbb{R}^n$ and λ runs through the spectrum of M and σ is spectral radius. The error should decline over the iteration; it can happen only if $\|P_s(M\Delta t)\| \leq 1$.

Exact solution of eq. 3.5 is

$$u(t+\Delta t) = \exp(\Delta t M)u(t)$$

To satisfy consistency, $P_s(z)$ approximates $\exp(z) \quad for \quad z \to 0$ so, $P_s(z) = exp(z) + \mathcal{O}(z^s)$.

Solution of the RK method is given by [57]

$$u^{n+1} = (1 + (M\Delta t)b^T e + (M\Delta t)^2 b^T Ae + ... + (M\Delta t)^s b^T A^{s-1}e)u^n \tag{3.8}$$

where $k = \Delta t$ and $e = (1, 1, ..., 1) \in \mathbb{R}^n$. Substituting $\beta_j = b^T A^{j-1}e$, eq. 3.8 becomes

$$u^{n+1} = (1 + (M\Delta t)\beta_1 + ... + (M\Delta t)^s\beta_s)u^n$$

If numerical gain is defined as $R\left(\Delta t\lambda\right) = \frac{u^{n+1}}{u^{n}}$, then numerical gain or amplification factor can be written as,

$$P_s = R\left(\Delta t\lambda\right) = \sum_{0}^{n} \frac{\left(\Delta t\lambda\right)^n}{n!} \tag{3.9}$$

This is stable only if $|R\left(\Delta t\lambda\right)| \leq 1$

3.3 Popular Optimized RK methods

In this section, we will review family of RK methods explored by various researchers. A brief introduction to a few optimized RK methods such as low dispersion and dissipation Runge-Kutta method (LDDRK), strong stability preserving Runge-Kutta method (SSPRK), explicit Runge-Kutta methods for parabolic and hyperbolic partial differential is presented.

3.3.1 Parabolic Runge Kutta Method

This RK approach was optimised using a technique similar to that discussed in this thesis. However, they used shifted Chebyshev polynomials to find optimal weights, but in our method, we used EA to evaluate the optimal weight. We also found that our polynomial covers a comparatively higher area in stability plot than this method. Also, it is not easy to apply this technique to orders greater than two (p $\geq$ 3) [47].

The absolute polynomial is defined as

$$P_s(z) = c_{0,s} + c_{1,s}z + c_{2,s}z^2 + ... + c_{s,s}z^s$$

Free coefficients of $c_{j,s}$ are used to maximize the real stability boundary $\beta(s)$. Second-order RK methods of this family are derived in [58] and its stability polynomial is

$$P_s(z) = \frac{2}{2-z} - \frac{z}{2-z}T_s\left[cos\frac{\pi}{s} + \frac{z}{2}\left(1 - cos\frac{\pi}{s}\right)\right], \quad s \geq 2$$

$$\beta(s) = \frac{2}{(tan\frac{\pi}{2s})^2} \approx 0.81s^2$$

Three and four stage polynomials of second-order accuracy of this kind are [47]

$$P_3(z) = 1 + z + \frac{1}{2}z^2 + \frac{1}{16}z^3$$

$$P_4(z) = 1 + z + \frac{1}{2}z^2 + \frac{2}{25}z^3 + \frac{1}{250}z^4$$

In general $c_{i,s}$ can be written as

$$c_{i,s} = 3\frac{1 - \frac{(i-1)^2}{s^2}}{i(2i-1)(1-\frac{1}{s^2})}c_{i-1,s} \quad for \quad i = 3, 4, ...s$$

More details of Runge Kutta Chebyshev(RKC) method can be found in [59].

3.3.2 Hyperbolic RK methods

In hyperbolic RK methods, instead of maximizing only the real stability boundary (parabolic RK method), both the real and the imaginary boundaries are optimized. The optimized stability polynomial of this kind proposed in [60] is

$$P_s^3(z) = \frac{1}{\beta^2+1}\left[1 + z + i^{s-1}\beta^2 T_{s-1}\left(\frac{iz}{\beta}\right) + \frac{1}{2}i^{s+2}\beta\{(s-2)T_s\left(\frac{iz}{\beta}\right) - sT_{s-2}\left(\frac{iz}{\beta}\right)\}\right]$$

3.3.3 LDDRK methods

Low dispersive and diffusive RK methods are RK methods optimized in the spectral plane by minimizing numerical diffusion and dispersion. In the past, researchers have optimized RK methods for temporal discretization. Later, they optimized it for combined spatial and temporal discretization [55]. This scheme is commonly used in computational aero-acoustics because these schemes may require lesser number of grid points to get lower dissipation and dispersion solution. Several optimized schemes are proposed for different wave-number range and order.

Temporal LDDRK methods:

Here the optimization is carried out only on a temporal plane. Further descriptions of this method are given in [45]. The numerical gain for the m-stage RK method is

$$G(\omega\Delta t) = \sum_{0}^{m} \gamma_m (i\omega\Delta t)^m$$

where, ω is wave number and Δt is the time step. For RK method of m^{th} order, $\gamma_m = \frac{1}{m!}$. Researchers attempted to minimize the spectral error by sacrificing a higher-order term to optimize it for a better spectral property. Then the resulting equation for optimization can be written as

$$min\left[\int |1 - G(\omega\Delta t)d(\omega\Delta t)| + \int |(\omega^*\Delta t) - (\omega\Delta t)d(\omega\Delta t)|\right]$$

Researchers have tried to optimize for a certain range of wavenumbers (e.g. $\frac{\pi}{6}$ to $\frac{\pi}{2}$ in [51]) since it is impossible to optimize for all wavenumbers from 0 to π. Here the terms in the first part of the integration correspond to the diffusion and the second part corresponds to the dispersion. ω^* is a modified wavenumber in the numerical scheme. Eventhough this method of optimization shows some improvement in results, the behaviour changes when space discretization is altered. In certain cases, the results are worse than the classical ones.

Spatial-temporal LDDRK methods:

Numerical dispersion and diffusion depend on combined spatial and temporal discretization, so some researchers have focused on this kind of optimization. For more detailed information on such approaches, refer to [54, 55, 61]. The RK method's numerical gain is

$$G_j = \sum_{0}^{m} \frac{(iA)^m}{m!}$$

where $i = \sqrt{-1}$. Unlike the temporal LDDRK process, A depends on spatial discretization, and G_j depends on both space and time. As in von Neumann stability analysis, A is

Fourier transform of spatial discretization. For example, A for the second-order difference approximation for the first derivative is $U\frac{\exp(ik\Delta x)-\exp(-ik\Delta x)}{\Delta x}$.

In this approach, researchers went further and applied the optimization method for low diffusion, group velocity and phase velocity of the linear convection equation. The phase speed is given as

$$\frac{c_j}{c} = \frac{\omega^* \Delta x}{\omega \Delta x}$$

where c is convection velocity in linear wave equation. The corresponding group velocity is given by

$$\frac{V_g}{c} = \frac{1}{CFL}\frac{d(\omega^* \Delta x)}{d(\Delta x)}$$

Then the final optimization equation is

$$min\left[\int |1 - G(\omega\Delta x)|\, d(\omega\Delta x) + \int \left|\frac{V_g}{c} - 1\right| d(\omega\Delta x) + \int \left|\frac{c_j}{c} - 1\right| d(\omega\Delta x)\right]$$

This type of optimization method can correctly predict the behaviour of the convection equation but cannot do so for the nonlinear Burgers equation. However, this demonstrates some improvement over the classical one. This optimization approach is time-consuming, and it is difficult to frame a numerical scheme to overcome a large range of wavenumbers for a given CFL number. Here also, we may not have an optimal system which suits well with any discretization.

3.3.4 SSPRK method

Strong stability RK methods are a family of RK methods with good stability as well as positivity properties that is essential for problems having discontinuity in the solutions. More details on this method can be found in [62]. The m stage RK method can be written as

$$u^i = \sum_{k=0}^{i-1}(\alpha_{ik}u^k + \Delta t \beta_{ik} L(u^k)), \qquad i = 1, ..m$$

$$u^{n+1} = u^m$$

A RK method is strong stability preserving if $\alpha_{ik} \geq 0$ and $\beta_{ik} \geq 0$, and the time step restriction is given by $\Delta t \leq c\Delta t_1$; where c is given as

$$c = min\frac{\alpha_{i,k}}{\beta_{i,k}}$$

This method is useful because it ensures positivity similar to that of Godunov scheme [63]. Here, we have compared the new RK methods presented in this work and SSPRK methods for some test cases.

3.4 Formulation of RK methods

This section presents the formulation of the RK methods used in this work.

3.4.1 Classical RK method

The formulation of CRK method is

$$u_1 = u^n + \Delta t F(u^n)$$

$$u_2 = u^n + 0.5\Delta t F(u_1)$$

$$u_3 = u^n + 0.5\Delta t F(u_2)$$

$$u_4 = u^n + \Delta t F(u_3)$$

$$u^{n+1} = u^n + \frac{\Delta t}{6}\left[F(u_1) + 2F(u_2) + 2F(u_3) + F(u_3)\right] \qquad \mathcal{O}(\Delta t^4)$$

3.4.2 Low storage RK method

Formulation of LRK method is as follows:

$$u_1 = u^n + \frac{1}{4}\Delta t F(u^n)$$

$$u_2 = u^n + \frac{1}{3}\Delta t F(u_1)$$

$$u_3 = u^n + \frac{1}{2}\Delta t F(u_2)$$

$$u^{n+1} = u^n + \Delta t F(u_3) \qquad \mathcal{O}(\Delta t^4)$$

3.4.3 SSPRK2

Formulation of SSPRK2 [64] method is:

$$u_1 = u^n + \Delta t F(u_1)$$

$$u^{n+1} = \frac{1}{2}u^n + \frac{1}{2}u_1 + \frac{1}{2}\Delta t F(u_1) \qquad \mathcal{O}(\Delta t^2)$$

3.4.4 SSPRK3

Formulation of SSPRK3 [64] method is as follows:

$$u_1 = u^n + \Delta t F(u_1)$$

$$u_2 = \frac{3}{4}u^n + \frac{1}{4}u_1 + \frac{1}{4}\Delta t F(u_1)$$

$$u^{n+1} = \frac{1}{3}u^n + \frac{2}{3}u_2 + \frac{2}{3}\Delta t F(u_2) \qquad \mathcal{O}(\Delta t^3)$$

3.4.5 SSPRK4

Formulation of SSPRK4 [62] method is given below.

$$u_0 = u^n$$

$$u_i = \left[u_{i-1} + \frac{\Delta t}{2}F(u_{i-1})\right] \qquad i = 1, ..., 3$$

$$u^{n+1} = \frac{2}{3}u_1 + \frac{2}{3}\left[u_3 + \frac{\Delta t}{2}F(u_3)\right] \qquad \mathcal{O}(\Delta t^4)$$

3.5 Formulation of HRK method

3.5.1 HRK31

In this thesis, we have reduced the order of accuracy for better stability by assigning higher-order term(s) in the stability equation as a free variable. Here the second-order three-stage RK method is defined as HRK31; where three (3) denotes the number of stages, and one (1) denotes the number of free variables in the stability equation. Then the Eq. 3.9 becomes

$$u^{n+1} = \left[1 + (h\lambda) + 0.5(h\lambda)^2 + a_1(h\lambda)^3\right] u^n \tag{3.17}$$

The coefficient a_1 is evaluated using an evolutionary algorithm where the stability plot area is maximized. Similar procedures were followed in [65, 60] but a damped shifted Chebyshev polynomial was used. Their technique is time-consuming for the higher-order method, as well as the achievement of a global maximum is tough and probably impossible for higher-order methods. The HRK31 formula is

$$u_1 = u^n + 0.071726918318082 \times hF(u^n)$$

$$u_2 = u^n + 0.5 \times hF(u_1)$$

$$u_3 = u^n + hF(u_2) \qquad \mathcal{O}(\Delta t^2)$$

3.5.2 HRK41

This is a four-stage, third-order RK method. The value of coefficient a_1 here is 0.020864242172733, and the HRK41 formula is

$$u_1 = u^n + 0.12518545303 \times hF(u^n) \tag{3.19a}$$

$$u_2 = u^n + \frac{1}{3} \times hF(u_1) \tag{3.19b}$$

$$u_3 = u^n + \frac{1}{2} \times hF(u_2) \tag{3.19c}$$

$$u^{n+1} = u^n + hF(u_3) \qquad \mathcal{O}(\Delta t^3) \tag{3.19d}$$

3.5.3 HRK42

This is a four-stage, second-order RK method. The HRK41 formula is

$$u_1 = u^n + 0.058812383 \times hF(u^n) \tag{3.20a}$$

$$u_2 = u^n + 0.180721032 \times hF(u_1) \tag{3.20b}$$

$$u_3 = u^n + \frac{1}{2} \times hF(u_2) \tag{3.20c}$$

$$u^{n+1} = u^n + hF(u_3) \qquad \mathcal{O}(\Delta t^2) \tag{3.20d}$$

3.6 Optimization algorithm

The steps used in the Evolutionary Optimization Algorithm (EA) are listed in this section. Since there are several peaks in the objective function (stability plot area), EA [66] is used. In order to decrease computing costs, the optimization algorithm is divided into three steps. The first and second stages are the EA algorithm, and the third stage is a form of local optimization to enhance the convergence rate.

Stage 1 The boundary of the variables is limited from 0 to 10. Since the algorithm's roots are the Taylors series coefficients, it should be less than 1 for the method's convergence. As stability is the primary goal, the search domain is increased from 0 to 10. In turn, the cost of computing increases. Here, only positive values have been considered to ensure positivity. If positivity is not a problem, negative numbers can also be used for the initial population.

1. The initial population size of variables (a_i) in eq. 3.17 is divided into 11 equally spaced (can be any value) values, ranging from -10 to 10.

2. The stability graph area was estimated for different values of a_i.

3. Using the Roulette-wheel selection process [67], the population of the next generation is determined.

4. Elitism [68] is followed, i.e. a set of a_i corresponding to the maximum area value is transferred to the next generation without mutation to ensure at least zero convergence.

5. A pseudo-random number generated between -1 and 1 is added to the population to increase the randomness of the population. The randomness range of the mutation operator is reduced instead of adding boundary constraint. Although no explicit cross-over operator is used for this problem, a random number around one order less than the population range is applied. Indirectly, it functions in the same way as a cross-over operator since it preserves a significant digit and applies randomness to less significant digits.

Stage 2 The outcome of stage 1 ensures that a_i is within 0 to 1. The value of a_i is in a positive plane, so it also gives positivity. The search domain is, therefore reduced to (0,1). Step 1 is followed here, but the pseudo-random number is generated between-0.1 and 0.1.

Stage 3 (*Multi section method*) Multi-section is a technique of optimization useful for convex-functions. It can be extended to a multiple-peak function. If the number of section is high, local maxima or minima can be cleverly avoided. Although the number of function evaluations is greater than the methods of bisection, it can easily be parallelized.

Algorithm of multi section method: We should know the outcome (a_i) of EA before beginning the multi-section, after which we should determine the approximate range (δa) where the desired roots are located.

1. Divide the interval $a_i + \delta a_i$ to $a_i - \delta a_i$ into n sections . We have used ten sections in the present work. It can be any number. The higher the number, it can easily bypass the local extremum. If we are not sure about the root range, this number should be elevated.

2. Calculate the area under the plot for a given population and obtain the a_i corresponding to the maximum area.

3. The next-generation δa_{i+1} is obtained using the following step: $\delta a_{i+1} \geq \frac{\delta a_i}{n}$

4. Go to step 1 above and repeat the process till the desired decimal accuracy or tolerance is acheived.

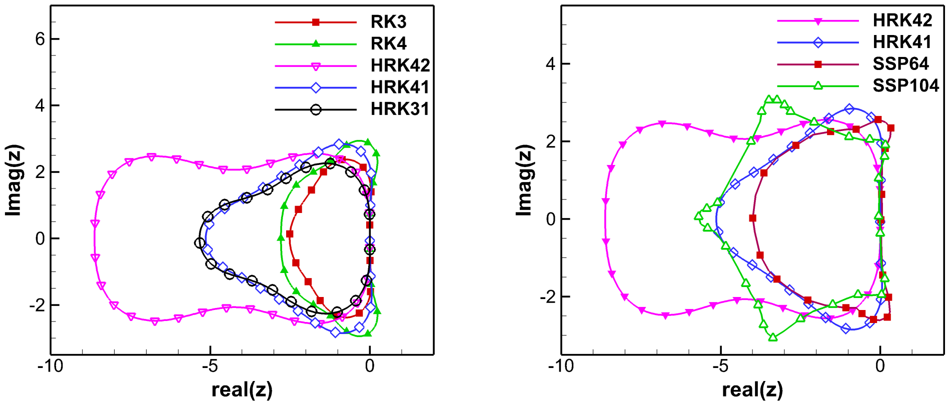
Because all the proposed RK methods are in the low storage format, at least one of $\beta_{i,j}$ of the HRK method is zero. If one of the $\beta_{i,j}$ is zero, the strong stability region of the method is zero. But it shows a better non-oscillatory property when a high CFL number is used than SSPRK methods.

3.7 Results and Discussion

In this section, we have presented some analysis, and some of the standard test cases tested using our schemes. The present schemes are tested on 1-D convection equation, 1-D shock tube problems, 2-D lid-driven cavity, and 2-D shock-diffraction problem. These schemes performed well on all the test cases considered. The RK methods used are SSPRK2 and SSPRK3 obtained from [69], and SSPRK4 is from [62]. CFL numbers tabulated in all the tables below are two decimal accurate.

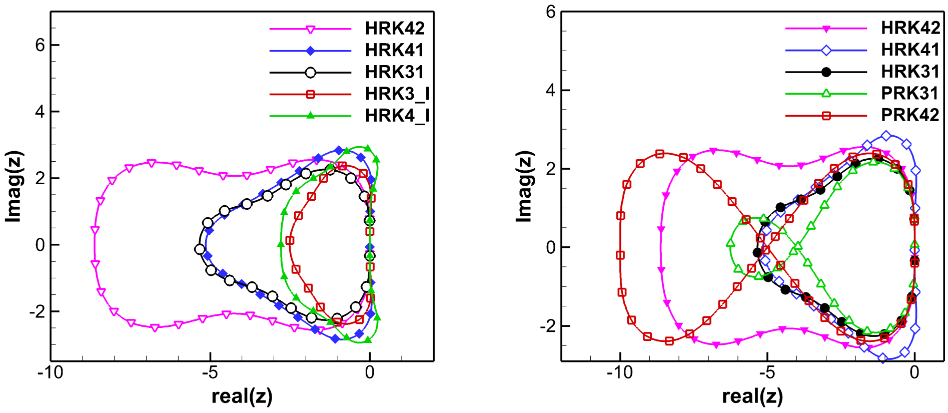
3.7.1 Stability plots

The stability graphs are obtained by replacing $\lambda h = x + jy$ with $j = \sqrt{-1}$. In figure 3.1 and figure 3.2, the stability graph of the various methods are shown. We may observe that the HRK methods have a better area of stability than most of the explicit methods of the same order. Although the methods shown here are second or third order, higher-order methods can be obtained by this technique. The PRK42 approach has better coverage on the real axis than HRK42, but both real and imaginary axis coverage is required for the hyperbolic equation. HRK42 is better than PRK42 in terms of the stability area. Also, compared to HRK methods presented in [60], our HRK methods have a high stability region coverage than others.



(a) Classical methods **(b)** Strong stability preserving RK method

Figure 3.1: Linear stability plots of different methods



(a) Hyperbolic RK methods **(b)** Parabolic RK methods

Figure 3.2: Linear stability plots of different methods

3.7.2 1-D convection equation

Governing equation of 1-D linear convection equation is

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{3.21}$$

Here, along with classical RK and HRK techniques for time integration, a first-order backward difference scheme in space is used for the discretization. The initial condition used is similar to that cited in [52, 70].

$$u_0 = exp\left[-\left(\frac{x}{3}\right)^2\right]$$

The $CFL = c\frac{\Delta t}{\Delta x}$, and dissipation error is

$$\epsilon_D = \frac{max(u_{numerical}) - max(u_{exact})}{max(u_{exact})} \times 100$$

Numerical dissipation (ϵ_D) is a function of Δx, CFL, Δt and the discretization used. All the values tabulated in table 3.1 are obtained from the numerical result. ϵ_D in table 3.1 is numerical dissipation corresponding to CFL_{max}. The CFL_{max} here is maximum CFL number at which the numerical scheme gives physical results. Our test on the 1-D convection equation shows that HRK42 is more stable than all the methods considered here. From table 3.1, the HRK42 method is approximately 2.6 times more stable than SSPRK2 and the low storage RK2, 2.1 times more stable than the low storage RK3, and 2.4 times more stable than the low storage RK4 for this problem based on the CFL_{max} criteria.

When we used second-order central difference in space (CD2), none of the above schemes could produce a good result with $n = 501$; this is because of numerical dispersion. For the case of CD4 scheme, there are intermittent intervals of CFL for which the time integration method may not be able to produce results. Hence, specifying a maximum of CFL when using CD4 for the space discretization is not possible.

Table 3.1
1-D convection equation using n = 501 points with c = 1

Methods	CFL_{max}	ϵ_D
SSPRK2	1.09	25.7975
RK2 low storage	1.09	25.6907
RK3 low storage	1.33	26.104
HRK31	1.33	26.1174
HRK42	2.83	26.5724
RK4 low storage	1.48	26.2028
RK4 classical	1.17	29.1903
HRK41	2.19	26.4474

3.7.3 Sod shock tube

When the present schemes tested on this test case are found to be less dispersive and more robust than other schemes considered in the thesis. We refer [5] for more details of the problem.

The 1-D Euler equation is

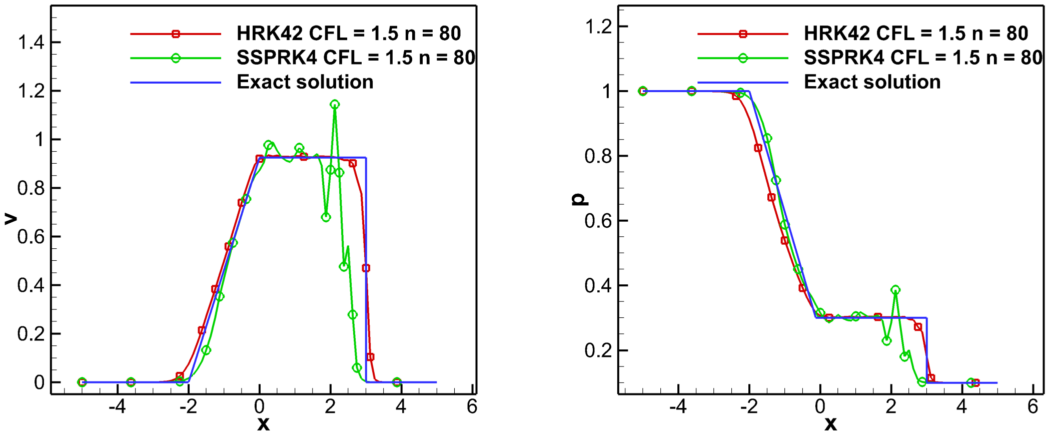
$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{3.22}$$

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho e_t \end{bmatrix} \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho e_t + p) \end{bmatrix}$$

The initial condition used is:

$$(\rho, u, p) = \begin{cases} (1, 0, 1) & x \leq 0 \\ (0.125, 0, 0.1) & x > 0 \end{cases}$$

Dual-volumed finite volume method [26] is used for the spatial discretization. Interpolation using the primitive variables works better and eliminates unnecessary oscillations than the interpolation of flux [71]. We used the second-order minmod limiter [12] to obtain primitive variables on the cell face. The Roe scheme [14] is used for the flux calculation. Table 3.2 shows the stability limit of various schemes using n = 80 points with the flow time of 1.7 s. In this table, the maximum CFL is based on CFL at which any of the primitive variables are approaching infinity. Please note that the CFL number



(a) Velocity

(b) Pressure

Figure 3.3: Solution plot of Sod problem at T = 1.7s using n = 80

should be much lower than the value tabulated to get an oscillation-free solution. Based on CFL_{max} criteria, HRK42 is more stable than all the methods considered here.

HRK42 is 1.9 times more stable than low storage RK2, 1.5 times more stable than RK3. HRK42 is 1.3 times more stable than RK4, and is two times more stable than SSPRK3 and SSPRK2. Figure 3.3 shows the results of different schemes for this problem. SSPRK4 demonstrates solution wiggles, but our systems showed no wiggles. It is recommended to use CFL less than 1.25, 0.95, 1.8, and 1.25 for SSPRK3, HRK31, HRK41, and HRK42, respectively, to obtain a reasonable oscillation free result for this problem while using 80 grid points. The positive attribute about SSPRK4 in this case is, it does not diverge. When the high value of CFL number is used, it may lead to poor time resolution. We shall note from the figure 3.3, SSPRK4 does not diverge even at CFL= 1.5 but it has introduced oscillations in the flow.

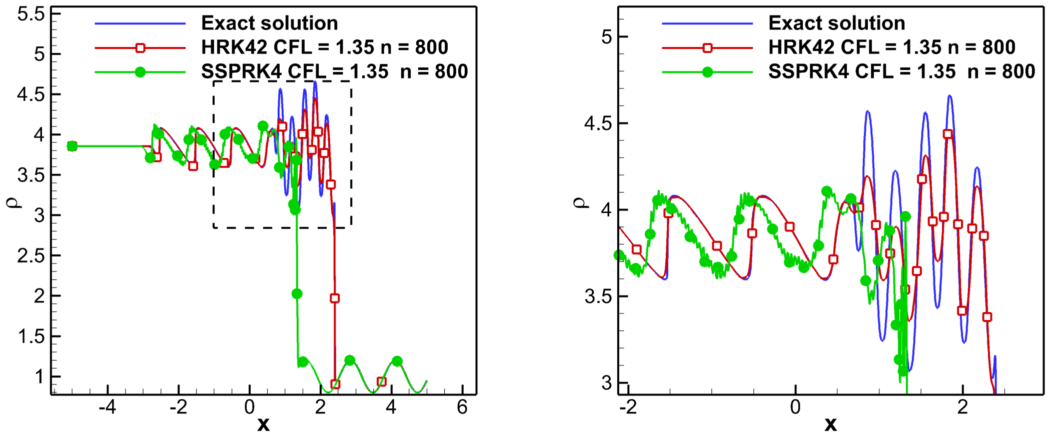
3.7.4 Shu-Osher problem

We have tested our schemes on Shu-Osher problem [72, 73, 74]. Initial condition used in the domain -5 to 5 is

$$(\rho, u, p) = \begin{array}{ll} (3.857143, 2.629369, 10.33333) & x \leq -4 \\ (1 + 0.2 \times sin(5x), 0, 1) & x > -4 \end{array}$$

Table 3.2
1-D Sod problem using n = 80 points of flow time 1.7 s and Shu-Osher problem using n = 800 of flow time 1.8 s

Methods	CFL_{max} **Sod**	CFL_{max} **Shu-Osher**
SSPRK3	1.25	1.19
SSPRK2	1.26	1.20
SSPRK4	not diverging	1.59
RK3 low storage	1.66	1.79
RK2 low storage	1.30	1.19
HRK31	1.61	2.33
HRK41	1.98	2.15
HRK42	2.51	2.4
RK4 low storage	1.85	1.67

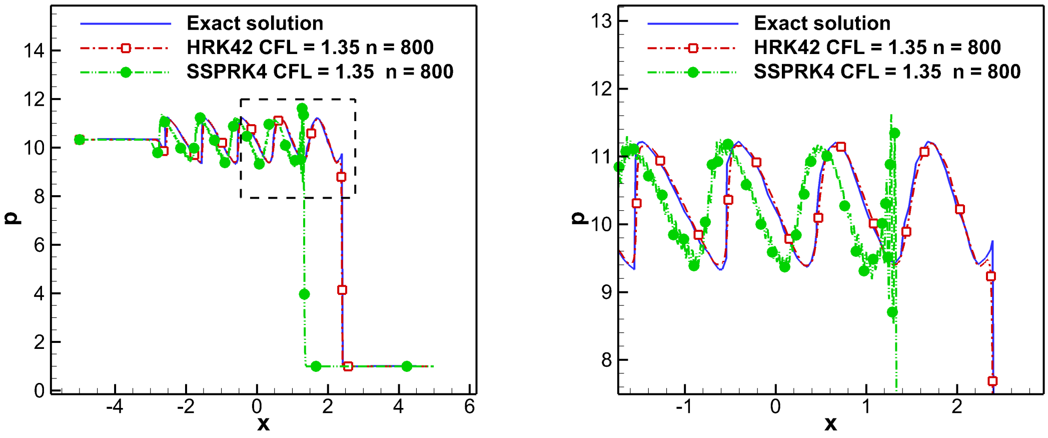


(a) Density using n = 800

(b) Density using n = 800

Figure 3.4: Density plot of Shu-Osher problem at T = 1.8 s

The same discretization procedure used for the Sod shock tube problem is used here. Current schemes have outperformed most of the other schemes used in literature. The results are tabulated in table 3.2. The discretization used is the same as the Sod problem. The best scheme of all the schemes considered here is HRK42. It is two times more stable than RK2, 1.3 times more stable than RK3, 1.4 times more stable than RK4, and two times more stable than SSPRK2 and SSPRK3 on the basis of a maximum of CFL number criterion. The results for this problem are shown in Figure 3.4 and Figure 3.5. Figure 3.4 shows that SSPRK4 is diffusive as well as gives wiggles with n= 800 at CFL= 1.35, but HRK42 did not give any wiggles.



(a) Pressure using n = 800

(b) Pressure using n = 800

Figure 3.5: Pressure plot of Shu-Osher problem at T = 1.8 s

3.7.5 Lid driven cavity

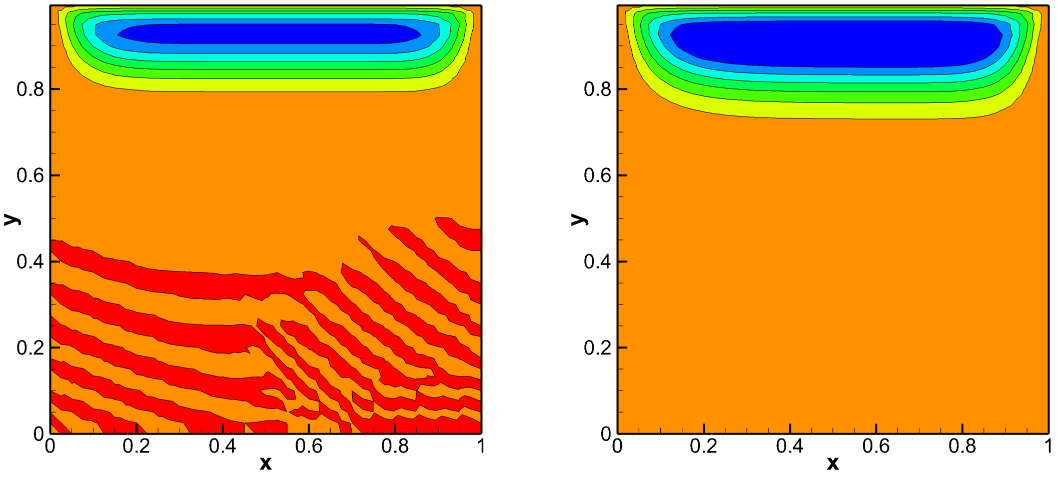
The new scheme has been tested on a 2-D lid-driven cavity problem. The lid-driven cavity is a common test case used to validate the numerical scheme [75]. More information on this test case can be found in [76]. Here, we used the stream function-vorticity (ψ-ω) formulation with the standard second-order central difference in space and our RK schemes for the time integration. A 51×51 grid on a square cavity of unit length and a Reynolds number (Re) = 1000 is used. Figure 3.6a and Figure 3.6b show the stream function of this problem using SSPRK4 and HRK42, respectively.

In SSPRK4 some oscillations are observed when using large Δt (Figure 3.6a). The comparison of the normalized u-velocity profile of the current scheme and the Erturk *et al.* [77] result at $y = 0.5$ line at $t = 150\ s$ is shown in Figure 3.7a. The velocity is normalized using maximum positive u-velocity. Figure 3.7b shows a comparison of the normalized vorticity profile of the current scheme and that presented in Erturk *et al.* [77] at $y = 0.5$ at $t = 150s$. The table 3.3 indicates the maximum CFL number that can be used on the basis of the lid velocity for this problem at $Re = 1000$. The flow is resolved using a 0.025 unit grid scale. All the present schemes are roughly one and a half times more robust than the other methods considered here.

Table 3.3

Maximum CFL number possible for Lid-driven cavity for different methods

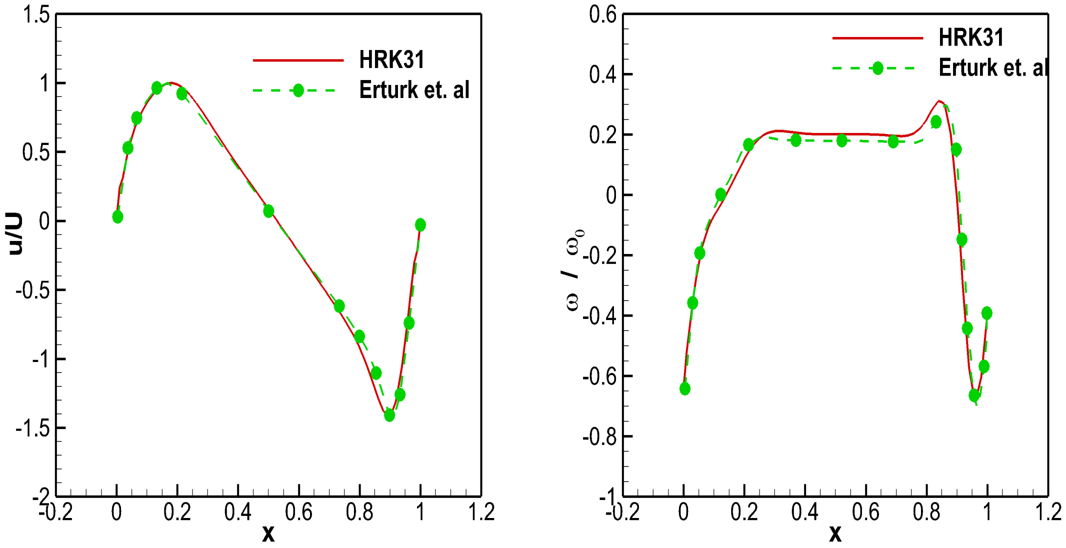
Method	CFL_{max}	**Stability %**
RK3	0.2142	108.02
SSPRK2	0.1983	100
SSPRK3	0.2142	108.01
SSPRK4	0.1824	91.98
HRK31	0.2935	148
HRK42	0.2935	148.00
HRK41	0.2697	136.00



(a) Stream function contour using SSPRK4

(b) Stream function contour using HRK42

Figure 3.6: Solution of Lid-driven cavity using 51×51 grid using $CFL = 0.238$ T = 5 s



(a) u-velocity plot at T = 150 s

(b) Vorticity plot at T = 150 s

Figure 3.7: Solution of Lid-driven cavity using 51×51 grid using HRK31 at T = 150 s

3.7.6 Supersonic flow past a 10° diamond

The results from the various schemes were also examined on a supersonic flow past a 10° diamond with the free-stream Mach number of 1.5. The Roe [14] Riemann solver is used with Harten's entropy fix [78]. Figure 3.8a and figure 3.8b shows the Mach number contour of SSPRK4 and HRK42 schemes. Table 3.4 shows the maximum CFL number possible for different schemes, the corresponding residue value, and the number of iteration required to achieve that. One can notice that HRK42 can go up to CFL = 5.32 without any numerical instability and the scheme converges in 607 iterations. For this test case, SSPRK3 and the low storage RK4 have convergence issues; the convergence stagnates once the residue is of the order of 10^{-5} for CFL number 4.19 and 4.82 respectively.

We have fixed 10000 as maximum iteration and 10^{-9} as the minimum residue for this analysis. Table 3.5 shows the minimum iteration required to achieve the residual value of 10^{-9} and corresponding CFL number. It should be noted that HRK42 can produce results up to CFL=5.32. This means that larger time steps can be allowed and the number of iterations can also be reduced. Although the current method is a second-order method, its convergence rate is better than RK4, which is a fourth-order method. This indicates that higher-order methods need not maintain a high convergence rate in the flows having

Table 3.4
Maximum CFL allowed for different schemes for supersonic flow past a 10° diamond (* indicates that scheme has convergence issue at that CFL)

Schemes	$\boldsymbol{CFL_{max}}$	**Residue**	**Iterations**
HRK31*	4.13	10^{-5}	10000
HRK42	5.32	10^{-9}	607
SSPRK2*	1.56	10^{-6}	10000
SSPRK3*	4.19	10^{-5}	10000
SSPRK4*	4.84	10^{-7}	10000
RK3* low storage	4.21	10^{-5}	10000
RK4* low storage	4.82	10^{-5}	10000

Table 3.5
Minimum iteration required to achieve the residual value of 10^{-9} and corresponding CFL number

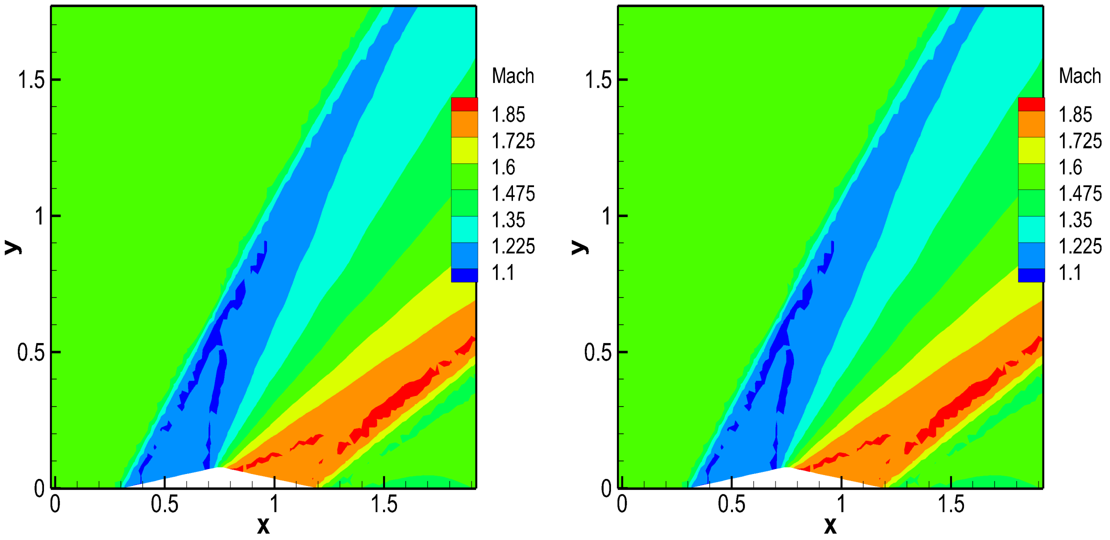
Schemes	$\boldsymbol{CFL}$	**Residue**	**Iterations**	**% wrt SSPRK2**
HRK31	3.98	10^{-9}	811	265.33
HRK42	5.32	10^{-9}	607	354.66
SSPRK2	1.5	10^{-9}	875	100.00
SSPRK3	Convergence issue 10^{-6}			
SSPRK4	Convergence issue 10^{-6}			
RK3 low storage	4.58	10^{-9}	705	305.33
RK4 low storage	4.15	10^{-9}	779	276.66

shocks (non-smooth solution). In these cases, the order of accuracy may be sacrificed for better stability. SSPRK3 and SSPRK4 have some convergence issues for this case; their residual value does not fall below 10^{-9}.

3.7.7 Temporal and spatial accuracy

In this section, the temporal and spatial accuracy of the schemes used are studied. A comparison of the numerical solution with the analytical solution of Sod shock tube problem is carried out, and the errors are estimated. Manhattan error (L_1^t) in temporal discretization is defined as

$$E_m^t = \frac{\sum_{j=1}^{j=m} \sum_{i=1}^{i=n} |u_{ex}(i,j) - u_{nu}(i,j)|}{m}$$



(a) SSPRK4 with CFL = 4.82

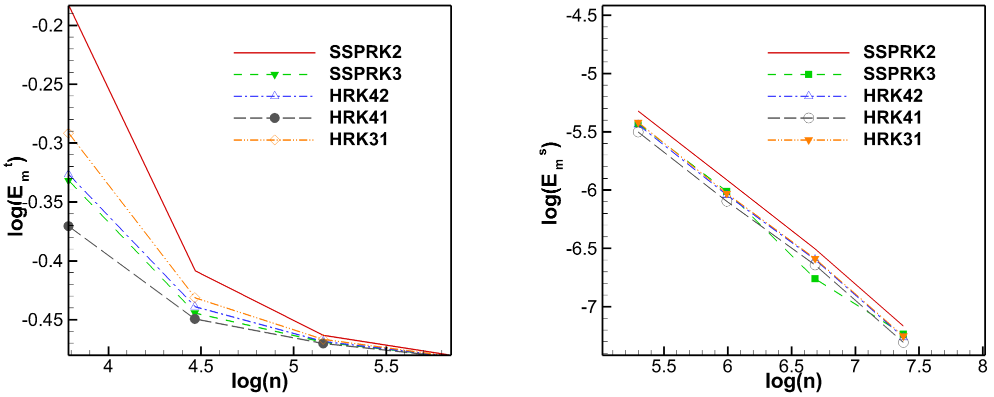
(b) HRK42 with CFL = 5.32

Figure 3.8: Mach number contour of supersonic wedge of 10°

where m is the number of steps in the temporal discretization and n is the number of points in the spatial discretization. $u_{ex}(i, j)$ is the exact solution and $u_{nu}(i, j)$ is the numerical solution. We need $\Delta t >> \Delta x$ to calculate the temporal error correctly such that the temporal error exceeds the spatial error. For an explicit approach, it might be difficult to meet this criterion, since it would reach the stability limit of all the methods considered here. Irrespective of the approach used, temporal discretization errors do not imply uniform convergence, so evaluating the temporal convergence of schemes is not meaningful. Since irrespective of the order of the RK method considered, we do not achieve good temporal convergence, the order of accuracy can be traded for better stability.

We have used absolute error for spatial discretization. Since uniform convergence has been shown by the spatial discretization, we have calculated the convergence rate of spatial discretization. As expected, this convergence rate is more or less the same for all the time integration scheme used. We have used the Sod shock tube problem to check convergence. All the errors are calculated at time $t = 1s$. We define absolute errors as

$$E_m^s = \frac{\sum_{i=1}^{i=n} |u_{ex}(i, t = 1s) - u_{nu}(i, t = 1s)|}{n}$$



(a) Temporal convergence (E_m^t) of different schemes

(b) Spatial convergence (E_m^s) of different schemes

Figure 3.9: Convergence plot of spatial and temporal discretization

Figure 3.9a and figure 3.9b show the spatial and temporal convergence of absolute error for different schemes. It is clear that spatial discretion has a uniform convergence, but temporal discretion does not. There is a comparatively lower initial error for higher-order methods than for lower-order methods. The current schemes reported an absolute error convergence of 0.87 and an RMS error convergence of about 0.44 for spatial discretization.

Popular non-linear stability analysis checks for the *positivity preserving property* [12] and for Lyapunov stability . The first is widely used in numerical analysis, which we have taken care of in our formulation. Although the present analysis is linear, there was some improvement over the classical one in all the problems discussed here. Linear and non-linear problems are considered in the present work. Note that the complete behaviour of the non-linear equation is not reliably predicted in this analysis. We are not aware of good non-linear stability analysis of partial differential equations. Extending the current methods to other RK families such as accelerated Runge-Kutta Methods (ARK) [79] and IRK [46] is straightforward. Extending this to the Embedded RK Methods (ERK) [32] requires some focus in the evolutionary algorithm.

3.8 Summary

Three RK methods with good stability property are proposed. We optimized the stability region here using EA. The test on linear convection equation and shock tube problems showed that HRK42 and HRK41 have a better stability property than the other methods. This means that a higher CFL number is allowed than the other methods considered here. HRK42 outperformed all other schemes in terms of stability, accuracy, and convergence for supersonic flow past 10° diamond airfoil. The results of supersonic flow past a 10° diamond airfoil have shown that the higher-order methods do not always guarantee a higher convergence rate on shock flows; the HRK42 converged faster than RK4. Note that HRK42 is a second-order method and that RK4 is a fourth-order method.

The temporal convergence test of the RK methods considered here does not indicate uniform convergence on flows with shocks, irrespective of the order of the method. Thus, for problems involving shock, compromising the order of accuracy is justified to achieve stability. The current approach may be seen as an alternative to other methods. Moreover, the current schemes are more stable than others. Here, we have used the low storage form of the RK method; it is much faster than the classical and SSPRK methods of the same order in terms of computational speed. Furthermore, the requirements for storage are considerably lower. Although this method is based on linear stability analysis, it can significantly improve stability over the other classical and optimized schemes of the same class.

4

Spatial Discretization Schemes

In this chapter, the spatial discretization used in this work is presented. First, we shall study the general finite volume framework. The spatial discretization can be split into two stages; they are reconstruction procedure (high-resolution scheme), and the Riemann solver. Both are essential for an accurate solution procedure. The main objective of the reconstruction step is to provide accurate input to the Riemann solver. The limiter and WENO schemes are two common reconstruction processes. Finite volume method and conservative finite difference method are mostly used in this work. The standard FVM does not require a non-oscillatory reconstruction step and the Riemann solvers, but they are mandatory for the problem involving shocks. In addition to that, FVM reduces to lower-orders when a poor mesh is used. The reason and proof for the reduction in the order of FVM are also presented here. FVM for high-speed flow is presented in chapter 5 and chapter 6.

4.1 Finite Volume Method

Finite volume method is a powerful numerical tool for solving the conservative form of the equation. It is in the integral form, so it is more suitable for the unstructured mesh, which is essential for handling complex geometry. It was introduced by A.N.Tikhonov and A.A.Samarskii [80] in 1963 for convection-diffusion equation. The method is extended to transient Euler equation in [81, 82]. We refer [83] for the brief history and development of FVM. FVM discretization is used in most of the computational fluid dynamics packages.

Integral form of conservative equation is:

$$\frac{\partial}{\partial t}\int_{\Omega} \mathbf{U} d\Omega = -\oint_{\Gamma} \mathbf{F} d\Gamma \tag{4.1}$$

where Ω is volume and Γ is surface area for the volume. Ω is area and Γ is length of the side of a cell for 2-D test case.

$$\frac{d\bar{\mathbf{U}}_{i,j}}{dt} = -\frac{1}{\Omega}\sum_{sides} \mathbf{F}\Delta\Gamma.\vec{n} \tag{4.2}$$

For square cell, this can be written as

$$\frac{d\bar{\mathbf{U}}_{i,j}}{dt} = -\frac{1}{\Omega}\left(\frac{\mathbf{F}_{i+0.5,j} - \mathbf{F}_{i-0.5,j}}{\Delta x} + \frac{\mathbf{F}_{i,j+0.5} - \mathbf{F}_{i,j-0.5}}{\Delta y}\right) \tag{4.3}$$

$\mathbf{F}_{i+0.5}$ in the above equation has to be obtained using a suitable monotone procedure. FVM method can be broadly divided into cell-centred and vortex-centred formulations. In this work, both vortex-centred and cell-centred are used. We refer [84, 85] for the advantages and the disadvantages of both the methods.

4.1.1 Cell-centered scheme

In the cell-centred scheme, the control volume is the same as a grid cell. Here, flow variables are defined in the centre (centroid) of the cell. We refer figure 4.1 for more details about the discretization.

There are three common flux calculation procedures. First one is based on following

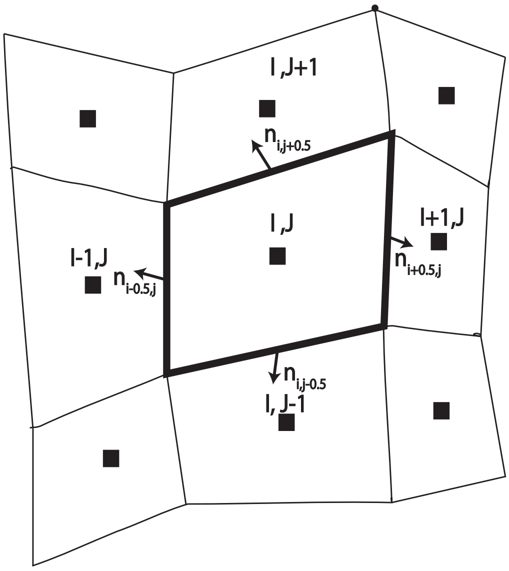


Figure 4.1: Cell-Centered FVM

formulation:

$$(\mathbf{F}\Delta S)_{I+0.5,J} \approx \frac{1}{2}\left[\mathbf{F}(\mathbf{U_{I,J}}) + \mathbf{F}(\mathbf{U_{I+1,J}})\right]\Delta S_{I+0.5,J} \tag{4.4}$$

In eq. 4.4, flux is calculated by taking the average of flux at two adjacent cell-centre. The second approach is instead of taking the average flux of the adjacent cell; we can take the average of the conservative variable or primitive variable and feed that to the flux.

$$(\mathbf{F}\Delta S)_{I+0.5,J} \approx \mathbf{F}(\mathbf{U_{I+0.5,J}})\Delta S_{I+0.5,J} \tag{4.5a}$$

$$\mathbf{U}_{I+0.5,J} \approx \frac{1}{2}\left[\mathbf{U_{I,J}} + \mathbf{U_{I+1,J}}\right]$$

The third one uses a Riemann solver. In this, we feed the left and the right state of the conservative variable or primitive variables to the Riemann solver to obtain the resultant flux.

$$(\mathbf{F}\Delta S)_{I+0.5,J} \approx F(\mathbf{U^L}, \mathbf{U^R}, \mathbf{\Delta S_{I+0.5,J}})$$

$$\mathbf{U}^L = \mathbf{U}(..., \mathbf{U_{I-1,J}}, \mathbf{U_{I,J}}, ...)$$

$$\mathbf{U}^R = \mathbf{U}(..., \mathbf{U_{I,J}}, \mathbf{U_{I+1,J}}, ...)$$

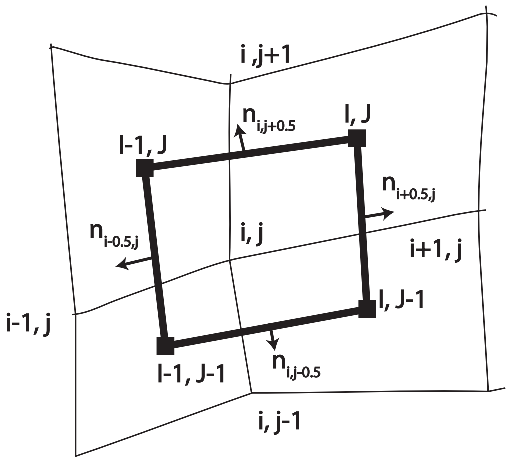


Figure 4.2: Vertex centered FVM

4.1.2 Cell-vertex scheme

In Cell-vertex scheme, we need a relation between flux or conservative variable at the node and the cell-centre. We refer figure 4.2 for a schematic representation of the discretization.

$$\mathbf{U}_{I,J-0.5} = \frac{1}{2}\left[\mathbf{U_{i,j}} + \mathbf{U_{i+1,j}}\right]$$

The flux is calculated using following formulation:

$$(\mathbf{F}_{I+0.5,J}\Delta S)_{I+0.5,J} \approx \mathbf{F}(\mathbf{U_{I,J-0.5}})\Delta S_{I,J-0.5} \tag{4.7}$$

4.2 Higher Order FVM

Higher-order FVM can be framed using two approaches; one is by calculating the left and the right state then calculating the flux at the cell interfaces. The second approach is, calculating the cell interface fluxes without calculating the left and the right states.

4.2.1 Standard FVM

General form of FVM in one-dimensional case is

$$\frac{d\bar{\mathbf{U}}}{dt} + \frac{\mathbf{F_{i+0.5}}(\mathbf{U}) - \mathbf{F_{i-0.5}}(\mathbf{U})}{\Delta x} = 0 \tag{4.8}$$

Because the left-hand side of eq. 4.8 is a cell-average quantity, we should write the right-hand side as a function of cell-averaged quantity.

$$\mathbf{U}_{i+0.5} = \mathbf{G}(..., \bar{\mathbf{U}}_{i-1}, \bar{\mathbf{U}}_{i}, ...) \tag{4.9}$$

Third-order FVM

The relation between cell-average quantity and point-wise quantity is

$$\bar{\mathbf{U}} = \frac{1}{\Omega}\int_{\Omega} \mathbf{U} d\Omega \tag{4.10}$$

The variation of point quantity within a cell can be approximated as a quadratic polynomial. We define averaged quantity in the cell $[i-0.5, i+0.5]$ as

$$\bar{\mathbf{U}}(x) = \frac{1}{\Delta x}\int_{x-0.5\Delta x}^{x+0.5\Delta x} \mathbf{U}(\xi) d\xi \tag{4.11}$$

The variation of $\mathbf{U}(\xi)$ over the local cells can be approximated as quadratic function.

$$\mathbf{U}(\xi) = a + b\xi + c\xi^2 \tag{4.12}$$

Substituting eq. 4.12 in eq. 4.11 and integrating gives

$$\bar{\mathbf{U}}(x) = a + bx + c\left(x^2 + \frac{\Delta x^2}{12}\right) \tag{4.13}$$

$$\bar{\mathbf{U}}_{i-1} = \frac{1}{\Delta x}\int_{-\frac{3\Delta x}{2}}^{-\frac{\Delta x}{2}} u(\xi) d\xi \tag{4.14a}$$

$$\bar{\mathbf{U}}_i = \frac{1}{\Delta x}\int_{-\frac{\Delta x}{2}}^{\frac{\Delta x}{2}} u(\xi)d\xi \tag{4.14b}$$

$$\bar{\mathbf{U}}_{i+1} = \frac{1}{\Delta x}\int_{\frac{\Delta x}{2}}^{\frac{3\Delta x}{2}} u(\xi)d\xi \tag{4.14c}$$

Substituting in eq. 4.12 in eq. 4.14 and solving for a, b and c gives

$$a = \frac{\bar{\mathbf{U}}_{i+1} - 2\bar{\mathbf{U}}_i + \bar{\mathbf{U}}_{i-1}}{2\Delta x^2} \tag{4.15a}$$

$$b = \frac{\bar{\mathbf{U}}_{i+1} - \bar{\mathbf{U}}_{i-1}}{2\Delta x} \tag{4.15b}$$

$$c = \frac{-\bar{\mathbf{U}}_{i-1} + 26\bar{\mathbf{U}}_i - \bar{\mathbf{U}}_{i+1}}{24} \tag{4.15c}$$

Once a, b and c are calculated, the interface values are calculated from

$$\mathbf{U}_{i+0.5} = a\left(\frac{\Delta x}{2}\right)^2 + \frac{\Delta x}{2}b + c \tag{4.16a}$$

$$\mathbf{U}_{i-0.5} = a\left(-\frac{\Delta x}{2}\right)^2 - \frac{\Delta x}{2}b + c \tag{4.16b}$$

Substituting eq. 4.15 in eq. 4.16 gives

$$\mathbf{U}_{i+0.5} = \frac{1}{6}\left(-\bar{\mathbf{U}}_{i-1} + 5\bar{\mathbf{U}}_i + 2\bar{\mathbf{U}}_{i+1}\right) \tag{4.17a}$$

$$\mathbf{U}_{i-0.5} = \frac{1}{6}\left(-\bar{\mathbf{U}}_{i-2} + 5\bar{\mathbf{U}}_{i-1} + 2\bar{\mathbf{U}}_i\right) \tag{4.17b}$$

Substituting eq. 4.17a and eq. 4.17b in eq. 4.8 with $\mathbf{F} = \mathbf{U}$ gives

$$\frac{d\bar{\mathbf{U}}}{dt} = -\frac{1}{\Delta x}\left(3\bar{\mathbf{U}}_i - 4\bar{\mathbf{U}}_{i-1} + \bar{\mathbf{U}}_{i-2}\right) \qquad \mathcal{O}(\Delta x^3) \tag{4.18}$$

4.2.2 FVM based on the left and the right state

In this approach, the left and the right states at the cell-faces are calculated using a cell-averaged interpolation method and the flux is calculated at the faces using a Riemann solver. Please refer [86] for more details. For simplicity, we shall restrict ourself to 1-D flow. The discrete equation is:

$$\frac{d\bar{\mathbf{U}}}{dt} + \frac{\mathbf{F}(\mathbf{U})_{i+0.5} - \mathbf{F}(\mathbf{U})_{i-0.5}}{\Delta x} = 0 \tag{4.19}$$

where $\mathbf{F}(\mathbf{U})_{i+0.5} = \mathbf{F}(\mathbf{U^L_{i+0.5}}, \mathbf{U^R_{i+0.5}})$ and $\mathbf{F}(\mathbf{U})_{i-0.5} = \mathbf{F}(\mathbf{U^L_{i-0.5}}, \mathbf{U^R_{i-0.5}})$.

The cell face value at $i + 0.5$ is a function of the cell averaged quantities in the neighbouring cells

$$\mathbf{U}^L_{i+0.5} = \mathbf{G}(..., \bar{\mathbf{U}}_{i-1}, \bar{\mathbf{U}}_i, ...)$$

$$\mathbf{U}^R_{i+0.5} = \mathbf{G}(..., \bar{\mathbf{U}}_i, \bar{\mathbf{U}}_{i+1}, ...)$$

Higher-order FVM can be formulated based on the way interface flux is calculated using Riemann solver and the order of the cell-averaged polynomial used.

Fourth-Order FVM

A cell-average flow variable is defined as:

$$\bar{\mathbf{U}} = \frac{1}{\Omega}\int_{\Omega} \mathbf{U} d\Omega$$

where Ω is the length of the cell in the one-dimensional test case, and $\mathbf{U}$ can be approximated by any polynomials. For a fourth-order scheme using this approach, we need quadratic polynomial, which is given in eq. 4.21.

$$\mathbf{U}(\xi) = a + b\xi + c\xi^2 \tag{4.21}$$

The procedure is same as third order FVM till eq. 4.15. Then the left and the right states at the cell-faces are

$$\bar{\mathbf{U}}^L_{i+0.5} = \frac{2\bar{\mathbf{U}}_{i+1} + 5\bar{\mathbf{U}}_i - \bar{\mathbf{U}}_{i-1}}{6} \tag{4.22a}$$

$$\bar{\mathbf{U}}^L_{i-0.5} = \frac{2\bar{\mathbf{U}}_i + 5\bar{\mathbf{U}}_{i-1} - \bar{\mathbf{U}}_{i-2}}{6} \tag{4.22b}$$

$$\bar{\mathbf{U}}^R_{i+0.5} = \frac{-\bar{\mathbf{U}}_{i+2} + 5\bar{\mathbf{U}}_{i+1} + 2\bar{\mathbf{U}}_i}{6} \tag{4.22c}$$

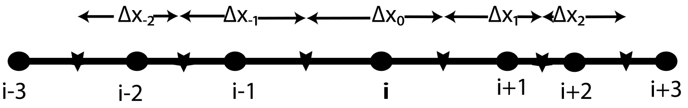


Figure 4.3: Grid nomenclature used for unstructured grid

$$\bar{\mathbf{U}}^{R}_{i-0.5} = \frac{-\bar{\mathbf{U}}_{i+1} + 5\bar{\mathbf{U}}_{i} + 2\bar{\mathbf{U}}_{i-1}}{6} \tag{4.22d}$$

The flux is calculated by

$$\mathbf{F}_{i+0.5} = \frac{1}{2}\left[\mathbf{F}(\bar{\mathbf{U}}^{L}_{i+0.5}) + \mathbf{F}(\bar{\mathbf{U}}^{R}_{i+0.5})\right] \tag{4.23a}$$

$$\mathbf{F}_{i-0.5} = \frac{1}{2}\left[\mathbf{F}(\bar{\mathbf{U}}^{L}_{i-0.5}) + \mathbf{F}(\bar{\mathbf{U}}^{R}_{i-0.5})\right] \tag{4.23b}$$

Substituting eq. 4.23 in eq. 4.19 with $\mathbf{F} = \mathbf{U}$ gives

$$\frac{\mathrm{d}\bar{\mathbf{U}}}{\mathrm{d}t} = -\frac{1}{12\Delta x}\left(-\bar{\mathbf{U}}_{i+2} + 8\bar{\mathbf{U}}_{i+1} - 8\bar{\mathbf{U}}_{i-1} - \bar{\mathbf{U}}_{i-2}\right) \qquad \mathcal{O}(\Delta x^{4}) \tag{4.24}$$

The right-hand side of the eq. 4.24 is a fourth-order central difference scheme. It is difficult to make the clear cut difference between FVM and conservative-FDM because both are based on the integral formulation. In this thesis we differentiate FVM and FDM based on how we evaluate cell-interface value. Please note that this is not universally used one but used in the papers of Prof. Chi-Wang Shu [87] and we are also following that. Based on that, in FVM we calculate cell-interface value using the flux calculated from the Riemann solver and the Riemann solver calculate flux using the left and the right state. In conservative FDM, flux is calculated using Riemann solver before calculating left and the right state [7] of $i + 0.5$ and $i - 0.5$ values.

4.3 FVM on non-uniform mesh

The grid nomenclature used for FVM for the non-uniform mesh is shown in figure 4.3. The discretization used for the unstructured mesh used in this section is not used in other chapters. The intention of this chapter is to justify the reduction of the order of accuracy of the conservative discretization on a arbitrarily varying non-uniform grid.

Volume averaged value at different cells are given as

$$\bar{\mathbf{U}}_{i-2} = \frac{1}{\Delta x_{-2}} \int_{-(\frac{\Delta x_0}{2}+\Delta x_{-1}+\Delta x_{-2})}^{-(\frac{\Delta x_0}{2}+\Delta x_{-1})} u(\xi)d\xi \tag{4.25a}$$

$$\bar{\mathbf{U}}_{i-1} = \frac{1}{\Delta x_{-1}} \int_{-(\frac{\Delta x_0}{2}+\Delta x_{-1})}^{-\frac{\Delta x_0}{2}} u(\xi)d\xi \tag{4.25b}$$

$$\bar{\mathbf{U}}_{i} = \frac{1}{\Delta x_{0}} \int_{-\frac{\Delta x_0}{2}}^{\frac{\Delta x_0}{2}} u(\xi)d\xi \tag{4.25c}$$

$$\bar{\mathbf{U}}_{i+1} = \frac{1}{\Delta x_{-1}} \int_{\frac{\Delta x_0}{2}}^{(\frac{\Delta x_0}{2}+\Delta x_{1})} u(\xi)d\xi \tag{4.25d}$$

$$\bar{\mathbf{U}}_{i+2} = \frac{1}{\Delta x_{-2}} \int_{(\frac{\Delta x_0}{2}+\Delta x_{1})}^{(\frac{\Delta x_0}{2}+\Delta x_{1}+\Delta x_{2})} u(\xi)d\xi \tag{4.25e}$$

Extension of standard FVM to non-uniform mesh

Here, we shall derive a third-order conservative discretization based on the procedure described in [88, 89] for a uniform grid where the derivative is approximated as

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x} = \frac{\mathbf{U}_{i+0.5} - \mathbf{U}_{i-0.5}}{\Delta x_0} \tag{4.26}$$

where a, b and c in eq. 4.13 are derived in terms of $\bar{\mathbf{U}}_{i-1}$, $\bar{\mathbf{U}}_{i}$ and $\bar{\mathbf{U}}_{i+1}$ using eq. (4.25). On substituting these in eq. 4.21 with $x = \frac{\Delta x_0}{2}$, we get

$$\mathbf{U}_{i+0.5} = a\left(\frac{\Delta x_0}{2}\right)^2 + b\left(\frac{\Delta x_0}{2}\right) + c \tag{4.27}$$

Similarly, a, b and c in eq. 4.13 is derived in terms of $\bar{\mathbf{U}}_{i-2}$, $\bar{\mathbf{U}}_{i-1}$ and $\bar{\mathbf{U}}_{i}$ using eq. 4.25. On substituting in eq. 4.21 with $x = -\frac{\Delta x_0}{2}$, we get

$$\mathbf{U}_{i-0.5} = a\left(-\frac{\Delta x_0}{2}\right)^2 + b\left(-\frac{\Delta x_0}{2}\right) + c \tag{4.28}$$

Extension of FVM based on the left and the right state to non-uniform grid

We can derive a, b and c in eq. 4.13 in terms of $\bar{\mathbf{U}}_{i-1}$, $\bar{\mathbf{U}}_i$ and $\bar{\mathbf{U}}_{i+1}$ using eq. (4.25) and on substituting in eq. 4.21, we get

$$\mathbf{U}^L_{i+0.5} = a\left(\frac{\Delta x_0}{2}\right)^2 + b\left(\frac{\Delta x_0}{2}\right) + c \tag{4.29}$$

Similarly the right state for $i + 0.5$ is derived in terms of $\bar{\mathbf{U}}_i$, $\bar{\mathbf{U}}_{i+1}$ and $\bar{\mathbf{U}}_{i+2}$ using eq. (4.25) and on substituting in eq. 4.21, we get

$$\mathbf{U}^R_{i+0.5} = a\left(\frac{\Delta x_0}{2}\right)^2 + b\left(\frac{\Delta x_0}{2}\right) + c \tag{4.30}$$

The right state for $i - 0.5$ is derived in terms of $\bar{\mathbf{U}}_{i-1}$, $\bar{\mathbf{U}}_i$ and $\bar{\mathbf{U}}_{i+1}$ using eq. 4.25 and on substituting in eq. 4.21, we get

$$\mathbf{U}^R_{i-0.5} = a\left(\frac{-\Delta x_0}{2}\right)^2 + b\left(-\frac{\Delta x_0}{2}\right) + c \tag{4.31}$$

The left state for $i - 0.5$ is derived in terms of $\bar{\mathbf{U}}_{i-2}$, $\bar{\mathbf{U}}_{i-1}$ and $\bar{\mathbf{U}}_i$ using eq. 4.25 and on substituting in eq. 4.21, we get

$$\mathbf{U}^L_{i-0.5} = a\left(\frac{-\Delta x_0}{2}\right)^2 + b\left(-\frac{\Delta x_0}{2}\right) + c \tag{4.32}$$

The derivative for non-uniform mesh can be approximated as

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x} = \frac{\mathbf{U}_{i+0.5} - \mathbf{U}_{i-0.5}}{\Delta x_0} \tag{4.33}$$

$$\mathbf{U}_{i+0.5} = \frac{\mathbf{U}^L_{i+0.5} + \mathbf{U}^R_{i+0.5}}{2} \qquad \mathbf{U}_{i-0.5} = \frac{\mathbf{U}^L_{i-0.5} + \mathbf{U}^R_{i-0.5}}{2}$$

This procedure is an extension of higher order conservative discretization in uniform mesh for the convection equation described in [86]. The final formulation is given in eq. A.1. Eq. A.1 reduced to the fourth-order central difference scheme if we substitute

$\Delta x_{-2} = \Delta x$, $\Delta x_2 = \Delta x$, $\Delta x_{-1} = \Delta x$ and $\Delta x_1 = \Delta x$. The fourth-order central difference scheme is

$$\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x} = \frac{1}{12\Delta x}(-\bar{\mathbf{U}}_{j+2} + 8\bar{\mathbf{U}}_{j+1} - 8\bar{\mathbf{U}}_{j-1} + \bar{\mathbf{U}}_{j-2}) \qquad \mathcal{O}(\Delta x^4)$$

Eq. A.1 is denoted here as N-FVM-CD5 because this is a non-uniform finite volume scheme. Here, CD5 does not mean a fifth-order accurate scheme, but it is a five-point scheme. Three-point N-FVM-CD3 ($i-1$, i and $i+1$) on arbitrarily varying non-uniform grid is given in eq. 4.34. This reduces to the standard second-order scheme when a uniform grid is used.

$$\frac{\mathrm{d}u}{\mathrm{d}x} = \frac{\bar{\mathbf{U}}_{i+1}\Delta x_0 - \bar{\mathbf{U}}_{i-1}\Delta x_0 - \bar{\mathbf{U}}_i\Delta x_{-1} + \bar{\mathbf{U}}_{i+1}\Delta x_{-1} + \bar{\mathbf{U}}_i\Delta x_1 - \bar{\mathbf{U}}_{i-1}\Delta x_1}{(\Delta x_0 + \Delta x_{-1})(\Delta x_0 + \Delta x_1)} \qquad \mathcal{O}(\Delta x) \tag{4.34}$$

Another three-point upwind N-FVM-UP3 framed using $i-2$, $i-1$ and i on a arbitrarily varying non-uniform grid is given in eq. 4.35.

$$\frac{\mathrm{d}u}{\mathrm{d}x} = \frac{\bar{\mathbf{U}}_i - \frac{(2\bar{\mathbf{U}}_{i-1}\Delta x_{-1} + \bar{\mathbf{U}}_{i-1}\Delta x_{-2} - \bar{\mathbf{U}}_{i-2}\Delta x_{-1})}{(\Delta x_{-1} + \Delta x_{-2})} + \frac{(\Delta x_0(\bar{\mathbf{U}}_i - \bar{\mathbf{U}}_{i-1}))}{(\Delta x_0 + \Delta x_{-1})}}{\Delta x_0} \qquad \mathcal{O}(\Delta x) \tag{4.35}$$

Eq. 4.35 reduced to the second-order upwind scheme when a uniform grid is used.

4.4 Challenges in extending higher-order FVM to arbitrarily varying non-uniform grid

Since FVM uses the integral equations, it can be used on the non-uniform and the unstructured mesh. However, when it is applied to an arbitrarily varying non-uniform grid, it faces some difficulties. FVM shows a drop in the order of convergence for non-uniform and unstructured mesh [90, 91, 92, 93]. When a discontinuity or singularity is present in the solution [94], the accuracy of a numerical scheme decreases to lower-order. When a higher-order method developed for a uniform grid is applied to a highly stretched grid, it may result in a lower-order solution [95].

The distinct source of errors in FVM is, errors due to poor mesh, interpolation/extrapolation, flux calculation, and others. Some of the significant causes of mesh error are non-orthogonality in the mesh, mesh skewness, and mesh non-uniformity [96]. In one of the test cases based on mesh adaptation in [97], the error increased due to the lack of orthogonality and mesh-to-flow alignment. The quality of the mesh plays a vital role in the accuracy of the solution. Another error source is the artificial viscosity introduced by Riemann solvers in calculating the flux at a cell interface. Also contributing to error is truncation error in the approximation of continuous functions in the discrete space.

Mesh shape plays a key role in the solution's accuracy. For example, in Cartesian geometry, a square mesh has a comparatively lower error than a triangle mesh. For discretization errors for different types of mesh, please refer to [96]. Various error estimates have been derived for FVM. Face Residual Error Estimator (FREE) [96], Numerical Diffusion Ratio (NDR) [98], and Richardson's extrapolation [99] are some of them. To measure local solution error [100], the imbalance between kinetic energy and angular momentum between the cells can be used. It is easy to obtain the convergence or order of accuracy of conservative discretization on a uniform mesh. However, obtaining the same for a non-uniform mesh is not easy.

The order of accuracy of the conservative schemes is discussed in sections 4.4.1. We have also studied the reasons behind the reduction in the order of accuracy in the conservative discretization when the quality of the mesh is poor. A theoretical evidence showing the need for a smooth variation in the FVM mesh distribution is provided.

4.4.1 Taylor series order analysis

The order analysis of a scheme can be performed using the polynomial basis [101]. We refer [102] for the derivation of FDM using a polynomial basis. First, we will review the polynomial order analysis for FDM derived for uniform grids, and then we will check the order analysis for the N-FVM developed for the non-uniform grid.

The fourth-order central scheme (CD4) is given as

$$\frac{\mathrm{d}U}{\mathrm{d}x} = \frac{1}{12\Delta x}(-U_{j+2} + 8U_{j+1} - 8U_{j-1} + U_{j-2}) \qquad \mathcal{O}(\Delta x^4) \tag{4.36}$$

As the CD4 scheme spans over five-points, maximum order possible is four for the first derivative approximation. The sum of the coefficients should be zero for the basic consistency test or zeroth-order accuracy, i.e., (-1 + 8 -8 +1) = 0.

1. First-order check:

   - First we start with the first-order polynomial, that is $U(x) = a_0 + a_1 x$. Its derivative at x =0 is $\frac{\mathrm{d}U}{\mathrm{d}x} = a_1$.
   - For the first-order polynomial, $U_{i-2} = U(-2\Delta x) = a_0 + a_1(-2\Delta x)$, $U_{i-1} = U(-\Delta x) = a_0 + a_1(-\Delta x)$, $U_i = U(0) = a_0 + a_1(0)$, $U_{i+1} = U(\Delta x) = a_0 + a_1(\Delta x)$ and $U_{i+2} = U(2\Delta x) = a_0 + a_1(2\Delta x)$.
   - On substituting the values of U_{i-2}, U_{i-1}, U_i, U_{i+1} and U_{i+2} in CD4 scheme and simplifying, it gives $\frac{\mathrm{d}U}{\mathrm{d}x} = a_1$. This is equivalent to the derivative of $U(x) = a_0 + a_1 x$, so it is at least first-order.

2. Second-order check:

   - Increment the order of polynomial $U(x) = a_0 + a_1 x + a_2 x^2$. Its derivative at $x = 0$ is $\frac{\mathrm{d}U}{\mathrm{d}x} = a_1$.
   - For the second-order polynomial, $U_{i-2} = U(-2\Delta x) = a_0 + a_1(-2\Delta x) + a_2(-2\Delta x)^2$, $U_{i-1} = U(-\Delta x) = a_0 + a_1(-\Delta x) + a_2(-\Delta x)^2$, $U_i = U(0) = a_0 + a_1(0) + a_2(0)^2$, $U_{i+1} = U(\Delta x) = a_0 + a_1(\Delta x) + a_2(\Delta x)^2$ and $U_{i+2} = U(2\Delta x) = a_0 + a_1(2\Delta x) + a_2(2\Delta x)^2$.
   - On substituting the values of U_{i-2}, U_{i-1}, U_i, U_{i+1} and U_{i+2} in CD4 scheme and simplifying, it gives $\frac{\mathrm{d}U}{\mathrm{d}x} = a_1$. This is equivalent to the derivative of $U(x) = a_0 + a_1 x$, so it is at least second-order.

3. Increment the order of polynomial and follow the steps till $\frac{\mathrm{d}U}{\mathrm{d}x} \neq a_1$. If $n + 1$ is the order of polynomial at which $\frac{\mathrm{d}U}{\mathrm{d}x} = a_1$, then the order of the scheme is n. In this case when $U(x) = a_0 + a_1 x + a_2 x^2 + ...a_5 x^5$, it will not satisfy $\frac{\mathrm{d}U}{\mathrm{d}x} = a_1$ condition so it is a fourth-order scheme.

Although the above approach is longer than finding the order of accuracy by expanding the Taylor series, this is relatively simple for the unstructured mesh. A symbolic computation is used to evaluate the truncation error and it is faster when compared to the term's expansion in the Taylor series. First, we create a third-order finite difference scheme for the non-uniform grid shown in figure 4.3 and check the scheme's order of accuracy.

For the grid shown in figure 4.3, the distance of $\mathbf{U}_{i-1}$ and $\mathbf{U}_{i-2}$ from origin is $\frac{\Delta x_0 + \Delta x{-1}}{2}$ and $\frac{-(\Delta x_0+\Delta x_{-1})}{2} - \frac{(\Delta x_{-2}+\Delta x_{-1})}{2}$ respectively. To frame a second-order finite difference scheme, we need second-order polynomial, $U(x) = a_0 + a_1 x + a_2 x^2$.

Then U_i, U_{i-1} and U_{i-2} are

$$U_i = U(0) \tag{4.37a}$$

$$U_{i-1} = U\left(\frac{\Delta x_0 + \Delta x{-1}}{2}\right) \tag{4.37b}$$

$$U_{i-2} = U\left(\frac{-(\Delta x_0 + \Delta x_{-1})}{2} - \frac{(\Delta x_{-2} + \Delta x_{-1})}{2}\right) \tag{4.37c}$$

Eq. 4.37 are functions of a_0, a_1, a_2, U_i, U_{i-1}, and U_{i-2}. We should solve for a_0, a_1 and a_2, where $\frac{\mathrm{d}U}{\mathrm{d}x} = a_1$. Third-order finite difference scheme is given in eq. A.3.

lemma 4.4.1. *Show that the order of accuracy of the three-point finite difference scheme in eq. A.3 is two.*

Proof. The condition for consistency property or zeroth order accuracy is $\sum a_i = 0$. If we substitute $\bar{\mathbf{U}}_{i-2} = 1$; $\bar{\mathbf{U}}_{i-1} = 1$; $\bar{\mathbf{U}}_i = 1$; $\bar{\mathbf{U}}_{i+1} = 1$; $\bar{\mathbf{U}}_{i+2} = 1$ in eq. A.3 it will become zero so it is at least zeroth-order accurate.

Consider the solution is linear then $\mathbf{U}(x) = a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-2} = \mathbf{U}(-((\Delta x_{-1} + \Delta x_0)/2 + (\Delta x_{-1} + \Delta x_{-2})/2))$; $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1} + \Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$; $\bar{\mathbf{U}}_{i+1} = \mathbf{U}((\Delta x_1 + \Delta x_0)/2)$; $\bar{\mathbf{U}}_{i+2} = \mathbf{U}((\Delta x_1 + \Delta x_0)/2 + (\Delta x_1 + \Delta x_2)/2)$ in eq. A.3, we get a_1 so it is at least first-order accurate.

Consider the solution is quadratic then $\mathbf{U}(x) = a_2x^2 + a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-2} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2+(\Delta x_{-1}+\Delta x_{-2})/2))$; $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$; $\bar{\mathbf{U}}_{i+1} = \mathbf{U}((\Delta x_1+\Delta x_0)/2)$; $\bar{\mathbf{U}}_{i+2} = \mathbf{U}((\Delta x_1+\Delta x_0)/2+(\Delta x_1+\Delta x_2)/2)$ in eq. A.3, we get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\Big|_{x=0} = a_1$, so it is at least second order accurate.

Consider the solution is cubic then $\mathbf{U}(x) = a_3x^3 + a_2x^2 + a_1\mathbf{x} + a_0$. If we substitute

$\bar{\mathbf{U}}_{i-2} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2+(\Delta x_{-1}+\Delta x_{-2})/2))$; $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$; $\bar{\mathbf{U}}_{i+1} = \mathbf{U}((\Delta x_1+\Delta x_0)/2)$; $\bar{\mathbf{U}}_{i+2} = \mathbf{U}((\Delta x_1+\Delta x_0)/2+(\Delta x_1+\Delta x_2)/2)$ in eq. A.3, we get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\big|_{x=0} = a_1$, so it is at least third-order accurate.

Consider the solution is quartic then $\mathbf{U}(x) = a_4x^4 + a_3x^3 + a_2x^2 + a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-2} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2+(\Delta x_{-1}+\Delta x_{-2})/2))$; $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$; $\bar{\mathbf{U}}_{i+1} = \mathbf{U}((\Delta x_1+\Delta x_0)/2)$; $\bar{\mathbf{U}}_{i+2} = \mathbf{U}((\Delta x_1+\Delta x_0)/2+(\Delta x_1+\Delta x_2)/2)$ in eq. A.3, we get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\big|_{x=0} \neq a_1$, so it is not a fourth-order accurate scheme and it is third-order accurate.

■

Based on the order analysis using Taylor series, it is proved that eq. A.3 is a third-order accurate scheme. Using this approach, we shall check the order of N-FVM-CD5(eq. A.1).

lemma 4.4.2. *Conservative discretization formed using $i-2$, $i-1$, i, $i+1$ and $i+2$ points, N-FVM-CD5(eq. A.2), cannot give the required order of accuracy on a non-uniform grid.*

Proof. For consistency or zeroth-order accuracy, the summation of coefficients should be zero $\sum a_i = 0$. For eq. A.2, the sum of the coefficient is zero so eq. A.2 is at least zeroth-order accurate.

Consider the solution is linear then $\mathbf{U}(x) = a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-2} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2+(\Delta x_{-1}+\Delta x_{-2})/2))$; $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{f}(0)$; $\bar{\mathbf{U}}_{i+1} = \mathbf{U}((\Delta x_1+\Delta x_0)/2)$; $\bar{\mathbf{U}}_{i+2} = \mathbf{U}((\Delta x_1+\Delta x_0)/2+(\Delta x_1+\Delta x_2)/2)$ in eq. A.2 we will get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\big|_{x=0} = a_1$, so it is at least first-order accurate.

Consider the solution is quadratic then $\mathbf{U}(x) = a_2\mathbf{x}^2 + a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-2} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2+(\Delta x_{-1}+\Delta x_{-2})/2))$; $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1}+\Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$; $\bar{\mathbf{U}}_{i+1} = \mathbf{U}((\Delta x_1+\Delta x_0)/2)$; $\bar{\mathbf{U}}_{i+2} = \mathbf{f}((\Delta x_1+\Delta x_0)/2+(\Delta x_1+\Delta x_2)/2)$ in eq. A.2, we get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\big|_{x=0} \neq a_1$. So eq. A.2 is first-order accurate. ■

Above proof shows that the order of N-FVM-CD5(eq. A.2) is one, but we do get better convergence for the uniformly stretched grid. Here, we shall explore the reasons behind it. In a Arithmetic Progression (AP) grid the grid point is defined by $x_{i+1} = x_i + d \times (i-1) + \Delta x_0$, where Δx_0 is grid size of the first cell, d is a common difference. In the case of AP grid, the truncation error of N-FVM5-CD5 is as follows:

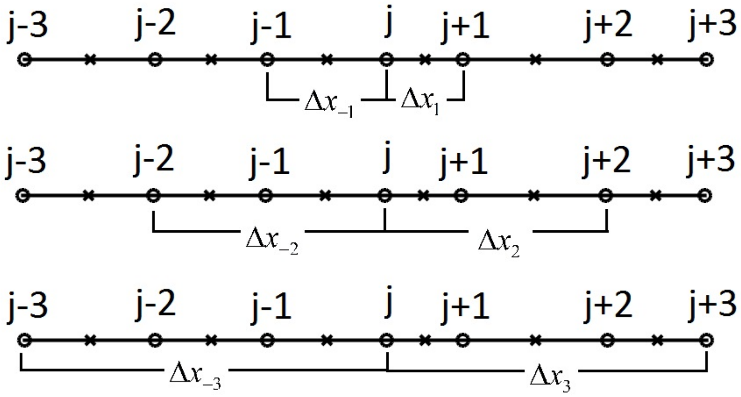


Figure 4.4: Grid nomenclature used for unstructured grid in [103]

$-(\frac{d}{12}\frac{\mathrm{d}^2\mathbf{U}}{\mathrm{d}x^2})-(\frac{11d^2}{72}\frac{\mathrm{d}^3\mathbf{U}}{\mathrm{d}x^3})-(\frac{129d^3}{64}\frac{\mathrm{d}^4\mathbf{U}}{\mathrm{d}x^4})-(\frac{31ad^2}{24}\frac{\mathrm{d}^4\mathbf{U}}{\mathrm{d}x^4})-(\frac{31a^2d}{144}\frac{\mathrm{d}^4\mathbf{U}}{\mathrm{d}x^4})-(\frac{a^4}{30}\frac{\mathrm{d}^5\mathbf{U}}{\mathrm{d}x^5})-(\frac{26131d^4}{5760}\frac{\mathrm{d}^5\mathbf{U}}{\mathrm{d}x^5})-$ $(\frac{1153ad^3}{240}\frac{\mathrm{d}^5\mathbf{U}}{\mathrm{d}x^5})-(\frac{2a^3d}{5}\frac{\mathrm{d}^5\mathbf{U}}{\mathrm{d}x^5})-(\frac{2881a^2d^2}{1440}\frac{\mathrm{d}^5\mathbf{U}}{\mathrm{d}x^5})$

So, N-FVM is a first-order accurate on AP grids. If d is a very small value, it may give third-order accurate scheme on the numerical test cases. This could be a reason for FVM method to show one order lower convergence than the theoretical maximum convergence rate on the uniform varying grid. If d is very small, it can give the theoretical maximum convergence. The second derivative terms in the truncation error is $-(\frac{d}{12}\frac{\mathrm{d}^2\mathbf{U}}{\mathrm{d}x^2})$. This term can act as a numerical viscosity to reduce/increase oscillations based on the CFL number and the diffusion number of the problem.

If we enforce, $\mathbf{U}_{i-0.5}(x) = \mathbf{U}_{i+0.5}(x-\Delta x)$, it leads to a second-order accurate scheme because of the central nature of the stencil. It is worth noting that, in case of uniform mesh, this approximation will lead to maximum order accurate scheme for this stencil. For example, if $\mathbf{U}_{i+0.5} = \frac{1}{6}\left(2\bar{\mathbf{U}}_{i-2} - 7\bar{\mathbf{U}}_{i-1} + 11\bar{\mathbf{U}}_{i}\right)$, then $\mathbf{U}_{i-0.5} = \frac{1}{6}\left(2\bar{\mathbf{U}}_{i-3} - 7\bar{\mathbf{U}}_{i-2} + 11\bar{\mathbf{U}}_{i-1}\right)$. For $\mathbf{U}_{i-0.5}$ co-efficient are same as $\mathbf{U}_{i+0.5}$, except the index (i) reduced by one. This we call as *index shifting property*.

If we use the grid nomenclature shown in figure 4.4 and derived a formulation for conservative discretization, the magnitude of oscillations is slightly higher. This could be because nodes are located at the midpoint of two consecutive cells used in the figure 4.3. We refer [103] for more details about conservative schemes for grid nomenclature shown in figure 4.4. Proof for the first-order convergence of other lower-order FVM on the non-uniform mesh is given below.

lemma 4.4.3. *The order of accuracy of N-FVM-CD3 (eq. 4.34) formed using $i-1$, i and $i+1$ is one.*

Proof. For consistency or zeroth-order accuracy, the summation of co-efficients should be zero $\sum a_i = 0$ and eq. 4.34 satisfies that property.

Consider the solution is linear; then, $\mathbf{U}(x) = a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1} + \Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$; $\bar{\mathbf{U}}_{i+1} = \mathbf{U}((\Delta x_1 + \Delta x_0)/2)$ in eq. 4.34 we get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\Big|_{x=0} = a_1$; so it is at least first-order accurate.

Consider the solution is quadratic; then, $\mathbf{U}(x) = a_2x^2 + a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1} + \Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$; $\bar{\mathbf{U}}_{i+1} = \mathbf{U}((\Delta x_1 + \Delta x_0)/2)$ in eq. 4.34, we get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\Big|_{x=0} \neq a_1$; so it is a first-order accurate scheme. ■

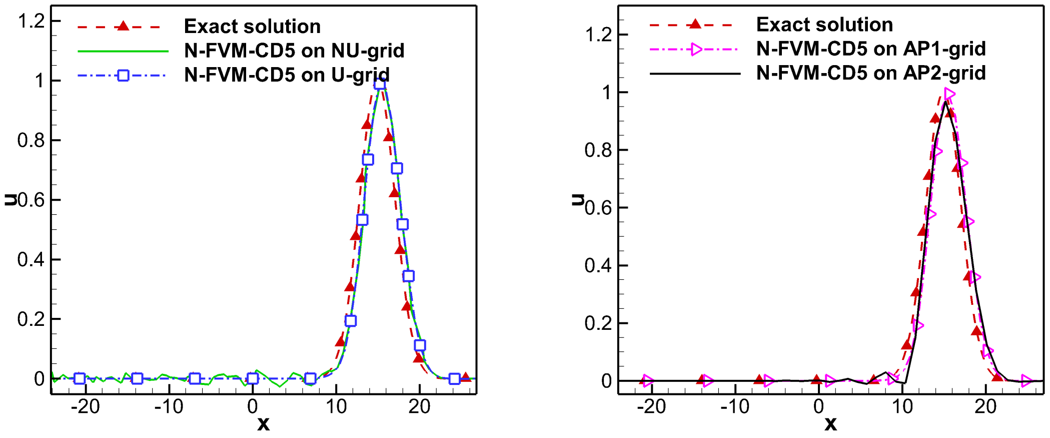
lemma 4.4.4. *The accuracy of N-FVM-UP3 (eq. 4.35) formed using $i-2$, $i-1$ and i upwind scheme is first-order.*

Proof. For consistency or zeroth-order accuracy, the summation of co-efficients should be zero $\sum a_i = 0$, and eq. 4.35 satisfies that property.

Consider the solution is linear; then, $\mathbf{U}(x) = a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-2} = \mathbf{U}(-((\Delta x_{-1} + \Delta x_0)/2 + (\Delta x_{-1} + \Delta x_{-2})/2))$; $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1} + \Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$ in eq. 4.34, we will get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\Big|_{x=0} = a_1$; so it is at least first-order accurate.

Consider the solution is quadratic; then, $\mathbf{U}(x) = a_2x^2 + a_1\mathbf{x} + a_0$. If we substitute $\bar{\mathbf{U}}_{i-2} = \mathbf{U}(-((\Delta x_{-1} + \Delta x_0)/2 + (\Delta x_{-1} + \Delta x_{-2})/2))$; $\bar{\mathbf{U}}_{i-1} = \mathbf{U}(-((\Delta x_{-1} + \Delta x_0)/2))$; $\bar{\mathbf{U}}_i = \mathbf{U}(0)$ in eq. 4.35, we will get $\frac{\mathrm{d}\mathbf{U}}{\mathrm{d}x}\Big|_{x=0} \neq a_1$; so it is a first-order accurate scheme. ■

When using a non-uniform grid, the fundamental conservative discretization of different orders only satisfies Taylor sequence up to the first-order term. This has been proved by Lemma 4.4.2, lemma 4.4.3 and lemma 4.4.4. The conservative cell-average discretization loses its accuracy on a non-uniform grid but maintains its accuracy and order on a uniform grid.



(a) N-FVM on uniform and arbitrarily varying non-uniform grid

(b) N-FVM on uniform varying arithmetic progression grids with different stretching

Figure 4.5: N-FVM on different grid configurations

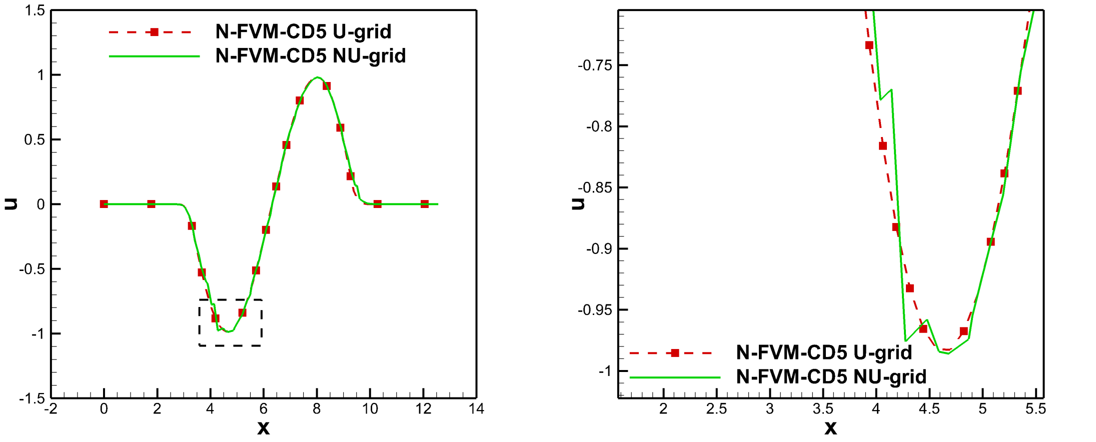
4.4.2 Numerical test cases

Linear convection equation

One-dimensional convection equation is solved using eq. A.1 for spatial discretization and the low storage RK4 for time integration. The initial condition is

$$u(x,0) = \exp\left[-\left(\frac{x}{3}\right)^2\right] \tag{4.38}$$

The problem is solved upto $T = 15$ s with CFL number 0.6. The solution of the equation in the uniform grid and the non-uniform arbitrary varying grid is shown in figure 4.5. N-FVM-CD5 is a non-uniform finite volume method given in eq. (A.1), NU-grid is arbitrarily varying non-uniform grid; U-grid is uniform grid; AP1-grid is arithmetic progression(AP) grid with $d = 0.001$. AP2-grid is a AP grid with $d = 0.01$. N-FVM-CD5 produced oscillations on arbitrarily varying non-uniform grid and did not produce any oscillations on the uniform grid as shown in figure 4.5a. It is clear from figure 4.5b that the AP grid with small stretching values did not produce oscillations, but when the stretching is high, it produces oscillations.



(a) Velocity plot of the viscous Burgers equation **(b)** Zoomed view of solution of Burgers equation

Figure 4.6: Solution of the viscous Burgers equation

4.4.3 Viscous Burgers equation

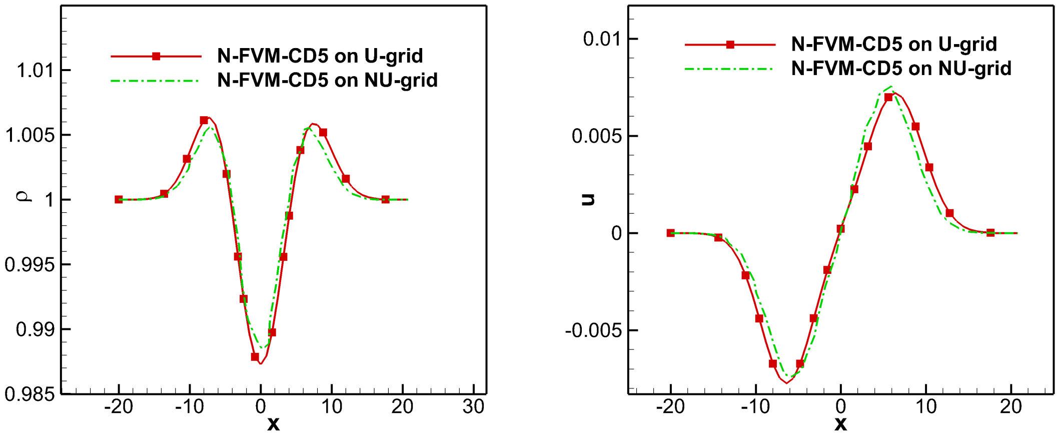
One-dimensional viscous Burgers equation is solved using $n = 100$ grid points up to flow time 0.25 s, over the domain 0 to 4π. The equation is

$$\frac{\partial u}{\partial t} + u\frac{\partial u}{\partial x} = 0.1\frac{\partial^2 u}{\partial x^2}$$

The initial condition used is

$$u = \begin{cases} 0 & 0 \leq x \leq \pi \\ \sin(x) & \pi < x < 3\pi \\ 0 & 3\pi \leq x \leq 4\pi \end{cases}$$

The convection term is discretized using N-FVM-CD5 and the diffusion term is discretized using a second-order central difference scheme using $CFL = 0.7$. The viscous Burgers equation solution is shown in the figure 4.6. The figure shows that when a non-uniform grid is used, N-FVM-CD5 generates oscillations and does not generate oscillations when a uniform grid is used.



(a) Density

(b) Pressure

Figure 4.7: Solution of Euler equation on different grid configuration

4.4.4 Euler equation

One-dimensional Euler equation is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{E}}{\partial x} = 0$$

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho e_t \end{bmatrix} \mathbf{E} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho e_t + p) \end{bmatrix}$$

The Euler equation is tested with an initial condition given in [104]. The initial condition used is

$$\rho = 1; \qquad u = 0 \quad p = \frac{1}{\gamma} + \Delta p; \qquad exp(-\alpha x^2)$$

where $\alpha = 0.05$ and $\Delta p = 0.015$. The problem is solved over the domain [-20, 20] using $CFL = 0.6$.

The solution to this problem is shown in Figure 4.7, where U-grid indicates a uniform grid, and NU-grid is a non-uniform grid. The HRK41 method [105] is used for time integration. N-FVM-CD5 produces oscillations on a non-uniform grid for shock-free solutions. We have already shown that conservative discretization could not provide the necessary convergence on an arbitrarily varying non-uniform grid.

4.4.5 Procedures to avoid oscillations in the higher-order conservative discretization

. In this section, a brief introduction to some ways to prevent oscillations in the higher-order conservative discretization on arbitrarily varying non-uniform meshes is provided.

Gradually stretched grid

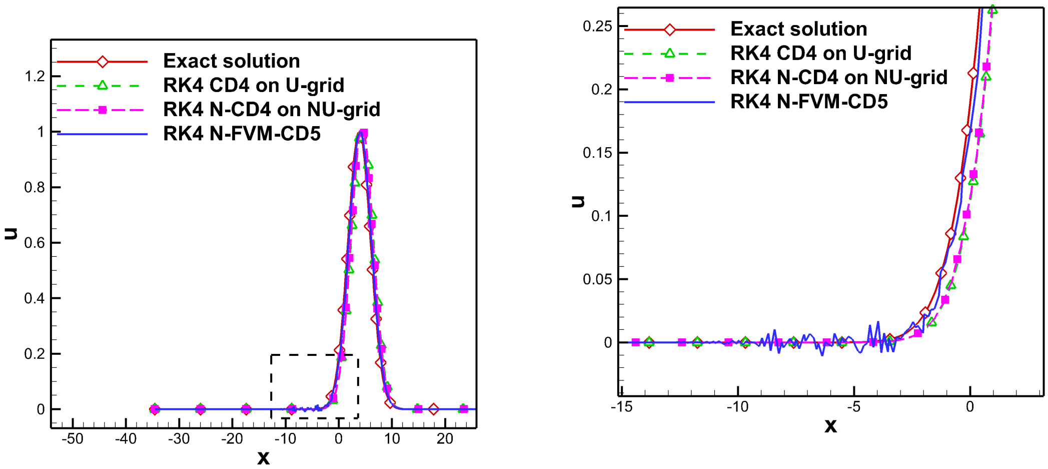
While the progressively stretched grid gives only first-order accuracy, it can produce a good result comparable to that on a uniform grid when the stretching is not high. From Figure 4.5b, it is clear that a highly stretched grid leads to oscillations in the solution. A common technique used in computational fluid dynamics is the use of small grid stretching. This is a common technique, and we can use a numerical scheme developed for a uniform grid on a non-uniform grid. If the grid is not highly stretched, it will provide the same convergence of the scheme as for the uniform grid [103]. Finding an ideal grid for a problem is a major drawback to this technique, and the mesh developer should make a considerable effort. It is difficult to determine the maximum grid length allowed for a problem because it depends entirely on the nature of the solution.

The truncation error of N-FVM-CD5 on AP-grid is $-(\frac{d}{12}\frac{\mathrm{d}^2\mathbf{U}}{\mathrm{d}x^2})$. The sign of the d determines whether the numerical viscosity can stabilize the solver or destabilize the solver. If the d is positive, that is a gradually stretched grid, it adds dissipation. If the value of d is negative, it adds anti-diffusion.

Calculating the derivative directly

Instead of using conservative discretization to determine derivatives, non-conservative discretization can be used for linear equations. Note that spatial conservative discretion does not always ensure physical conservation because conservative quantities depend on space and time.

Although this strategy has worked well for linear equations, it may not always work



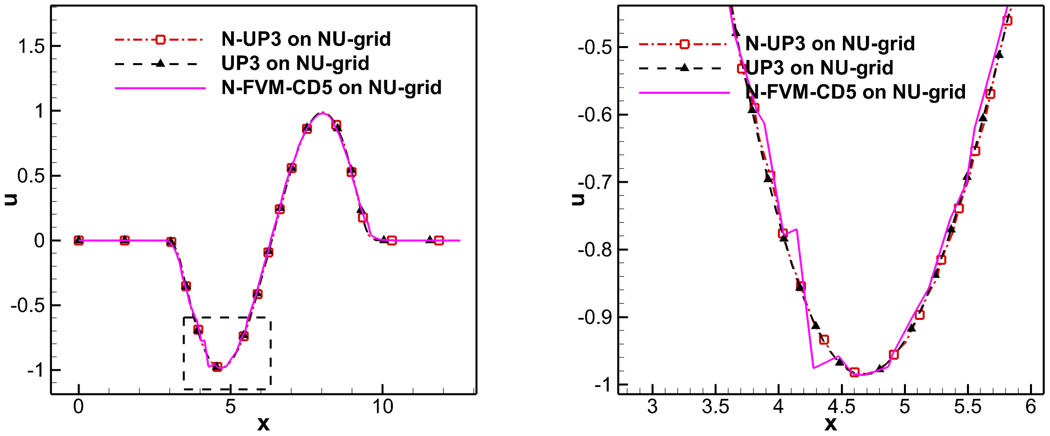
(a) Solution of linear convection equation

(b) Solution of linear convection equation (zoomed view)

Figure 4.8: Solution of linear convection equation with different schemes on different grid configuration

on non-linear equations discretized on arbitrary grids. The solution of linear convection equation with I.C given in eq. 4.38 is shown in figure 4.8. The equation is solved with $CFL = 1$ up to a flow time of 0.5 s. In Figure 4.8, CD4 is the fourth-order central difference scheme; N-CD4 is the central fourth-order finite difference scheme derived using points $i-2$, $i-1$, i, $i+1$ and $i+2$ on the non-uniform grid; U-grid is a uniform grid and NU-grid is an arbitrarily varying non-uniform grid. Time integration is carried out using a low storage RK4 scheme. N-FVM-CD5 produced spurious oscillations, as shown in 4.8, but other schemes produced no oscillations.

The description of the governing equation and initial condition is given in the section 4.4.3. The solution of the viscous Burgers equation is shown in figure 4.9, where UP3 is third-order upwind scheme framed using the grid points $i-2$, $i-1$, i and $i+1$ on a uniform grid. N-UP3 is third-order upwind scheme framed using grid points $i-2$, $i-1$, i and $i+1$ on the non-uniform grid. N-FVM-CD5 is a conservative scheme given in eq. A.2. When using the third-order N-UP3 scheme, no oscillations are produced, but oscillations occur when N-FVM-CD5 is used. Although direct derivative calculation worked for the non-linear Burgers equation, it does not work for the Euler equation in the presence of shock. In the subsection 4.4.5, we study a common way of finding a solution to the Euler equation by adding artificial viscosity. It can be noted that the addition of



(a) Solution of Burgers equation

(b) Solution of Burgers equation (zoomed view)

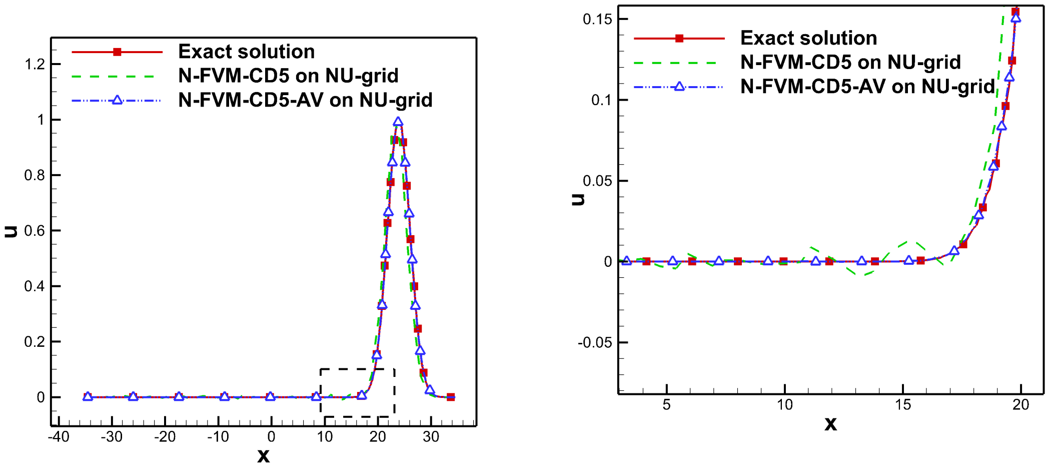
Figure 4.9: Solution of Burgers equation with different schemes

artificial viscosity reduces the accuracy of the scheme.

Adding artificial viscosity

The addition of artificial viscosity is an easy and efficient way to remove oscillations in the solution. An adequate amount of artificial viscosity can provide sufficient bias in the numerical scheme and eliminate oscillations. However, the solution may be diffused and may make the scheme relatively lower order.

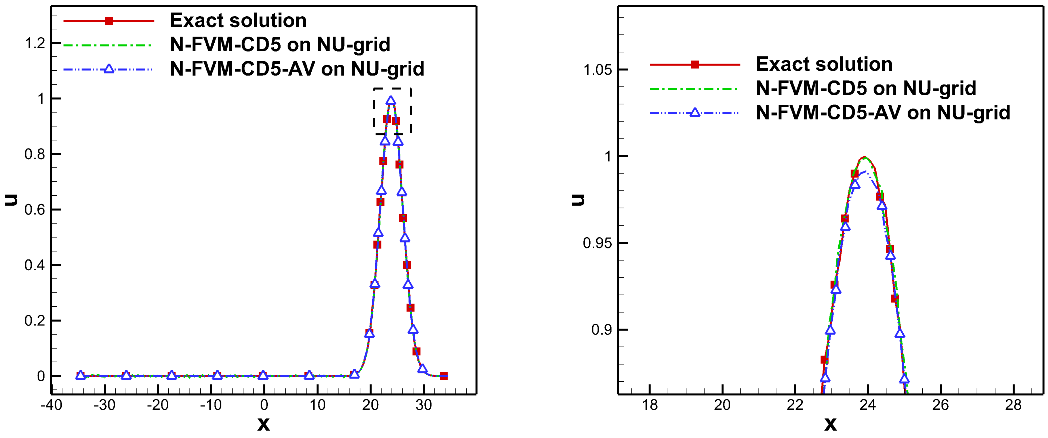
Linear convection equation The governing equation and discretization procedure described in the section 4.4.2 is used here. Figure 4.10 shows the solution of linear convection equation using N-FVM-CD5 with and without artificial viscosity. N-FVM-CD5-AV is N-FVM-CD5 scheme with artificial viscosity, where second-order diffusion term is added with the coefficient of viscosity value 0.01. From figure 4.10b, we can observe that the artificial viscosity can remove the oscillations present in the N-FVM-CD5 scheme. Adding artificial viscosity can diffuse the result that is shown in figure 4.11b. Because a very small amount of oscillations are present in the solution, artificial viscosity has a little impact on the solution.



(a) Solution of linear convection equation

(b) Solution of linear convection equation (zoomed view)

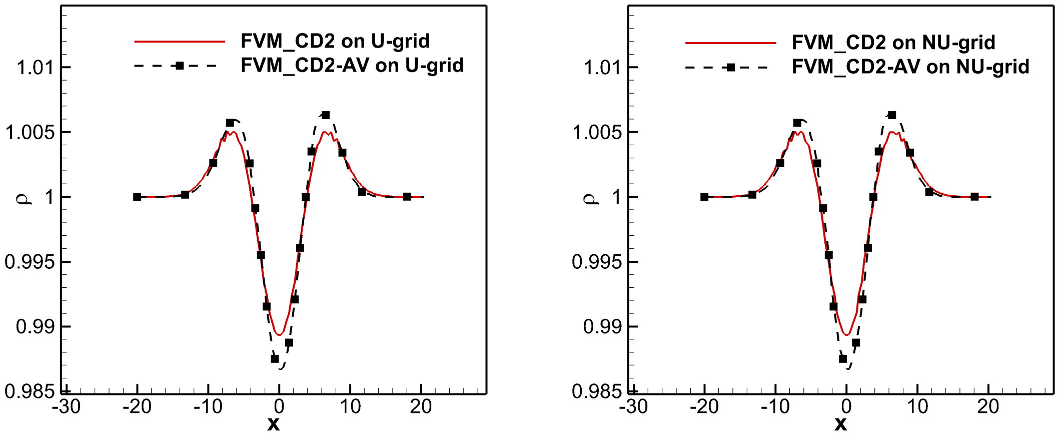
Figure 4.10: Solution of linear convection equation with and without artificial viscosity



(a) Solution of linear convection equation

(b) Linear convection equation (zoomed view)

Figure 4.11: Solution of linear convection equation with and without artificial viscosity



(a) Density plot using uniform grid

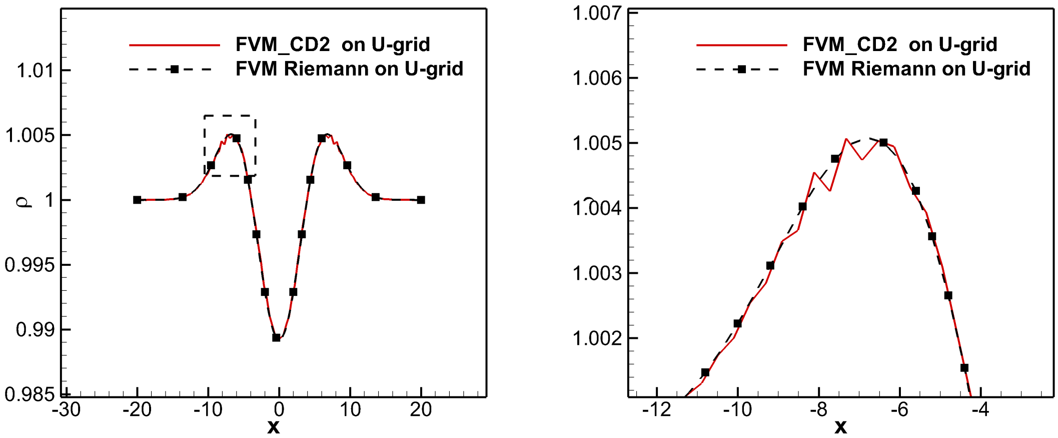
(b) Density plot using non-uniform grid

Figure 4.12: Solution of Euler equation with and without artificial viscosity

Euler equation Euler equation is solved with the initial condition and discretization given in section 4.4.4. A small amount of artificial viscosity (0.2) is added to remove any oscillations present. Figure 4.12 shows the solution of the Euler equation with and without artificial viscosity. Because of the lack of viscosity in the Euler equation, it produces oscillations even on the uniform grid that is stabilized by adding artificial viscosity (figure 4.12a). Artificial viscosity can eliminate oscillations in a uniform, finite-volume scheme when applied to a non-uniform grid, as shown in figure 4.12b.

Riemann solver

A good Riemann solver can remove the oscillations present in the solution. Here, we shall study the elegance of Riemann solver in eliminating oscillations. Figure 4.13 shows the solution of the Euler equation with and without Riemann solver on a uniform grid. From figure 4.13b, it is clear that Riemann solver can remove oscillations in the solution. However, Riemann solver has a little impact than adding artificial viscosity on removing oscillations on a non-uniform grid. If these are combined, they could perform even better at the cost of numerical diffusion.



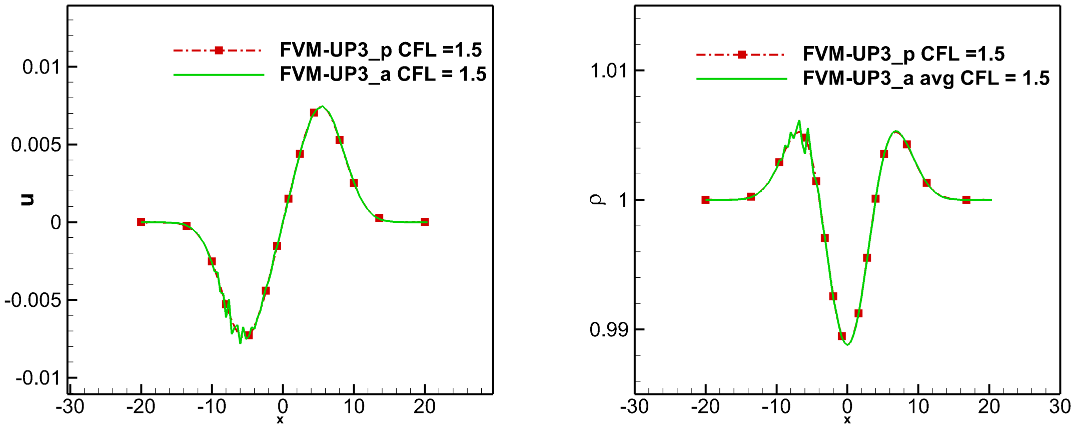
(a) Density plot using uniform grid

(b) Density plot using uniform grid (zoomed)

Figure 4.13: Solution of Euler equation with and without Riemann solver

Point-based reconstruction

Conservative discretion is based on the cell-average reconstruction. Cell-average and point-based reconstruction lead to a different formulation when the order of the scheme is more than two, even on a uniform grid. The Euler equation is solved with the initial condition defined in section 4.4.4. A limiter is applied on the first and second derivatives of the primitive variables to obtain a third-order accurate scheme. Figure 4.14b shows a comparison of density for the averaged and point-based third-order scheme. Average and point-based FVM gave more or less the same result when low CFL was used. The average process leads to numerical instability when a high CFL number is used. Therefore, point-based FVM methods could be used for the derivative calculations in the high-speed flow problems when higher-order schemes are used with a Riemann solver. Cell-average FVM may be acceptable in the low-speed shock-free problems because one can avoid Riemann solver and retain the accuracy of the flux derivatives. This method can also be viewed as a cell-averaged method, but the average is carried over to one cell instead of multiple cells.



(a) Velocity plot averaging and point-based FVM at CFL = 1.5 with n = 102

(b) Density plot of averaging and point-based FVM at CFL = 1.5 with n = 102

Figure 4.14: Solution of the Euler equation

4.5 Summary

Finite volume and conservative finite difference schemes are the two popular schemes based on the conservative discretization. The accuracy of the conservative discretization depends on the accuracy of the interpolation, Riemann solver, orthogonality of mesh, the skewness of mesh, non-uniformity of mesh etc. Among them, non-uniformity is a common source of error in both the schemes when a unstructured grid is used. FVM and conservative-FDM reduce to the same formulation for the one-dimensional case. They also share the same formulation in the two-dimensional case when rectangle mesh is used and in the three-dimensional case when a cuboid mesh is used. In this work, we have given a theoretical proof for the reduction in the order of conservative discretization on an arbitrarily varying non-uniform grid. We have shown the presence of oscillations in conservative discretization on a non-uniform grid. Conservative discretization on the non-uniform grid can give the required convergence when the grid is slightly stretched, but it may show oscillations when the grid is highly stretched.

5

Higher-Order High-Resolution Slope Limiter

High-resolution schemes are based on a non-linear switch and are essential for shock resolution. The parabolic basis of reconstruction is prone to Runge's phenomena, and shocks cannot be resolved. Limiters are the first high-resolution schemes. Limiters are diffusive as they reduce to first-order in the non-smooth regions. Essentially Non-Oscillatory scheme (ENO) is less diffusive than the limiter. The first ENO scheme was first developed by Harten, Engquist, Osher and Chakravarthy in 1987 [87]. ENO uses the best and smoothest among the available polynomials for the reconstruction. The smoothness is determined based on the magnitude of the derivatives. The ENO algorithm uses conditional statements that increase the computational time and decrease the program's readability. Liu, Osher and Chan in 1994 suggested the Weighted Essential Non-Oscillatory (WENO) algorithm. The WENO algorithm gained more popularity among researchers as it avoided the conditional statements by giving weightage to the polynomials based on its smoothness. The formulation and the need for limiters and WENO schemes are demonstrated in this chapter. New limiters are also introduced in this chapter.

5.1 Limiter

Due to the robustness, accuracy and greater stability than other high-resolution schemes, limiters are common and widely used in Computational Fluid Dynamics (CFD). It is based on the total variation diminishing (TVD) principle, i.e. it will not allow the solution value to increase by limiting its value. The reduction in the accuracy of TVD schemes at shocks is to maintain monotonicity [106]. The fundamentals of limiters can be found in [12, 107].

The need for a the limiter is illustrated using the following example. Let us assume we are doing one reconstruction step function (refer figure 5.1) that is extrapolating solution value at $i + 0.5$ using a backward in space method. This can be written as

$$y_{i+0.5} = y_i + \frac{1}{2}\mathrm{d}f_i \tag{5.1}$$

If an extrapolation using the backward difference scheme $(y_i - y_{i-1})$ is performed at $i + 0.5$, the maximum value of the step-function will be surpassed because the derivative at i is positive. Using the forward difference scheme, $(y_{i+1} - y_i)$, the derivative at i is zero, so this derivative information should be used at i. Similarly, in the case of $i - \frac{3}{2}$, the minimum value of interpolation value decreases below the minimum value of the actual step function, if we use the forward difference scheme. The backward difference scheme should be used in that case. In both cases, we have a slope leading to a solution increase, and another leading to a diffusive outcome, so we can hope for a better solution if we take a slope in between the two. A minmod limiter proposed in [108] is a limiter that uses the least value of the slopes.

The classical MUSCL scheme is a second-order scheme which is extended to the

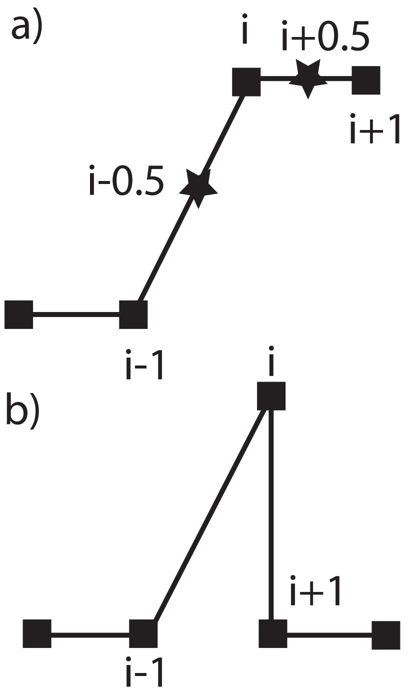


Figure 5.1: a) Simple differentiable step like function b) The derivative of the step function

third-order in [109]. A fourth-order reconstruction polynomial with monotonicity retaining property is presented in [110]. Since parabolic reconstructions are prone to Runge's phenomena, hyperbolic reconstructions are explored in [111]. It is possible to express all the second-order schemes in a general form in conservative discretization; such schemes are called κ-schemes [12, 112].

$$\mathbf{U}_{i+0.5} = \frac{1}{2}(\mathbf{U}_{i+1} + \mathbf{U}_i) - \frac{1-\kappa}{4}(\mathbf{U}_{i+1} - 2\mathbf{U}_i + \mathbf{U}_{i-1}) \tag{5.2}$$

Eq. 5.2 can be viewed as a central difference scheme with the inclusion of an artificial dissipation term. When combined with explicit Euler time integration [102], all schemes will become unstable without the artificial dissipation term, so higher-order schemes use bias that leads to numerical dissipation. This bias is needed to provide upwinding for hyperbolic equations [113]. Eq. 5.2 reduces to different limiters for different values of κ. For example, for $\kappa = -1$ it reduces to linear-upwind interpolation [114]. Eq. 5.2 becomes Fromm scheme [115] for $\kappa = 0$ so the dispersion term in truncation error disappears. Cubic-upwind interpolation [116] is recovered for $\kappa = \frac{1}{3}$. Quadratic-upwind interpolation scheme [117] can be obtained by substituting $\kappa = \frac{1}{2}$. A central difference scheme is obtained for $\kappa = 1$. Comparison of different limiters is presented in [112, 118].

The minmod limiter [119] is relatively more diffusive but can handle poor quality mesh and a variety of flow conditions. Superbee limiter [120] was proposed to improve the shock resolving property of the limiter. Due to aggressive switching nature of superbee, it may produce oscillations in the solution. To make one compromise between resolution and dispersion a limiter which is smoothly varying in the Sweby diagram [107] is proposed in [121]. Harmonic limiter [106] does a harmonic averaging of the upwind and central gradient. A popular limiter that will not fall into Sweby diagram for the negative value of r is introduced in[122]. CHARM [123] is another limiter in which the limiter value can go up to three. This maximum value of flux-limiter can view as an anti-diffusion term for slope correction in the face flux calculation. This maximum value cannot be arbitrarily increased as it leads to unphysical oscillations in the solution.

The maximum anti-diffusion value is increased up to 3 in HCUS limiter [124]; this limiter is comparatively computationally economical than CHARM. HQUICK is another limiter [124], where the anti-diffusion term is pushed up to 4. A flux limiter method based on monotone first-order weighted (FORWE) scheme is presented in [125]. Hybrid low-dissipation and adaptive MUSCL reconstruction techniques are presented in [126]. Third-order accurate reconstruction using three points were developed in [127]. A class of three-point based reconstruction polynomial with limiters that can achieve higher-order can be found in [112]. A variant of superbee limiter with good shock resolving property is proposed in [118]. Minmod limiter with sign preserving property can be found in [128]. Other popular flux-limiters are Koren limiter [129], Chakravarthy limiter [130], OSPRE [124], SMART [131], and UMIST [132]. A limiter which is smoothly varying in the Sweby diagram and good for unstructured grid is proposed in [133].

An efficient TVD scheme for shallow water problem with source terms is applied in [134]. Most of the TVD schemes that ensure monotonicity in one-dimensional problems may not work well for multidimensional problems [135]. The results can be even worse on the unstructured grids. A multidimensional unstructured edge-based limiter is presented in [135] and a non-local flux limiter is proposed in [136]. A comparison of

TVD, ENO and UNO on the unstructured mesh is studied in [137]. An efficient high-resolution relaxation scheme for hyperbolic systems of conservation laws is explored in [138]. Flux-limiters for the coupling of radiation and hydrodynamic models are explored in [139]. The effect of the limiters on different time marching schemes is studied in [140].

TVD second-order schemes with added artificial viscosity are studied in [141]. The performance of the various limiters on Zalesak [142] ellipse problem is discussed in [143] and concludes that the spreading limiters has yielded better results for smooth solutions and the non-spreading limiters has yielded better results when discontinuities are present. A flux limiter based on the Richtmyer-Lax-Wendroff method coupled with a conservative upwind method and a non-conventional flux-limiter function is provided in [144]. A unified, universal total stability region and a new flux-limiter are proposed in [145]. In [146], non-negative positive preservation, flux-limiter technique is given for the worst-case pricing model in financial mathematics. A vertex-based slope-limiter with adaptive p-refining on discontinuous Galerkin methods is provided in [147, 148].

5.1.1 Governing equation and discretization

Governing differential equation considered in this chapter is the Euler equation. The one-dimensional form of the Euler equation is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{5.3}$$

$$\mathbf{U} = \begin{bmatrix} \rho \\ \rho u \\ \rho E \end{bmatrix} \mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + p \\ u(\rho E + p) \end{bmatrix}$$

Formulating integral form of the equation by doing volume averaging in the cell $I_i \in [i - 0.5, i + 0.5]$ using the method of line leads to

$$\frac{\mathrm{d}\bar{\mathbf{U}}_i}{\mathrm{d}t} = -\frac{1}{\Delta x}(\mathbf{F}_{i+0.5} - \mathbf{F}_{i-0.5}) \tag{5.4}$$

where $\mathbf{F}_{i+0.5}$ is numerical flux at the cell interface computed from $\mathbf{F}_{i+0.5} = \mathbf{F}(\mathbf{U}^R_{i+0.5}, \mathbf{U}^L_{i+0.5})$ and $\mathbf{U} \in \mathbb{R}^3$. The second-order numerical flux may be achieved by evaluating $\mathbf{U}^R_{i+0.5}$, $\mathbf{U}^L_{i+0.5}$ given in eq. 5.5 but it may end up in oscillations in the non-smooth data because of non-monotonicity.

$$\begin{aligned} \mathbf{U}^L_{i+0.5} &= \mathbf{U}_i + \frac{\Delta x}{2}\mathrm{d}\mathbf{U} + \frac{\Delta x^2}{8}\mathrm{d}^2\mathbf{U} \qquad \mathcal{O}(\Delta x^3) \\ \mathbf{U}^R_{i+0.5} &= \mathbf{U}_{i+1} - \frac{\Delta x}{2}\mathrm{d}\mathbf{U} + \frac{\Delta x^2}{8}\mathrm{d}^2\mathbf{U} \qquad \mathcal{O}(\Delta x^3) \end{aligned} \tag{5.5}$$

Limiter function is used on the gradients

$$\begin{aligned} \mathbf{U}^L_{i+0.5} &= \mathbf{U}_i + \frac{\Delta x}{2}L[\mathrm{d}\mathbf{U}] + \frac{\Delta x^2}{8}L[\mathrm{d}^2\mathbf{U}] \\ \mathbf{U}^R_{i+0.5} &= \mathbf{U}_{i+1} - \frac{\Delta x}{2}L[\mathrm{d}\mathbf{U}] + \frac{\Delta x^2}{8}L[\mathrm{d}^2\mathbf{U}] \end{aligned} \tag{5.6}$$

where L is the limiting operator. In the presence of shocks in the compressible flow, we can not calculate $\mathbf{U}^L_{i+0.5}$ or $\mathbf{U}^R_{i+0.5}$ directly. It is necessary to restrict the value of $\mathrm{d}\mathbf{U}$ and $\mathrm{d}^2\mathbf{U}$ using the slope-limiter, otherwise it will produce non-physical oscillations in the solution. Using the limiters the left and right states are determined using the Monotonic Upwind Scheme for Conservation Laws (MUSCL) approach. Harten entropy fix based on the jump present in the conserved variable is incorporated in the Roe scheme is used as the Riemann solver. Hyperbolic Runge-Kutta-method [105] is used for the time integration.

5.2 Limiters used in the present work

This section discusses the second and the third-order limiters used in this thesis.

5.2.1 Second-order slope limiter

A second-order limiter is introduced in this section, and the numerical properties of these limiters are discussed. Second-order flux-limiters are widely used in CFD, but achieving a high-resolution result without significant wiggles in the solution is often a difficult task when using an aggressive limiter. A computationally economical second-order limiter is presented in this section. The second-order limiter used here is comparable to the form given in [112]. The representation and implementation of the limiter used in the present work vary slightly from the flux-limiters in [107]. It is possible to write the interpolation scheme as

$$\mathbf{U}_{i+0.5} = \mathbf{U}_i + \frac{\Delta x}{2} L[\partial \mathbf{U}(\mathbf{U}^+, \mathbf{U}^-))] \tag{5.7}$$

where $L[\partial \mathbf{U}] = \psi(r)\partial \mathbf{U}(\mathbf{U}^+, \mathbf{U}^-)$ [112]. $\psi(r)$ is the limiter function, $r = \frac{\mathbf{U}_1^-}{\mathbf{U}_1^+}$, $\mathbf{U}_1^+ = \mathbf{U}_{i+1} - \mathbf{U}i$ and $\mathbf{U}_1^- = \mathbf{U}(i) - \mathbf{U}(i-1)$.

MMF1

This limiter is a modified minmod [107] limiter, but a different approach is used in choosing limiting parameter.

$$L[\partial \mathbf{U}(\mathbf{U}^+, \mathbf{U}^-)] = \begin{cases} \mathbf{U}_1^+ + c(\mathbf{U}_1^- - \mathbf{U}_1^+) abs\left(\frac{\mathbf{U}_1^+}{\mathbf{U}_1^- + \epsilon}\right)\psi, & \text{if} \quad abs(\mathbf{U}_1^+) < abs(\mathbf{U}_1^-) \\ \mathbf{U}_1^- + c(\mathbf{U}_1^+ - \mathbf{U}_1^-) abs\left(\frac{\mathbf{U}_1^-}{\mathbf{U}_1^+ + \epsilon}\right)\psi, & \text{if} \quad abs(\mathbf{U}_1^+) \geq abs(\mathbf{U}_1^-) \end{cases} \tag{5.8}$$

where $\psi = abs\left(\frac{sign(\mathbf{U}_1^-) + sign(\mathbf{U}_1^+)}{2}\right)$

Since the classical minmod chooses the absolute minimum of the slopes among the right and the left slopes, the second-order scheme is diffused. There are oscillations if the highest slope is chosen. In order to minimize dissipation, a small anti-diffusion term is applied to the slope obtained from minmod. The resulting value of the slope should be between the highest and lowest original values. The parameter r controls the magnitude of the anti-diffusion so that the limiter remains in the monotonous region. The current limiter is always in the monotone region, regardless of the value of r.

5.2.2 Basic properties of the MMF1 limiter

Condition to be satisfied by the second-order limiter is

$$0 \leq \psi(r) \leq min(2r, 2) \tag{5.9}$$

lemma 5.2.1. *MMF1 (eq. 5.8) always lie in monotone region in Sweby diagram for $c = 1$.*

Proof. Most extreme case is $\psi(r) = 2$, since $r = \frac{\mathbf{U}^-}{\mathbf{U}^+} \geq 1$

$$2\mathbf{U}^+ \geq \mathbf{U}^+ + (\mathbf{U}^- - \mathbf{U}^+) abs\left(\frac{\mathbf{U}^+}{\mathbf{U}^-}\right) \tag{5.10a}$$

$$\mathbf{U}^+ \geq (\mathbf{U}^- - \mathbf{U}^+)\frac{\mathbf{U}^+}{\mathbf{U}^-} \tag{5.10b}$$

$$0 \geq -\frac{\mathbf{U}^{+^2}}{abs(\mathbf{U}^-)} \tag{5.10c}$$

Even for most extreme case $\psi(r) = 2$, the present limiter satisfies eq. 5.10c for all value of $\mathbf{U}^+$ so it is monotone. ■

lemma 5.2.2. *MMF1 always satisfies the minimum requirement of a second-order limiter* $\psi(r)|_{r=1}$ *= 1.*

Proof. MMF1 expression in terms of r is

$$\mathbf{U}(r) = 1 + c(r-1) * \frac{1}{\mathbf{U}^-} \tag{5.11}$$

when $r = 1$, $\mathbf{U}(r) = 1$, so it will not add any anti-diffusion when the solution is smooth.

■

lemma 5.2.3. *MMF1 is symmetric.*

Proof. Symmetricity of limiter is defined as

$$\partial\mathbf{U}(\mathbf{U}^+, \mathbf{U}^-) = \partial\mathbf{U}(\mathbf{U}^-, \mathbf{U}^+) \tag{5.12}$$

Because MMF1 chooses the slope based on the magnitude of the left and the right slopes regardless of the position in the argument of the limiting function so it is symmetric. ■

lemma 5.2.4. *MMF1 is homogeneous.*

Proof. Homogeneity property of limiter is defined as

$$\partial\mathbf{U}(\lambda\mathbf{U}^+, \lambda\mathbf{U}^-) = \lambda\partial\mathbf{U}(\mathbf{U}^-, \mathbf{U}^+) \tag{5.13}$$

Multiplying λ in the arguments of MMF1,

$$\lambda \mathbf{U}^{+} + \lambda(\mathbf{U}^{-} - \mathbf{U}^{+})\frac{\mathbf{U}^{+}}{\mathbf{U}^{-}} = \lambda\partial\mathbf{U}(\mathbf{U}^{-}, \mathbf{U}^{+}) \tag{5.14}$$

So MMF1 satisfies homogeneity property. ■

lemma 5.2.5. *MMF1 satisfies consistency property.*

Proof. Consistency property of limiter is defined as:

$$\partial\mathbf{U}(a, a) = \partial\mathbf{U} \tag{5.15}$$

Substituting $\mathbf{U}^{+}$ in the arguments of MMF1

$$\partial\mathbf{U}(\mathbf{U}^{+}, \mathbf{U}^{+}) = \mathbf{U}^{+} + \lambda(\mathbf{U}^{+} - \mathbf{U}^{+})\frac{\mathbf{U}^{+}}{\mathbf{U}^{+}} = \mathbf{U}^{+} \tag{5.16}$$

So MMF1 satisfies consistency property. ■

5.2.3 Third-order slope-limiters

A third-order TVD limiter is presented where the first and second derivatives are limited in the interpolation of the left and right states. The general third-order scheme to calculate face value at $i + 0.5$ is

$$\mathbf{U}^{L}_{i+0.5} = \mathbf{U}_{i} + \frac{\Delta x}{2}L[\partial\mathbf{U}(\mathbf{U}^{+}, \mathbf{U}^{-}, \mathbf{U}^{0})] + \frac{\Delta x^2}{8}L[\partial^2\mathbf{U}] \tag{5.17}$$

$$\mathbf{U}^{R}_{i+0.5} = \mathbf{U}_{i+1} - \frac{\Delta x}{2}L[\partial\mathbf{U}] + \frac{\Delta x^2}{8}L[\partial^2\mathbf{U}(\mathbf{U}^{+}, \mathbf{U}^{-}, \mathbf{U}^{0})] \tag{5.18}$$

where L is the limiter operator which operates on the gradients. In the stencil [$i{-}2,\ i{+}2$], there are three ways to calculate $\partial^2\mathbf{U}$. They are

$$\begin{aligned} \mathbf{U}^{-}_{2} &= \mathbf{U}_{j-2} - 2\mathbf{U}_{j-1} + \mathbf{U}_{j} && \mathcal{O}(\Delta x) \\ \mathbf{U}^{0}_{2} &= \mathbf{U}_{j-1} - 2\mathbf{U}_{j} + \mathbf{U}_{j+1} && \mathcal{O}(\Delta x^2) \\ \mathbf{U}^{+}_{2} &= \mathbf{U}_{j} - 2\mathbf{U}_{j+1} + \mathbf{U}_{j+2} && \mathcal{O}(\Delta x) \end{aligned} \tag{5.19}$$

5.2.4 Minmod_s2

Eq. 5.19 provides three possible second derivative expressions for the stencil $[i-2, i+2]$. The best option is to take the minimum among $abs(\mathbf{U2}^+)$, $abs(\mathbf{U2}^-)$, and $abs(\mathbf{U2}^0)$.

$$L[\partial^2\mathbf{U}(\mathbf{U}_2^+, \mathbf{U}_2^-, \mathbf{U}_2^0)] = \begin{cases} \mathbf{U}_2^+\psi, & \text{if } abs(\mathbf{U}_2)^+ < abs(\mathbf{U}_2^-) \quad \& \quad abs(\mathbf{U}_2^))+ < abs(\mathbf{U}_2^0) \\ \mathbf{U}_2^-\psi, & \text{if } abs(\mathbf{U}_2^+) \geq abs(\mathbf{U}_2^-) \quad \& \quad abs(\mathbf{U}_2^0) \geq abs(\mathbf{U}_2^-) \\ \mathbf{U}_2^0\psi & \textit{else} \end{cases}$$

where $\psi = \frac{sign(\mathbf{U}_2^+)+sign(\mathbf{U}_2^-)}{2}$

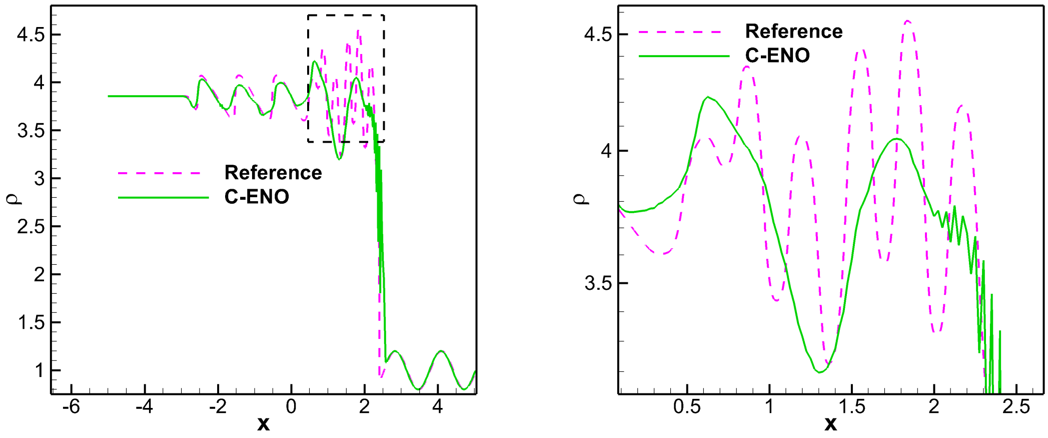
This limiter is similar to convex-ENO (C-ENO) proposed in [149]. However, they have used limiters on $\mathbf{U}_2^-$ and $\mathbf{U}_2^0$ and not on $\mathbf{U}_2^+$. In the present limiter, extra conditions are implemented to improve the stability and accuracy of the scheme, but the limiter is computationally expensive than C-ENO. C-ENO and the present scheme is compared for Shu-Osher problem. The initial condition used for this problem is:

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.33333) & x \leq -4 \\ (1 + 0.2 \times sin(5x), 0, 1) & x > -4 \end{cases} \tag{5.20}$$

Figure 5.2 shows the solution of the Shu-Osher problem with 400 grid points at a flow time of 1.8 s. The problem is solved by using CFL=0.8. The reference solution is obtained by using a minmod limiter on 2000 grid points. C-ENO produced oscillations when a large CFL number was used. When CFL number is less than 0.7 convex-ENO did not generate any oscillations on this grid. Table 5.1 lists different combinations of limiters used in this thesis for the first and second-order derivative terms.

5.3 Results of the second-order limiter

The performance of the MMF1 limiter on various standard test cases for 1-D and 2-D Euler equations with different shock structures is studied in this section. The performance



(a) Density

(b) zoomed view of density

Figure 5.2: Density plot of Shu-Osher problem with 400 grid points with $CFL = 0.8$

Table 5.1

List of the reconstruction schemes used

Notation	**1$^{\text{st}}$ derivative**	**2$^{\text{nd}}$ derivative**
MMF1	MMF1	-
Cada [150]	Cada	-
MM-S2	minmod	minmod_s2
MMF1-S2	MMF1	minmod_s2
Convex-ENO [149]	minmod	minmod
Venkadakrishnan [133] (VK)	VK	-

of the current limiters on the unstructured triangular grid is also assessed.

5.3.1 Sod shock tube problem

The initial conditions for the problem are

$$(\rho, u, p) = \begin{cases} (1, 0, 1) & x \leq 0.5 \\ (0.125, 0, 0.1) & x > 0.5 \end{cases} \tag{5.21}$$

This problem is solved using 200 grid points with $CFL = 0.5$. A second-order four-stage hyperbolic Runge-Kutta method (HRK42 [105]) is used for time integration. Interpolation is carried out in the primitive variables using the MUSCL scheme using different limiters. The standard Roe scheme [151] with entropy fix is used for Riemann solver.

A comparison of various limiters with MMF1 is presented in figure 5.3. From the figure 5.3, it is clear that MMF1 has better discontinuity resolving capability than all the limiters considered in this work. The result of MMF1 is also comparable to the superbee limiter. L_2 error of the solution with different limiters is shown in table 5.2. The error is calculated using eq. 5.22

$$E_\rho = \sqrt{\left(\frac{\sum_{i=1}^{i=n}(\rho_i^{exact} - \rho_i^{numerical})^2}{n^2}\right)} \tag{5.22}$$

The relative error (% $\mathbf{E}_\rho$)is calculated using eq. 5.23.

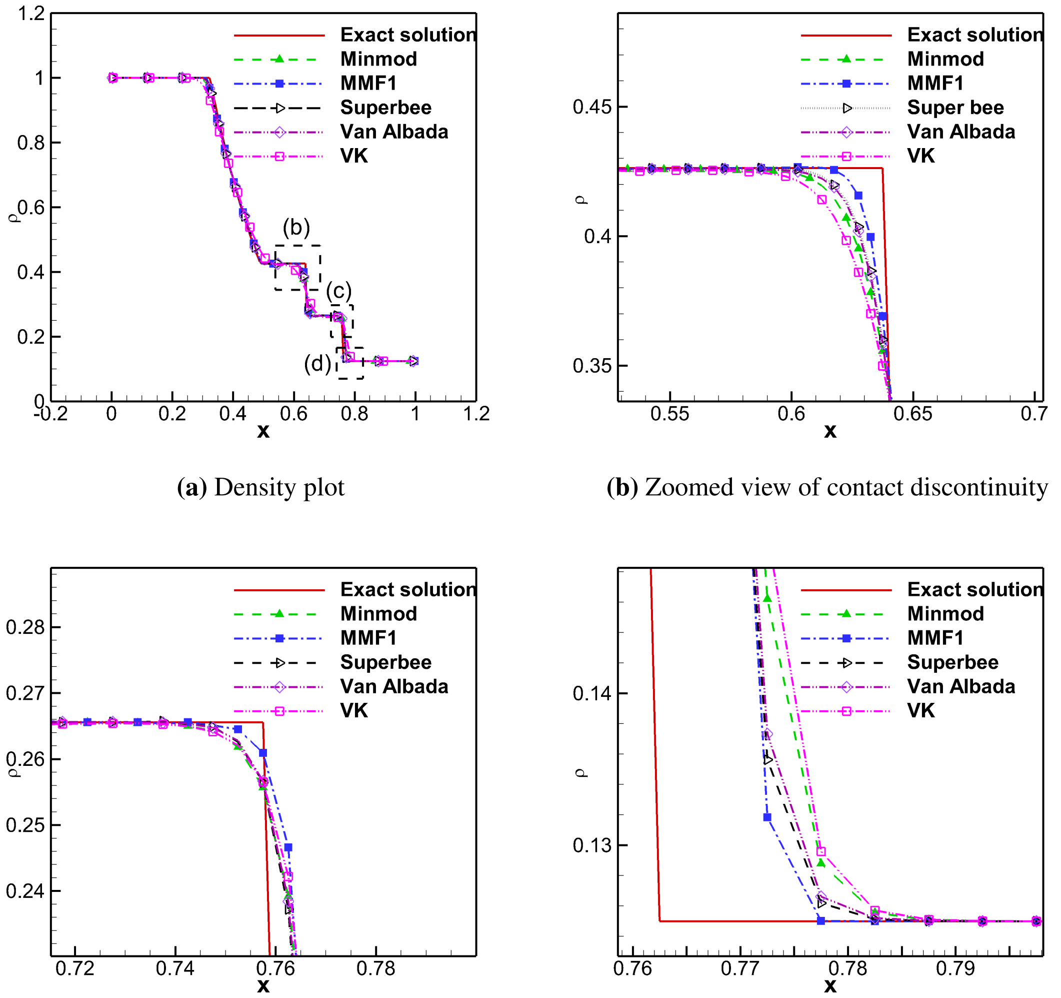
$$\%\mathbf{E}_\rho = \frac{\mathbf{E}_\rho^{\mathbf{limiter}}}{\mathbf{E}_\rho^{\mathbf{MMF1}}} \times 100 \tag{5.23}$$

In eq. 5.23, $\mathbf{E}_\rho^{\mathbf{MMF1}}$ is root mean square (RMS) error of MMF1. $\mathbf{E}_\rho^{\mathbf{limiter}}$ is the RMS error of other limiters. For this test case, MMF1 is relatively more accurate than other limiters. The result of suberbee and MMF1 is more or less the same, but MMF1 is relatively more accurate than superbee in velocity (table 5.2). The current limiter is 25% more accurate than minmod limiter. Venkatakrishnan limiter which perform well for unstructured grid is relatively more diffusive than the other limiter.

Table 5.2

L_2 error in Sod shock tube problem using different limiters

Limiters	$\mathbf{E}_\rho$	$\mathbf{E_u}$	$\mathbf{E_p}$	% $\mathbf{E}_\rho$
Minmod	0.029949108	0.097347997	0.034681065	125.7823
MMF1	0.023810274	0.079651239	0.026100415	100
Superbee	0.023958118	0.087255857	0.026877834	100.6209
Van Albada	0.028097104	0.093829356	0.032254258	118.0041
Venkatakrishnan	0.0220426951	0.09443749	0.02402663	92.57



(a) Density plot

(b) Zoomed view of contact discontinuity

(c) Density plot

(d) Zoomed view of contact discontinuity

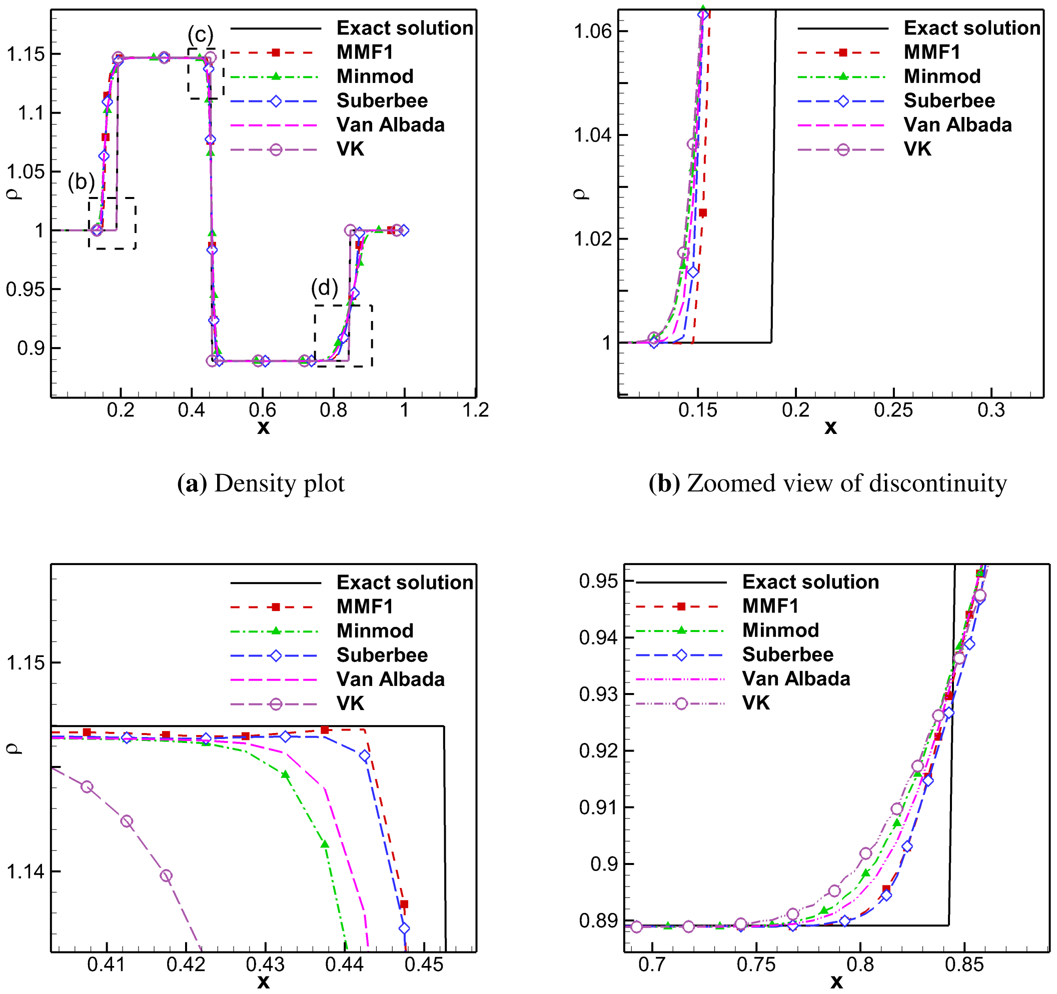
Figure 5.3: Density plot of Sod shock tube problem with 200 grids $CFL = 0.5$ at $t = 0.15s$

5.3.2 Right expansion and left strong shock

This case has a only jump in the pressure in the initial condition; there are no jumps in other variables. The initial condition used is

$$(\rho, u, p) = \begin{cases} (1, 0, 7) & x \leq 0.5 \\ (1, 0, 10) & x > 0.5 \end{cases}$$

Numerical simulation is done using $CFL = 0.9$ over 200 grid points. The solution is obtained up to a flow time of $t = 0.1s$, and the results are shown in figure 5.4. The L_2



(a) Density plot

(b) Zoomed view of discontinuity

(c) Zoomed view of discontinuity

(d) Zoomed view of discontinuity

Figure 5.4: Density plot of Right Expansion and left strong shock tube problem with 200 grids $CFL = 0.9$ at $t = 0.1s$

error is presented in the table 5.3 for this test case. The performance of all the limiters are identical, but there is a lower density error for superbee. In velocity and pressure fields, MMF1 is slightly more effective than others. In general, the MMF1 limiter is more accurate than the superbee limiter on the coarse grid, and the superbee limiter is more accurate on the fine grid.

Table 5.3
L_2 error in right expansion and left strong shock problem using different limiters

Limiters	$\mathbf{E}_\rho$	$\mathbf{E_u}$	$\mathbf{E_p}$	% $\mathbf{E}_\rho$
MMF1	0.032392426	0.086743897	0.294213352	100
Minmod	0.034556739	0.087621733	0.296819026	106.6815
Superbee	0.031561244	0.087006966	0.294756368	97.434
Van Albada	0.033673969	0.087865768	0.297727905	103.9563
Venkadakrishnan (VK)	0.0345159	0.08787372	0.29816845	106.55

5.3.3 Mach number 3 test case

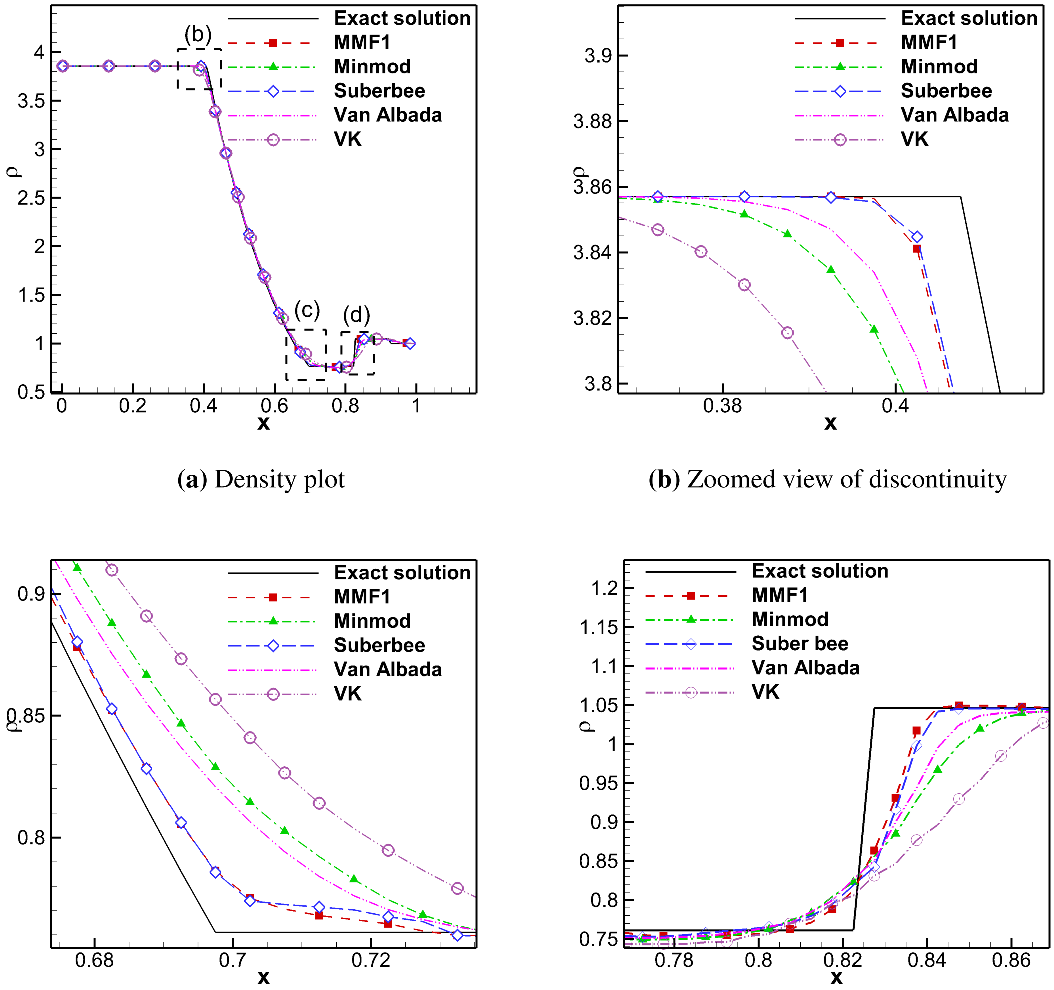
The initial condition used is

$$(\rho, u, p) = \begin{cases} (3.857, 0.92, 10.333) & x \leq 0.5 \\ (1, 3.55, 1) & x > 0.5 \end{cases}$$

The problem is solved by using 200 grid points with $CFL = 0.9$. The problem is simulated to a flow time of $t = 0.09s$, and the results are shown in Figure 5.5. For this problem, the MMF1 limiter has outperformed other limiters. The RMS error of different limiters for this test case is shown in the table 5.4. The MMF1 limiter is 37% more accurate than the minmod limiter.

Table 5.4
L_2 error in Mach number 3 problem using different limiters

Limiters	$\mathbf{E}_\rho$	$\mathbf{E_u}$	$\mathbf{E_p}$	% $\mathbf{E}_\rho$
MMF1	0.058996677	0.057073058	0.170774375	100
Minmod	0.081226185	0.073182142	0.234189558	137.6793
Superbee	0.060195562	0.039913532	0.154488691	102.0321
Van Albada	0.073034229	0.063915602	0.202328364	123.7938
Venkadakrishnan (VK)	0.0688619	0.0637810	0.1764672	120.6557



(a) Density plot

(b) Zoomed view of discontinuity

(c) Zoomed view of discontinuity

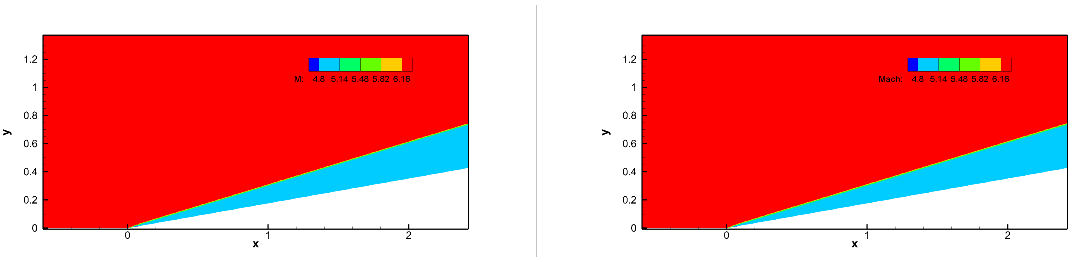
(d) Zoomed view of discontinuity

Figure 5.5: Density plot of Mach = 3 test with 200 grids $CFL = 0.9$ at $t = 0.09s$

5.3.4 Supersonic flow over a wedge

Supersonic flow at Mach number 6.5 over a 10° wedge is solved. The problem is solved using a 401×401 quadrilateral grid. The boundary conditions used are slip wall on the wedge and *free-stream* condition at the other boundaries. The derivatives are limited using MMF1 limiter. Time integration is carried out using the implicit Euler method. Mach number contours are shown in figure 5.6. There is no significant variation in the solution between minmod1 and minmod, but the resolution of minmod1 is slightly better than minmod, as shown in figure 5.7. Root mean square (RMS) error of minmod and MMF1 is 0.1068824 and 0.097375, respectively. In terms of RMS error, MMF1 is 9.7%

h



(a) Using MMF1 limiter

(b) Using minmod limiter

Figure 5.6: Supersonic flow past 10° wedge with $M = 6.5$

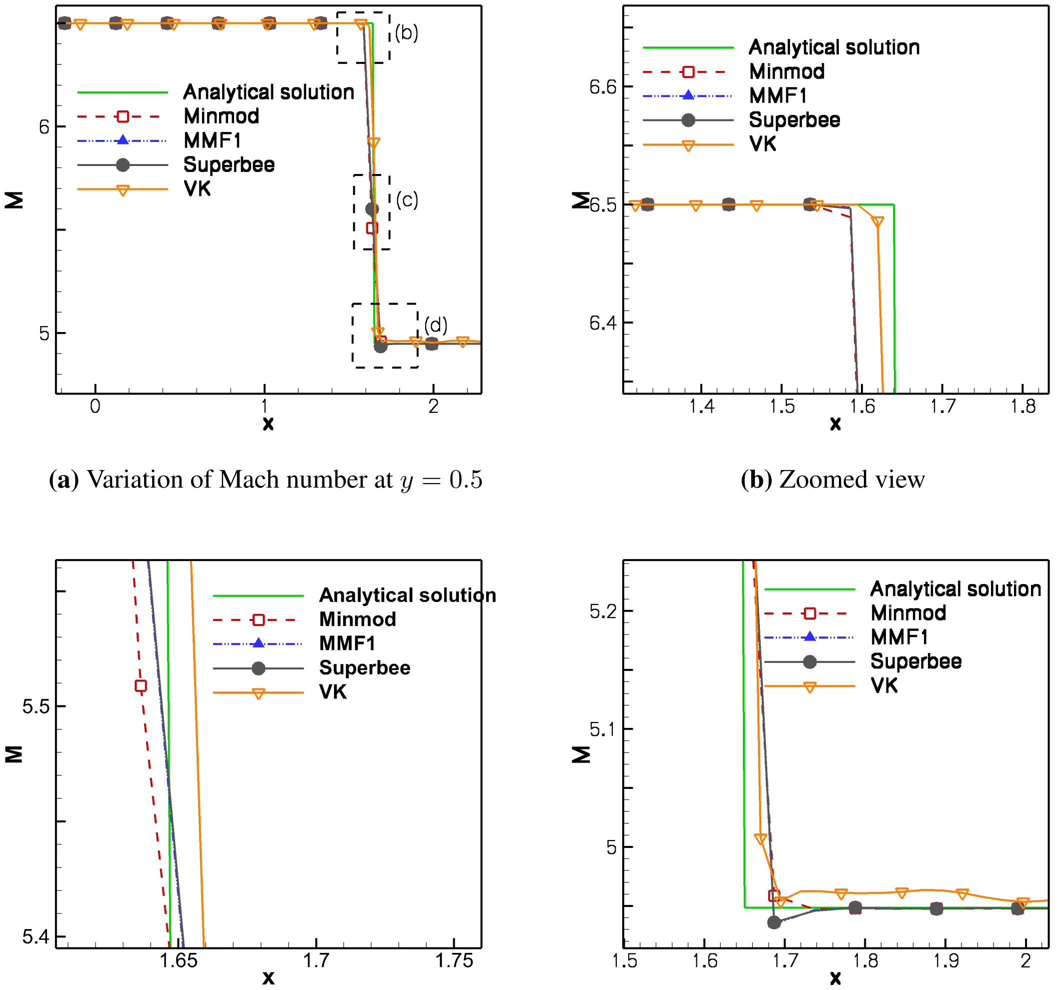
more accurate than minmod. For this problem, Venkatakrishnan limiter shows a better shock resolving property at the top surface but it shows a poor resolution at the lower side of the shock.

5.3.5 Supersonic flow over a wedge with expansion fan

The limiter is evaluated on a wedge with a turning angle of 26.6° followed by a shoulder. The simulation is carried on a 500×200 grid at a free-stream Mach number of 1.44. The boundary condition is zero normal velocity on the lower boundary (wall) and *free-stream* condition at other boundaries. Moving least square method with a Gaussian kernel is used for interpolation. Time integration is carried out using the implicit Euler method. For isotopic mesh, the vertex-centred system is more precise than the cell-centred system, so the vertex-centred system is used. Here, an unstructured triangular grid is generated, and the edge-based limiting procedure is followed. The pressure contours for the test case are shown in figure 5.8a. The comparison of the experimental [152] and the theoretical results with the computational results are shown in the figure 5.8b.

5.3.6 Shock reflection

Supersonic flow with an inflow Mach number 4.5 is passed over a wedge with angle 16.7°. The domain [-1, 15] × [0, 1.5] is divided into 300 × 100 grid points. The problem is solved using implicit Euler method and solved upto the steady state. The boundary conditions used are slip wall on the lower and the upper surface. *Free-stream* condition is used at the other boundaries. The performance of different limiters are compared.



(a) Variation of Mach number at $y = 0.5$

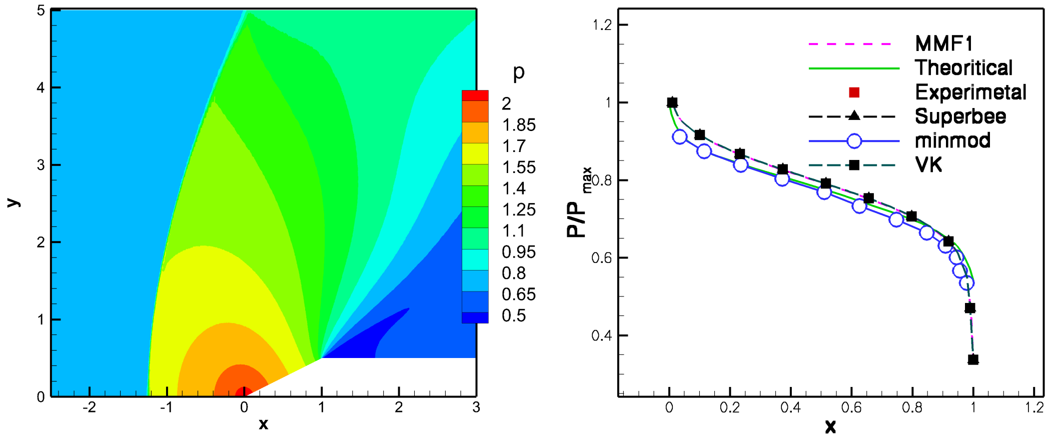
(b) Zoomed view

(c) Zoomed view

(d) Zoomed view

Figure 5.7: Supersonic flow past 10° wedge with $M = 6.5$

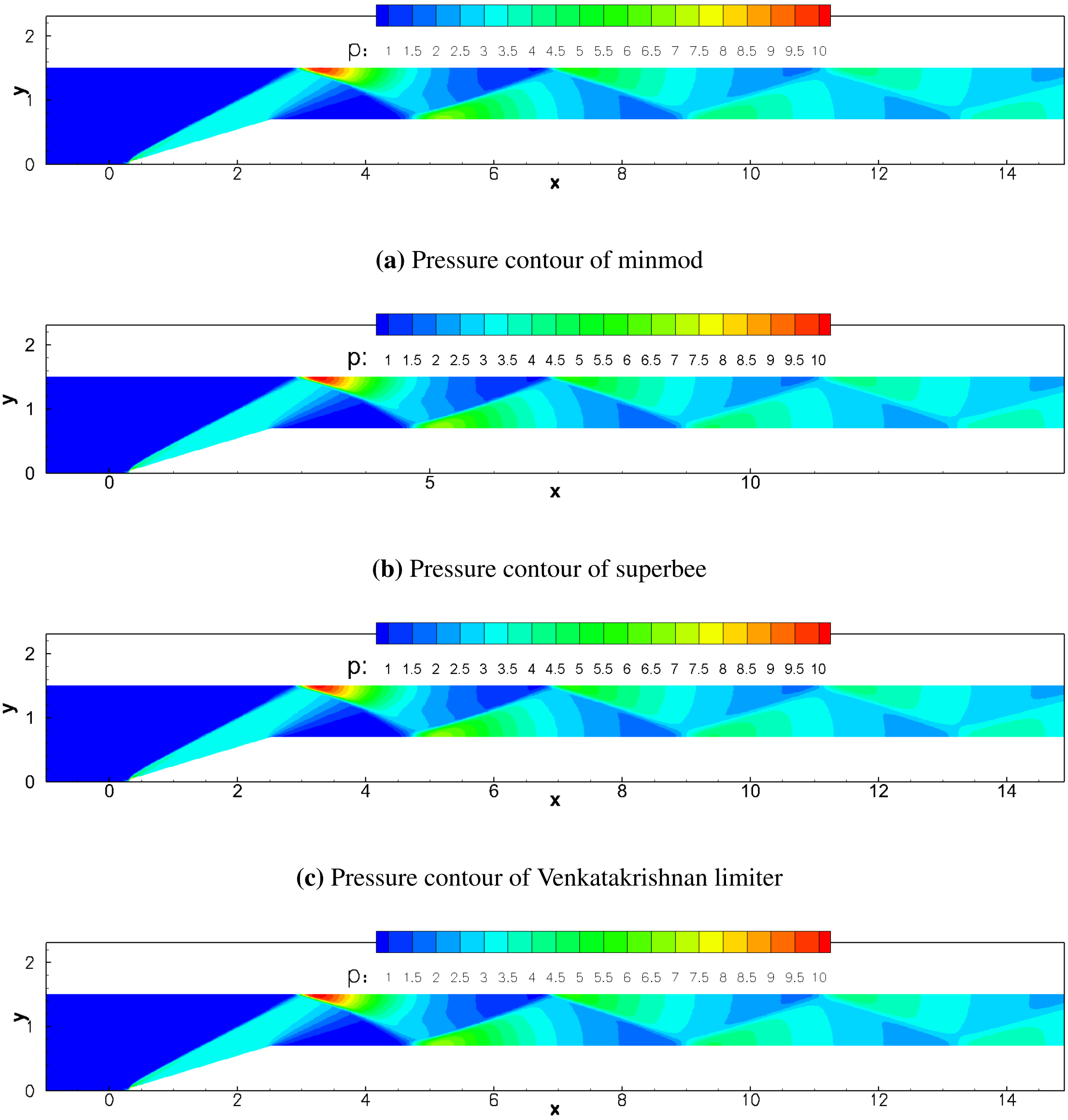
Figure 5.9 shows the pressure contour of different limiters for this test case. Figure 5.10 shows the pressure at $y = 1.1$ line. For this test case superbee is slightly better than MMF1 limiter. Other limiters are more diffusive than MMF1 limiter.



(a) Pressure contour of the wedge

(b) Pressure on the surface of the wedge

Figure 5.8: Pressure plots of 26.6° semi-angle wedge at M = 1.44



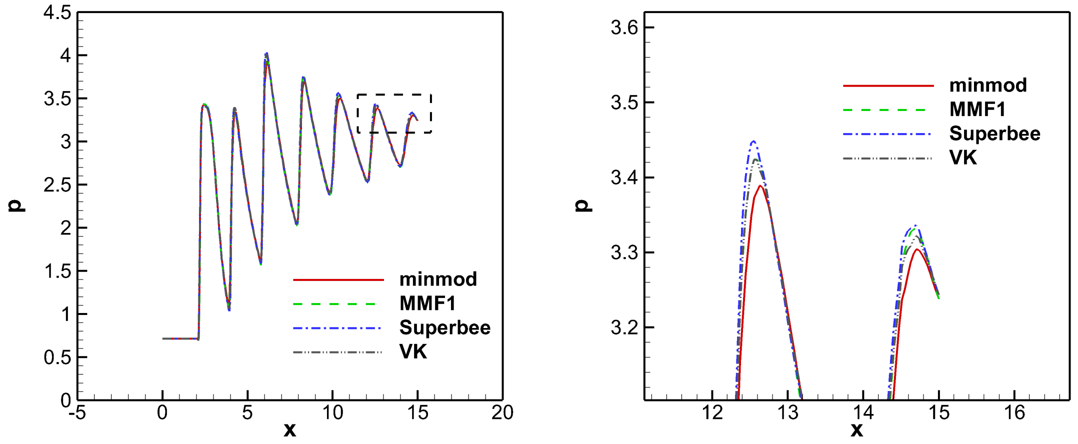
(a) Pressure contour of minmod

(b) Pressure contour of superbee

(c) Pressure contour of Venkatakrishnan limiter

(d) Pressure contour of MMF1 limiter

Figure 5.9: Pressure plots of 16.7° semi-angle wedge at M = 4.5



(a) Pressure plot

(b) Zoomed view

Figure 5.10: Pressure plots at $y = 1.1$ line of $16.7°$ semi-angle wedge at M = 4.5

5.4 Results of the third-order limiter

The results of the third-order limiters are presented in this section. Compared to the Sod shock tube problem, the results of third-order limiters are better than the second-order limiters on the Shu-Osher problem and blast waves interaction problem. In general, for problems with curvatures, third-order limiters function well.

5.4.1 Sod shock tube problem

Sod shock tube problem is solved using 200 grid points and $CFL = 0.5$. The test case is solved up to a flow time of 0.1 s using HRK42 method. The initial condition is given in eq. 5.21. The density plot of the test case is shown in figure 5.11. From the figure, it is clear that third-order limiter MM-S2 is relatively more oscillatory than others, but it has better shock resolution capability. Cada limiter [150] has shown a better trade between shock resolution and over-shoots. L_2 error of the test case is shown in table 5.5. Although MMF1-S2 limiter has some over-shoots in the contact discontinuity when compared to other limiters, it has given the least RMS error. The results from MMF1 are comparable to that from the second-order limiter. MMF1-S2 is 16% more accurate than Cada limiter.

Table 5.5
L_2 error of Sod shock tube problem

Limiters	$\mathbf{E}_\rho$	$\mathbf{E_u}$	$\mathbf{E_p}$	% $\mathbf{E}_\rho$
MMF1	0.012085929	0.055583416	0.012893991	100
Cada	0.014126912	0.059664611	0.014680221	116.8873
MM-S2	0.015526093	0.059992679	0.015825881	128.4642
MMF1-S2	0.012049288	0.054684266	0.012724591	99.6968
Convex-ENO	0.01010391	0.03618296	0.0090118	83.6006
Venkatakrishnan	0.01492193	0.06068145	0.0153616	123.46531

5.4.2 Shu-Osher problem

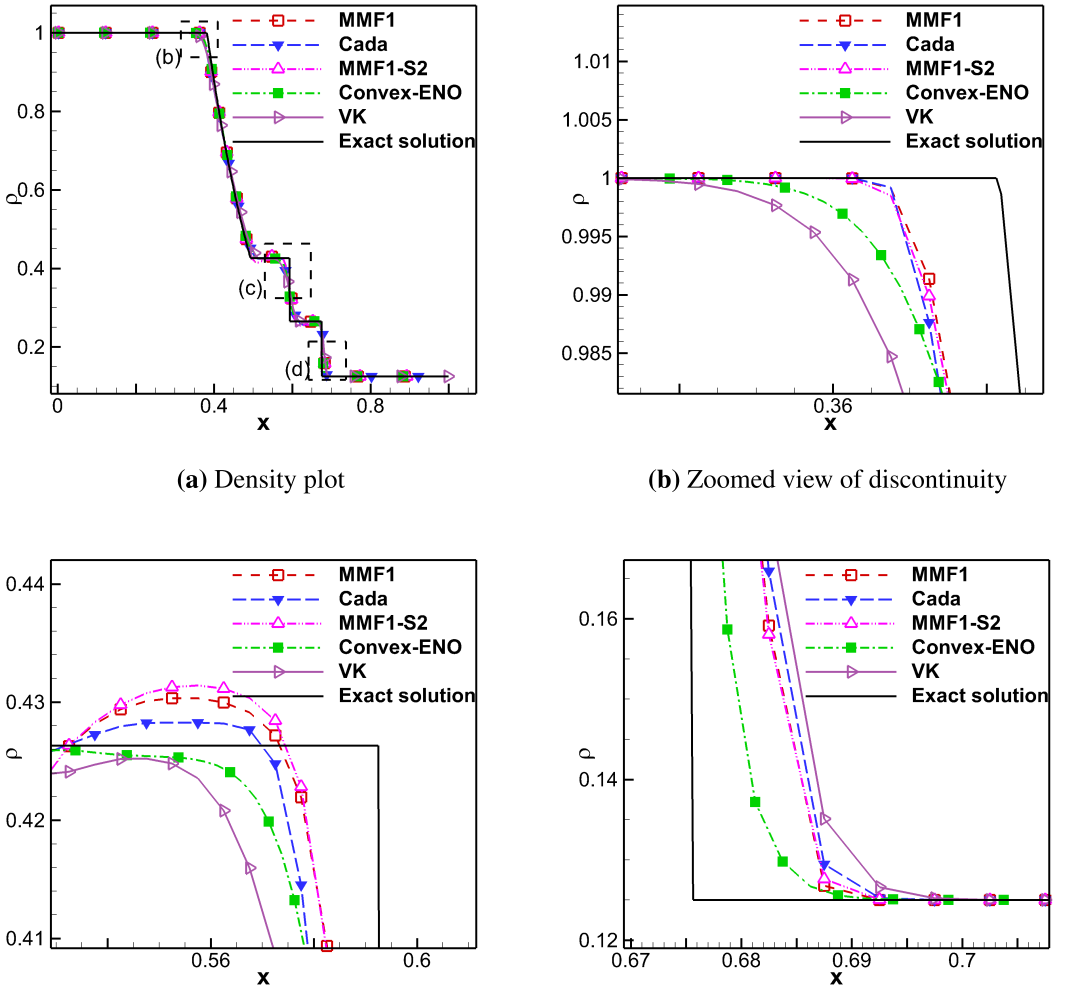
The initial condition used is

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.33333) & x \leq -4 \\ (1 + 0.2 \times \sin(5x), 0, 1) & x > -4 \end{cases}$$

The problem is solved in the domain [-5, 5]. The simulation is carried out up to a flow time of 1.8 s using 400 grid points. The value of CFL number used is 0.5. It can be seen from the results that the MMF1 limiter is unable to resolve some of the peaks present in the solution. However, all second-order limiters are surpassed by the third-order limiter MMF1-S2. The comparison of the different limiters is shown in figure 5.12. The reference solution is obtained by using the minmod limiter on 2000 grid points. The performance of Cada limiter and MMF1-S2 is good. The RMS error for this problem is shown in table 5.6. The MMF1-S2 limiter is more accurate than any other limiter and has the least RMS error. It is 53% and 20% more accurate than the MMF1 limiter and Cada limiter respectively.

Table 5.6
L_2 error of Shu-Osher problem

Limiters	$\mathbf{E}_\rho$	$\mathbf{E_u}$	$\mathbf{E_p}$	% $\mathbf{E}_\rho$
MMF1	0.121069366	0.047402636	0.190404816	100
Cada	0.094859715	0.022425722	0.094337962	78.3515
MM-S2	0.091178647	0.020078345	0.092606075	75.3111
MMF1-S2	0.078841668	0.021502648	0.094778111	65.1211
Convex-ENO	0.11988841	0.04358392	0.18415129	99.0246
Venkatakrishnan	0.1688152	0.08746029	0.227610	139.4367



(a) Density plot

(b) Zoomed view of discontinuity

(c) Zoomed view of discontinuity

(d) Zoomed view of discontinuity

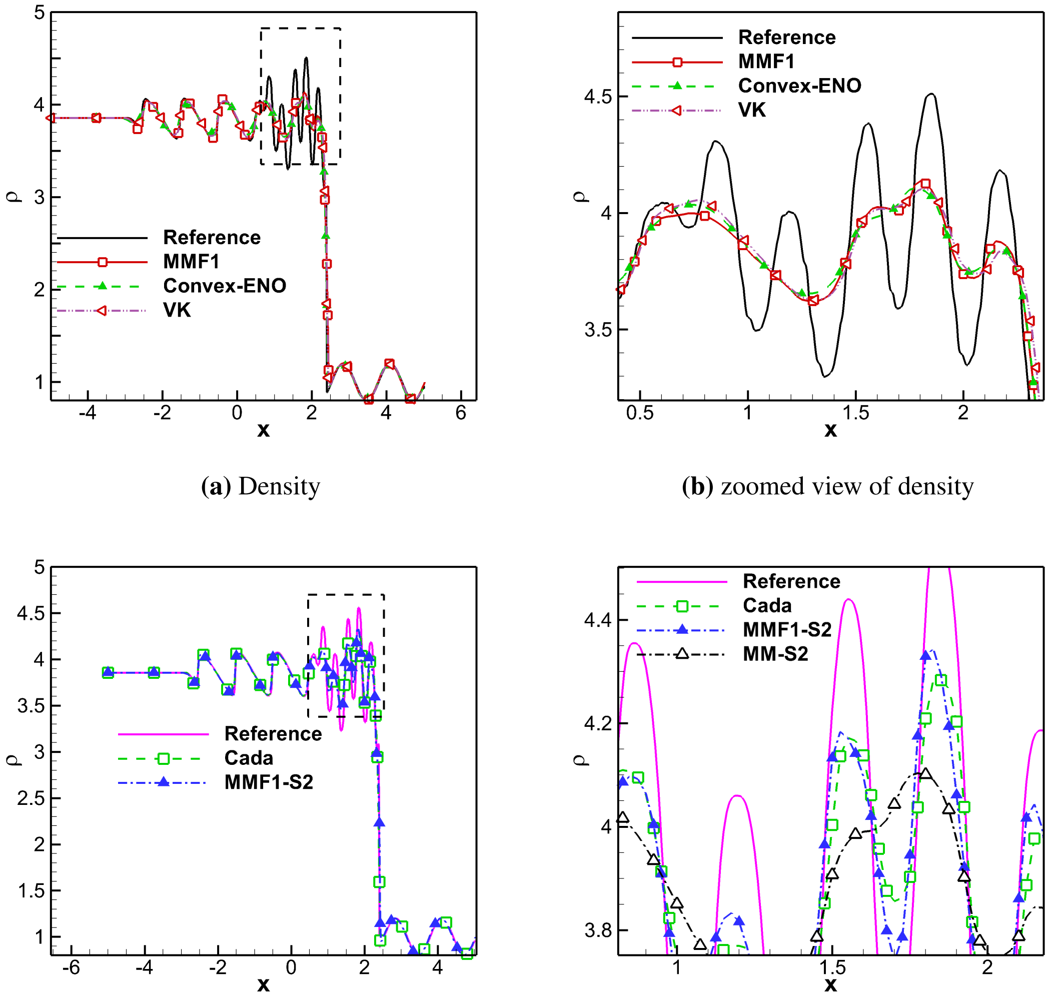
Figure 5.11: Density plot of Sod shock tube problem using 400 grids $CFL = 0.5$ at $t = 0.1s$

5.4.3 Blast waves interaction

In this test case, a low-pressure region is placed between two high-pressure regions on either side. The initial condition for this problem is

$$(\rho, u, p) = \begin{cases} (1, 0, 1000) & x \geq 0 \quad and \quad x \leq 0.1 \\ (1, 0, 0.01) & x > 0.1 \quad and \quad x <= 0.9 \\ (1, 0, 1000) & x > 0.9 \quad and \quad x <= 1 \end{cases}$$

The initial condition leads to the formation of two blast waves moving towards the central low-pressure zone. When these waves come into contact with each other, they create



(a) Density

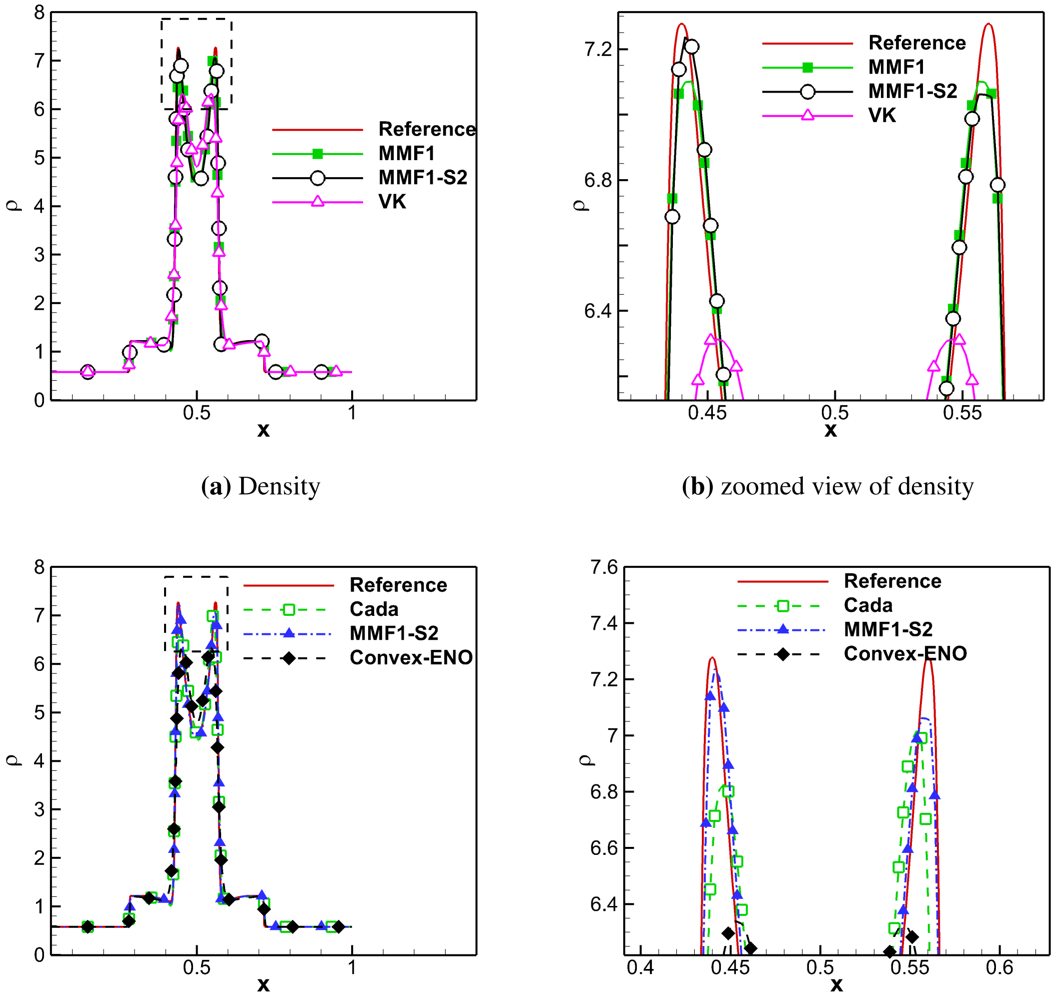
(b) zoomed view of density

(c) Density

(d) zoomed view of density

Figure 5.12: Comparison of limiters on Shu-Osher problem.

a high-pressure zone. The waves travel outward after this, and the pressure at the core decreases. This problem has a 10^5 pressure ratio, and the interacting blast wave will increase the pressure at the end of the shock tube by more than three times of the initial pressure. The solution is obtained by discretizing the domain [0, 1] with 400 grid points. The problem was solved up to 0.025 s. The reference solution is obtained by using 2000 grid points. Figure 5.13 shows the solution. The L_2 error of different schemes in this test case is shown in the table 5.7.



(a) Density

(b) zoomed view of density

(c) Density

(d) zoomed view of density

Figure 5.13: Density plot of blast wave interaction at t = 0.025 s

5.4.4 Summary

This chapter discusses new second-order and third-order limiters. The shock resolution properties and the accuracy of the new limiters are compared with other common limiters for standard benchmark problems. Although higher-order limiters are available in the limiter form used in [153], few limiters have successfully used second-derivative terms when expressed in the form used in [112]. Without considering the second-order derivative data, most of the limiters use higher-order first derivative information. In this work, second derivative terms are successfully integrated into the limiter without introducing any oscillations. Present limiters are checked on a uniform and an unstructured grid. Present

Table 5.7
L_2 error of Blast wave interaction test case

Limiters	$\mathbf{E}_\rho$	$\mathbf{E_u}$	$\mathbf{E_p}$	% $\mathbf{E}_\rho$
MMF1	0.07925544	21.44495695	21.7792556	100
Cada	0.19745523	21.40694564	30.2437603	249.1378
MM-S2	0.33655261	21.37375978	38.0867291	424.6429
MMF1-S2	0.07987625	21.44441891	18.9531758	100.7833
Convex-ENO	0.3359265	21.3678743	39.063563	423.8529
Venkatakrishnan (VK)	0.3407564	21.365641	40.90038	429.9470

limiters have shown improved property and accuracy in shock resolution compared to other limiters. On blast-wave problems, the performance of the current limiter is 247% and 420% more accurate than the Cada limiter and Convex-ENO methods respectively.

6
Weighted Essentially Non-Oscillatory Scheme

The classical Fourier series and higher-order polynomial cannot exactly reconstruct step-functions without spurious oscillations. The former one is called as Gibbs phenomena and the latter is called as Runge's Phenomena. Because of that, relatively lower-order methods (but high resolution) are used in the non-smooth regions and high accurate (higher-order) methods are used in the smooth regions. Essentially non-oscillatory(ENO) scheme uses smoothest among all the polynomials for the reconstruction [78]. Since conditional statements are computationally expensive, weighted essentially non-oscillatory schemes are studied in [154, 7] where weights are given to all the polynomials based on the smoothness of the solution [89]. To reduce the computational cost of finite volume WENO, finite difference WENO is explored in [155]. The classical fifth-order WENO procedure uses quadratic polynomial and can retain fifth-order accuracy in the smooth region [154, 7, 156]. Similarly, a seventh-order WENO is studied in [89, 157].

Because WENO is a versatile method that allows different ways to estimate smoothness indicator (β), weights to the polynomial (w) and ϵ (small number added to the denominator to avoid getting unphysical number), several researchers contributed to WENO in the development of various formulations for those basic parameters. Mapped-WENO [89] is an alternative to classical WENO, where a mapping function is suggested to obtain the weights so that it can maintain higher-order accuracy at the critical points and it is further refined in [158]. Different values of ϵ change the order of convergence of the scheme (i.e) high value of ϵ causes the schemes tend to higher-order and small value makes it degrade to ENO3 [159]. WENO-P [160] is a modified WENO scheme where a global higher-order smoothness indicator is used to find the variation in the smoothness indicator of the different polynomials and weights are adjusted based on that. WENO-Z [161] is similar to WENO-P with a modification in the global smoothness indicator and the smoothness indicator. In WENO-RL [162] a relative total variation limiter and relative smoothness limiter are introduced to improve grid-convergence of WENO. Since classical seventh order WENO performed poorly when two shocks present in the stencil, that is improved by TENO [163].

A new class of limiter that includes ENO scheme as one of the cases with limiters, where TVD is applied at the second derivative term and it is named as Power-ENO [164]. Parabolic reconstruction is common in WENO procedure, apart from parabolic reconstruction, WENO scheme uses wavelet basis is studied in [165], Lagrange basis in [166] and Hermite polynomial based HWENO is developed in [167]. An adaptive septic B-spline quasi-interpolation WENO scheme is presented in [168]. Moving least-square WENO is explored in [169]. The classical WENO procedure gives non-monotone negative weights in a staggered grid which is overcome in [170]. WENO scheme is improved by grid staggering the level-set function in [171].

The classical WENO scheme is more dissipative and it is not recommended for direct numerical simulation (DNS), so researchers attempted to improve the spectral property of WENO scheme. The spectral analysis of WENO is carried out in [172]. Initially, WENO scheme is optimized for better spectral property [173], eventually, the researchers found a better way to reduce numerical diffusion of WENO by adding one more point in WENO stencil that is called as Bandwidth WENO (BWENO) [174]. BWENO is optimized to minimize the spectral errors and it is explored in [175]. Both the WENO and BWENO had a biased stencil so a central WENO that can adapt its stencil between central and upwind is presented in [176, 177].

A sophisticated WENO for Large Eddy Simulation (LES) is studied in [178]. WENO with controllable artificial dissipation and dispersion is presented in [179]. Schemes that satisfy energy stability has better stability in the numerical results [180] and the energy stable WENO construction procedure is described in [181]. Numerical schemes that satisfy entropy inequality is called entropy stable scheme. WENO schemes satisfying such property is presented in [182, 183]. WENO method can also act as a limiter to Discontinuous Galerkin (DG) method [184]. The extension of WENO to the unstructured mesh is studied in [185, 186].

Apart from discretization and reconstruction, the accuracy of the solution also depends on Riemann solver. We refer [187] for the comparison of different Riemann solvers, WENO and MUSCL interpolation schemes. An adaptive characteristic-wise reconstruction of WENO-Z is explored in [188]. Where they proposed a hybrid component and characteristic based WENO scheme that can retain the accuracy of characteristic-wise reconstruction method with the lesser computational cost. Since the WENO scheme is computationally expensive, a hybrid WENO scheme that makes a switch between linear and non-linear scheme(WENO) is presented in [189]. WENO scheme is applied for traffic flows and sedimentation models to resolve multi-scale structures [190, 191].

6.1 Discretization of the classical schemes

The general form of the scalar equation is:

$$\frac{\partial \psi}{\partial t} + \frac{\partial u}{\partial x} = 0$$

The integral formulation is formulated by carrying volume averaging in the cell $I_i \equiv [i-0.5, i+0.5]$ and using method of line leads to,

$$\frac{\mathrm{d}\bar{\psi}_i}{\mathrm{d}t} = -\frac{1}{\Delta x}(u_{i+0.5} - u_{i-0.5}) \qquad \mathcal{O}(\Delta x^5)$$

where, $u_{i+0.5}$ is numerical flux at the cell interface and $u \in \mathbb{R}^3$.

$u_{i+0.5}$ is calculated using reconstruction scheme and the reconstruction scheme used here is WENO. Following [7], $i+0.5$ interface reconstruction flux for fifth-order scheme can be written as,

$$u_{i+0.5} = \frac{1}{60}(2\bar{u}_{j-2} - 13\bar{u}_{j-1} + 47\bar{u}_j + 27\bar{u}_{j+1} - 3\bar{u}_{j+2}) \tag{6.1}$$

and reconstruction flux at $i-0.5$ cell-face is

$$u_{i-0.5} = \frac{1}{60}(2\bar{u}_{j-3} - 13\bar{u}_{j-2} + 47\bar{u}_{j-1} + 27\bar{u}_j - 3\bar{u}_{j+1}) \tag{6.2}$$

From eq. 6.1 and eq. 6.2, we can notice that, for interpolating flux at $i+0.5$ and $i-0.5$, the coefficients are same but index is shifted. Here, this kind of behaviour is denoted as *index shifted stencil*. This index shifting property is essential for numerical conservativeness [103]. *Index shifted stencil* will appear only if the function is smooth, else that can be enforced by substituting $u_{i-0.5} = u_{i+0.5}(x - \Delta x)$, but this will end up in a reduction in the order of accuracy when one of the polynomials is not smooth. That means it will be dropped to second/third order in the non-smooth region instead of fourth-order. Finite difference ENO procedure with *reduced order polynomial* which is used to retain the order of accuracy of derivative. This occur when the left and the right face polynomial are not index shifted which is presented in [192]. In the present work, instead of using the *reduced order polynomials* for face flux calculation, the classical WENO polynomials are used.

The main difference between the classical WENO and the present scheme is: in the classical WENO schemes, smoothness indicator directly related to the weighting function of the polynomial but in the present procedure smoothness indicators are used only to decide the smooth polynomials. Once the smooth polynomials are identified using one computational economical switch, smoothness indicator will not play any role in further calculations. It is worth noting that the present switch does not use any conditional statements and can also act as a step-filter. The classical WENO scheme is very sensitive to a small perturbation in weights. This problem was overcome in WENO-Z scheme by introducing a global smoothness indicator. WENO-ZS scheme [193] can achieve fourth-order accuracy but it used the conditional statements. The present formulation is different from [166], where they presented one order-adaptive WENO scheme based on the Legendre-polynomial which can make the reconstruction and smoothness indicator compact. Here, the classical polynomials are used and the weighting procedure is also different. The present method can be extended to WENO scheme based on a non-polynomial basis.

A WENO scheme that can achieve fourth-order accuracy by including a fourth-order polynomial as a WENO scheme basis is presented in [194]. The advantage of the present scheme over this is: the present scheme uses three third-order polynomials and it can achieve fourth-order accuracy but WENO-AON(5,4,3) [194] used four polynomial basis to achieve the fourth-order accuracy. So the present scheme is computationally more economical. The present scheme is similar to WENO-EZ scheme proposed in [195] but in WENO-EZ decoupling the direct relationship between the weights and the smoothness indicators are not carried out, so that may not always perfectly satisfy Taylor series. In the presnt scheme, a standard value for the tuning parameter is proposed which works well for numerical test cases. In addition to that, further tuning can be easily achieved by changing it. Though the objective of the present scheme, convex-ENO [149], WENO-ZS [193], WENO-AON(5,4,3) [194] and WENO-EZ [195] are the same, the way in which the present scheme implemented is relatively simpler than others. The present schemes are easy to implement and it has a tuning parameter that can be changed based on the requirement of the problem. To the best of our knowledge, the present scheme is the first WENO scheme that decouples the direct relation between smoothness indicator and weight. Because of that, it can exactly satisfy Taylor series regardless of smoothness of that data. Though some of the other adaptive WENO schemes may show the expected rate of convergence on numerical test cases, they may not always satisfy Taylor series expansion. This is illustrated in a test case presented in the subsection 6.2.1.

6.2 The classical WENO formulation

For the third-order approximation, any variable can be written as

$$u(\xi) = a + b\xi + c\xi^2,$$

where ξ is the local co-ordinate of the variable u. Following[7, 89], the cell-average quantity is defined as

$$\bar{u}(x) = \frac{1}{\Delta x}\int_{x-0.5\Delta x}^{x+0.5\Delta x} u(\xi)\,\mathrm{d}\xi,$$

where $u \in \mathrm{I\!R}^n$;

$$\bar{u}(x) = a + bx + c\left(x^2 + \frac{\Delta x^2}{12}\right), \tag{6.3}$$

where $\Delta x = x_{i+0.5} - x_{i-0.5}$ for a uniform mesh. Here a, b and c are calculated by substituting $x_i = x_j + (i-j)\Delta x$ and the derivatives of any variable can be calculated as

$$\frac{\partial u}{\partial x} = \frac{u_{i+0.5} - u_{i-0.5}}{\Delta x}$$

where $x_{i+0.5} = \frac{1}{2}(x_i + x_{i+1})$; $i + 0.5$ is cell-center value; $u_{i+0.5} = a + b(\frac{\Delta x}{2}) + c(\frac{\Delta x}{2})^2$. a, b and c are calculated from averaged u (that is $\bar{u}$) and $u_{i+0.5}$ is calculated from non-averaged (point-based) u because of weak formulation followed in conservative schemes.

The equation formed using eq. 6.3 are

$$p^0_{i+0.5} = \frac{1}{6}(2\bar{u}_{i-2} - 7\bar{u}_{i-1} + 11\bar{u}_i) = h_{i+0.5} - \frac{1}{4}p'''(0)\Delta x^3 + O(\Delta x^4) \tag{6.4a}$$

$$p^1_{i+0.5} = \frac{1}{6}(-\bar{u}_{i-1} + 5\bar{u}_i + 2\bar{u}_{i+1}) = h_{i+0.5} + \frac{1}{12}p'''(0)\Delta x^3 + O(\Delta x^4) \tag{6.4b}$$

$$p^2_{i+0.5} = \frac{1}{6}(2\bar{u}_i + 5\bar{u}_{i+1} - \bar{u}_{i+2}) = h_{i+0.5} - \frac{1}{12}p'''(0)\Delta x^3 + O(\Delta x^4) \tag{6.4c}$$

$$h(x+0.5) = \sum_{k=0}^{2} \gamma_k p_{i+0.5}^k + O(\Delta x^5)$$

The ideal weights of those polynomials are $\gamma_0 = \frac{1}{10}$, $\gamma_1 = \frac{3}{5}$ and $\gamma_2 = \frac{3}{10}$

The final reconstruction formula of WENO-JS scheme is

$$u_{i+0.5} = \sum_j \omega_j p_{i+0.5}^j, \quad \tilde{\omega}_j = \frac{\gamma_j}{(\epsilon + \beta_j)^2}$$

The smoothness indicators [7] used here are

$$\beta_0 = \frac{13}{12}\left(\bar{u}_{i-2} - 2\bar{u}_{i-1} + \bar{u}_i\right)^2 + \frac{1}{4}\left(\bar{u}_{i-2} - 4\bar{u}_{i-1} + 3\bar{u}_i\right)^2, \tag{6.5}$$

$$\beta_1 = \frac{13}{12}\left(\bar{u}_{i-1} - 2\bar{u}_i + \bar{u}_{i+1}\right)^2 + \frac{1}{4}\left(\bar{u}_{i-1} - \bar{u}_{i+1}\right)^2, \tag{6.5}$$

$$\beta_2 = \frac{13}{12}\left(\bar{u}_i - 2\bar{u}_{i+1} + \bar{u}_{i+2}\right)^2 + \frac{1}{4}\left(3\bar{u}_i - 4\bar{u}_{i+1} + \bar{u}_{i+2}\right)^2. \tag{6.5}$$

The WENO scheme weights are

$$\omega_j = \frac{\tilde{\omega}_j}{\sum_j \tilde{\omega}_j}.$$

In case of WENO-Z scheme, the weights are calculated from

$$\tilde{\omega}_j^z = \gamma_j \left[1 + \frac{\tau}{(\epsilon + \beta_j)}\right],$$

where $\tau = |\beta_0 - \beta_2|$ and

$$\omega_j^z = \frac{\tilde{\omega}_j^z}{\sum_j \tilde{\omega}_j^z}. \tag{6.6}$$

Final reconstruction formula of WENO-Z scheme is

$$u_{i+0.5} = \sum_j \omega_j^z p_{i+0.5}^j.$$

In case of WENO-ZS scheme, the weights are calculated from

$$\tilde{\omega}_j^{zs} = \gamma_j \left[1 + \frac{\tau}{(\epsilon + \beta_j^{zs})} \right],$$

$$\beta_k^{zs} = \frac{\beta_j + \epsilon}{\beta_j + \epsilon + \tau},$$

$$\omega_j^{zs} = \frac{\tilde{\omega}_j^{zs}}{\sum_j \tilde{\omega}_j^{zs}}.$$

The reconstruction formulation of WENO-ZS is

$$u_{i+0.5}^{zs1} = \sum_j \omega_j^{zs} p_{i+0.5}^j.$$

The final reconstruction formulation of WENO-ZS scheme is

$$u_{i+0.5} = \begin{cases} h_0^4, & \text{if} \quad \tau_4^0 \leq min(\beta_i) \quad \text{and} \quad \tau_4^1 > min(\beta_i) \\ h_1^4, & \text{if} \quad \tau_4^0 > min(\beta_i) \quad \text{and} \quad \tau_4^1 \leq min(\beta_i) \\ u_{i+0.5}^{zs1}, & \text{otherwise} \end{cases}$$

where,

$$h_0^4 = \frac{1}{12}u_{i-2} - \frac{5}{12}u_{i-1} + \frac{13}{12}u_i + \frac{1}{4}u_{i+1},$$

$$h_1^4 = -\frac{1}{12}u_{i-1} + \frac{7}{12}u_i + \frac{7}{12}u_{i+1} - \frac{1}{12}u_{i+2}.$$

We refer to [163] for TENO5 scheme and TENO5-opt scheme. We also refer to [195] for WENO-EZ scheme and WENO-EJS scheme.

6.2.1 Some of the limitations of WENO scheme

In general, the reconstruction formulation can be written as

$$p_{i\pm0.5}^k = h_{i\pm0.5} + A_k \Delta x^3 + O(\Delta x^4) \quad k = 0, 1, 2,$$

where the quantities of A_k is given in eq. 6.4. The necessary and sufficient conditions for the fifth-order convergence are [161, 196]

$$\sum_{k=0}^{2} \omega_k^{\pm} - \gamma_k = O(\Delta x^6), \tag{6.7a}$$

$$\sum_{k=0}^{2} A_k(\omega_k^{+} - \omega_k^{-}) = O(\Delta x^3), \tag{6.7b}$$

$$\omega_k^{\pm} - \gamma_k = O(\Delta x^2). \tag{6.7c}$$

where ω^+ and ω^- are the right and left state weights of WENO scheme. WENO-JS scheme cannot satisfy the conditions in eq. 6.7a if one of the polynomials is not smooth. That is illustrated in the following examples. The general form of the fifth-order WENO scheme is

$$u_{i+0.5} = A_{i-2}\bar{u}_{j-2} + A_{i-1}\bar{u}_{j-1} + A_i\bar{u}_j + A_{i+1}\bar{u}_{j+1} + A_{i+2}\bar{u}_{j+2}$$

Let us consider one test case which has discontinuity and curvatures, and study the behavior of WENO schemes. The test case is

$$y = \sin(2x) + \tanh(2x) + 5 \times \tanh(10000x)$$

Interpolation is carried over the interval $[-2\pi, 2\pi]$ with 101 grid points. All the WENO schemes work well on $\sin(x)$ function so we have chosen “sin” as a basis. Because WENO schemes are designed for the problems having shocks, $\tanh(10000x)$ is incorporated to introduce a step-like curve in the solution. To make a test case moderately smooth at all the points except shock, $\tanh(2x)$ is added. Now the problem has moderate smoothness and discontinuity like data. We shall consider one point which is smooth that is $x = -4.021238596594936$ and its index is $i = 90$. So all the points from $i - 2$ to $i + 2$ should be smooth. The corresponding smoothness indicators are $\beta_0 = 0.067435814878532$, $\beta_1 = 0.072121290715112$ and $\beta_2 = 0.067435814878532$. The coefficients of different interpolation WENO schemes and cell-averaged WENO schemes are tabulated in Table 6.1 and Table 6.2, respectively. The given function is smooth except at the location of discontinuity, WENO-JS scheme can only satisfy Taylor series up to third-order terms and this limitation is overcome by WENO-Z scheme. Though WENO-JS scheme gives non-oscillatory good derivative calculation on smooth function like $f(x) = \sin(x)$, it is unable to give the good derivative approximation on semi-smooth step-like functions (for example $f(x) = \tanh(kx)$ when $k = 5$ to 25). This can be partially rectified by modifying the weighting function similar to WENO-Z scheme but a permanent solution is to decouple the direct relation between the smoothness indicator and the weighting function. The smoothness indicator can take a spectrum of value depends on the smoothness of data but Taylor series supposed to be a fixed value for a given order else it may introduce a small error in the coefficients of WENO schemes. This is the reason why WENO-JS scheme is unable to give the desired values but it is closer to that.

The coefficients of different schemes at $x = -4.021238596594936$ are tabulated in Table 6.3. The smoothness indicator at the point are $\beta_0 = 0.067435814878532$, $\beta_1 = 0.072121290715112$ and $\beta_2 = 0.067435814878532$. Here, the cell-average formulation is considered because interpolation-WENO is not provided in the literature of the WENO schemes considered in the table. For this small perturbation case, WENO-EZ scheme and TENO5 scheme can able to achieve the desired formulation but WENO-EJS and TENO5-opt unable to achieve the desired formulation. It is worth noting that TENO5-opt scheme is optimized for high wavenumber resolution so it may not satisfy the desired order.

Lets include the discontinuity point in WENO stencil and study the behavior of different WENO schemes. The point is $x = -0.376991118430776$ and its grid point index $i = 49$. Here, the function in the grid points index 49,50 and 51 are not smooth because of step-like curve but the grid points 48 to 50 are smooth. The corresponding smoothness indicators for grid points 49 to 51 are $\beta_0 = 0.332650488368829$, $\beta_1 = 0.374653137007571$ and $\beta_2 = 34.163939596847023$ respectively. The coefficients of different interpolation WENO schemes and cell-average WENO schemes at $x = -0.376991118430776$ is tabulated in Table 6.4 and Table 6.5 respectively.

WENO-JS scheme and WENO-Z schemes are different from ideal fourth-order reconstruction scheme but WENO-ZS scheme can able to achieve a fourth-order accuracy.

Table 6.1
Coefficients of different interpolation WENO schemes for oscillating step function at grid point index = 90 ($x = -4.021238596594936$)

A_x	**WENO-JS**	**WENO-Z**	**WENO-ZS**	**Ideal fifth order**
A_{i-2}	$\frac{14662848604299963}{576460752303423488} \approx$ 0.025435988	$\frac{3}{128} =$ 0.0234375	$\frac{3}{128} =$ 0.0234375	$\frac{3}{128} =$ 0.0234375
A_{i-1}	$-\frac{45804029421831337}{288230376151711744} \approx$ -0.15891465	$-\frac{5}{32} =$ -0.15625	$-\frac{5}{32} =$ -0.15625	$-\frac{5}{32} =$ -0.15625
A_i	$\frac{403019867018972067}{576460752303423488} \approx$ 0.69912802	$\frac{45}{64} =$ 0.703125	$\frac{45}{64} =$ 0.703125	$\frac{45}{64} =$ 0.703125
A_{i+1}	$\frac{536765969787069}{1125899906842624} \approx$ 0.468750000081537	$\frac{15}{32} =$ 0.46875	$\frac{15}{32} =$ 0.46875	$\frac{15}{32} =$ 0.46875
A_{i+2}	$-\frac{6109520251791289}{144115188075855872} \approx$ -0.052083333364694	$-\frac{5}{128} =$ -0.0390625	$-\frac{5}{128} =$ -0.0390625	$-\frac{5}{128} =$ -0.0390625

Table 6.2
Coefficients of different cell-averaged WENO schemes for oscillating step function at grid point index = 90 ($x = -4.021238596594936$)

A_x	**WENO-JS**	**WENO-Z**	**WENO-ZS**	**Ideal fifth-order**
A_{i-2}	$\frac{2597867956555469}{72057594037927936} \approx$ 0.03605266	$\frac{1}{30} \approx$ 0.033333333	$\frac{1}{30} \approx$ 0.033333333	$\frac{1}{30} \approx$ 0.033333333
A_{i-1}	$-\frac{31812801882309515}{144115188075855872} \approx$ - 0.22074566	$-\frac{13}{60} \approx$ -0.2166666667	$-\frac{13}{60} \approx$ -0.2166666667	$-\frac{13}{60} \approx$ -0.2166666667
A_i	$\frac{112302386193548207}{144115188075855872} \approx$ 0.77925434	$\frac{47}{60} \approx$ 0.783333	$\frac{47}{60} \approx$ 0.783333	$\frac{47}{60} \approx$ 0.783333
A_{i+1}	$\frac{8277933965146523}{18014398509481984} =$ 0.45951764	$\frac{9}{20} =$ 0.45	$\frac{9}{20} =$ 0.45	$\frac{9}{20} =$ 0.45
A_{i+2}	$-\frac{974200483708243}{18014398509481984} =$ - 0.05407899	$-\frac{1}{20} =$ -0.05	$-\frac{1}{20} =$ -0.05	$-\frac{1}{20} =$ -0.05

Table 6.3
Coefficients of different cell-averaged WENO schemes for oscillating step function at grid point index = 90 ($x = -4.021238596594936$)

A_x	**WENO-EJS**	**WENO-EZ**	**TENO5-opt**	**TENO5**	**Ideal fifth order**
A_{i-2}	$\frac{7398008828397373}{216172782113783808} \approx$ 0.034222665573612	$\frac{1}{30} \approx$ 0.03333333	$\frac{11}{60} \approx$ 0.1833333	$\frac{1}{30} \approx$ 0.0333333	$\frac{1}{30} \approx$ 0.0333333
A_{i-1}	$-\frac{94251620523120931}{432345564227567616} \approx$ -0.218000665027087	$-\frac{13}{60} \approx$ -0.21666667	$-\frac{17}{24} \approx$ - -0.7083333	$-\frac{13}{60} \approx$ -0.21666666	$-\frac{13}{60} \approx$ -0.21666666
A_i	$\frac{338093943704450191}{432345564227567616} \approx$ 0.781999334972921	$\frac{47}{60} \approx$ 0.7395833	$\frac{163}{120} \approx$ 1.3583333	$\frac{47}{60}$ = 0.703125	$\frac{47}{60}$ = 0.703125
A_{i+1}	$\frac{48975312468658715}{108086391056891904} \approx$ 0.453112662840970	$\frac{9}{20}$ = 0.45	$\frac{7}{40} =$ 0.175	$\frac{9}{20}$ = 0.45	$\frac{9}{20}$ = 0.45
A_{i+2}	$-\frac{5548506621297811}{108086391056891904} \approx$ -0.051333998360416	$-\frac{1}{20}$ = - 0.05	$-\frac{1}{120} \approx$ -0. -0.00833333	$-\frac{1}{20}$ = -0.05	$-\frac{1}{20}$ = -0.05

Table 6.4
Coefficients of different interpolation WENO schemes for oscillating-step function at grid point index = 49 ($x = -0.376991118430776$)

A_x	**WENO-JS**	**WENO-Z**	**WENO-ZS**	**Ideal fourth-order**
A_{i-2}	$\frac{562949954063591}{27021597764222976} \approx$ 0.020833333357102	$\frac{601279451362269}{18014398509481984} \approx$ 0.033377714556819	$\frac{1}{16}$ = 0.0625	$\frac{1}{16}$ = 0.0625
A_{i-1}	$-\frac{3190049739061259}{18014398509481984} \approx$ -0.177083333500264	$-\frac{9554215325978265}{36028797018963968} \approx$ -0.265182745928190	$-\frac{5}{16}$ = -0.3125	$-\frac{5}{16}$ = -0.3125
A_i	$\frac{6661574452045037}{9007199254740992} \approx$ 0.739583333691511	$\frac{1605930795259626629}{1729382256910270464} \approx$ 0.928615283776993	$\frac{15}{16}$ = 0.9375	$\frac{15}{16}$ = 0.9375
A_{i+1}	$\frac{50665495774342165}{108086391056891904} \approx$ 0.468749999689360	$\frac{1054254984927821929}{3458764513820540928} \approx$ 0.304806812003311	$\frac{5}{16}$ = 0.3125	$\frac{5}{16}$ = 0.3125
A_{i+2}	$-\frac{1876499841292507}{36028797018963968} \approx$ -0.052083333237710	$-\frac{5593044994176917}{3458764513820540928} \approx$ -0.001617064408932	0 = 0	0 = 0

Table 6.5
Coefficients of different cell-averaged WENO schemes for oscillating-step function at grid point index = 49 ($x = -0.376991118430776$)

A_x	**WENO-JS**	**WENO-Z**	**WENO-ZS**	**Ideal fourth order**
A_{i-2}	$\frac{2095780213163101}{36028797018963968} \approx$ 0.058169586	$\frac{2818262907093287}{54043195528445952} \approx$ 0.052148339	$\frac{1}{12} \approx$ 0.083333	$\frac{1}{12} \approx$ 0.083333
A_{i-1}	$-\frac{24583684197797899}{72057594037927936} \approx$ -0.34116715	$-\frac{34760130846314659}{108086391056891904} \approx$ -0.32159581	$-\frac{5}{12}$ = -0.416667	$-\frac{5}{12}$ = -0.416667
A_i	$\frac{74363789361233028515}{73786976294838206464} \approx$ 1.0078173	$\frac{851920277382308059}{864691128455135232} \approx$ 0.98523074	$\frac{13}{12} \approx$ 1.083333	$\frac{13}{12} \approx$ 1.083333
A_{i+1}	$\frac{40610664138388541487}{147573952589676412928} \approx$ 0.27518856	$\frac{494140904351336815}{1729382256910270464} \approx$ 0.28573261	$\frac{1}{4}$ = 0.25	$\frac{1}{4}$ = 0.25
A_{i+2}	$-\frac{1220787204155811}{147573952589676412928} \approx$ -0.000008272	$-\frac{873840563877601}{576460752303423488} \approx$ -0.0015158717	0 = 0	0 = 0

Table 6.6
Coefficients of different cell-averaged WENO schemes for oscillating step function at grid point index = 49 ($x = -0.376991118430776$)

A_x	**WENO-EJS**	**WENO-EZ**	**TENO5-opt**	**TENO5**	**Ideal 4^{th} order**
A_{i-2}	$\frac{2420991034576273}{27021597764222976}$ $\approx$ 0.0895946	$\frac{2675451579191675}{27021597764222976}$ $\approx$ 0.099011598	$\frac{11}{57}$ 0.19298246	$\frac{1}{21} \approx$ 0.047619048	$\frac{1}{12} \approx$ 0.08333
A_{i-1}	$-\frac{234972721471584443}{54043195528445952}$ $\approx$ -0.43478688	$-\frac{250575962371317785}{54043195528445952}$ $\approx$-0.46365867	$-\frac{85}{114} \approx$ - -0.74561404	$-\frac{13}{42} \approx$ -0.30952381	$-\frac{5}{12} \approx$ -0.416666
A_i	$\frac{634179440384448387}{576460752303423488}$ $\approx$1.1001259	$\frac{3909233169316943459}{3458764513820540928}$ $\approx$1.1302398	$\frac{161}{114} \approx$ 1.4122807	$\frac{41}{42}$ = 0.97619048	$\frac{13}{12}$ = 1.08333
A_{i+1}	$\frac{849922328668799821}{3458764513820540928}$ $\approx$ 0.24573	$\frac{1621815402160180655}{6917529027641081856}$ $\approx$ 0.2344501	$\frac{8}{57} =$ 0.14035088	$\frac{2}{7}$ = 0.28571429	$\frac{1}{4}$ = 0.25
A_{i+2}	$-\frac{2295892162571945}{3458764513820540928}$ $\approx$-0.000663	$-\frac{98666357728353}{2305843009213693952}$ $\approx$- 0.0000427	0 = 0	0 = 0	0 = 0

WENO-JS and WENO-Z lead to a third-order accurate interpolation formulation instead of a fourth-order accurate formulation. This limitation cannot be overcome by straightforward modifications in WENO weighting procedure, so a new weighting function is proposed to take care of these limitations. In WENO-ZS scheme, it is taken care by explicitly using conditional statement which may not desirable in WENO scheme. Because the ideal WENO scheme not supposed to use any conditional statements.

Coefficients of several cell-average WENO schemes at $x = -0.376991118430776$ is shown in Table 6.6. From the table, it is clear that no WENO schemes considered here are able to give the desired fourth-order formulation. WENO-EZ and WENO-EJS are derived for the fourth-order accuracy but they failed to give a fourth-order accurate formulation in this test case. The performance of WENO-E scheme needs improvement when one of the polynomials is not smooth but it can give a fourth-order convergence when one of the polynomials is extremely non-smooth. For the test case considered here, WENO-E scheme supposed to give a fourth-order convergence but gave a third-order convergence. WENO-E scheme can give a third-order convergence in the numerical test case but they did not exactly satisfy the Taylors series up to third order term. It has a truncation error order of 10^{-10}. The formulation of WENO-E scheme is good but its sensitivity is not good to detect the non-smoothness present in the simulation. It does not provide an adjusting parameter to control the order adaptive capability.

In the present scheme, we have explicitly provided a parameter(c_o) to control the order adaptive property so that the user can tune it based on their requirement. We have also provided a standard value which works well for most of the test cases. TENO-opt scheme and TENO scheme gave a third-order accuracy. None of the schemes considered here is able to achieve the fourth-order accuracy. A study on WENO-JS scheme and upwind scheme on shock-heavy bubble interaction has shown that WENO-JS scheme performed poorly than up-wind scheme [156]. So it is clear that the weighting procedure of WENO-JS needs more refinement.

The summary of this section is given below. The order of accuracy of different WENO schemes is evaluated in two cases.

- In the first case, the solution is smooth but the variation in the smoothness indicator is less than 5%. In that case, all the WENO schemes based on JS-family are unable to produce desired formulation but Z-family of WENO schemes are able to handle small perturbations in the weights.

- In case two, the order of accuracy is studied, where one of the polynomials is not smooth.

- In case two, WENO-ZS is the only polynomial able to give desired formulation others are unable to provide the desired formulation.

- Though WENO-E is an order adaptive scheme, it is unable to achieve order adaptive property for this case. It can achieve order adaptive only when one of the polynomials is extremely non-smooth.

To overcome these limitations, we have proposed a scheme which can decouple the direct effect of the smoothness indicator on the WENO weights, so it can exactly satisfy Taylor series regardless of the smoothness of data. In addition to that, it has a better order adaptive character.

6.3 Methodology

The different types of order adaptive WENO scheme (WENO-OA) presented in this work are:

1. FD-WENO-OA
    - It is a conservative finite difference order-adaptive scheme and it is based on cell-average interpolation.
2. FD-WENO-OA-p
    - It is a finite difference order-adaptive scheme which is based on point-valued interpolation.
3. FV-WENO-OA

- It is a finite volume order-adaptive scheme that calculates the left and the right state using cell-average data.

4. FV-WENO-OA-p

   - It is a finite volume order-adaptive scheme which calculates left and the right state using point-valued reconstruction.

6.3.1 FD-WENO-OA scheme

Table 6.7 shows the desired final reconstruction polynomial ($u_{i+0.5}$) and weights(α_i) to the reconstruction polynomials so that the order of accuracy of the derivative is high. In table 6.7, "good" stands for smooth polynomial and "bad" stands for the non-smooth polynomial. If all the polynomials are "good", the weights are same as ideal weights of WENO scheme. From the studies in subsection 6.2.1, it is clear that WENO-JS, WENO-Z and TENO schemes did not produce the desired fourth-order polynomial when the two consecutive WENO polynomial basis are smooth. A new weighting function that passes through the values of α_i tabulated in the Table 6.7 is presented. The lower-order accurate formulation is only used when the centre polynomial is not smooth. Before formulating the weights, we should distinguish "good" and "bad" polynomials. This information can be obtained from weights or smoothness indicator. The basis polynomial is "good" if the weight obtained using eq. 7.6 is higher or almost equivalent to the ideal weights. Similarly, the basis polynomial is "bad", if that is much lower than the ideal weights. Here, we should fix a cut-off with respect to the ideal weights. If the weight is higher than the cut-off value the basis polynomial is "good" else that is "bad". Using simple algebraic manipulation, the following formulation is proposed to decide the "good" and "bad" polynomials.

$$a_i = \frac{\text{floor}\left(\frac{\gamma_i}{\omega_i c_0}\right)}{\text{floor}\left(\frac{\gamma_i}{\omega_i c_0}\right) + \epsilon}, \quad i = 0, 1, 2 \tag{6.8}$$

Eq. 6.8 returns "zero" or "one", where "one" stands for "good" polynomial and "zero"

Table 6.7
Ideal weights and desired final reconstruction formulation for $u_{i+0.5}$ (cell-averaged)

$\mathbf{p^0_{i+0.5}}$	$\mathbf{p^1_{i+0.5}}$	$\mathbf{p^2_{i+0.5}}$	**Desired final $\mathbf{u_{i+0.5}}$**	$\alpha^{\mathbf{n}}_{\mathbf{0}}$	$\alpha^{\mathbf{n}}_{\mathbf{1}}$	$\alpha^{\mathbf{n}}_{\mathbf{2}}$
good	good	good	$\frac{47\bar{u}_i}{60} - \frac{13\bar{u}_{i-1}}{60} + \frac{\bar{u}_{i-2}}{30} + \frac{9\bar{u}_{i+1}}{20} - \frac{\bar{u}_{i+2}}{20}$	$\frac{1}{10}$	$\frac{6}{10}$	$\frac{3}{10}$
bad	good	good	$\frac{7\bar{u}_i}{12} - \frac{\bar{u}_{i-1}}{12} + \frac{7\bar{u}_{i+1}}{12} - \frac{\bar{u}_{i+2}}{12}$	0	$\frac{1}{2}$	$\frac{1}{2}$
good	good	bad	$\frac{13\bar{u}_i}{12} - \frac{5\bar{u}_{i-1}}{12} + \frac{\bar{u}_{i-2}}{12} + \frac{\bar{u}_{i+1}}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	0

stands for “bad” polynomial. ϵ is a small number added to the denominator to avoid getting an unphysical number. Any number lower than 10^{-18} for ϵ can serve the purpose for most of the double precision computation. A tuning parameter $c_o = \frac{3}{10}$ is good for most of the test cases considered here but we used $c_o = \frac{1}{90}$ to $c_o = \frac{7}{10}$. If the problem has a strong or complex shock structure like Sedov test case or double Mach reflection, a high value of c_o is recommended. If the problem is relatively simpler like Sod shock tube problem low value is recommended. A low value of c_o can make the scheme biased to higher-order accuracy and vice versa. The effect of c_o of the scheme on the solution is presented in subsection 6.6.4. A low value of c_o can also sharpen the shock and may give oscillations. Higher the value may dissipate the signal and reduces the oscillations. Here, $floor$ function is used because $floor$ is a round-off operation and it is based on chopping the decimal values. This *switch* can be used as a step-filter. Because it is not using any conditional statements, it is computationally less expensive than the traditional *switch* used in numerical methods. This concept can be employed to remove the conditional statement from TENO and convex-ENO schemes so that TWENO or convex-WENO scheme can be formulated. Once a_i are determined, the final weights are framed such that, it passes through the values tabulated in table 6.7. An elegant and easy way to calculate this weights in linear space is pretested in section 6.4. Both the formulation will lead to the same result but the formulation provided here is compact.

$$\alpha_0^n = t_1\left(a_0\frac{1}{4} - t_2\frac{3}{20}\right) + t_3\frac{a_0}{s_a}, \tag{6.9a}$$

$$\alpha_1^n = t_1\left(a_0a_1\frac{3}{4} + a_1a_2\frac{1}{2} - t_2\frac{13}{20}\right) + t_3\frac{a_1}{s_a}, \tag{6.9b}$$

$$\alpha_2^n = t_1\left(\frac{1}{2}a_2 - t_2\frac{1}{5}\right) + t_3\frac{a_2}{s_a}, \tag{6.9c}$$

where $t_1 = a_0a_1 + a_1a_2$; $t_1 = \frac{t_1}{(t_1+10^{-18})}$; $s_a = a_0 + a_1 + a_2$; $t_2 = a_0a_1a_2$; $t_3 = |t_1 - 1|$; Here t_1 will give “one”, if two consecutive polynomials are smooth else it will give “zero”. If t_1 is “zero”, equal weights will be given to other smooth polynomials.

The final reconstruction polynomial is

$$p_{i+0.5} = \sum_{j=0}^{2} \alpha_j^n p_{i+0.5}^j. \tag{6.10}$$

The cut-off could be set once the smoothness indicators are determined but in this work that is used after the standard WENO procedure.

6.3.2 WENO-OA-p scheme (interpolation WENO scheme)

The WENO scheme is a powerful interpolation scheme for discontinuous data thanks to its non-linear switching procedure. Note that in FV-WENO scheme, $u_{i+0.5}$ is calculated from cell-average values and interpolation WENO scheme is formulated from point-values. The interpolation polynomials are [155]:

$$r_{i+0.5}^0 = \frac{1}{8}(3u_{i-2} - 10u_{i-1} + 15u_i); \quad r_{i+0.5}^1 = \frac{1}{8}(-u_{i-1} + 6u_i + 3u_{i+1}); \quad r_{i+0.5}^2 = \frac{1}{8}(3u_i + 6u_{i+1} - u_{i+2});$$

and the ideal weights are

$\gamma_0 = \frac{1}{16}$, $\gamma_1 = \frac{5}{8}$ and $\gamma_2 = \frac{5}{16}$

$$\tilde{\omega}_j = \frac{\gamma_j}{(\epsilon + \beta)^2}, \quad \omega_j = \frac{\tilde{\omega}_j}{\sum_{j=0}^{2}\tilde{\omega}_j}$$

Similar to FD-WENO-OA, cut-off or switch step is

$$a_i = \frac{\text{floor}\left(\frac{\omega_i}{\omega_i c_0}\right)}{\text{floor}\left(\frac{\omega_i}{\omega_i c_0}\right) + \epsilon}, \quad i = 0, 1, 2 \tag{6.11}$$

Table 6.8
Ideal weights and desired interpolation formulation for $u_{i+0.5}$ (interpolation)

$\mathbf{r^0_{i+0.5}}$	$\mathbf{r^1_{i+0.5}}$	$\mathbf{r^2_{i+0.5}}$	**Desired final** $\mathbf{u_{i+0.5}}$	$\alpha^{\mathbf{n}}_{\mathbf{0}}$	$\alpha^{\mathbf{n}}_{\mathbf{1}}$	$\alpha^{\mathbf{n}}_{\mathbf{2}}$
good	good	good	$\frac{45u_i}{64} - \frac{5u_{i-1}}{32} + \frac{3u_{i-2}}{128} + \frac{15u_{i+1}}{32} - \frac{5u_{i+2}}{128}$	$\frac{1}{16}$	$\frac{5}{8}$	$\frac{5}{16}$
bad	good	good	$\frac{9u_i}{16} - \frac{u_{i-1}}{16} + \frac{9u_{i+1}}{16} - \frac{u_{i+2}}{16}$	0	$\frac{1}{2}$	$\frac{1}{2}$
good	good	bad	$\frac{15u_i}{16} - \frac{5u_{i-1}}{16} + \frac{u_{i-2}}{16} + \frac{5u_{i+1}}{16}$	$\frac{1}{6}$	$\frac{5}{6}$	0

Eq. 6.11 gives “zero” if the polynomial is not smooth else it gives “one”. The ideal interpolation formulation and corresponding weights are given in Table 6.8. Using an algebraic mapping function such that the final weights satisfy the weights given in Table 6.8 based on the smoothness of the solution. The final weights for WENO-OA-p are:

$$\alpha_0^n = t_1\left(a_0\frac{1}{6} - t_2\frac{5}{48}\right) + t_3\frac{a_0}{s_a}, \tag{6.12a}$$

$$\alpha_1^n = t_1\left(a_0a_1\frac{5}{6} + a_1a_2\frac{1}{2} - t_2\frac{17}{24}\right) + t_3\frac{a_1}{s_a}, \tag{6.12b}$$

$$\alpha_2^n = t_1\left(\frac{1}{2}a_2 - t_2\frac{3}{16}\right) + t_3\frac{a_2}{s_a}, \tag{6.12c}$$

Where $t_1 = a_0a_1 + a_1a_2$; $t_1 = \frac{t_1}{(t_1+10^{-18})}$; $s_a = a_0 + a_1 + a_2$; $t_2 = a_0a_1a_2$; $t_3 = |t_1 - 1|$; The final reconstruction formulation is

$$r^l_{i+0.5} = \sum_{j=0}^{2} \alpha_j^n r^j_{i+0.5}$$

It is worth noting that eq. 6.9 and eq. 6.12 are different. The former one is for FD-WENO-OA (conservative FD-WENO-OA) which is commonly called as finite-difference WENO in most of the literature and the later one is for interpolation WENO scheme(FD-WENO-OA-p).

6.3.3 FV-WENO-OA scheme

Here, WENO-OA for finite volume method is presented. Two ways of interpolation to calculate interface value is considered. The first one is the cell-averaged interpolation, the second one is a point-valued interpolation used in [197] to solve hyperbolic equations. The former one is commonly used in WENO schemes to solve conservative equations.

FV-WENO-OA scheme based on the cell average value

Finite volume method for compressible flow needs the left and the right state calculation and the final flux is determined using a Riemann solver [16]. For the left state calculation, the procedure described in subsection 6.3.1 is used. Here, the procedure to calculate the right state using the points $i-1$ to $i+3$ is presented. The polynomials used to calculate right state are

$$p^0_{i+0.5} = \frac{1}{6}(-\bar{u}_{i-1}+5\bar{u}_i+2\bar{u}_{i+1}); \quad p^1_{i+0.5} = \frac{1}{6}(2\bar{u}_i+5\bar{u}_{i+1}-\bar{u}_{i+2}); \quad p^2_{i+0.5} = \frac{1}{6}(2\bar{u}_{i+3}-7\bar{u}_{i+2}+11\bar{u}_{i+1})$$

and the ideal weights are

$\gamma_1 = \frac{3}{10}$, $\gamma_2 = \frac{6}{10}$ and $\gamma_3 = \frac{1}{10}$

$$\tilde{\omega}_j = \frac{\gamma_j}{(\epsilon+\beta)^2} \qquad \omega_j = \frac{\tilde{\omega}_j}{\sum_{j=0}^{2}\tilde{\omega}_j} \tag{6.13}$$

The smooth and the non-smooth polynomials are identified using following expressions

$$a_i = \frac{\text{floor}\left(\frac{\gamma_i}{\omega_i c_0}\right)}{\text{floor}\left(\frac{\gamma_i}{\omega_i c_0}\right)+\epsilon}, \quad i = 0,1,2 \tag{6.14}$$

where $\epsilon = 10^{-18}$ is used to avoid getting unphysical numbers.

Once the smooth and non-smooth polynomials are identified, the final weights are calculated from eq. 6.15. The ideal weights and formulation for right state of $u_{i+0.5}$ is given in table 6.9.

Table 6.9
Ideal weights and desired final reconstruction formulation for FV-WENO-OA to calculate right state of $u_{i+0.5}$

$\mathbf{p^0_{i+0.5}}$	$\mathbf{p^1_{i+0.5}}$	$\mathbf{p^2_{i+0.5}}$	**Desired final** $\mathbf{u_{i+0.5}}$	$\alpha^{\mathbf{p}}_{\mathbf{0}}$	$\alpha^{\mathbf{p}}_{\mathbf{1}}$	$\alpha^{\mathbf{p}}_{\mathbf{2}}$
good	good	good	$\frac{47\bar{u}_{i+1}}{60} + \frac{9\bar{u}_i}{20} - \frac{\bar{u}_{i-1}}{20} - \frac{\bar{u}_{i+2}}{60} + \frac{\bar{u}_{i+3}}{30}$	$\frac{3}{10}$	$\frac{6}{10}$	$\frac{1}{10}$
bad	good	good	$\frac{13\bar{u}_{i+1}}{12} + \frac{\bar{u}_i}{4} - \frac{5\bar{u}_{i+2}}{12} + \frac{\bar{u}_{i+3}}{12}$	0	$\frac{3}{4}$	$\frac{1}{4}$
good	good	bad	$\frac{7\bar{u}_{i+1}}{12} + \frac{7\bar{u}_i}{12} - \frac{\bar{u}_{i-1}}{12} - \frac{\bar{u}_{i+2}}{12}$	$\frac{1}{2}$	$\frac{1}{2}$	0

$$\alpha_0^p = t_1\left(a_0\frac{1}{2} - t_2\frac{1}{5}\right) + t_3\frac{a_0}{s_a}, \tag{6.15a}$$

$$\alpha_1^p = t_1\left(a_0a_1\frac{1}{2} + a_1a_2\frac{3}{4} - t_2\frac{13}{20}\right) + t_3\frac{a_1}{s_a}, \tag{6.15b}$$

$$\alpha_2^p = t_1\left(\frac{1}{4}a_2 - t_2\frac{3}{20}\right) + t_3\frac{a_2}{s_a}, \tag{6.15c}$$

Where $t_1 = a_0a_1 + a_1a_2$; $t_1 = \frac{t_1}{(t_1+10^{-18})}$; $s_a = a_1 + a_2 + a_0$; $t_2 = a_1a_2a_0$; $t_3 = |t_1 - 1|$;

The final reconstruction polynomial is

$$p^r_{i+0.5} = \sum_{j=0}^{2} \alpha_j^p p^j_{i+0.5} \tag{6.16}$$

FV-WENO-OA-p scheme (Point-based WENO scheme)

Though the finite volume weak form formulation is based on cell-averaging [86], it is possible to obtain results without averaging. Point-based interpolation leads to higher-order accuracy in the interpolation and lower-order of accuracy in the derivative calculation. Since the accuracy of the solution depends on the order of accuracy of the input of the Riemann solver, point-based interpolation could be an alternative to cell-average interpolation. A brief study on the effect of averaging and point-based reconstruction on some of the numerical test cases is presented in subsection 6.6.3.

Interpolation polynomials are [155]

Table 6.10
Ideal weights and desired interpolation formulation for FV-WENO-OA-p to calculate right state of $u_{i+0.5}$

$\mathbf{r^0_{i+0.5}}$	$\mathbf{r^1_{i+0.5}}$	$\mathbf{r^2_{i+0.5}}$	**Desired final $\mathbf{u_{i+0.5}}$**	$\alpha^{\mathbf{p}}_{\mathbf{0}}$	$\alpha^{\mathbf{p}}_{\mathbf{1}}$	$\alpha^{\mathbf{p}}_{\mathbf{2}}$
good	good	good	$\frac{45u_{i+1}}{64} + \frac{15u_i}{32} - \frac{5u_{i-1}}{128} - \frac{u_{i+2}}{32} + \frac{3u_{i+3}}{128}$	$\frac{5}{16}$	$\frac{5}{8}$	$\frac{1}{16}$
bad	good	good	$\frac{15u_{i+1}}{16} + \frac{5u_i}{16} - \frac{5u_{i+2}}{16} + \frac{u_{i+3}}{16}$	0	$\frac{5}{6}$	$\frac{1}{6}$
good	good	bad	$\frac{9u_{i+1}}{16} + \frac{9u_i}{16} - \frac{u_{i-1}}{16} - \frac{u_{i+2}}{16}$	$\frac{1}{2}$	$\frac{1}{2}$	0

$$r^0_{i+0.5} = \frac{1}{8}(-u_{i-1}+6u_i+3u_{i+1}); \quad r^1_{i+0.5} = \frac{1}{8}(3u_i+6u_{i+1}-u_{i+2}); \quad r^2_{i+0.5} = \frac{1}{8}(15u_{i+1}-10u_{i+2}+3u_{i+3})$$

and the ideal weights are

$\gamma_0 = \frac{5}{16}$, $\gamma_1 = \frac{5}{8}$ and $\gamma_2 = \frac{1}{16}$

$$\tilde{\omega}_j = \frac{\gamma_j}{(\epsilon+\beta)^2}, \quad \omega_j = \frac{\tilde{\omega}_j}{\sum_{j=0}^{2}\tilde{\omega}_j} \tag{6.17}$$

$$a_i = \frac{\text{floor}\left(\frac{\gamma_i}{\omega_i c_0}\right)}{\text{floor}\left(\frac{\gamma_i}{\omega_i c_0}\right)+\epsilon}, \quad i = 0,1,2 \tag{6.18}$$

Table 6.10 shows the ideal formulation and weights for the right state reconstruction of $u_{i+0.5}$ using FV-WENO-OA-p. The final weights are formulated to match with the α^p_i in Table 6.10.

$$\alpha^p_0 = t_1\left(a_0\frac{1}{2} - t_2\frac{3}{16}\right) + t_3\frac{a_0}{s_a}, \tag{6.19a}$$

$$\alpha^p_1 = t_1\left(a_0a_1\frac{1}{2} + a_1a_2\frac{5}{6} - t_2\frac{17}{24}\right) + t_3\frac{a_1}{s_a}, \tag{6.19b}$$

$$\alpha^p_2 = t_1\left(\frac{1}{6}a_2 - t_2\frac{5}{48}\right) + t_3\frac{a_2}{s_a}, \tag{6.19c}$$

Where $t_1 = a_1a_0+a_2a_1$; $t_1 = \frac{t_1}{(t_1+10^{-18})}$; $s_a = a_1+a_2+a_0$; $t_2 = a_1a_2a_0$; $t_3 = |t_1 - 1|$;

The final reconstruction step is

$$r^l_{i+0.5} = \sum_{j=0}^{2} \alpha^p_j r^j_{i+0.5}$$

6.3.4 Order of accuracy of WENO-OA scheme

Similar to WENO scheme, WENO-OA scheme can give a fifth-order accurate derivative scheme when the solution is smooth and it reduced to third-order when only one polynomial is smooth. The WENO-JS, WENO-Z, TENO and TENO-opt procedures leads to lower-order accuracy when one of the polynomials is not smooth. In some cases, WENO-EZ and WENO-EJS schemes also end up in lower-order accuracy (Table 6.6). Here, the effects of those weights on WENO-OA scheme are presented. For the test case in table 6.1, FD-WENO-OA scheme leads to the following formulation,

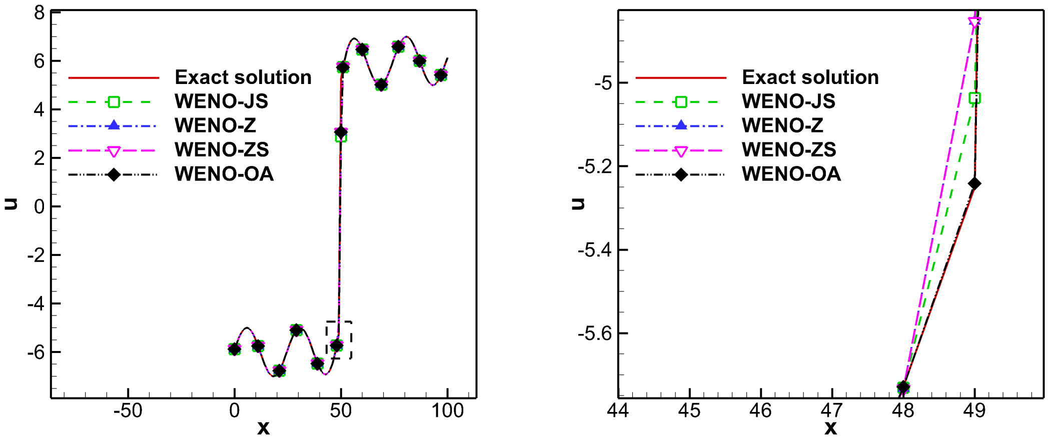
$$u_{i+0.5} = \frac{1}{4}\bar{u}_{i+1} + \frac{13}{12}\bar{u}_i - \frac{5}{12}\bar{u}_{i-1} + \frac{1}{12}\bar{u}_{i-2} \qquad \mathcal{O}(\Delta x^2) \tag{6.20}$$

Eq. 6.20 leads to fourth-order accuracy in the derivative calculation. Similarly, WENO-OA-p scheme leads to the formulation

$$u_{i+0.5} = \frac{5}{16}u_{i+1} + \frac{15}{16}u_i - \frac{5}{16}u_{i-1} + \frac{1}{16}u_{i-2} \qquad \mathcal{O}(\Delta x^4) \tag{6.21}$$

Eq. 6.21 gives a fourth-order accurate formulation of $u_{i+0.5}$ in the interpolation. Note that WENO-OA formulation can give the exact polynomial required for WENO but other WENO procedures may give weighting coefficients closer to WENO-OA when one of the polynomials is not smooth but they may not exactly satisfy the Taylor series. Figure 6.1 shows the result of interpolation on the oscillating step function using different WENO schemes using 101 grid points. From Figure 6.1b, it is clear that interpolation carried out by WENO-OA-p is better than WENO-ZS-p, WENO-JS-p and WENO-Z-p. Since the interpolation formulation is not provided for WENO-E and TENO family of WENO schemes, we have compared those WENO on the derivative calculation of a differentiable step-like function. Test function considered is:

$$y(x) = \tanh(50x) + \sin(x) \tag{6.22}$$



(a) Interpolation on oscillating step function

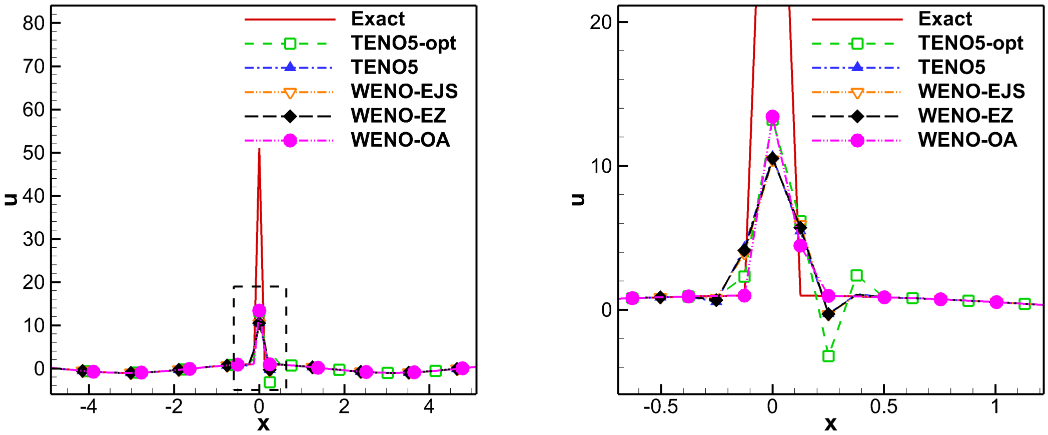
(b) Zoomed view of discontinuity

Figure 6.1: Interpolation WENO schemes on oscillating step function-1 using 101 grid points

and its derivative is $\frac{\mathrm{d}y}{\mathrm{d}x} = \cos(x) - 50(\tanh(50x)^2 - 1)$. The derivative approximation of eq. 6.22 is shown in figure 6.2. From the figure 6.2, it is clear that WENO-E and TENO are producing oscillations but WENO-OA is not producing any oscillations. In that case, the result of TENO and WENO-E are comparable but the result of TENO-opt is relatively more oscillatory compared to others. This oscillation could be removed if TENO and WENO-E had a tuning parameter to control its order adaptive nature. Though tuning parameter of TENO family is derived from approximate dispersion relation preserving property, it is unable to remove oscillation because that analysis is carried by assuming the temporal discretization is exact or spectral. In most of the computations involving practical applications use finite single/multi-step methods. This could be the reason why approximate dispersion relation preserving optimization may not always an optimal one.

6.4 Framing customized weighting function

Because WENO-OA weights are different from the classical WENO schemes, framing these weights can be carried through different ways. This weights cannot be framed using straight forward usage of linear algebra because this system does not have any solution in the linear space because the weight equation is non-linear. We can split the non-linear



(a) derivative of oscillating step function

(b) Zoomed view of discontinuity

Figure 6.2: Derivative approximation of cell-averaged WENO schemes on oscillating step function-2 using $n = 101$.

components into a set of linear basis with some switching functions. Weights presented in the previous sections are relatively more compact than weights formed using linear algebra.

6.4.1 Formulating weighting function using linear algebra

Once we assigned "one" to smooth polynomial and "zero" to non-smooth polynomial using eq. 6.8, we should find the mapping function. We can write weights and possible combinations of "good" and "bad" polynomials in matrix form. Here, we did not consider all "bad" because that is hypothetical.

$$A_f^T = \begin{pmatrix} 1 & 0 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{pmatrix} \tag{6.23}$$

There are $2^3 - 1 = 7$ (excluding all zero weights) possibilities and corresponding weights of ideal polynomials are

$$\alpha_{f0}^T = \begin{pmatrix} \frac{1}{16} & 0 & \frac{1}{6} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & 1 \end{pmatrix} \tag{6.24}$$

When A_f matrix is multiplied with some B_{f0} matrix we should get α_{f0}.

$$A_f B_{f0} = \alpha_{f0} \tag{6.25}$$

Our objective is to find the B_{f0} matrix. Because A_f matrix is not a square matrix, we could find the system solution using least-square solution approach or finding the left inverse of A_f matrix and solving it for B_{f0}. Unfortunately, none of them can give the required weights because the system is ill-posed. If we break the system into two or more sub-matrix where we can find the linear solution, the problem can be simplified. Because we cannot achieve higher-order accuracy in the derivative calculation when there is a critical point at the centre or only when one polynomial is "good", we can skip those system equation. So the system matrix A_f can be written into two set of matrix one can achieve higher-order that is matrix A_0 another one A_{f1} Rewriting the system equation in two

$$A_0 = \begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix} \qquad A_{f1} = \begin{pmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{6.26}$$

so that $A_f = \begin{pmatrix} A_0 \\ \\ A_{f1} \end{pmatrix}$ and $\alpha_f = \begin{pmatrix} \alpha_0 \\ \\ \alpha_{f1} \end{pmatrix}$. where, $\alpha_0 = \begin{pmatrix} \frac{1}{16} \\ \\ 0 \\ \\ \frac{1}{6} \end{pmatrix}$. Similarly other weights can be obtained from the table 6.8 and can be written in matrix form.

$$\alpha = \begin{pmatrix} | & | & | \\ \\ \alpha_0 & \alpha_1 & \alpha_2 \\ \\ | & | & | \end{pmatrix} = \begin{pmatrix} \frac{1}{16} & \frac{5}{8} & \frac{5}{16} \\ \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \\ \frac{1}{6} & \frac{5}{6} & 0 \end{pmatrix}; \qquad B_0 = A_0^{-1}\alpha_0 = \begin{pmatrix} \frac{1}{16} \\ \\ \frac{5}{48} \\ \\ -\frac{5}{48} \end{pmatrix} \tag{6.27}$$

Similarly, we can find $B_1 = A_0^{-1}\alpha_1$ and $B_2 = A_0^{-1}\alpha_2$.

$$B = \begin{pmatrix} | & | & | \\ \\ B_0 & B_1 & B_2 \\ \\ | & | & | \end{pmatrix} = \begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{-3}{16} \\ \\ \frac{5}{48} & \frac{17}{24} & \frac{3}{16} \\ \\ -\frac{5}{48} & -\frac{5}{24} & \frac{5}{16} \end{pmatrix} \tag{6.28}$$

The accuracy of B can be tested by multiplying A_0 with B so that we get α

$$\alpha = \begin{pmatrix} 1 & 1 & 1 \\ \\ 0 & 1 & 1 \\ \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} | & | & | \\ \\ B_0 & B_1 & B_2 \\ \\ | & | & | \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ \\ 0 & 1 & 1 \\ \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} \frac{1}{16} & \frac{1}{8} & \frac{-3}{16} \\ \\ \frac{5}{48} & \frac{17}{24} & \frac{3}{16} \\ \\ -\frac{5}{48} & -\frac{5}{24} & \frac{5}{16} \end{pmatrix} = \begin{pmatrix} \frac{1}{16} & \frac{5}{8} & \frac{5}{16} \\ \\ 0 & \frac{1}{2} & \frac{1}{2} \\ \\ \frac{1}{6} & \frac{5}{6} & 0 \end{pmatrix} \tag{6.29}$$

Weights are (from B matrix):

$$\alpha_0 = \frac{1}{16}a_0 + \frac{5}{48}a_1 - \frac{5}{48}a_2, \tag{6.30a}$$

$$\alpha_1 = \frac{1}{8}a_0 + \frac{17}{24}a_1 - \frac{5}{24}a_2, \tag{6.30b}$$

$$\alpha_2 = -\frac{3}{16}a_0 + \frac{3}{16}a_1 + \frac{5}{16}a_2. \tag{6.30c}$$

For other possibilities of the “good” and “bad”, we can use the following function that is giving equal weights for all the “good” polynomials. The formulation is:

$$\alpha_0^n = \frac{a_0}{s_a}; \quad \alpha_1^n = \frac{a_1}{s_a}; \quad \alpha_2^n = \frac{a_2}{s_a} \tag{6.31}$$

$s_a = a_0 + a_1 + a_2$

We should club both the terms using a non-linear switch, so that eq. 6.30 is retained when two consecutive polynomials are “good”, else eq. (6.31) is retained. The simple formulation is:

$$t_1 = a_0 a_1 + a_1 a_2; \qquad t_1 = \frac{t_1}{t_1 + \epsilon} \tag{6.32}$$

where, ϵ is a small number added to avoid unphysical number. Eq. 6.31 should be *on* when eq. 6.30 is *off* and vice versa, so switching statement is complement of t_1 that is

$$t_3 = |t_1 - 1| \tag{6.33}$$

Combining both, the final formulation is

$$\alpha_0 = t_1\left(\frac{1}{16}a_0 + \frac{5}{48}a_1 - \frac{5}{48}a_2\right) + t_3\frac{a_0}{s_a}, \tag{6.34a}$$

$$\alpha_1 = t_1\left(\frac{1}{8}a_0 + \frac{17}{24}a_1 - \frac{5}{24}a_2\right) + t_3\frac{a_1}{s_a}, \tag{6.34b}$$

$$\alpha_2 = t_1\left(-\frac{3}{16}a_0 + \frac{3}{16}a_1 + \frac{5}{16}a_2\right) + t_3\frac{a_2}{s_a}. \tag{6.34c}$$

6.4.2 Framing WENO-OA-467

WENO-OA-467 can go upto seventh-order, minimum order is four and it also has an order adaptiveness of six. All possible cases of WENO-OA-467 is

$$A^T = \begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{6.35}$$

It is almost impossible to solve the whole system in linear space so we shall split the matrix into two. Because we need sixth-order accuracy, only two cases satisfy that. The A_0 matrix is:

$$A_0 = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \tag{6.36}$$

and

$$\alpha = \begin{pmatrix} | & | & | & | \\ \alpha_0 & \alpha_1 & \alpha_2 & \alpha_3 \\ | & | & | & | \end{pmatrix} = \begin{pmatrix} \frac{1}{35} & \frac{12}{35} & \frac{18}{35} & \frac{4}{35} \\ \frac{1}{15} & \frac{8}{15} & \frac{2}{5} & 0 \\ 0 & \frac{1}{5} & \frac{3}{5} & \frac{1}{5} \end{pmatrix} \tag{6.37}$$

$B_0 = A_0^{-1}\alpha_0 = \begin{pmatrix} \frac{1}{35} \\ \frac{4}{105} \\ 0 \\ -\frac{4}{105} \end{pmatrix}$ where, A_0^{-1} is left inverse of A_0. Here, we used left inverse because we cannot find direct inverse of A_0. Similarly, we can find $B_1 = A_0^{-1}\alpha_1$, $B_2 = A_0^{-1}\alpha_2$ and $B_3 = A_0^{-1}\alpha_3$.

Solving this we get

$$B = \begin{pmatrix} \frac{1}{35} & \frac{1}{7} & \frac{-3}{35} & \frac{-3}{35} \\ \frac{4}{105} & \frac{41}{105} & \frac{17}{35} & \frac{3}{35} \\ 0 & 0 & 0 & 0 \\ -\frac{4}{105} & -\frac{4}{21} & \frac{4}{35} & \frac{4}{35} \end{pmatrix}$$

The accuracy of B can be checked by multiplying A_0 with B so that we get the ideal weights (α) that is given in eq. 6.38.

$$\alpha = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} | & | & | & | \\ B_0 & B_1 & B_2 & B_3 \\ | & | & | & | \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} \frac{1}{35} & \frac{1}{7} & \frac{-3}{35} & \frac{-3}{35} \\ \frac{4}{105} & \frac{41}{105} & \frac{17}{35} & \frac{3}{35} \\ 0 & 0 & 0 & 0 \\ -\frac{4}{105} & -\frac{4}{21} & \frac{4}{35} & \frac{4}{35} \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{35} & \frac{12}{35} & \frac{18}{35} & \frac{4}{35} \\ \frac{1}{15} & \frac{8}{15} & \frac{2}{5} & 0 \\ 0 & \frac{1}{5} & \frac{3}{5} & \frac{1}{5} \end{pmatrix} \tag{6.38}$$

Using B, the weight equations are framed that is:

$$\alpha_0 = \left(\frac{1}{35}a_0 + \frac{4}{105}a_1 - \frac{4}{105}a_3\right), \tag{6.39a}$$

$$\alpha_1 = \left(\frac{1}{7}a_0 + \frac{41}{105}a_1 - \frac{4}{21}a_3\right), \tag{6.39b}$$

$$\alpha_2 = \left(-\frac{3}{35}a_0 + \frac{17}{35}a_1 + \frac{4}{35}a_3\right), \tag{6.39c}$$

$$\alpha_3 = \left(-\frac{3}{35}a_0 + \frac{3}{35}a_1 + \frac{4}{35}a_3\right). \tag{6.39d}$$

We can give equal weights to "good" polynomials in other equations so fourth-order accuracy will be achieved by other possibilities of "good" and "bad" polynomials. The weight equation for other possible combinations are:

$$\alpha_0^n = \frac{a_0}{s_a}; \qquad \alpha_1^n = \frac{a_1}{s_a}; \qquad \alpha_2^n = \frac{a_2}{s_a}; \qquad \alpha_3^n = \frac{a_3}{s_a} \tag{6.40}$$

We should find a switch that connects both set of equations in eq. 6.39 and eq. 6.40.

Non-linear switch to check three consecutive "good" is:

$$t_1 = a_0 a_1 a_2 + a_1 a_2 a_3; \qquad t_1 = \frac{t_1}{t_1 + \epsilon} \tag{6.41}$$

To switch "on" other possibilities when this condition is not satisfied, the formulation is

$$t_3 = |t_1 - 1| \tag{6.42}$$

The final formulation is:

$$\alpha_0 = t_1 \left(\frac{1}{35} a_0 + \frac{4}{105} a_1 - \frac{4}{105} a_3 \right) + t_3 \frac{a_0}{s_a}, \tag{6.43a}$$

$$\alpha_1 = t_1 \left(\frac{1}{7} a_0 + \frac{41}{105} a_1 - \frac{4}{21} a_3 \right) + t_3 \frac{a_1}{s_a}, \tag{6.43b}$$

$$\alpha_2 = t_1 \left(-\frac{3}{35} a_0 + \frac{17}{35} a_1 + \frac{4}{35} a_3 \right) + t_3 \frac{a_2}{s_a}, \tag{6.43c}$$

$$\alpha_3 = t_1 \left(-\frac{3}{35} a_0 + \frac{3}{35} a_1 + \frac{4}{35} a_3 \right) + t_3 \frac{a_3}{s_a}. \tag{6.43d}$$

Where, $s_a = a_0 + a_1 + a_2 + a_3$.

6.5 Convergence and error study

Different WENO schemes are tested for convergence and error study on the function $f(x) = x^3 + sin(x)$. This function has critical points at $f'(0) = 0$ and $f'''(0) = 6$ [89]. The derivatives are calculated over the domain 0 to 2π and compared with the analytical solution. Table 6.11, shows the error and the convergence of different schemes. In this test, WENO-OA has shown a better convergence and minimum RMS error when compared with other schemes considered in this work. The schemes are also tested on $f(x) = \sin(\pi x - \sin(\pi x)/\pi)$. The results are shown in table 6.12. WENO-OA scheme shows a better convergence and lesser error compared to other schemes considered.

6.6 Numerical test cases

The present schemes are tested on the Euler equation and compared with other WENO schemes. The present schemes show a better result in most of the one-dimensional and two-dimensional test cases. No explicit dissipation is added to any of the numerical schemes. Most of the WENO schemes are able to produce a reasonably good result without adding explicit dissipation term. The diffusion added by the standard Riemann solver and inbuilt diffusion of the schemes are the only source of numerical diffusion in the test cases. We did not consider any test case where the addition of explicit artificial viscosity is required.

6.6.1 One-dimensional test case

Different WENO schemes are solved on the standard one-dimensional test case and the result is compared with the analytical result. WENO-OA outperformed other schemes, in term of resolution and accuracy. Different norms to calculate error is presented here, Root mean square (RMS) error at a particular time is defined as:

$$\phi_{rms} = \frac{\sqrt{\sum_{i=1}^{n}(\phi_{nu}(i,t=t_1) - \phi_{ex}(i,t=t_1))^2}}{n}$$

L_2 error at a particular time is defined as:

$$\phi_m = \frac{\sum_{i=1}^{n} |\phi_{nu}(i,t=t_1) - \phi_{ex}(i,t=t_1)|}{n}$$

Where ϕ is the variable in which error is calculated. ϕ_{ex} is the exact solution, ϕ_{nu} is the numerical solution, n is the number of grid points and t_1 is time at which errors are evaluated.

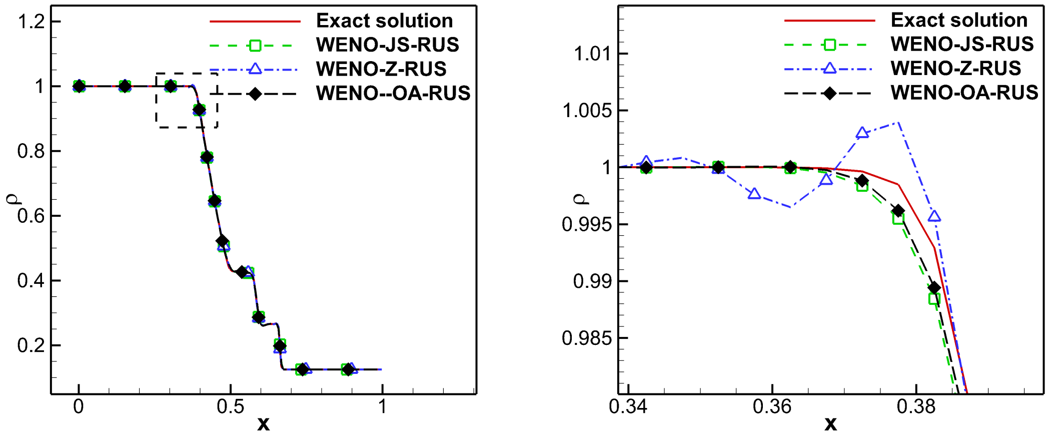
Table 6.11

Convergence and error study on the spatial derivative of $f(x) = x^3 + \sin(x)$

Schemes	**n**	**RMS error**	**Convergence**	**Mean convergence**
WENO-JS	51	2.49E-03		
	101	2.38E-04	3.43E+00	
	201	5.32E-06	5.52E+00	5.005
	401	8.11E-08	6.06E+00	
WENO-Z	51	8.93E-04		
	101	5.22E-05	4.1566	
	201	1.47E-06	5.1898	4.5013
	401	8.30E-08	4.1576	
WENO-EZ	51	2.87E-03		
	101	2.16E-04	3.783	
	201	3.76E-06	5.8852	4.6898
	401	1.80E-07	4.4013	
TENO	51	5.63E-04		
	101	1.10E-08	15.8631	
	201	3.53E-10	5.0032	8.6188
	401	1.12E-11	4.9901	
TENO-opt	51	1.37E-04		
	101	1.82E-05	2.9612	
	201	2.32E-06	2.9904	2.9823
	401	2.93E-07	2.9954	
WENO-OA	51	3.83E-03		
	101	1.74E-06	11.2653	
	201	3.53E-10	12.3537	9.5363
	401	1.12E-11	4.9901	

Table 6.12
Convergence and error study on the spatial derivative of $f(x) = sin(\pi x - sin(\pi x)/\pi)$

Schemes	n	RMS error	Convergence	Mean convergence
WENO-JS	51	2.26E-02		
	101	1.12E-03	4.39E+00	
	201	5.64E-05	4.35E+00	4.2916
	401	3.24E-06	4.13E+00	
WENO-Z	51	5.39E-03		
	101	2.08E-04	4.7609	
	201	6.90E-06	4.9533	4.8645
	401	2.37E-07	4.8793	
WENO-EZ	51	6.81E-03		
	101	2.87E-04	4.6338	
	201	9.22E-06	4.9971	4.7956
	401	3.45E-07	4.7557	
TENO	51	4.47E-03		
	101	1.51E-04	4.96E+00	
	201	4.77E-06	5.02E+00	5
	401	1.49E-07	5.02E+00	
TENO-opt	51	4.36E-02		
	101	5.40E-03	3.0572	
	201	6.66E-04	3.0419	3.0403
	401	8.26E-05	3.0218	
WENO-OA	51	5.18E-03		
	101	1.51E-04	5.1704	
	201	4.77E-06	5.0228	5.0717
	401	1.49E-07	5.0217	



(a) Density plot using different FV-WENO schemes using Rusnov flux

(b) Density plot using different FV-WENO schemes using Rusnov flux (zoomed view)

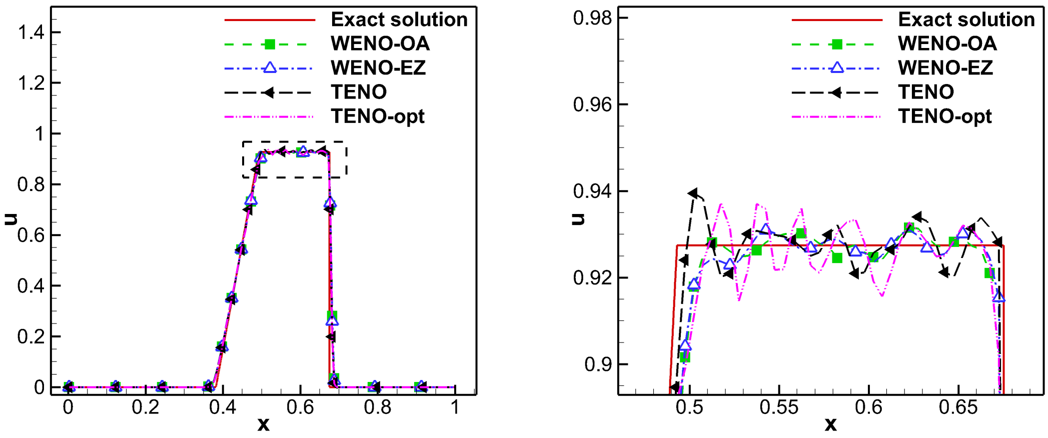
Figure 6.3: Solution of Sod shock tube problem using finite volume schemes with $CFL = 0.9$ at 0.1 s using $n = 200$.

Sod shock tube problem

Simulation is carried out using $n = 200$, $CFL = 0.9$ with a flow time of $T = 0.1$ s. CFL number is calculated from $\frac{|u+a|\,\Delta t}{\Delta x}$, where a is speed of sound and u is velocity of the flow. Riemann solver used is Rusanovs scheme with finite volume discretization and Lax-Friedrichs scheme for finite difference method. For time integration, HRK method [105] and SSPRK3 [64] are used. Solution of this problem using different schemes are shown in figure 6.3. Initial condition for this problem is:

$$(\rho, u, p) = \begin{cases} (1, 0, 1) & x \leq 0.5 \\ (0.125, 0, 0.1) & x > 0.5 \end{cases}$$

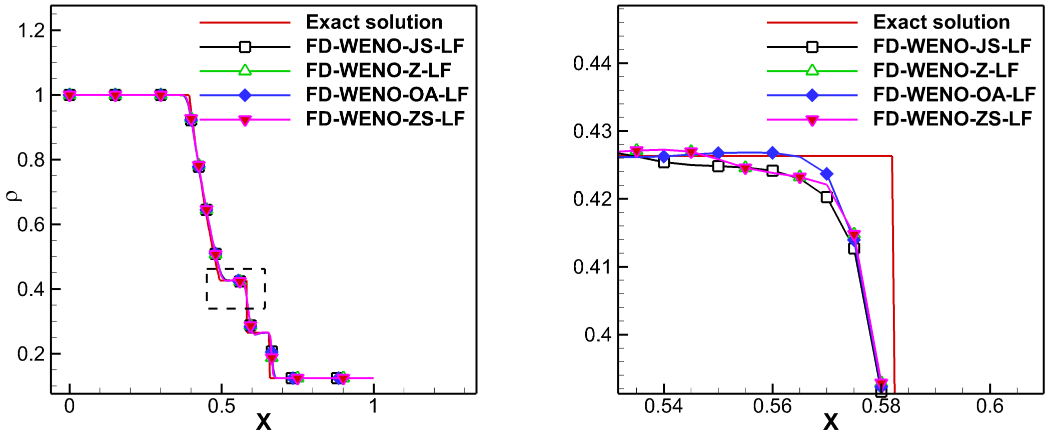
From figure 6.3 and figure 6.4, it is clear that FV-WENO-OA has given a better result than others. FV-WENO-Z has better shock resolving property but it is shows some oscillations in finite-volume discretization and it does not show any oscillations in finite-difference discretization. The result of WENO-OA and WENO-EZ are more or less similar. Figure 6.5b shows the density plot of Sod shock tube problem using finite difference scheme. FD-WENO-OA has a better resolution at contact-discontinuity than others.



(a) Velocity plot using different FV-WENO schemes using Rusnov flux

(b) Velocity plot using different FV-WENO schemes using Rusnov flux (zoomed view)

Figure 6.4: Solution of Sod shock tube problem using finite volume schemes with $CFL = 0.9$ at 0.1 s using $n = 200$.



(a) Density plot using different FD-WENO schemes using Lax-Friedrichs scheme

(b) Density plot using different FD-WENO schemes using Lax-Friedrichs scheme (zoomed view)

Figure 6.5: Solution of Sod shock tube problem using finite difference schemes with $CFL = 0.9$ at 0.1 s using $n = 200$.

WENO-Z and WENO-ZS did not produce any significant difference in the result. This may be because, the switching function may be not appropriate for this problem. Root mean square error and Manhattan norm of error of Sod shock tube problem is tabulated in table 6.13 and table 6.14 respectively. WENO-OA has shown lesser error compared to other schemes for this problem when $c_o = \frac{1}{90}$ is used.

Table 6.13
Root mean square error in Sod shock tube problem at $T = 0.1$ s with $CFL = 0.8$ using 200 grid points

Schemes	$\rho_{\mathbf{rms}}$	$\mathbf{u_{rms}}$	$\mathbf{p_{rms}}$
WENO-OA	0.00988088	0.048314105	0.010747067
WENO-Z	0.010976269	0.052433822	0.012003003
WENO-EZ	0.011369047	0.055959165	0.012908202
WENO-JS	0.012276472	0.058729794	0.013787214
TENO	0.010584637	0.049817081	0.011537524
TENO-opt	0.009696902	0.050672151	0.01176616

Right Expansion and left strong shock (RELSS)

Numerical test is carried on $n = 200$, $CFL = 0.7$ with the flow time of $T = 0.1$ s. The simulation result of the finite volume scheme is shown in figure 6.6. Here also the present scheme outperformed others. In this test case, FD-WENO-Z and WENO-EZ outperformed other schemes when $c_0 = \frac{3}{10}$ is used for WENO-OA. WENO-OA outperformed other scheme when $c_0 = \frac{1}{45}$ is used. Initial condition used is:

$$(\rho, u, p) = \begin{cases} (1, 0, 7) & x \leq 0.5 \\ \\ (1, 0, 10) & x > 0.5 \end{cases}$$

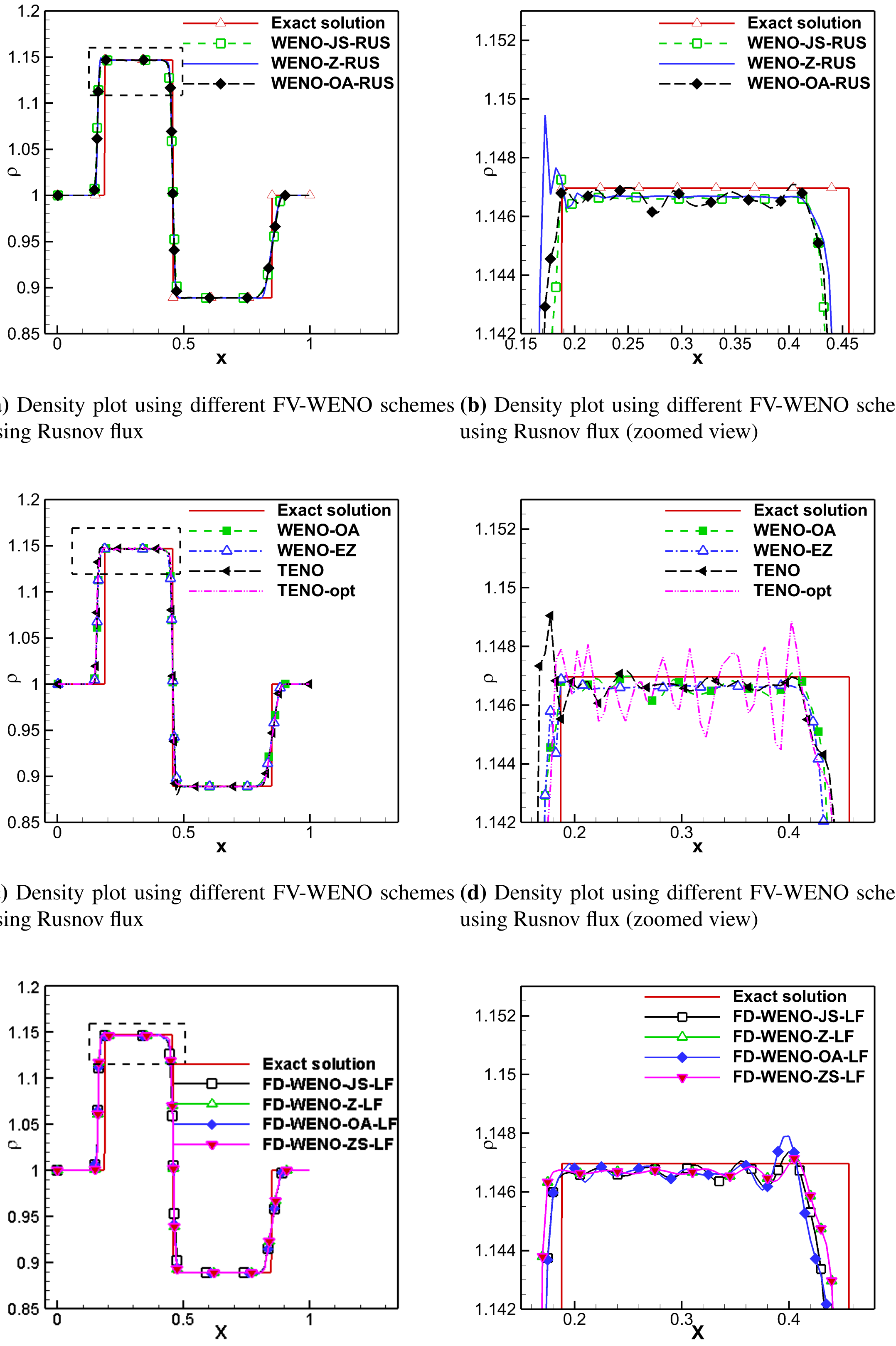
Root mean square error of this problem for different schemes is shown in table 6.15. WENO-OA shows lesser error compared to other schemes with $c_o = \frac{1}{45}$.

Table 6.14
Manhattan norm of error in Sod shock tube problem at $T = 0.1$ s with $CFL = 0.8$ using 200 grid points

Schemes	$\rho_{\mathbf{m}}$	$\mathbf{u_m}$	$\mathbf{p_m}$
WENO-OA	0.002759398	0.006285947	0.002418984
WENO-Z	0.002696686	0.006710168	0.002553439
WENO-EZ	0.00285168	0.007655313	0.002860007
WENO-JS	0.003225898	0.00826728	0.003121195
TENO	0.002801088	0.00678148	0.002534338
TENO-opt	0.002864696	0.006781949	0.002553073

Table 6.15
Root mean square error in Right Expansion and left strong shock test case at $T = 0.1$ s with $CFL = 0.8$ using 200 grid points

Schemes	$\rho_{\mathbf{rms}}$	$\mathbf{u_{rms}}$	$\mathbf{p_{rms}}$
WENO-OA	0.025695907	0.075162366	0.256409756
WENO-Z	0.02698291	0.07693153	0.262184325
WENO-EZ	0.026876803	0.076216971	0.259344129
WENO-JS	0.027323114	0.076569512	0.260526666
TENO	0.026703638	0.076467775	0.260307523
TENO-opt	0.025719869	0.075745195	0.257286225



(a) Density plot using different FV-WENO schemes using Rusnov flux

(b) Density plot using different FV-WENO schemes using Rusnov flux (zoomed view)

(c) Density plot using different FV-WENO schemes using Rusnov flux

(d) Density plot using different FV-WENO schemes using Rusnov flux (zoomed view)

(e) Density plot using different FD-WENO schemes using Lax-Friedrichs scheme

(f) Density plot using different FV-WENO schemes using Lax-Friedrichs scheme (zoomed view)

Figure 6.6: Solution of right expansion and left strong shock problem using finite volume and finite difference method with $CFL = 0.7$ at 0.1 s using $n = 200$.

Shu-Osher problem

Higher-order schemes work well on problems with curvatures so the present schemes are tested on the Shu-Osher problem. Initial condition used for the Shu-Osher problem is [7]

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.33333) & x \leq -4 \\ \\ (1 + 0.2 \times sin(5x), 0, 1) & x > -4 \end{cases}$$

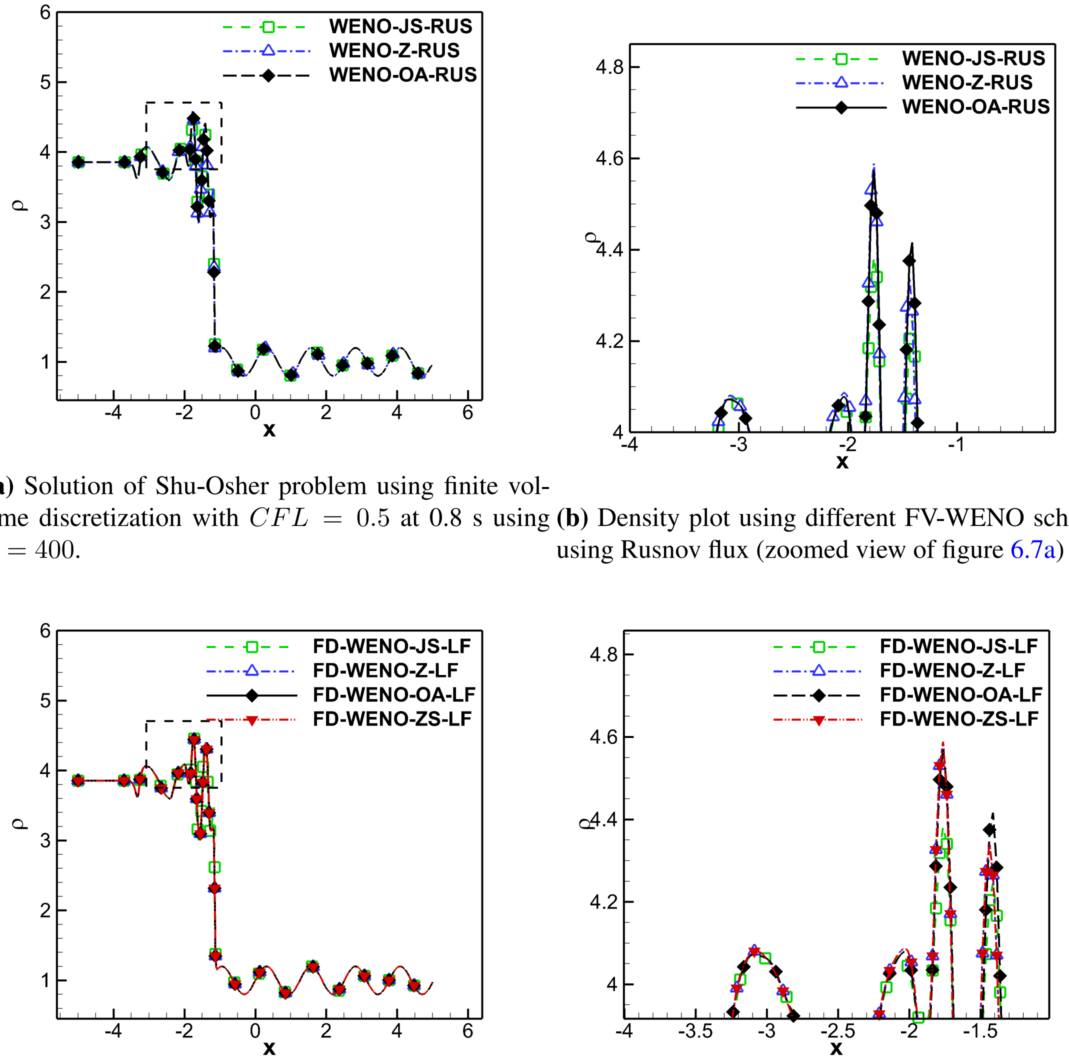
Simulation is carried on $n = 400$ with a $CFL = 0.5$ and solved up to $T = 0.8s$. The result of this test case using FVM is shown in figure 6.7. For this test case, WENO-OA is less-oscillatory and less dissipative compared to other schemes considered. Figure 6.7a shows the solution of Shu-Osher problem using finite difference discretization. Here also FD-WENO-OA outperformed others.

Noh test case

This problem is an implosion problem [198]. This test case is used to check the robustness of the scheme to preserve spherical symmetry. For this problem, the solution is an infinite-strength shock wave expands outward from the origin with a constant velocity of $\frac{1}{3}$. Figure 6.8 shows the solution of Noh test case using 200 grid points with CFL number 0.8 at flow time 1 s. For this test case, except TENO and TENO-opt, other schemes produced more or less the same result but all the scheme exactly identified the location of shock. Since the performance of Roe-solver is better than others, the Roe scheme is used as the Riemann solver. RMS error of different schemes for this problem is tabulated in table 6.16. WENO-OA has given the least error than the other schemes when $c_o = \frac{1}{40}$ is used.

Initial condition used is:

$$(\rho, u, p) = \begin{cases} (1, 1, 10^{-6}) & x \leq 0.5 \\ \\ (1, -1, 10^{-6}) & x > 0.5 \end{cases}$$



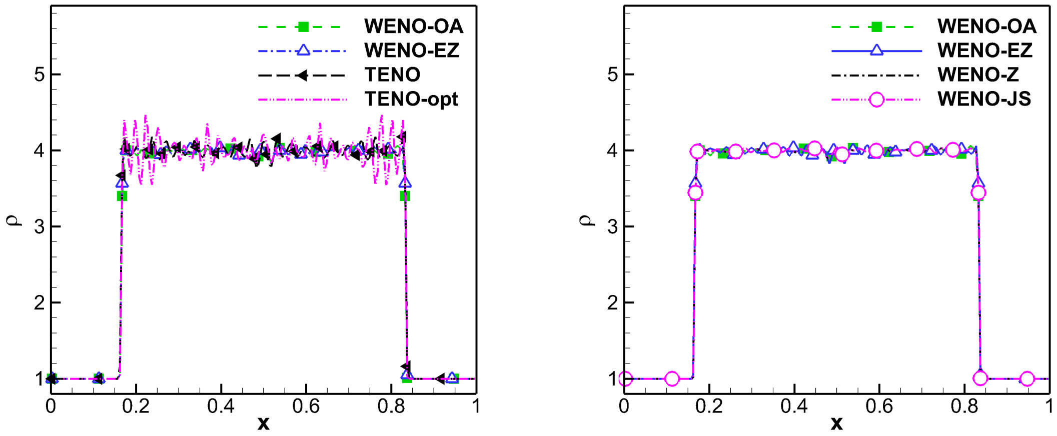
(a) Solution of Shu-Osher problem using finite volume discretization with $CFL = 0.5$ at 0.8 s using $n = 400$.

(b) Density plot using different FV-WENO schemes using Rusnov flux (zoomed view of figure 6.7a)

(c) Density plot using different FD-WENO schemes using Lax-Friedrichs scheme

(d) Density plot using different FD-WENO schemes using Lax-Friedrichs scheme (zoomed view)

Figure 6.7: Solution of Shu-Osher problem using finite difference and finite volume discretization with $CFL = 0.5$ at 0.8 s using $n = 400$.



(a) Density plot of Noh test

(b) Density plot of Noh test case

Figure 6.8: Density plot of Noh test case at 1 s with $CFL = 0.8$ using $n = 200$

Table 6.16
Root mean square error in Noh test case at $T = 1$ s with $CFL = 0.8$ using 200 grid points

Schemes	$\rho_{\mathbf{rms}}$	$\mathbf{u_{rms}}$	$\mathbf{p_{rms}}$
WENO-OA	0.040015493	0.007789326	0.019837074
WENO-Z	0.043634591	0.008384231	0.022471241
WENO-EZ	0.050953473	0.016127981	0.021547521
WENO-JS	0.050554665	0.026901231	0.020175966
TENO	0.07262997	0.009513642	0.030441726
TENO-opt	0.106553153	0.041361497	0.053483004

6.6.2 Comparison of FD-WENO-OA and FV-WENO-OA with different Riemann solvers

The effect of the Riemann solver on the cell-average and point-based reconstruction of WENO-OA is studied. The FV discretization of 1-D conservation equation is

$$\frac{\mathrm{d}\bar{u}_i}{\mathrm{d}t} = -\frac{1}{\Delta x}\left[f(u_{i+0.5}) - f(u_{i-0.5})\right]$$

Where $\bar{u}_i$ is cell-average value. The finite volume WENO-OA needs left and right state, that can be calculated from eq. 6.10 and eq. 6.16. Once the left and the right state are determined, that is passed to Riemann solver. The Riemann solver will give the resultant flux at the interface so the flux can be written as $f_{i+0.5}(u^l_{i+0.5}, u^r_{i+0.5})$. Lax-Friedrichs is used as Riemann solver and it is

$$f^{LF}(u^l_{i+0.5}, u^r_{i+0.5}) = \frac{1}{2}\left[f(u^r_{i+0.5}) + f(u^l_{i-0.5}) - \alpha(u^r_{i+0.5} - u^l_{i+0.5})\right]$$

The finite difference formulation can be applied through flux-splitting procedure so the final flux is given by

$$f(u) = f^+(u) + f^-(u)$$

$$\frac{\mathrm{d}f^+(u)}{\mathrm{d}u} \geq 0 \qquad \frac{\mathrm{d}f^-(u)}{\mathrm{d}u} \leq 0$$

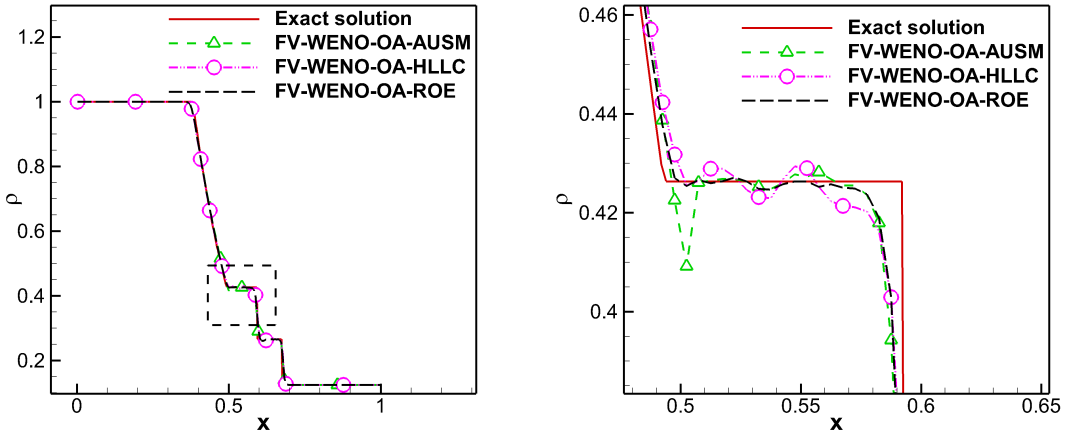
Then the Lax-Friedrich splitting is

$$f^{\pm} = \frac{1}{2}\left[f(u) \pm \alpha u\right]$$

$$\alpha = \max_{\lambda} |f'(u)|$$

We refer [11] for more details about the difference between FV-WENO and FD-WENO scheme.

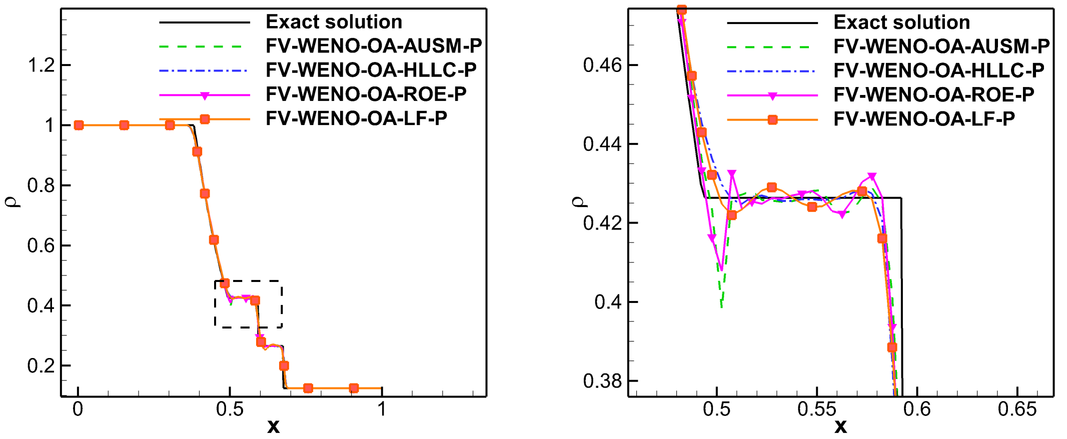
Figure 6.9 shows the comparison of different Riemann solver using FV-WENO-OA scheme. Riemann solver used here are HLLC, AUSM and Roe scheme. Among all the solvers considered, Roe solver has given a better result than others for cell-averaged



(a) Comparison of different Riemann solver with FV-WENO-OA scheme

(b) Comparison of different Riemann solver with FV-WENO-OA scheme (zoomed view)

Figure 6.9: Density plot of Sod shock tube problem for different Riemann solvers using cell-averaged reconstruction with $CFL = 0.9$ at 0.1 s using $n = 200$.



(a) Comparison of different Riemann solver with FV-WENO-OA-P scheme

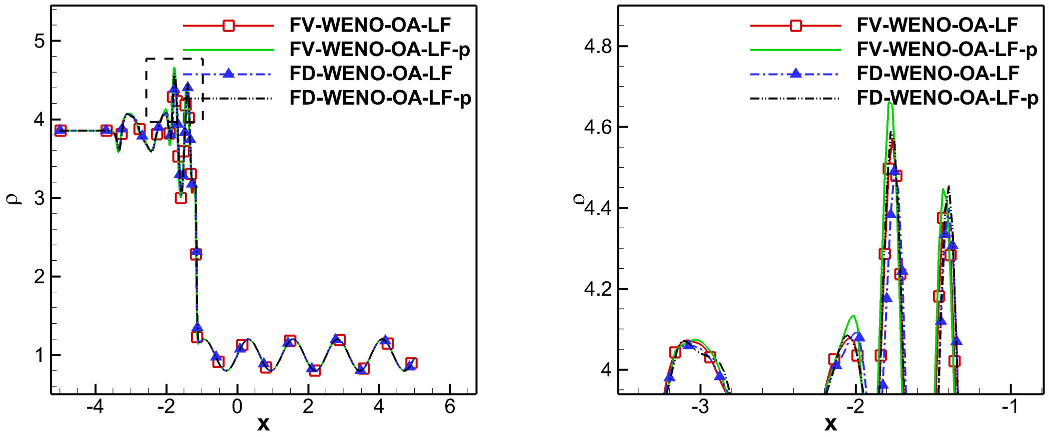
(b) Comparison of different Riemann solver with FV-WENO-OA-P scheme (zoomed view)

Figure 6.10: Density plot of Sod shock tube problem for different Riemann solvers using point-based reconstruction with $CFL = 0.9$ at 0.1 s using $n = 200$.

reconstruction procedure. Figure 6.10 shows the comparison of different Riemann solver on point-based interpolation. In figure 6.10, FV-WENO-OA-P represents point-based order adaptive WENO scheme. For point-based reconstruction procedure, HLLC solver outperformed other Riemann solvers. Please note that in cell-averaged FV-WENO-OA, Roe solver outperformed other solvers. From this, we can notice that the solution of Riemann solver also sensitive to reconstruction procedure used.

6.6.3 The effect of cell-average and point-wise reconstruction procedure on FV-WENO-OA and FD-WENO-OA

It is clear that the point-based and cell-averaged reconstructions lead to a different formulation so the effect of those reconstructions is studied on the shock tube problems. The cell-average reconstruction procedure leads to higher-order accuracy in the derivative calculation and point-wise reconstruction leads to higher-order accuracy in the interpolation. Figure 6.11a shows the comparison of WENO-OA schemes based on cell-average and point-wise reconstruction schemes on Shu-Osher problem. It is clear that point-wise reconstruction procedure is less diffusive than the cell-average reconstruction. For Sod shock tube problem, the solution of averaging and point wise reconstruction are almost the same. Table 6.17 shows the maximum CFL number can be used for different schemes for different test cases. For finite volume schemes, HLLC flux is used to calculate resultant flux at the interface. For the finite-difference scheme, Lax-Friedrichs scheme is used. All the calculations are carried out with n = 100 grid points. It is clear that averaging in reconstruction has reduced the maximum possible CFL for the scheme. FD-WENO is relatively less stable than FV-WENO in terms of maximum possible CFL but the computational cost of FV-WENO is higher than FD-WENO. Please note that CFL here is based on CFL number at which any of the primitive variables become unphysical, so to get a good result CFL number used for that problem should be much lower than the value tabulated in the table 6.17.



(a) Desnity plot of Shu-Osher problem

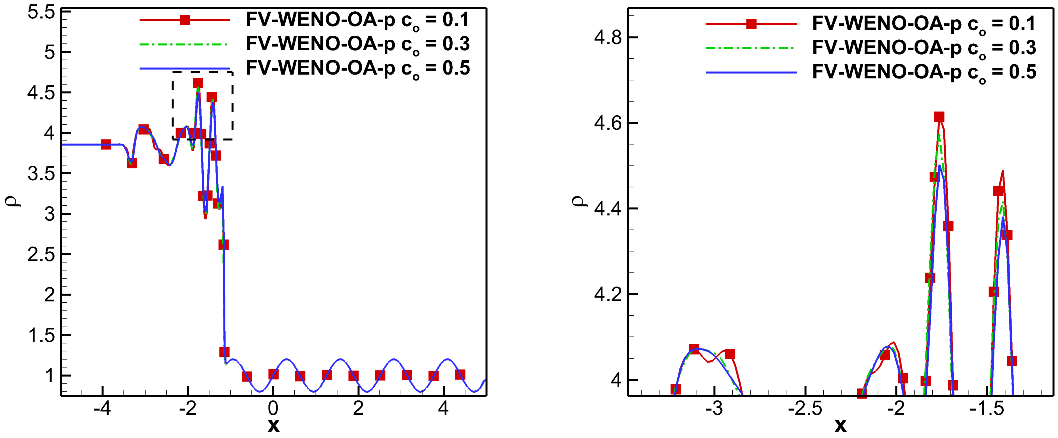
(b) Zoomed view of figure 6.11a

Figure 6.11: Comparison of FD-WENO-OA and FV-WENO-OA on point-wise and averaged-based interpolation with $CFL = 0.5$ at 0.8 s with $n = 400$.

Table 6.17

Maximum CFL number possible for different scheme for different test cases

Schemes	**Sod**	**RELSS**	**Mach 3**	**Shu-Osher**
FD-WENO-JS	1.54	1.78	1.64	1.68
FD-WENO-JS-p	1.6	2.14	1.84	2.02
FV-WENO-JS	1.81	2.23	1.94	1.82
FV-WENO-JS-p	1.96	3.03	2	2.02
FV-WENO-OA-p	1.93	2.59	2	2.17
FV-WENO-OA	1.92	2.42	1.92	1.87
FD-WENO-OA	1.59	1.89	1.69	1.59
FD-WENO-OA-p	1.79	2.33	1.89	2.02
FV-WENO-Z	1.75	2.59	1.99	2.08
FD-WENO-Z	1.68	2.08	1.87	1.98
FD-WENO-ZS	1.68	2.08	1.87	1.98
FV-WENO-EZ	1.69	2.25	1.89	1.87
FV-TENO	1.83	2.1	1.66	1.84
FV-TENO-opt	1.79	2.12	1.6	1.71



(a) Density plot.

(b) Zoomed view

Figure 6.12: Effect of c_o on FV-WENO-OA for Shu-Osher problem

6.6.4 The effect of c_o on the solution

WENO-OA schemes need one cut-off value (eq. 6.8) to determine the smooth and non-smooth polynomial. If the weight is higher than the ideal weight, the polynomial is considered as "good" polynomial. The polynomial whose weight is slightly lower than the ideal weights can be considered as "good" polynomial. This is set by the cut-off value (c_o). c_o value plays an important role in the resolution of the solution. Low the value leads to higher-order reconstruction but may cause oscillations in the solution. High value of c_o leads to a lower-order scheme but may be diffusive. Figure 6.12a shows the effect of variation of c_o on Shu-Osher problem. Since low c_o is oscillatory and high value is dissipative, $c_o = 0.3$ is suggested for better result. It can be changed based on the problem and users requirement. In the test cases, $c_o = \frac{1}{90}$ to $\frac{7}{10}$ is used but for most of the test cases $c_o = \frac{3}{10}$ is used.

6.6.5 2-D benchmark problems

Here, WENO discretization is carried out by decomposing two dimensions into two 1-D WENO[11].

2-D Riemann problem configuration 1

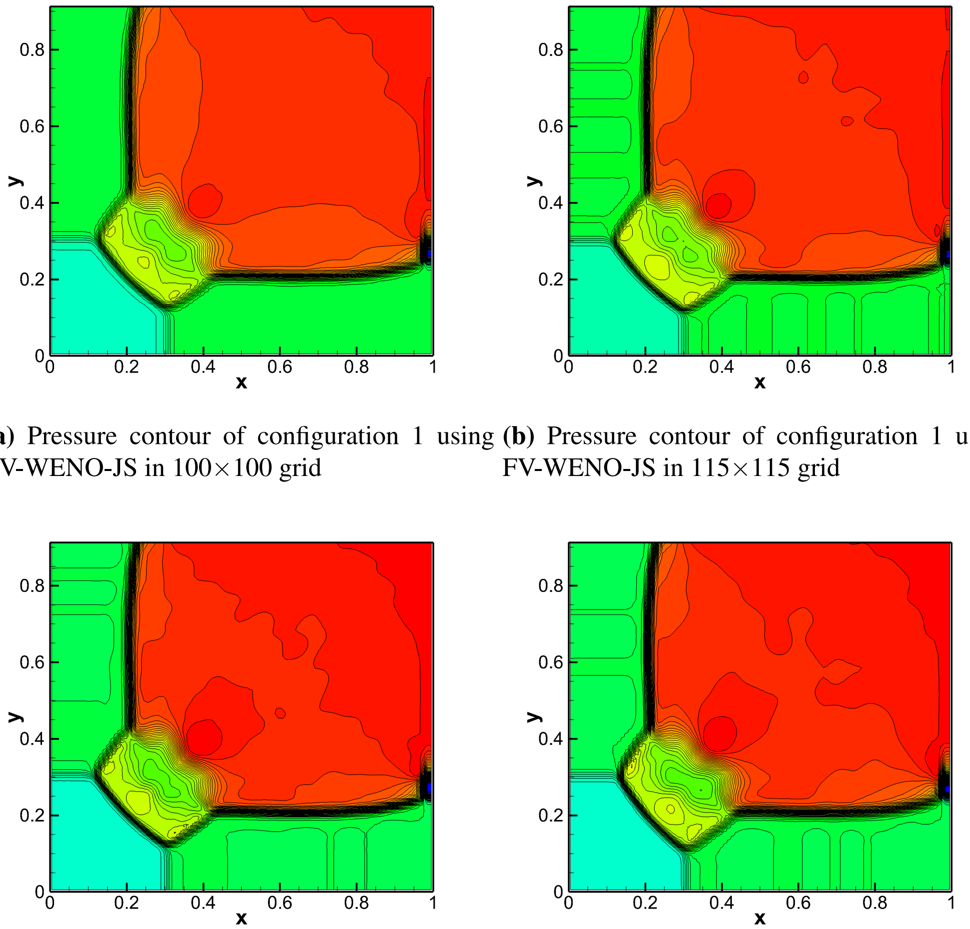
The present schemes are tested on 2-D Riemann problem. Initial condition used is:

$$(\rho, u, v, p) = \begin{cases} (1.5, & 0, & 3, & 1.5) & if \;\; 0.5 \leq x \leq 1, \;\; 0.5 \leq y \leq 1 \\ (0.5323 & 1.206, & 0, & 0.3) & if \;\; 0 < x \leq 0.5, \;\; 0.5 \leq y \leq 1 \\ (0.138, & 1.206, & 1.206, & 0.029) & if \;\; 0 \leq x < 0.5, \;\; 0 \leq y \leq 0.5 \\ (0.5323, & 0, & 1.206, & 0.3) & if \;\; 0.5 < x \leq 1, \;\; 0 \leq y < 0.5 \end{cases}$$

The flow is solved up to the flow time $T = 0.45$ s using Rusanov-flux. The solution of the present test case is shown in figure 6.13. Figure 6.13a and figure 6.13b show the pressure contour of the the present test case in 100×100 grids and 115×115 grid points respectively using FV-WENO-JS. It is clear that some of the flow features present in 115×115 are missing in 100×100 grid points. Figure 6.13c shows the solution in 100×100 grid points by FV-WENO-OA scheme. The solution of FV-WENO in 115×115 grid points is similar to FV-WENO-OA in 100×100 grid points. Zero gradient boundary condition is used at the boundaries. From the figure, it is clear that FV-WENO-JS requires more grid point to produce the same result of FV-WENO-OA or WENO-Z. FV-WENO-JS in 115×115 grid points diverges when CFL = 0.5 is used so CFL = 0.3 is used but FV-WENO-OA and FV-WENO-Z did not diverge using 115×115 grid points with CFL = 0.5 so FV-WENO-OA is more stable and less-diffusive than FV-WENO-JS. Here, reconstruction based on cell-averaging and point-based did not show any significant difference. For this test case, WENO-Z and WENO-OA behaved more or less the same and produced the same result. For this problem, WENO-EZ, TENO and TENO-opt failed to give the result with the same solver procedure followed for WENO-OA.

2-D Riemann problem configuration

The initial condition used is



(a) Pressure contour of configuration 1 using FV-WENO-JS in 100×100 grid

(b) Pressure contour of configuration 1 using FV-WENO-JS in 115×115 grid

(c) Pressure contour of configuration 1 using FV-WENO-OA in 100×100 grid

(d) Pressure contour of configuration 1 using FV-WENO-Z in 100×100 grid

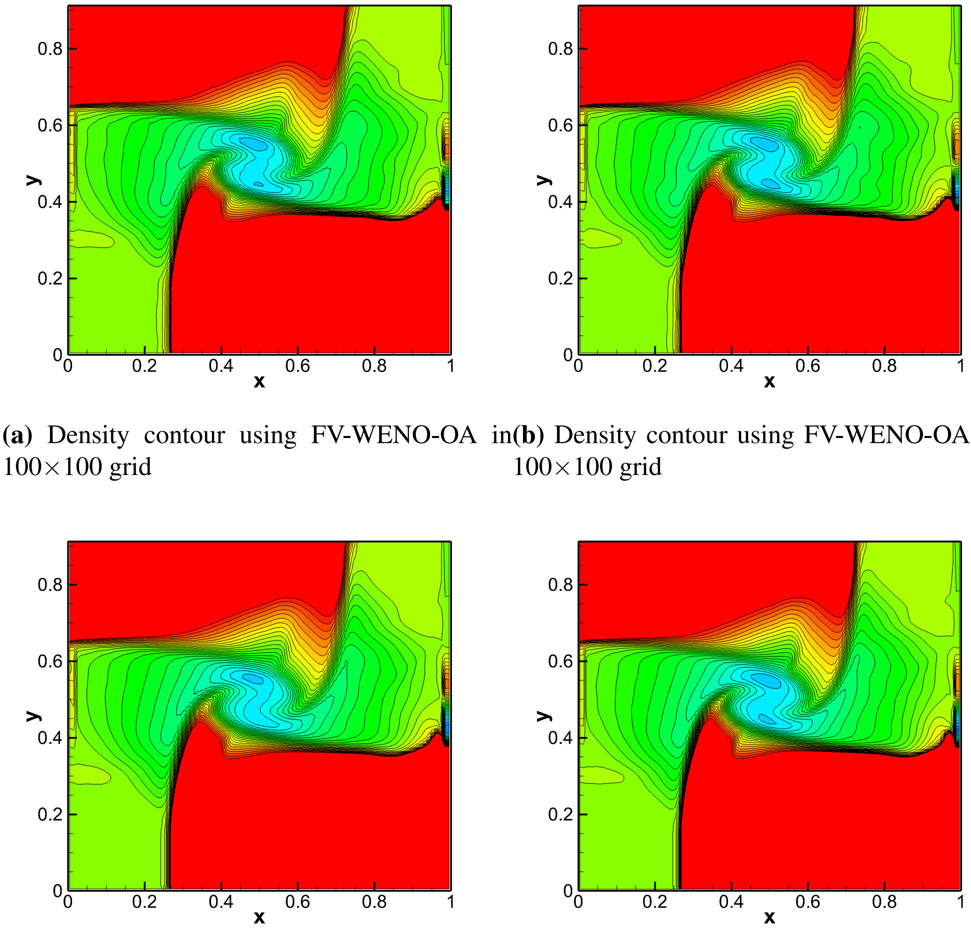
Figure 6.13: Pressure contour of configuration 1 at $T = 0.45$ s

$$
(\rho, u, v, p) = \begin{cases}
(1, \quad 0.75, \quad -0.5, \quad 1) \quad if \quad 0.5 \leq x \leq 1, \quad 0.5 \leq y \leq 1 \\
(2 \quad 0.75, \quad 0.5, \quad 1) \quad if \quad 0 < x \leq 0.5, \quad 0.5 \leq y \leq 1 \\
(1, \quad -0.75, \quad 0.5, \quad 1) \quad if \quad 0 \leq x < 0.5, \quad 0 \leq y \leq 0.5 \\
(3, \quad -0.75, \quad -0.5, \quad 1) \quad if \quad 0.5 < x \leq 1, \quad 0 \leq y < 0.5
\end{cases}
$$

The flow is solved using $CFL = 0.5$ up to the flow time $T = 0.3$ s using Rusanov-flux. Zero gradient boundary condition is used at the boundaries. The density contour of this configuration is shown in figure 6.14. Figure 6.14a and figure 6.14c shows the density contour of the present test case using 100×100 grid points using FV-WENO-OA and FV-WENO-Z respectively. Though there is no big difference in the solution of FV-WENO-OA and FV-WENO-Z, FV-WENO-Z is under-resolved in $100{\times}100$ grid points. To obtain the same result of FV-WENO-OA-p, FV-WENO-M required $120{\times}120$ grid points (figure 6.14d). FV-WENO-JS, WENO-EZ, TENO and TENO-opt unable to solve this problem with the grid and CFL number used for FV-WENO-OA and FV-WENO-Z. Here also point-based and cell-averaged based reconstruction did not show any significant improvement in terms of resolution.

Isentropic vortex convection

Isentropic vortex convection test case is studied over the domain [0,10]×[0,10] up to the flow time 50 s. CFL number used is 0.5 with $120{\times}120$ grid points. The bottom and the top boundary conditions are zero gradients and the left and the right boundary conditions are outlet. Initial condition used is



(a) Density contour using FV-WENO-OA in 100×100 grid

(b) Density contour using FV-WENO-OA-p in 100×100 grid

(c) Density contour using FV-WENO-Z in 100×100 grid

(d) Density contour using FV-WENO-Z in 120×120 grid

Figure 6.14: Density contour of configuration 2

$$\begin{Bmatrix} \rho \\ \\ u \\ \\ v \\ \\ p \end{Bmatrix} = \begin{Bmatrix} \left[1 - \frac{(\gamma-1)b^2 \exp\left(1-r^2\right)}{(8\gamma\sigma^2)}\right]^{2.5} \\ \\ -\frac{b}{2\sigma} \exp\left(\frac{1-r^2}{2}\right)(y-5) \\ \\ \frac{b}{2\sigma} \exp\left(\frac{1-r^2}{2}\right)(x-5) \\ \\ \rho^\gamma \end{Bmatrix}$$

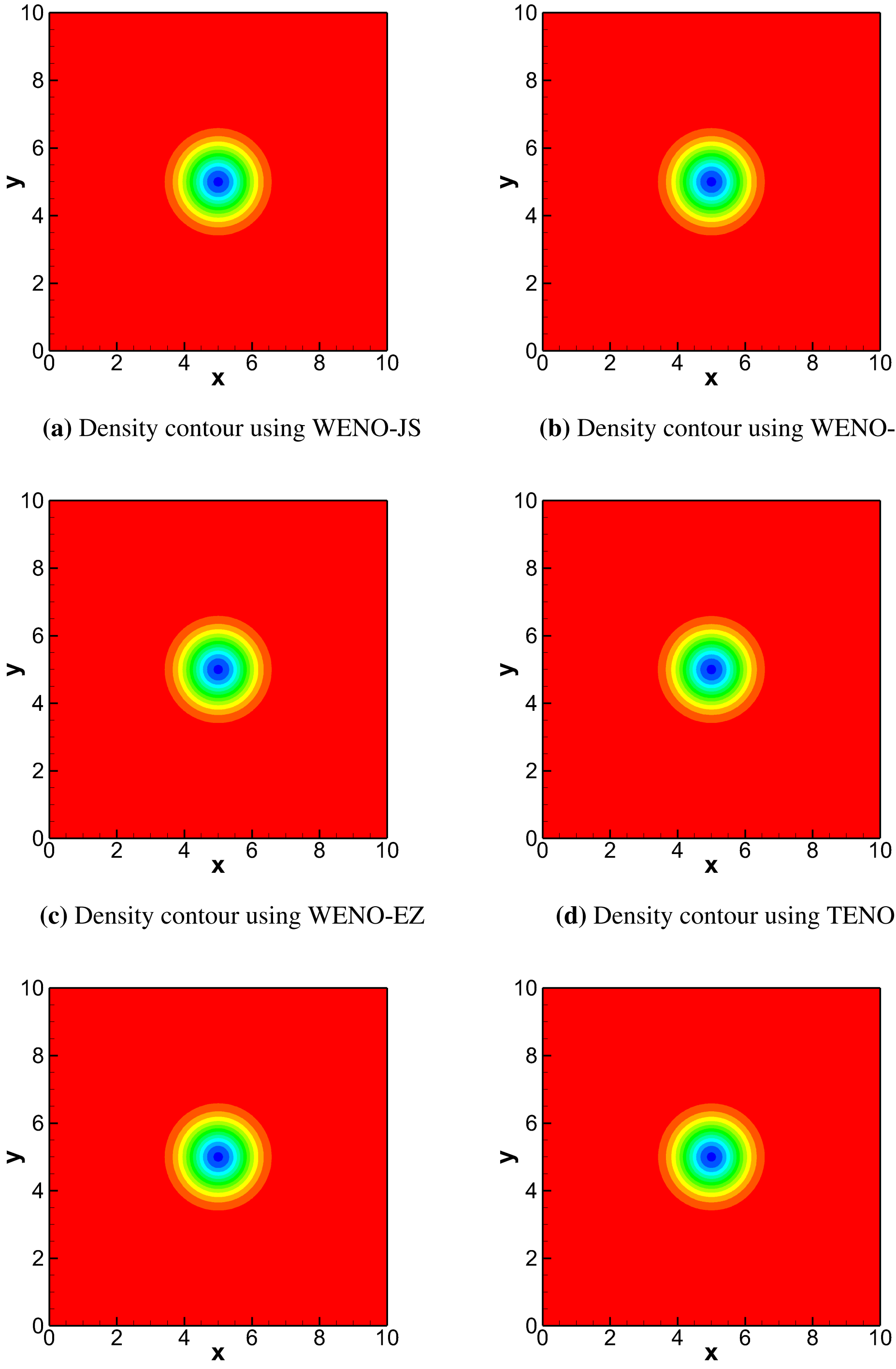
Where $\sigma = 4\tan^{-1}(1)$, $b = \frac{1}{2}$ and $r = \sqrt{x^2 + y^2}$. The solution of the isentropic vortex convection test case is shown in figure 6.15. For this problem, there is no difference in the solution of the different WENO schemes considered.

Double Mach reflection

Double Mach reflection test case is commonly used to check the small scale resolution capacity of the scheme. The size of computational domain is [0, 4]×[0, 1] using 1024× 256 grid points. The left end of the domain is the inlet and the right end is the outlet. The bottom side is reflecting wall starting from $x = 0.6$. The initial condition is a right moving incident shock of Mach number 10 with an angle $60°$. The post-shock values are assigned to the left side of the shock. We refer to [199] for more details of the present test case. Problem is solved up to the flow time 0.2 s with CFL number 0.1. The local-Lax-Friedrich scheme is used as Riemann solver. Initial condition used is:

$$(\rho, u, v, p) = \begin{cases} (1.4, 0, 0, 1) & \text{if} \qquad y < 1.732(x - 0.1667) \\ \\ (8, 7.145, -4.125, 116.833) & \text{otherwise} \end{cases}$$

The solution of the present test case is shown in figure 6.16. For this problem except the solution of TENO, other schemes performed well. Tuning parameter used for WENO-OA is 0.7.



(a) Density contour using WENO-JS

(b) Density contour using WENO-Z

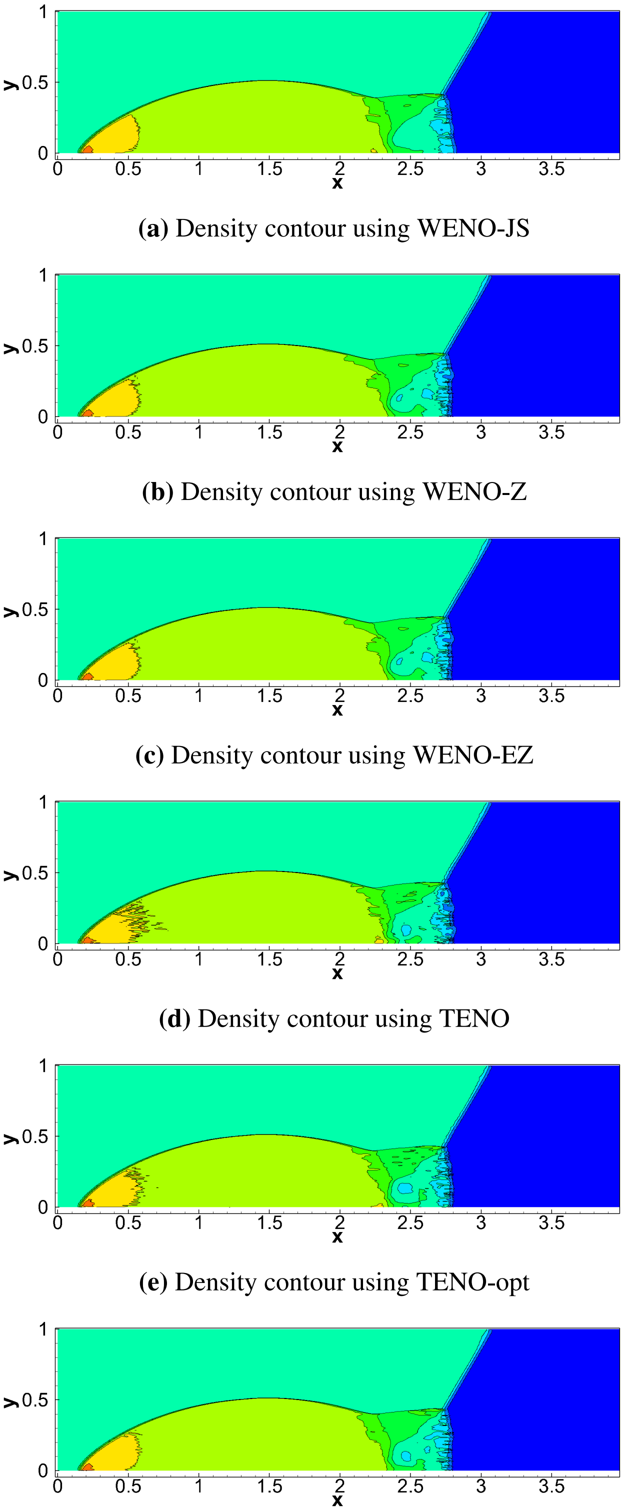
(c) Density contour using WENO-EZ

(d) Density contour using TENO

(e) Density contour using TENO-opt

(f) Density contour using FV-WENO-OA

Figure 6.15: Density contour of isentropic vortex convection using 120× 120 grid points at 50 s



(a) Density contour using WENO-JS

(b) Density contour using WENO-Z

(c) Density contour using WENO-EZ

(d) Density contour using TENO

(e) Density contour using TENO-opt

(f) Density contour using FV-WENO-OA

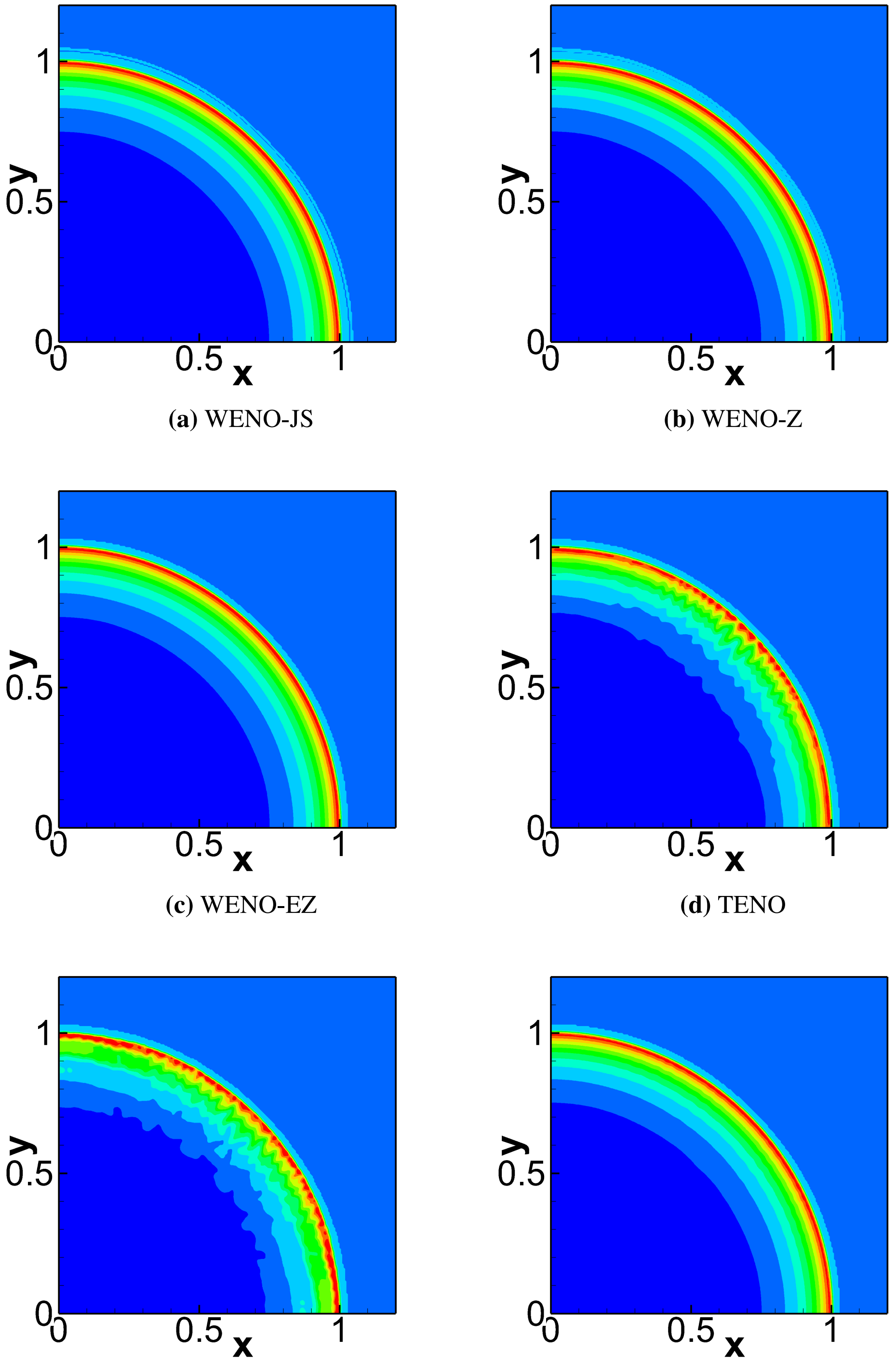
Figure 6.16: Density contour of Double Mach reflection problem using 1024× 256 grid points at 0.2 s

Sedov problem

Sedov problem is a point-blast in a uniform medium. This test case is used to identify the robustness of the scheme to preserve cylindrical symmetry and handling strong shocks. An exact solution of the test case is presented in [200]. The problem is solved over the domain [0, 1.2] × [0, 1.2]. The domain is discretized using 240×240 grid points. The problem is solved up to 1 s with CFL number 0.5. Initial condition used is $\rho_0 = 1$; $u_0 = 0$; $v_0 = 0$ and $p_0 = 10^{-6}$ except at the origin where pressure is set to $p_0 = \frac{(\gamma-1)\rho_0\epsilon_s}{\Delta x \Delta y}$; where $\epsilon_s = 0.244816$, $\gamma = 1.4$, Δx and Δy are grid size in x and y direction. The solution is an infinite strength shock wave propagating in the radial direction. The density contour of Sedov problem is shown in figure 6.17. For this problem, excessive oscillations are present in TENO, TENO-opt and WENO-OA but WENO-OA is able to produce a good result because it has a tuning parameter. That is set to 0.7 instead of standard 0.3, else WENO-OA also produce oscillations but it can able to predict the peak value of density better than others. It is noteworthy that others are unable to predict it well. Figure 6.18 shows the density contour of WENO-OA with different tuning parameter. Figure 6.19 shows the variation of density at $y = 0$ line in the solution. From the figure it is clear that WENO-OA is less diffusive than other schemes. WENO-Z outperformed other schemes for this test case.

Shock/shear layer interaction

Shock/shear layer interaction is to study the resolving capacity of the scheme on preserving small scale vortices when interacting with a shock wave. A shear layer with Mach number 0.6 impact on an oblique shock. The computational domain is [0, 200] ×[-20, 20] is solved up to flow time 120 s with CFL number 0.5. The number of grid points used is 480×120. Initially, the vortices are passed through an oblique shock, again the disturbed vortices are passed through the reflected shock by the lower boundary. Initial condition used is [188]:



(a) WENO-JS

(b) WENO-Z

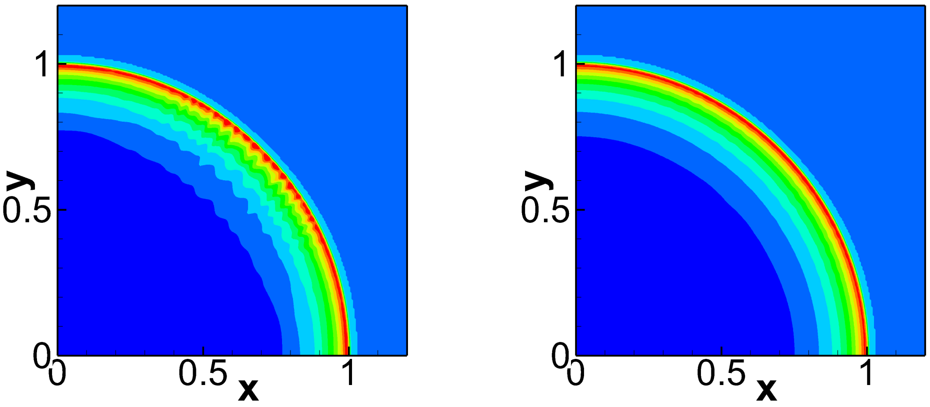
(c) WENO-EZ

(d) TENO

(e) TENO-opt

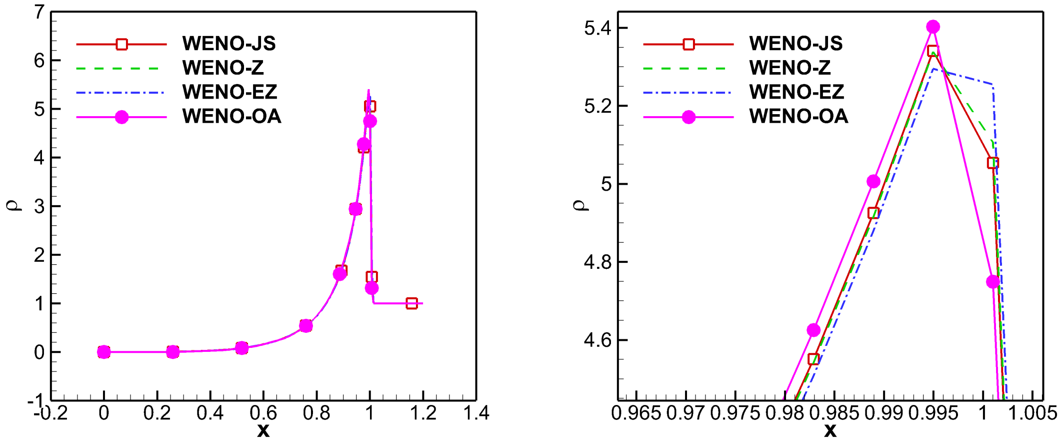
(f) FV-WENO-OA with $c_o = 0.7$

Figure 6.17: Density contour of Sedov problem case using 240× 240 grid points at 1 s



(a) FV-WENO-OA with $c_o = 0.3$ **(b)** FV-WENO-OA with $c_o = 0.7$

Figure 6.18: Density contour of Sedov problem case using 240× 240 grid points at 1 s with WENO-OA scheme



(a) Density plot of Sedov test **(b)** Zoomed view of density plot

Figure 6.19: Density plot of Sedov test case at $T = 1s$ at $y = 0$ line

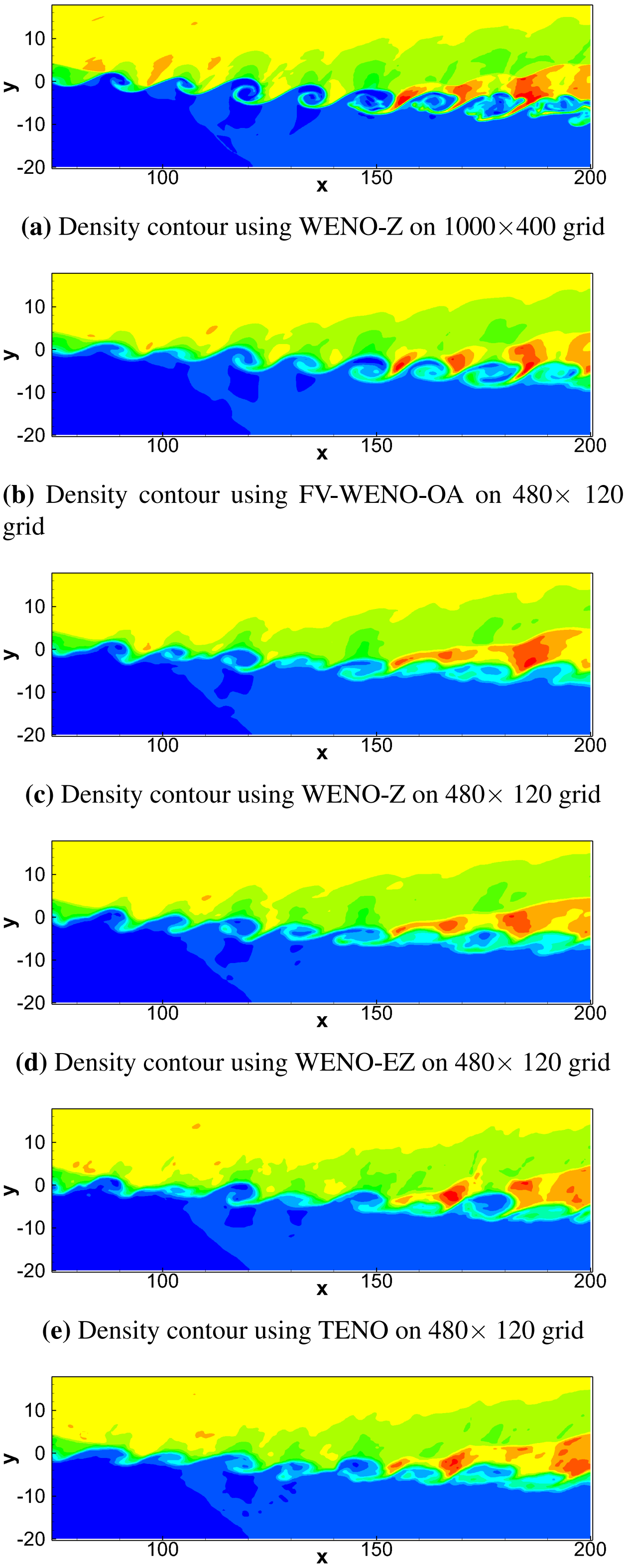
$$\left\{\begin{array}{c} \rho \\ \\ u \\ \\ v \\ \\ p \end{array}\right\} = \left\{\begin{array}{llll} 0.3626 & \text{if } y < 0 & \text{else} & 1.6374 \\ \\ 2.5 + 0.5\tanh(2y) & \text{if } y < 0 & \text{else} & 2.5 + 0.5\tanh(2y) \\ \\ v' & \text{if } y < 0 & \text{else} & v' \\ \\ 0.3327 & \text{if } y < 0 & \text{else} & 0.3327 \end{array}\right.$$

The upper boundary is set as post-shock condition (ρ, u, v, p) = (2.1101, 2.9709, -0.1367, 0.4754). The lower boundary is set as a slip wall. Fluctuation is added to the y-component velocity that is:

$$v' = 0.05\cos\left(\frac{2\pi kt \times 2.68}{30}\right)\exp\left(-y^2/10\right) + 0.05\cos\left(\frac{2\pi kt \times 2.68}{30} + \frac{\pi}{2}\right)\exp\left(-y^2/10\right)$$

.

The density and vorticity contour of different WENO schemes are shown in figure 6.20 and figure 6.21 respectively. Reference solution is carried on 1000×400 are shown in figure 6.20a and figure 6.21a. Solution obtained using WENO-OA is more closer to the other schemes even on a course grid. Other schemes unable to retain the structure of the vortices but WENO-OA able to retain the structures on the course grid. For this problem, WENO-OA outperformed other schemes and solution is comparable to the reference solution presented in [188] with lesser number of grid points. Since WENO-OA is able to retain the structure on the coarse grid and it is computationally efficient.



(a) Density contour using WENO-Z on 1000×400 grid

(b) Density contour using FV-WENO-OA on 480× 120 grid

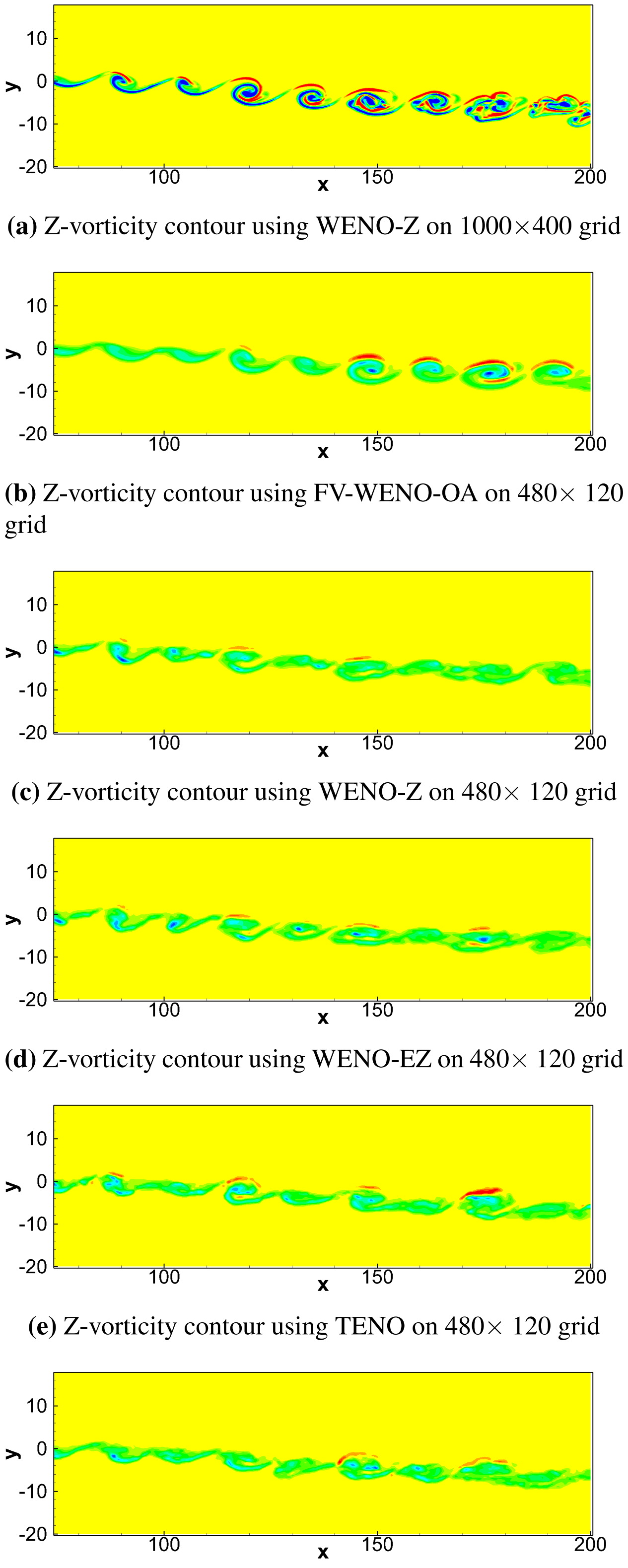
(c) Density contour using WENO-Z on 480× 120 grid

(d) Density contour using WENO-EZ on 480× 120 grid

(e) Density contour using TENO on 480× 120 grid

(f) Density contour using TENO-opt on 480× 120 grid

Figure 6.20: Density contour of Shock/shear-layer interaction problem at 0.2 s



(a) Z-vorticity contour using WENO-Z on 1000×400 grid

(b) Z-vorticity contour using FV-WENO-OA on 480× 120 grid

(c) Z-vorticity contour using WENO-Z on 480× 120 grid

(d) Z-vorticity contour using WENO-EZ on 480× 120 grid

(e) Z-vorticity contour using TENO on 480× 120 grid

(f) Z-vorticity contour using TENO-opt on 480× 120 grid

Figure 6.21: Z-vorticity contour of Shock/shear-layer interaction problem at 0.2 s

6.7 Computational cost and accuracy

Though the number of operations per time WENO-OA function evaluation is higher than WENO-JS, WENO-Z and WENO-EZ, it is relatively more accurate than those schemes in most of the test cases. Since it is more accurate than others, it require less number of grid points than other schemes for a given result where it can save computational cost. Table 6.18 shows the computational cost of different WENO schemes for shock-shear layer interaction test case. WENO-Z schemes with 2000×600 is taken as reference to calculate RMS error. RMS error is calculated on density data at flow time 120 s. For a given RMS error approximately 0.17, computational time and grid points for different schemes are calculated. In table 6.18, η is computational efficiency which is the ratio of computational time of a given scheme and computational time of WENO-OA. From this we can notice that, WENO-OA is 4.9 and 2.4 times more economical than TENO and TENO-opt schemes respectively.

Table 6.18
Computational efficiency of different schemes

Scheme	**Time (s)**	**Grid**	**Iteration**	**RMS error**	η
WENO-JS	4701.75	500×300	5416	0.174367	177.7621
WENO-Z	4637.25	500×300	5416	0.174367	175.3237
WENO-OA	2644.96	400×200	3856	0.163596	100
WENO-EZ	3467.21	400×300	4816	0.175394	131.0873
TENO	13047.42	500×300	9697	0.178497	493.2924
TENO-opt	6567.69	500×200	7402	0.1752	248.309

6.8 Three level seventh-order WENO scheme

The three-level seventh-order WENO scheme is an extension of the five-order three-level WENO scheme presented in chapter 6. This scheme is referred as WENO-OA-467. It can achieve a fourth or sixth or seventh-order based on the smoothness of the solution. The derivation of this scheme is presented in section 6.4.2. Here, we will study the performance of the WENO-OA-467 in different test cases.

6.8.1 Sod shock tube problem

Sod shock tube problem is one of the classic benchmark problems for the Euler equation. It was introduced in [5] in 1978 to study the performance of some finite difference scheme in hyperbolic equations. Simulation is carried out with 200 grid points up to 0.1 s flow time. The Lax-Frederich scheme is used as a Riemann solver. Time marching is carried out using the method HRK42 [105]. The component-wise solver that uses a conservative variable is used in WENO algorithm stencil selection.

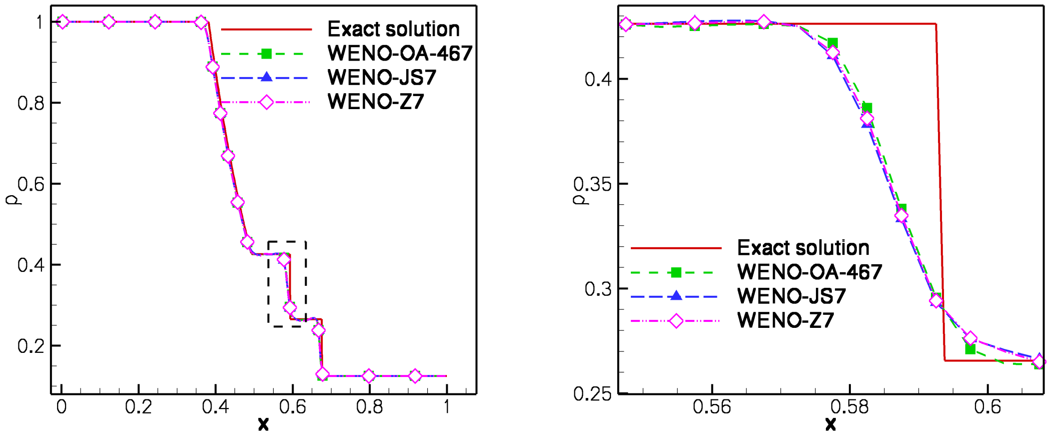
Initial condition used for this problem is

$$(\rho, u, p) = \begin{cases} (1, 0, 1) & x \leq 0.5 \\ \\ (0.125, 0, 0.1) & x > 0.5 \end{cases}$$

The problem is solved over the domain 0 to 1. The density plot at 0.1 s is shown in figure 6.22. In that figure, WENO-OA-467 shows a better resolution than others. Table 6.19 shows the L_1 error of density, velocity and pressure of Sod shock tube problem.

Table 6.19
L_1 error of Sod shock tube problem using $n = 200$

Schemes	ρ_{m}	$\mathrm{u_m}$	$\mathrm{p_m}$
WENO-OA-467	0.005059	0.008106	0.004515
WENO-JS7	0.005447	0.00852	0.004668
WENO-Z7	0.005174	0.008243	0.004508



(a) Density plot

(b) Zoomed view of contact discontinuity

Figure 6.22: Density plot of Sod shock tube problem with 200 grids $CFL = 0.5$ at $t = 0.10s$

6.8.2 Right expansion and left strong shock

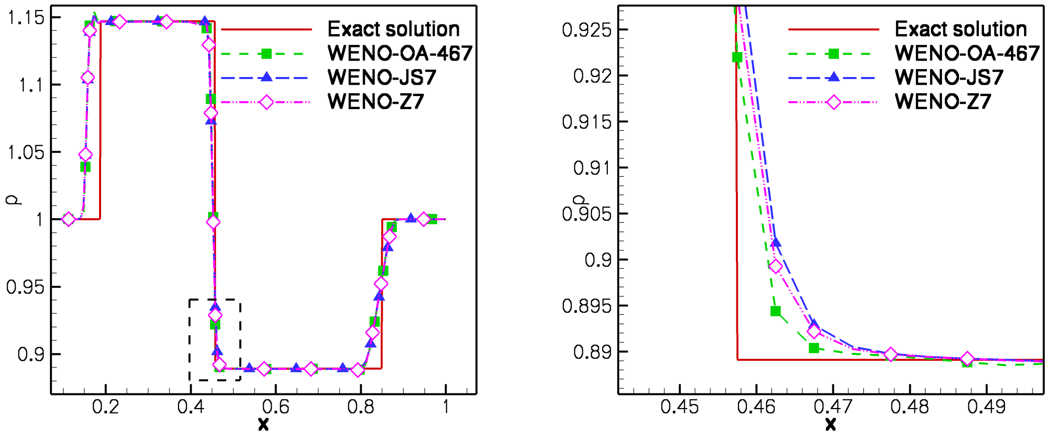
The Initial condition for right expansion and left strong shock is

$$(\rho, u, p) = \begin{cases} (1, 0, 7) & x \leq 0.5 \\ \\ (1, 0, 10) & x > 0.5 \end{cases}$$

The problem has been solved up to $t = 0.1$s using $n = 200$ over the domain (0,1). The performance of different WENO schemes is shown in figure 6.23. The Riemann solver used is the Lax-Frederich scheme. Component wise interpolation is carried out on a conservative variable. WENO-OA-467 has a better result than other schemes. Also, L_1 error of WENO-OA-467 is lower than the other schemes presented in table 6.20.

6.8.3 Mach number 3 test case

This test case is introduced in [201]. The initial condition used is,



(a) Density plot

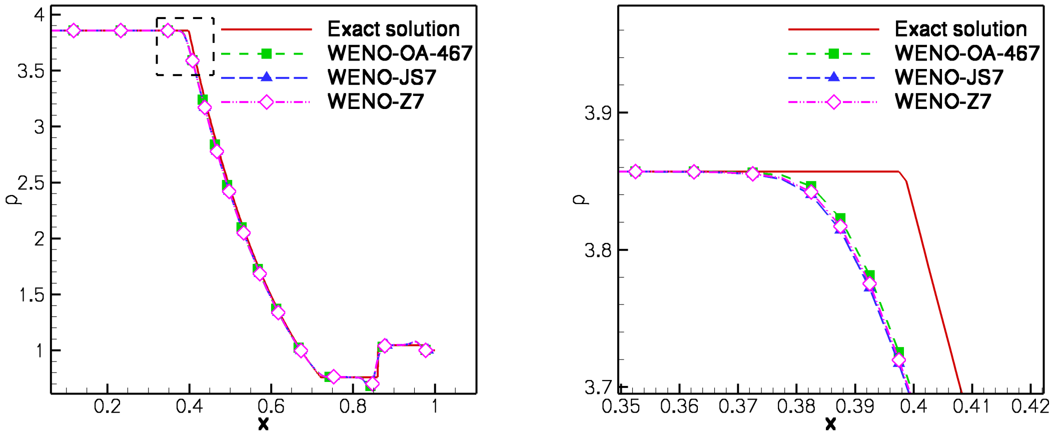
(b) Zoomed view of discontinuity

Figure 6.23: Density plot of right expansion and left strong shock tube problem with 200 grids $CFL = 0.9$ at $t = 0.1s$

Table 6.20

L_1 error of right expansion and left strong shock tube problem using $n = 200$

Schemes	$\rho_{\mathbf{m}}$	$\mathbf{u_m}$	$\mathbf{p_m}$
WENO-OA-467	0.007985	0.021041	0.071759
WENO-JS7	0.008122	0.021004	0.071499
WENO-Z7	0.008028	0.021056	0.07165



(a) Density plot

(b) Zoomed view of discontinuity

Figure 6.24: Density plot of Mach = 3 test with 200 grids $CFL = 0.9$ at $t = 0.09s$

$$(\rho, u, p) = \begin{cases} (3.857, 0.92, 10.333) & x \leq 0.5 \\ \\ (1, 3.55, 1) & x > 0.5 \end{cases}$$

The problem is solved using $n = 200$ grid point over the domain 0 to 1. Riemann solver used is Lax-Frederich's scheme and solved up to 0.1 s. Component reconstruction is carried on the conservative variable. The solution to this problem is shown in figure 6.24. Here also WENO-OA-467 performed better than other schemes. L_1 error of the present test case is shown in table 6.21. WENO-OA-467 shows relatively lower error than other schemes.

Table 6.21
L_1 error of Mach 3 test case using $n = 200$

Schemes	$\rho_\mathbf{m}$	$\mathbf{u_m}$	$\mathbf{p_m}$
WENO-OA-467 0.3	0.017924	0.011125	0.046796
WENO-JS7	0.018389	0.011351	0.04711
WENO-Z7	0.018266	0.011332	0.046971

The performance of the seventh-order three-level order adaptive WENO scheme is studied for the one-dimensional test case. WENO-OA-467 shows a better result compared to other schemes. Unfortunately, the preset scheme didn't show significant improvement in the result in two-dimensional test case that is not included in the present work.

6.9 Summary

The standard WENO procedures lead to lower order accuracy when one of the candidate polynomials is not smooth. In addition to that JS-WENO and WENO-Z is relatively much more sensitive to small variation in the smoothness indicator so a new weighting procedure is presented to cure those problems. In most of the WENO schemes, the weighting function is directly related to the smoothness indicator. Here, the importance to decouple the direct relation between weighting function and the smoothness indicator is emphasized with an example. A novel computationally economical switch without logical statements to eliminate the direct relation between the weighting function and the smoothness indicator is also proposed. This switch can also act as a step-filter to remove a certain range of frequencies present in the data. The formulation of WENO-OA for finite difference and finite volume schemes are presented. In addition to that interpolation based WENO-OA is also discussed. To the best of the knowledge of the authors, this is the only WENO which has a tuning parameter to control the order adaptive property and it can adapt any order ranging from three to five.

The test on 1-D and 2-D problems have shown that WENO-OA scheme is better than other WENO schemes in terms of resolution and accuracy. The present point-based WENO-OA (FV-WENO-OA-p) has better stability than FV-WENO-JS, FV-WENO-OA, FV-WENO-JS-p, FD-WENO-JS, FD-WENO-JS-p, FD-WENO-OA-p, FD-WENO-OA, FV-WENO-Z, FD-WENO-Z, TENO, TENO-opt and WENO-EZ. Though FV formulation is based on cell-averaging, the test on different 1-D Euler equation with Riemann problem like initial condition shows poor resolution and lower stability than the point-based reconstruction. From these observations, point-based reconstruction is recommended over cell-averaged reconstruction for the Euler equation which having a discontinuity in the solution. If there is no discontinuity, cell-average reconstruction may work better than point-based method and it can maintain higher order derivative approximation on uniform meshes.

There is no significant difference between the results of WENO method based on finite difference and finite volume but finite volume is better than the finite difference in terms of stability. In the case of WENO-Z and WENO-ZS, FD-WENO scheme is better than FV-WENO scheme. Please note that FV based WENO is computationally more expensive than FD-WENO. Though WENO-ZS used a fourth order polynomial basis, it is unable to give a significant improvement over WENO-Z for the Euler equation. For the test cases considered, TENO is better than TENO-opt, though the later one is optimized for lower spectral errors. The results of WENO-EZ and WENO-OA are more or less similar but tuning parameter present in WENO-OA made WENO-OA more robust and more accurate than WENO-EZ.

The test on Riemann solvers using finite volume WENO-OA scheme has shown that HLLC has a better resolution than others when point-based reconstruction is used. Roe scheme has given a better result than the others when reconstruction is carried out using cell-averaged values. Point-based reconstruction does not show some significant improvement over cell-averaged reconstruction in finite difference WENO but shows a better resolution and stability with finite volume based WENO. Though WENO-OA schemes show a marginal improvement in the one-dimensional test cases, it shows some significant improvement in two-dimensional test cases in terms of stability and the resolution over WENO-JS, WENO-Z and TENO schemes. The present WENO-OA scheme is computationally more expensive than WENO-JS, WENO-Z and WENO-EZ for a fixed grid but it needs lesser number of grid points to achieve the same result of those schemes. All other families of WENO schemes are unable to solve some of the test cases considered but WENO-OA is able to handle all the problem because of the tuning parameter.

7

Discontinuity Preserving Scheme

The order barrier theorem of Godunov [63] stated that all schemes higher than the first-order are non-monotonous. Van Leer [108] extended the SCS to second-order scheme for shock problems using the concept of Total Variation Diminishing (TVD). Because TVD schemes introduce excessive dissipation and reduce to lower-order at shocks, it gives diffused results. This limitation was partially overcome by Liu et al. [154] using the WENO scheme, which is less diffusive than the limiters.

Most of the SCS are based on Piece-wise Parabolic Reconstruction (PPR) polynomials. It is known that the PPR polynomials suffer from Runge's phenomenon, so Marquina [202] investigated a non-parabolic reconstruction. Although some of the non-parabolic reconstructions show some improvement over PPR, they need a switch mechanism at the shocks. The reconstruction procedure used by Deng *et al.* [203] considerably eliminated the numerical dissipation present in the reconstruction basis. However, this procedure cannot reduce the artificial dissipation caused by the Riemann-solver. It is impossible to solve problems involving shocks without Riemann-solver or artificial dissipation using existing algorithms, so eliminating Riemann-solver or reducing artificial dissipation in the solver may lead to carbuncle phenomenon or reduces the robustness of the solver. It is common practice to refine the grid or try to achieve a higher-order reconstruction to obtain an accurate shock structure. The former increases the cost of computation, and the latter leads to oscillations in the solution. To overcome this issue, we have proposed a scheme to reduce the numerical dissipation present in the schemes.

The conservative equations are solved using SCS in this method. WENO [11] is being used in the present study for the Euler equation, and backward in space discretization is used for the convective equation. To preserve the discontinuities present in the solution, DPS is applied to the solution of SCS. This approach is similar to the JST-scheme introduced by Jameson [13], where artificial diffusion or filtering is carried out on a solution obtained from the regular SCS to eliminate the oscillations present in the solution. We have validated our scheme for linear convection equation and non-linear Euler equations and found some significant improvement over others in terms of resolution. The present scheme can resolve the discontinuity with three grid points, irrespective of the SCS and Riemann solver used.

7.1 Methodology

The details of the conservative equation and numerical schemes are described here. We have used WENO for interpolation. As the exact Riemann solvers are computationally expensive, Rusanov's scheme (also referred to as Local Lax-Fredrich / LLF) [204] is used as an approximate Riemann solver. The values at the cell-interface are calculated by interpolating conservative variables.

7.1.1 Discretization of Euler equation

Euler equation locally produces step-function like discontinuity and has the exact solution for some cases. One-dimensional Euler equation is

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}}{\partial x} = 0 \tag{7.1}$$

$$\mathbf{U} = \begin{bmatrix} \rho \\ \\ \rho u \\ \\ \rho e_t \end{bmatrix} \mathbf{F} = \begin{bmatrix} \rho u \\ \\ \rho u^2 + p \\ \\ u(\rho e_t + p) \end{bmatrix}$$

Using conservative discretization and method of lines it will become

$$\frac{\mathrm{d}\bar{U}_i}{\mathrm{d}t} = \frac{\mathbf{F}(\mathbf{U_{i+0.5}}) - \mathbf{F}(\mathbf{U_{i-0.5}})}{\Delta x} \tag{7.2}$$

Once the spatial discretization is computed, the time marching is carried out using HRK31 [105] or forward Euler method.

7.1.2 WENO

Since the flux (**F**) is a function of conservative variable (**U**), the face values of **U** (i.e) $\mathbf{U}_{i+0.5}$ is calculated using WENO scheme. WENO basis are [11]

$$p^0_{i+0.5} = \frac{1}{6}(2\bar{\mathbf{U}}_{i-2} - 7\bar{\mathbf{U}}_{i-1} + 11\bar{\mathbf{U}}_i) \tag{7.3a}$$

$$p^1_{i+0.5} = \frac{1}{6}(-\bar{\mathbf{U}}_{i-1} + 5\bar{\mathbf{U}}_i + 2\bar{\mathbf{U}}_{i+1}) \tag{7.3b}$$

$$p^2_{i+0.5} = \frac{1}{6}(2\bar{\mathbf{U}}_i + 5\bar{\mathbf{U}}_{i+1} - \bar{\mathbf{U}}_{i+2}) \tag{7.3c}$$

Ideal weights of those polynomials are $\gamma_0 = \frac{1}{10}$, $\gamma_1 = \frac{3}{5}$ and $\gamma_2 = \frac{3}{10}$

The final reconstruction function is

$$\mathbf{U}_{i+0.5} = \sum_{j=0}^{2} \omega_j p^j_{i+0.5} \tag{7.4}$$

$$\tilde{\omega}_j = \frac{\gamma_j}{(\epsilon + \beta_j)^2} \tag{7.5}$$

$$\omega_j = \frac{\tilde{\omega}_j}{\sum_j \tilde{\omega}_j} \tag{7.6}$$

7.1.3 Discontinuity preserving scheme

Once the conservative equation is solved using SCS, the solution is refined using DPS. The DPS first detects the location of the shock by calculating the slopes. The slopes are calculated using the following

$$\zeta = \frac{(\mathbf{U}_{i+1} - \mathbf{U}_{i-1})}{2\Delta x}; \quad \mathbf{d}\zeta = \frac{(\zeta_{i+1} - \zeta_{i-1})}{2\Delta x} \tag{7.7}$$

The position of the critical points (shocks) is obtained by calculating the first derivative of ζ using local maxima calculating procedure. Once the shock locations are identified, a hyperbolic-tangent function is used to fit the shock. The function is similar to the function proposed by Deng *et al.* [203] that is

$$\bar{U}_i = \bar{U}_{min} + \frac{\bar{U}_{max}}{2}[1 + \theta \times tanh(k\tilde{x}_i)] \tag{7.8}$$

Let k_1 is the location of discontinuity. It takes a value $i+0$ (centre of discontinuity) locally and s_t is the span of smoothing function. $s_t = 5$ to 15 and $k = 10000$ gives a good result for most of the cases. If two shocks are very close s_t should be small. If the scheme is very dissipative s_t should be a big number.

$$\begin{aligned} \bar{U}_{min} = &\ min([U(k_1 - s_t), U(k_1 - s_t + 1) \\ & ..., U(k_1 + s_t)]) \end{aligned} \tag{7.9}$$

$$\begin{aligned} \bar{U}_{max} = &\ max([U(k_1 - s_t), U(k_1 - s_t + 1) \\ & ..., U(k_1 + s_t)]) - \bar{U}_{min} \end{aligned} \tag{7.10}$$

$$\theta = sign[U(k_1 + s_t) - U(k_1 - s_t)] \tag{7.11}$$

$$\tilde{x}_i = x_i - x_{i-s_t} \tag{7.12}$$

The eq. 7.8 is applied over the interval $k_1 - s_t$ to $k_1 + s_t$ on the solution of the SCS. That means DPS is only applied at the location of shocks.

7.2 Results and discussion

To check the robustness of the present algorithm, we have tested the scheme on linear convection equation and shock tube problems.

7.2.1 Linear convection equation

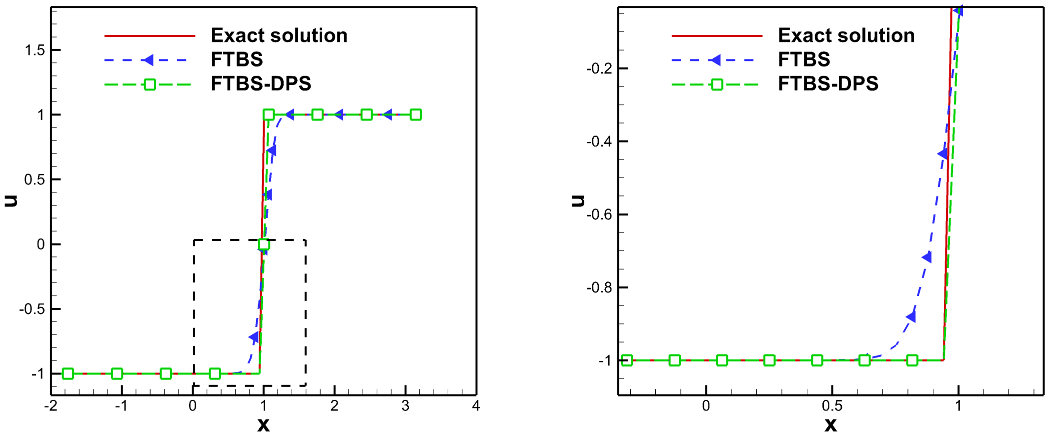
One-dimensional linear convection equation is

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial x} = 0 \tag{7.13}$$

The initial condition is

$$u(x, 0) = tanh(10000x) \tag{7.14}$$

The problem is solved in the domain $[-\pi$ to $\pi]$. Forward Euler method is used for time integration and the backward Euler for spatial discretization (FTBS). The number of grid points (n) used is 101 with the CFL = 0.8 and simulated up to the flow time 1.0053 s. Figure 7.1 shows the solution of the convection equation. It is clear that the present scheme is less dissipative and dispersive than SCS. It is worth noting that the present



(a) Solution of convection equation

(b) Zoomed view of the discontinuity

Figure 7.1: Solution of linear convection equation

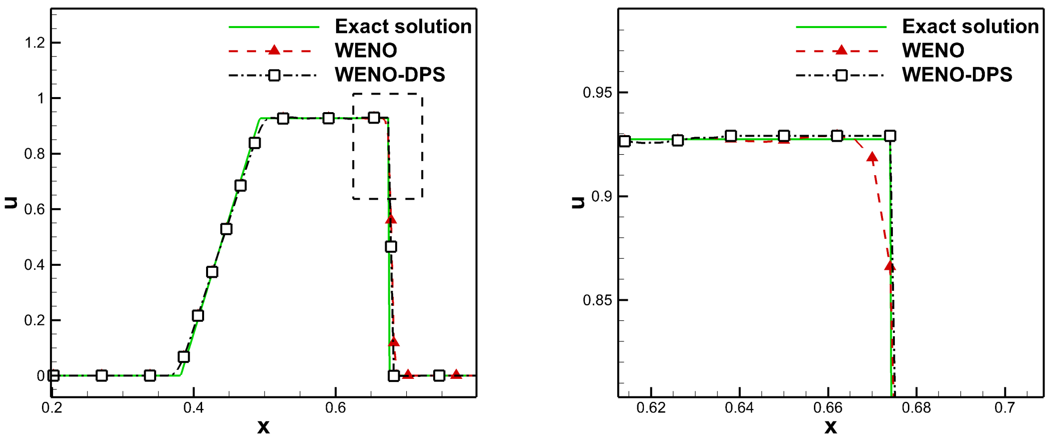
scheme (FTBS-DPS) used three-point to resolve the discontinuity, but FTBS used twelve points to resolve it.

7.2.2 Sod shock-tube problem

The present scheme is tested on Sod shock tube problem [5]. The problem is solved using 400 grid points (n) with CFL number 0.5, up to flow time = 0.1 s. The initial condition of the problem is

$$(\rho, u, p) = \begin{cases} (1, 0, 1) & x \leq 0.5 \\ (0.125, 0, 0.1) & x > 0.5 \end{cases}$$

Euler equation is solved using finite volume method (FVM), interpolation is carried out using WENO and Rusanov's scheme is used to calculate interface flux. Velocity plot of Sod shock tube problem is shown in the figure 7.2. It is evident from figure 7.2 that the current model has a better high-resolution property than the standard SCS and is comparable with the analytical solution. To solve the discontinuity, the WENO scheme used eight grid points for this problem, but three points were used by the proposed scheme.



(a) Velocity plot of Sod shock tube problem

(b) Zoomed view of the discontinuity

Figure 7.2: Solution of Sod shock tube problem

7.2.3 Mach 3 test case

Mach 3 test case [6] is solved using n = 250 with the CFL = 0.7 up-to the flow time 0.1 s. Initial condition used for this case is

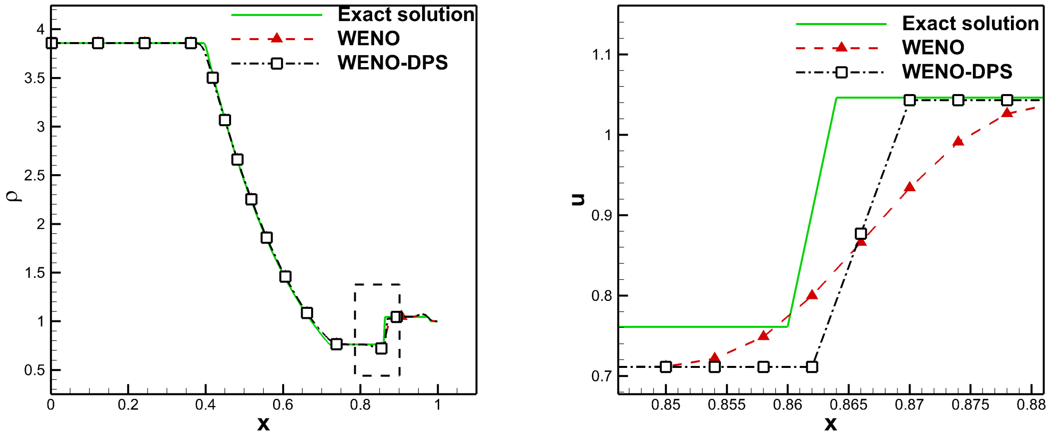
$$(\rho, u, p) = \begin{cases} (3.857, 0.92, 10.333) & x \leq 0.5 \\ (1, 3.55, 1) & x > 0.5 \end{cases}$$

The discretization procedure used here is the same as that for the Sod shock tube problem. Figure 7.3 shows the velocity plot of Mach 3 test case. The present scheme shows a better shock resolving property than WENO. Incorrect shock strength is because the weighting procedure of WENO scheme is not good. It originated from the reconstruction of density in the left-state of ($i - 2$ to $i + 2$) the density reconstruction. In this case, with some dissipation, the limiters can capture the shock strength correctly.

7.2.4 Lax test case

Lax test case [205] is solved using a total number of n = 400 grid points with the CFL = 0.7 up to a flow time of 0.1s. Initial condition used is

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528) & x \leq 0.5 \\ (0.5, 0.0, 0.571) & x > 0.5 \end{cases}$$



(a) Density plot of Mach 3 test case

(b) Zoomed view of the discontinuity

Figure 7.3: Solution of Mach 3 test case

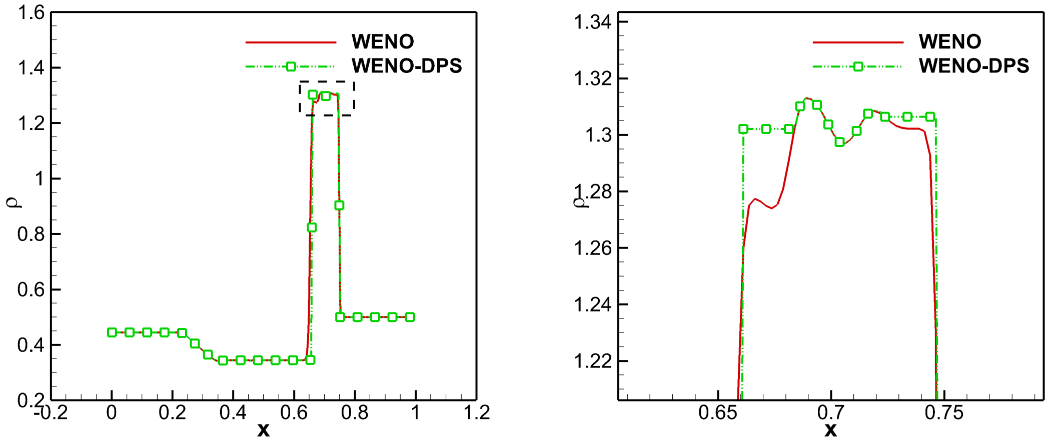
The Lax test case solution can be found in 7.4. It is evident from Figure 7.4b that WENO-DPS cannot completely remove the oscillations present in the solution; however, this can be cured by carefully adjusting the span of WENO-DPS shown in Figure 7.5, where $s_t = 20$ is used. FTBS used twelve points to resolve the discontinuity in the convection equation, but the current algorithm (FTBS-DPS) used three points in order to achieve the same.

7.2.5 Right expansion and left strong shock

The problem is solved using 400 grid points with CFL = 0.8 up to flow time 0.01 s. The Rusnove Riemann solver is used here. Initial condition used is

$$(\rho, u, p) = \begin{cases} (1, 0, 7) & x \leq 0.5 \\ \\ (1, 0, 10) & x > 0.5 \end{cases}$$

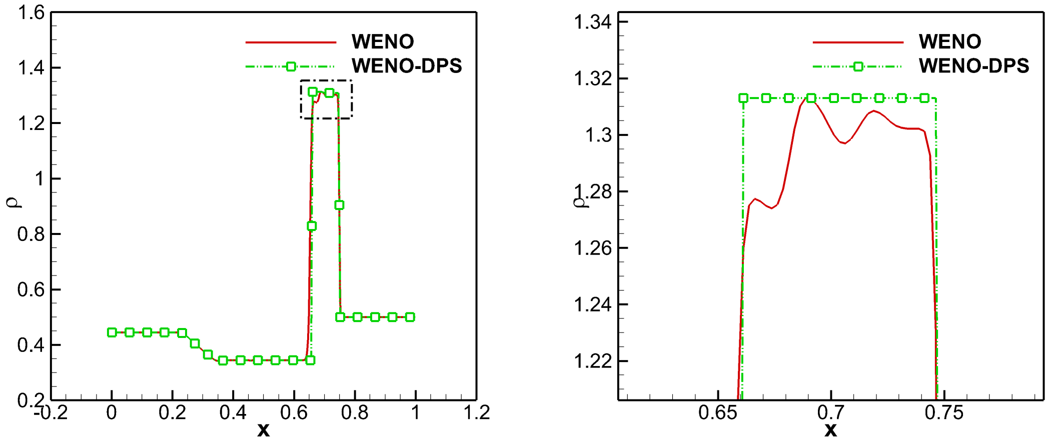
The solution to this problem using WENO, WENO-DPS and the exact solution is shown in figure 7.6. Here, WENO scheme used 13 grid points to resolve the discontinuity. In this case, WENO and WENO-DPS are unable to estimate the second shock speed accurately. It can be made better if a good Riemann solver is designed.



(a) Density plot of lax test case

(b) Zoomed view of the discontinuity

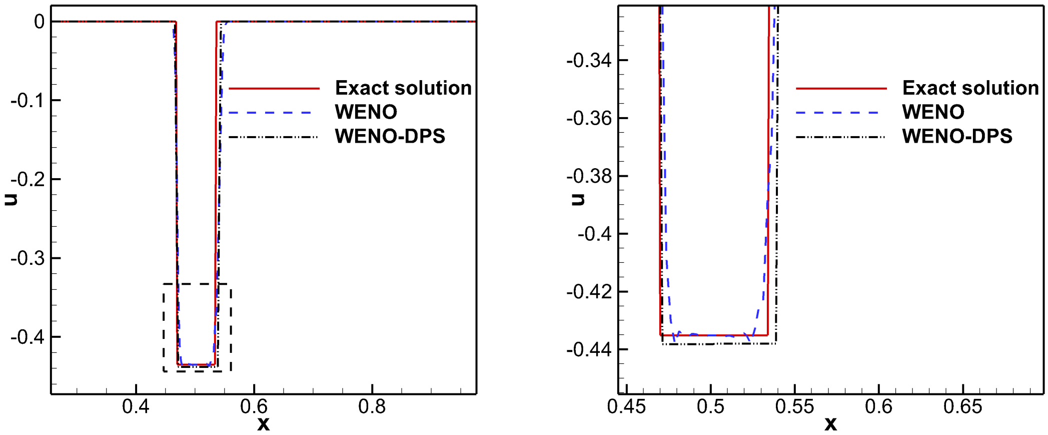
Figure 7.4: Solution of Lax test case with $s_t = 10$



(a) Density plot of lax test case

(b) Zoomed view of the discontinuity

Figure 7.5: Solution of Lax test case with $s_t = 20$



(a) Density plot of RELS test case

(b) Zoomed view of the discontinuity

Figure 7.6: Solution of RELS test case with $s_t = 10$

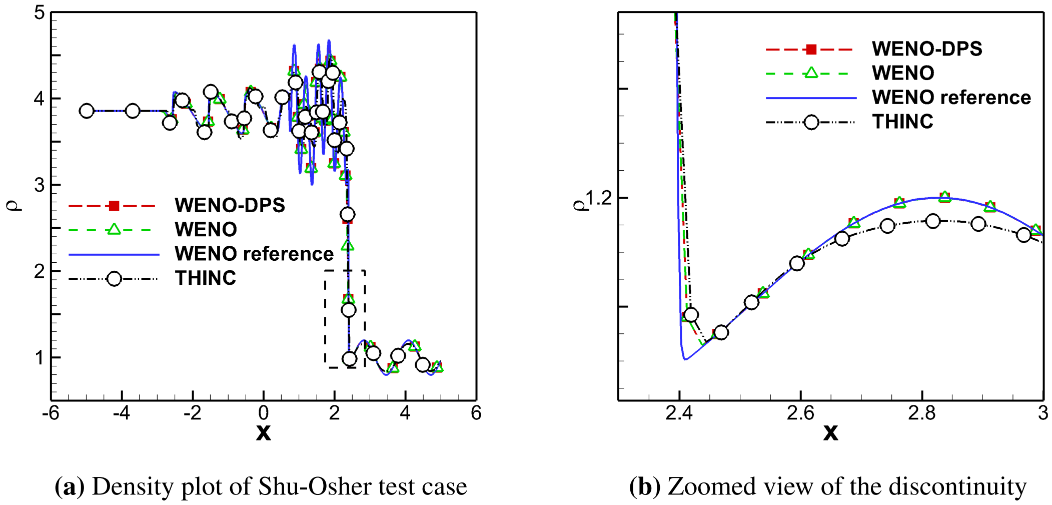
In the Sod shock tube problem, WENO-Rusanov scheme used eight points to resolve the discontinuity, but the proposed algorithm used three points. Similarly, for the Mach 3 test case, the current scheme could resolve shocks with three points, but the WENO scheme used twelve points so that the current scheme could be bundled with a SCS to avoid numerical dissipation of shock. This algorithm can result in convergence of the fifth-order in a smooth region and the third order in a non-smooth region like in the WENO scheme. This can be checked by using the Taylor series.

7.2.6 Shu-Osher Problem

Shu-Osher problem is solved using 400 grid points and solved using up to flow time 1.8 s over the domain [-5 5]. $s_t = 3$ is used for this problem. The initial condition used is:

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.33333) & x \leq -4 \\ \\ (1 + 0.2 \times sin(5x), 0, 1) & x > -4 \end{cases}$$

The density solution to this problem is shown in figure 7.7a and figure 7.7b. For this problem, WENO-DPS did not show significant improvement in the solution over WENO and THINC.



(a) Density plot of Shu-Osher test case **(b)** Zoomed view of the discontinuity

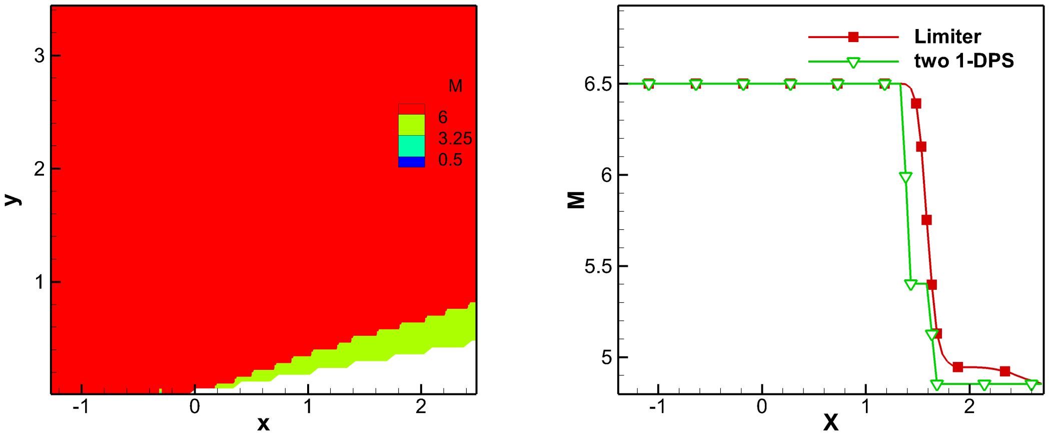
Figure 7.7: Solution of Shu-Osher test case with $s_t = 3$

7.2.7 Two-Dimensional DPS

The test case used is the flow past a wedge of angle 10 degrees. Inflow Mach number is 5. Roe scheme is used as Riemann solver. Edge base limiting strategy is followed using minmod1 limiter. Time advancing is carried using the implicit Euler method. Unstructured quadrilateral grid is used with the node-centred second-order finite-volume method. The domain $[-2, 3] \times [0, 5]$ is discretized using 100×100 quadrilateral cells. A second-order Gaussian kernel moving least square procedure is employed to obtain slope values. Normalized pressure and density are used, so the initial condition for the density and pressure is one. CFL ramping is employed for better stability at the initial transition region.

We could split the 2-dimensional problem using two 1-dimensional problems and do the DPS, but it will lead to the severe stair-case effect. Figure 7.8 shows the solution of the above test case post-processed using 1-D DPS. In this case, DPS algorithm performed worse than the standard shock capturing algorithm. We need some modifications in the present 1-D DPS algorithm, so an alternative 2-D DPS algorithm is presented. Here, the location of discontinuous is calculated using eq. 7.7 in horizontal and its extension of that to vertical directions. Once the discontinuities are located, we fit the following function using moving least square algorithm. The function used is

$$y = a \times \tanh\left[1000000(x \times tan(b) + c \times y)\right] + d \tag{7.15}$$



(a) Mach number contour using 1-D DPS on 2-D test case

(b) Mach number plot at y = 0.5 location using 1-D DPS on 2-D test case.

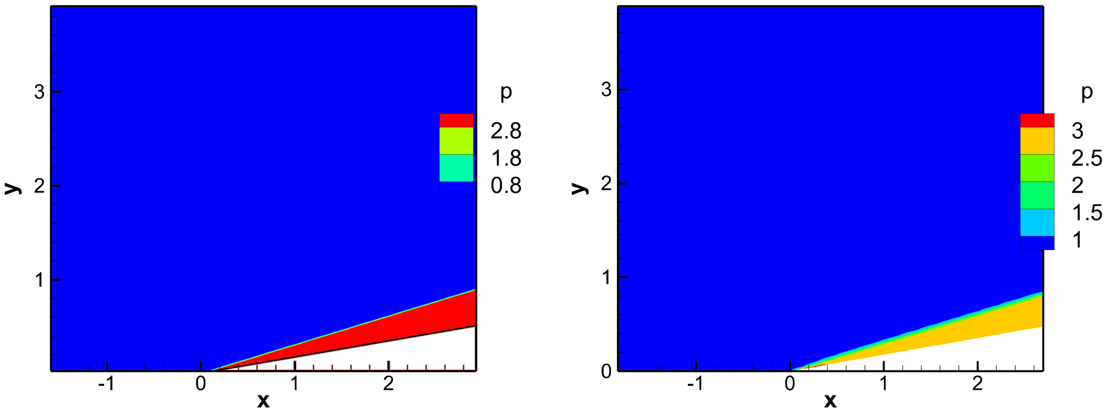
Figure 7.8: Supersonic flow past 10 degree wedge

where a, b, c and d are calculated using Levenberg-Marquardt algorithm. Figure 7.9 the pressure contour of SCS and SCS post-proceeded with 1-DSP is presented. The solution of DPS algorithm is shown in figure 7.10. Pressure plot at $y = 0.5$ of SCS-DPS is shown in figure 7.10b. In post-processing, the geometry is removed from the control volume to make the interpolation simple. In this case, DPS is able to resolve the shock with three grid points but SCS took 12 points to resolve the discontinuity.

7.3 Advantages and Disadvantages

7.3.1 Advantages

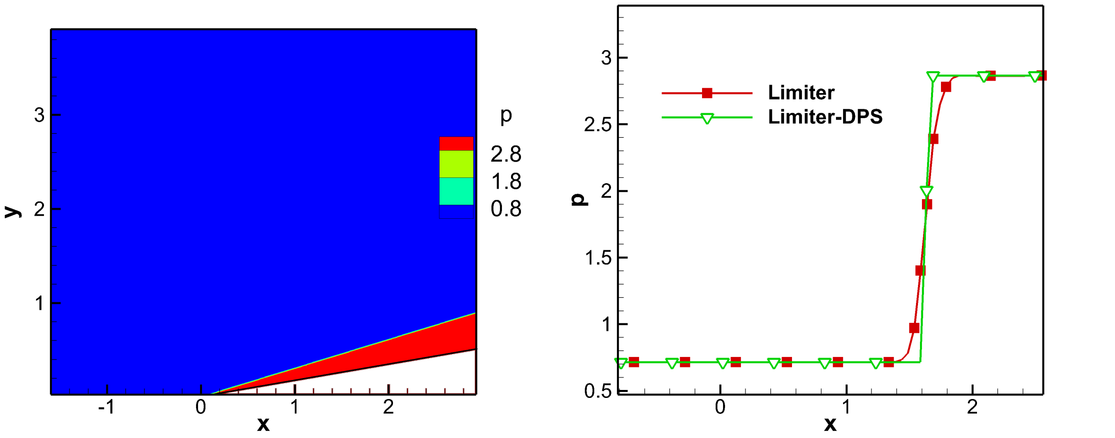
- Excellent shock resolving capacity regardless of grid size/ CFL/ Riemann solver etc.
- **No numerical dissipation** or dispersion at shocks.
- Computationally **more economical** because it can able to retain shock structure even in course grid.
- Some of the Riemann solvers may not have enough artificial viscosity to suppress oscillations that can be cured using the present approach.



(**a**) Pressure number contour using 2-D DPS on 2-D test case

(**b**) Pressure number contour using SCS on 2-D test case

Figure 7.9: Supersonic flow past 10 degree wedge



(**a**) Pressure contour using 2-D DPS on 2-D test case (**b**) Pressure plot at y = 0.5 location using 2-D DPS.

Figure 7.10: Supersonic flow past 10 degree wedge

- Well tuned DPS can produce result **matches with analytical solution**.

7.3.2 Disadvantages

- Cut-off should be tuned based on the problem, discretization and grid size.
- Not recommended for **very-diffusive schemes** (first order schemes)
- If not tuned well, it may **remove physical oscillations** present in the vicinity of the shocks.
- Will **not** work on **multi dimensional unstructured grids**.
- If grid is not aligned with the shock, it may lead to stair-case effect.

7.4 Summary

A discontinuity preserving scheme (DPS) is introduced in the present study. The aim is to preserve the discontinuities present in the solution that would otherwise be dissipated by the numerical dissipation of shock capture schemes (SCS). It is still impossible to eliminate the artificial dissipation of the Riemann solvers by using existing approaches. This algorithm is similar to the artificial dissipation method and the spatial filters used in large eddy simulation to remove specific or unwanted oscillations present in the solution.

Instead of applying DPS at each iteration, it can be applied at a few time iterations or at the end of time iteration to reduce the computational cost. For linear and non-linear test cases, the current scheme shows better shock-resolving properties than SCS. It can resolve the shock with a lesser number of grids than standard SCS. This algorithm offers a cost-effective way to solve the shocks with fewer grid points than the regular SCS. This scheme is a hybrid shock fitting shock capturing algorithm. The standard shock fitting algorithm relies on analytical expression, but the present scheme relies on numerical solution obtained using SCS. It is much simpler than the standard shock fitting technique.

8
Conclusions

In this work, higher-order schemes with the high-resolution property are explored for the hyperbolic equations. The advantages of higher-order schemes are highlighted with basic test cases and the limitations of higher-order schemes are also outlined. Importance of well stable schemes is also explained using the test cases. Different discretization procedures used in FVM are described and the procedure to obtain higher-order schemes are also presented. The importance of some sophisticated tools like reconstruction schemes and Riemann solvers are also described with the purpose. We tried to extend the higher-order conservative discretization on the non-uniform grid and we found some issues in it. The reasons and the proof for the reduction in the order of FVM on the non-uniform grid are also presented.

Three hyperbolic Runge-Kutta methods are formed by maximizing the area covered in the linear stability analysis. The optimization is carried using an evolutionary algorithm because of the non-linear nature of the area covered by the stability equation. An easily parallelizable convex optimization algorithm called multi-section method is proposed. It is suitable for problems having discontinuities in the solution. Though the HRK method is based on linear stability analysis, it is better than other schemes in terms of computational time and convergence. In addition to that storage requirements of the HRK methods are also much lower than the classical and SSPRK methods. The present schemes are up to 8 times more stable than RK2 schemes in the area covered in the linear stability analysis.

Non-linear reconstruction schemes like limiters and WENO schemes are studied in this work. Novel second and third-order slope limiters are presented. The performance of the slope limiters is studied for simple and complicated test cases involving simple and unstructured grids. The present limiters outperformed other limiters in most of the test cases. In one test case, the present algorithm is up to 400% more accurate than the classical schemes. In this work three-level order adaptive, WENO scheme is presented which can achieve fifth, fourth and third-order accuracy. WENO schemes presented in this work is up to 5 times more accurate than other schemes for certain test cases.

Discontinuity preserving algorithm is proposed in this work which can resolve the shocks using three grid points. Several existing schemes are explored and some novel conventional and non-conventional schemes are presented for hyperbolic equation having discontinuity in the solution. Several linear and non-linear test cases are studied in uniform and unstructured grids.The schemes explored here are computationally economical and having better shock resolving property for the one-dimensional and two-dimensional problems.

Bibliography

[1] P. L. Bhatnagar, E. P. Gross, and M. Krook, "A model for collision processes in gases. i. small amplitude processes in charged and neutral one-component systems," *Phys. Rev.*, vol. 94, pp. 511–525, May 1954. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRev.94.511

[2] J. J. Zipay, C. T. Modlin, and C. E. Larsen, "The ultimate factor of safety for aircraft and spacecraft-its history, applications and misconceptions," in *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2016, p. 1715. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20150003482.pdf

[3] G. B. Whitham, *Linear and nonlinear waves*. John Wiley & Sons, 2011, vol. 42.

[4] R. J. LeVeque, *Finite volume methods for hyperbolic problems*. Cambridge university press, 2002, vol. 31.

[5] G. A. Sod, "A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws," *Journal of Computational Physics*, vol. 27, no. 1, pp. 1 – 31, 1978. [Online]. Available: https://doi.org/10.1016/0021-9991(78)90023-2

[6] M. Arora and P. L. Roe, "A well-behaved tvd limiter for high-resolution calculations of unsteady flow," *Journal of Computational Physics*, vol. 132, no. 1, pp. 3 – 11, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199919695514X

[7] G.-S. Jiang and C.-W. Shu, “Efficient implementation of weighted eno schemes,” *Journal of Computational Physics*, vol. 126, no. 1, pp. 202 – 228, 1996. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999196901308

[8] H. BATEMAN, “Some recent researches on the motion of fluids,” *Monthly Weather Review*, vol. 43, no. 4, pp. 163–170, 1915. [Online]. Available: https://doi.org/10.1175/1520-0493(1915)43<163:SRROTM>2.0.CO;2

[9] S. Chandrasekhar, *On the decay of plane shock waves*. Ballistic Research Laboratories, 1943, no. 423.

[10] E. Hopf, “The partial differential equation $u_t + uu_x = \mu u_{xx}$,” *Communications on Pure and Applied Mathematics*, vol. 3, no. 3, pp. 201–230, 1950. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160030302

[11] C. Shu, “High order weighted essentially nonoscillatory schemes for convection dominated problems,” *SIAM Review*, vol. 51, no. 1, pp. 82–126, 2009. [Online]. Available: https://doi.org/10.1137/070679065

[12] C. Hirsch, *Numerical computation of internal and external flows: The fundamentals of computational fluid dynamics*. Butterworth-Heinemann, 2007.

[13] A. Jameson, “Origins and further development of the jameson–schmidt–turkel scheme,” *AIAA Journal*, pp. 1487–1510, 2017. [Online]. Available: https://doi.org/10.2514/1.J055493

[14] P. Roe, “Approximate riemann solvers, parameter vectors, and difference schemes,” *Journal of Computational Physics*, vol. 43, no. 2, pp. 357 – 372, 1981. [Online]. Available: https://doi.org/10.1016/0021-9991(81)90128-5

[15] A. Harten, “High resolution schemes for hyperbolic conservation laws,” *Journal of Computational Physics*, vol. 49, no. 3, pp. 357–393, mar 1983. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0021999183901365

[16] E. F. Toro, *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[17] M.-S. Liou and C. J. Steffen, “A new flux splitting scheme,” *Journal of Computational Physics*, vol. 107, no. 1, pp. 23 – 39, 1993. [Online]. Available: https://doi.org/10.1006/jcph.1993.1122

[18] M.-S. Liou, “A sequel to ausm: Ausm+,” *Journal of Computational Physics*, vol. 129, no. 2, pp. 364 – 382, 1996. [Online]. Available: https://doi.org/10.1006/jcph.1996.0256

[19] M. S. Liou, “A sequel to ausm, part ii: Ausm+-up for all speeds,” *Journal of Computational Physics*, vol. 214, no. 1, pp. 137 – 170, 2006. [Online]. Available: https://doi.org/10.1016/j.jcp.2005.09.020

[20] E. Shima and K. Kitamura, “Parameter-free simple low-dissipation ausm-family scheme for all speeds,” *AIAA journal*, vol. 49, no. 8, pp. 1693–1709, 2011. [Online]. Available: https://doi.org/10.2514/1.J050905

[21] J. R. Edwards and M.-S. Liou, “Low-diffusion flux-splitting methods for flows at all speeds,” *AIAA journal*, vol. 36, no. 9, pp. 1610–1617, 1998. [Online]. Available: https://doi.org/10.2514/2.587

[22] K. H. Kim, J. H. Lee, and O. H. Rho, “An improvement of ausm schemes by introducing the pressure-based weight functions,” *Computers and Fluids*, vol. 27, no. 3, pp. 311 – 346, 1998. [Online]. Available: https://doi.org/10.1016/S0045-7930(97)00069-8

[23] K. Masatsuka, *I do Like CFD, vol. 1*. Lulu. com, 2013, vol. 1. [Online]. Available: http://ossanworld.com/cfdbooks/cfdbooks.html

[24] C. B. Laney, *Computational gasdynamics*. Cambridge university press, 1998.

[25] M. Berzins and J. M. Ware, “Positive cell-centered finite volume discretization methods for hyperbolic equations on irregular meshes,” *Applied Numerical Mathematics*, 1995.

[26] J. Blazek, *Computational fluid dynamics: principles and applications*. Butterworth-Heinemann, 2015.

[27] J. Sun, K. Lee, and H. Lee, "Comparison of implicit and explicit finite element methods for dynamic problems," *Journal of Materials Processing Technology*, vol. 105, no. 1, pp. 110 – 118, 2000. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S092401360000580X

[28] J. C. Butcher, *Numerical methods for ordinary differential equations*. John Wiley & Sons, 2016.

[29] C. Runge, "Ueber die numerische auflösung von differentialgleichungen," *Mathematische Annalen*, vol. 46, no. 2, pp. 167–178, Jun 1895. [Online]. Available: https://doi.org/10.1007/BF01446807

[30] J. C. Butcher, *The numerical analysis of ordinary differential equations: Runge-Kutta and general linear methods*. Wiley-Interscience, 1987.

[31] E. Fehlberg, "Low-order classical runge-kutta formulas with stepsize control and their application to some heat transfer problems," 1969. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19690021375.pdf

[32] Erwin-Fehlberg, "Klassische runge-kutta-formeln vierter und niedrigerer ordnung mit schrittweiten-kontrolle und ihre anwendung auf waermeleitungsprobleme," *Computing*, vol. 6, no. 1, pp. 61–71, 1970. [Online]. Available: https://doi.org/10.1007/BF02241732

[33] J. R. Cash and A. H. Karp, "A variable order runge-kutta method for initial value problems with rapidly varying right-hand sides," *ACM Trans. Math. Softw.*, vol. 16, no. 3, pp. 201–222, Sep. 1990. [Online]. Available: http://doi.acm.org/10.1145/79505.79507

[34] L. Shampine and M. Reichelt, "The matlab ode suite," *SIAM Journal on Scientific Computing*, vol. 18, no. 1, pp. 1–22, 1997. [Online]. Available: https://doi.org/10.1137/S1064827594276424

[35] J. R. Dormand, *Numerical methods for differential equations: a computational approach*. CRC Press, 1996, vol. 3.

[36] U. M. Ascher, S. J. Ruuth, and R. J. Spiteri, "Implicit-explicit runge-kutta methods for time-dependent partial differential equations," *Applied Numerical Mathematics*, vol. 25, no. 2, pp. 151 – 167, 1997, special Issue on Time Integration. [Online]. Available: https://doi.org/10.1016/S0168-9274(97)00056-1

[37] R. Alexander, "Diagonally implicit runge–kutta methods for stiff o.d.e.'s," *SIAM Journal on Numerical Analysis*, vol. 14, no. 6, pp. 1006–1021, 1977. [Online]. Available: https://doi.org/10.1137/0714068

[38] X. Zhong, "Additive semi-implicit runge–kutta methods for computing high-speed nonequilibrium reactive flows," *Journal of Computational Physics*, vol. 128, no. 1, pp. 19 – 31, 1996. [Online]. Available: https://doi.org/10.1006/jcph.1996.0193

[39] L. Pareschi and G. Russo, "Implicit–explicit runge–kutta schemes and applications to hyperbolic systems with relaxation," *Journal of Scientific Computing*, vol. 25, no. 1, pp. 129–155, Oct 2005. [Online]. Available: https://doi.org/10.1007/s10915-004-4636-4

[40] S. Jin and Z. Xin, "The relaxation schemes for systems of conservation laws in arbitrary space dimensions," *Communications on Pure and Applied Mathematics*, vol. 48, no. 3, pp. 235–276. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpa.3160480303

[41] S. Jin, "Runge-kutta methods for hyperbolic conservation laws with stiff relaxation terms," *Journal of Computational Physics*, vol. 122, no. 1, pp. 51 – 67, 1995. [Online]. Available: https://doi.org/10.1006/jcph.1995.1196

[42] J. Williamson, "Low-storage runge-kutta schemes," *Journal of Computational Physics*, vol. 35, no. 1, pp. 48 – 56, 1980. [Online]. Available: https://doi.org/10.1016/0021-9991(80)90033-9

[43] M. H. Carpenter and C. A. Kennedy, "Fourth-order 2n-storage runge-kutta schemes," 1994. [Online]. Available: https://ntrs.nasa.gov/search.jsp?R=19940028444

[44] M. Calvo, J. Franco, and L. Rández, "Minimum storage runge-kutta schemes for computational acoustics," *Computers & Mathematics with Applications*, vol. 45, no. 1, pp. 535 – 545, 2003. [Online]. Available: https://doi.org/10.1016/S0898-1221(03)80035-4

[45] F. Hu, M. Hussaini, and J. Manthey, "Low-dissipation and low-dispersion runge–kutta schemes for computational acoustics," *Journal of Computational Physics*, vol. 124, no. 1, pp. 177 – 191, 1996. [Online]. Available: https://doi.org/10.1006/jcph.1996.0052

[46] F. Rabiei and F. Ismail, "New improved runge-kutta method with reducing number of function evaluations," in *International Conference on Software Technology and Engineering, 3rd (ICSTE 2011)*. ASME press, 2011. [Online]. Available: https://doi.org/10.1115/1.859797.paper14

[47] J. Verwer, "Explicit runge-kutta methods for parabolic partial differential equations," *Applied Numerical Mathematics*, vol. 22, no. 1, pp. 359 – 379, 1996, special Issue Celebrating the Centenary of Runge-Kutta Methods. [Online]. Available: https://doi.org/10.1016/S0168-9274(96)00022-0

[48] B. Cockburn and C.-W. Shu, "Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework," *Mathematics of computation*, vol. 52, no. 186, pp. 411–435, 1989. [Online]. Available: https://doi.org/10.1090/S0025-5718-1989-0983311-4

[49] S. Ruuth, "Global optimization of explicit strong-stability-preserving runge-kutta methods," *Mathematics of Computation*, vol. 75, no. 253, pp. 183–207, 2006. [Online]. Available: https://doi.org/10.1090/S0025-5718-05-01772-2

[50] D. Ketcheson, "Highly efficient strong stability-preserving runge–kutta methods with low-storage implementations," *SIAM Journal on Scientific Computing*, vol. 30, no. 4, pp. 2113–2136, 2008. [Online]. Available: https://doi.org/10.1137/07070485X

[51] C. Bogey and C. Bailly, “A family of low dispersive and low dissipative explicit schemes for flow and noise computations,” *Journal of Computational Physics*, vol. 194, no. 1, pp. 194 – 214, 2004. [Online]. Available: https://doi.org/10.1016/j.jcp.2003.09.003

[52] A. R. Appadu, “Optimized low dispersion and low dissipation runge-kutta algorithms in computational aeroacoustics,” *Appl. Math*, vol. 8, no. 1, pp. 57–68, 2014. [Online]. Available: http://dx.doi.org/10.12785/amis/080106

[53] A. G. Neelan and M. T. Nair, “Optimized dispersion relation schemes for spatial and temporal discretization,” *International Conference on Trends in Technology and Engineering (ICTTE’16)*, pp. 4–7, 2016.

[54] J. Ramboer, T. Broeckhoven, S. Smirnov, and C. Lacor, “Optimization of time integration schemes coupled to spatial discretization for use in caa applications,” *Journal of Computational Physics*, vol. 213, no. 2, pp. 777 – 802, 2006. [Online]. Available: https://doi.org/10.1016/j.jcp.2005.08.033

[55] M. K. Rajpoot, T. K. Sengupta, and P. K. Dutt, “Optimal time advancing dispersion relation preserving schemes,” *Journal of Computational Physics*, vol. 229, no. 10, pp. 3623 – 3651, 2010. [Online]. Available: https://doi.org/10.1016/j.jcp.2010.01.018

[56] D. Stanescu and W. Habashi, “2n-storage low dissipation and dispersion runge-kutta schemes for computational acoustics,” *Journal of Computational Physics*, vol. 143, no. 2, pp. 674 – 681, 1998. [Online]. Available: https://doi.org/10.1006/jcph.1998.5986

[57] J. Mead and R. Renaut, “Optimal runge–kutta methods for first order pseudospectral operators,” *Journal of Computational Physics*, vol. 152, no. 1, pp. 404 – 419, 1999. [Online]. Available: https://doi.org/10.1006/jcph.1999.6260

[58] P. J. V. D. Houwen and B. P. Sommeuer, “A Special Class of Multistep Runge—Kutta Methods with Extended Real Stability Interval,” *IMA Journal of*

Numerical Analysis, vol. 2, no. 2, pp. 183–209, 04 1982. [Online]. Available: https://dx.doi.org/10.1093/imanum/2.2.183

[59] P. J. van Der Houwen and B. P. Sommeijer, "On the internal stability of explicit, m-stage runge-kutta methods for large m-values," *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik*, vol. 60, no. 10, pp. 479–485. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/zamm.19800601005

[60] I. P. Kinnmark and W. G. Gray, "One step integration methods of third-fourth order accuracy with large hyperbolic stability limits," *Mathematics and Computers in Simulation*, vol. 26, no. 3, pp. 181 – 188, 1984. [Online]. Available: https://doi.org/10.1016/0378-4754(84)90056-9

[61] T. K. Sengupta, A. Sengupta, and K. Saurabh, "Global spectral analysis of multi-level time integration schemes: Numerical properties for error analysis," *Applied Mathematics and Computation*, vol. 304, pp. 41 – 57, 2017. [Online]. Available: https://doi.org/10.1016/j.amc.2017.01.026

[62] S. Gottlieb and L.-A. J. Gottlieb, "Strong stability preserving properties of runge–kutta time discretization methods for linear constant coefficient operators," *Journal of Scientific Computing*, vol. 18, no. 1, pp. 83–109, Feb 2003. [Online]. Available: https://doi.org/10.1023/A:1020338228736

[63] S. K. Godunov, "A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics," *Matematicheskii Sbornik*, vol. 89, no. 3, pp. 271–306, 1959.

[64] S. Gottlieb, C. Shu, and E. Tadmor, "Strong stability-preserving high-order time discretization methods," *SIAM Review*, vol. 43, no. 1, pp. 89–112, 2001. [Online]. Available: https://doi.org/10.1137/S003614450036757X

[65] P. van der Houwen, "The development of runge-kutta methods for partial differential equations," *Applied Numerical Mathematics*, vol. 20, no. 3, pp. 261 – 272, 1996. [Online]. Available: https://doi.org/10.1016/0168-9274(95)00109-3

[66] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, “A fast and elitist multiobjective genetic algorithm: Nsga-ii,” *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, April 2002. [Online]. Available: https://ieeexplore.ieee.org/document/996017

[67] A. Lipowski and D. Lipowska, “Roulette-wheel selection via stochastic acceptance,” *Physica A: Statistical Mechanics and its Applications*, vol. 391, no. 6, pp. 2193 – 2196, 2012. [Online]. Available: https://doi.org/10.1016/j.physa.2011.12.004

[68] S. Baluja and R. Caruana, “Removing the genetics from the standard genetic algorithm,” in *Machine Learning Proceedings 1995*, A. Prieditis and S. Russell, Eds. San Francisco (CA): Morgan Kaufmann, 1995, pp. 38 – 46. [Online]. Available: https://doi.org/10.1016/B978-1-55860-377-6.50014-1

[69] S. Gottlieb and C.-W. Shu, “Total variation diminishing runge-kutta schemes,” *Mathematics of computation of the American Mathematical Society*, vol. 67, no. 221, pp. 73–85, 1998. [Online]. Available: https://doi.org/10.1090/S0025-5718-98-00913-2

[70] K. Tselios and T. Simos, “Runge–kutta methods with minimal dispersion and dissipation for problems arising from computational acoustics,” *Journal of Computational and Applied Mathematics*, vol. 175, no. 1, pp. 173 – 181, 2005, selected Papers of the International Conference on Computational Methods in Sciences and Engineering. [Online]. Available: https://doi.org/10.1016/j.cam.2004.06.012

[71] E. Johnsen and T. Colonius, “Implementation of weno schemes in compressible multicomponent flow problems,” *Journal of Computational Physics*, vol. 219, no. 2, pp. 715 – 732, 2006. [Online]. Available: https://doi.org/10.1016/j.jcp.2006.04.018

[72] D. S. Balsara and C.-W. Shu, “Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy,” *Journal of*

Computational Physics, vol. 160, no. 2, pp. 405 – 452, 2000. [Online]. Available: https://doi.org/10.1006/jcph.2000.6443

[73] M. Martín, E. Taylor, M. Wu, and V. Weirs, “A bandwidth-optimized weno scheme for the effective direct numerical simulation of compressible turbulence,” *Journal of Computational Physics*, vol. 220, no. 1, pp. 270 – 289, 2006. [Online]. Available: https://doi.org/10.1016/j.jcp.2006.05.009

[74] N. K. Yamaleev and M. H. Carpenter, “Third-order energy stable weno scheme,” *Journal of Computational Physics*, vol. 228, no. 8, pp. 3025 – 3047, 2009. [Online]. Available: https://doi.org/10.1016/j.jcp.2009.01.011

[75] D. A. Barcarolo, “Improvement of the precision and the efficiency of the sph method: theoretical and numerical study,” Ph.D. dissertation, Ecole Centrale de Nantes (ECN), 2013.

[76] U. Ghia, K. Ghia, and C. Shin, “High-resolutions for incompressible flow using the navier-stokes equations and a multigrid method,” *Journal of Computational Physics*, vol. 48, no. 3, pp. 387 – 411, 1982. [Online]. Available: https://doi.org/10.1016/0021-9991(82)90058-4

[77] E. Erturk, T. C. Corke, and C. Gökçöl, “Numerical solutions of 2-d steady incompressible driven cavity flow at high reynolds numbers,” *International Journal for Numerical Methods in Fluids*, vol. 48, no. 7, pp. 747–774. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.953

[78] A. Harten, “High resolution schemes for hyperbolic conservation laws,” *Journal of Computational Physics*, vol. 49, no. 3, pp. 357 – 393, 1983. [Online]. Available: https://doi.org/10.1016/0021-9991(83)90136-5

[79] F. E. Udwadia and A. Farahani, “Accelerated runge-kutta methods,” *Discrete Dynamics in Nature and Society*, vol. 2008, 2008. [Online]. Available: http://dx.doi.org/10.1155/2008/790619

[80] A. Tikhonov and A. Samarskii, “Homogeneous difference schemes on non-uniform nets,” *USSR Computational Mathematics and Mathematical Physics*,

vol. 2, no. 5, pp. 927 – 953, 1963. [Online]. Available: https://doi.org/10.1016/0041-5553(63)90505-6

[81] R. Maccormack and A. Paullay, *Computational efficiency achieved by time splitting of finite difference operators*. [Online]. Available: https://arc.aiaa.org/doi/abs/10.2514/6.1972-154

[82] *The Computation of Transonic Flow Through Two-Dimensional Gas Turbine Cascades*, ser. Turbo Expo: Power for Land, Sea, and Air, vol. ASME 1971 International Gas Turbine Conference and Products Show, 03 1971, v001T01A089. [Online]. Available: https://doi.org/10.1115/71-GT-89

[83] R. Eymard, T. Gallouet, and R. Herbin, "Finite volume methods," in *Solution of Equation in R3 (Part 3), Techniques of Scientific Computing (Part 3)*, ser. Handbook of Numerical Analysis. Elsevier, 2000, vol. 7, pp. 713 – 1018. [Online]. Available: https://doi.org/10.1016/S1570-8659(00)07005-8

[84] B. Diskin, J. L. Thomas, E. J. Nielsen, H. Nishikawa, and J. A. White, "Comparison of node-centered and cell-centered unstructured finite-volume discretizations: Viscous fluxes," *AIAA Journal*, vol. 48, no. 7, pp. 1326–1338, 2010. [Online]. Available: https://doi.org/10.2514/1.44940

[85] B. Diskin and J. L. Thomas, "Comparison of node-centered and cell-centered unstructured finite-volume discretizations: Inviscid fluxes," *AIAA Journal*, vol. 49, no. 4, pp. 836–854, 2011. [Online]. Available: https://doi.org/10.2514/1.J050897

[86] H. Lomax, T. H. Pulliam, and D. W. Zingg, *Fundamentals of Computational Fluid Dynamics*. Springer Science & Business Media, 2013.

[87] A. Harten, B. Engquist, S. Osher, and S. R. Chakravarthy, *Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 218–290. [Online]. Available: https://doi.org/10.1007/978-3-642-60543-7_12

[88] ——, “Uniformly high order accurate essentially non-oscillatory schemes, iii,” *Journal of Computational Physics*, vol. 131, no. 1, pp. 3 – 47, 1997. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999196956326

[89] A. K. Henrick, T. D. Aslam, and J. M. Powers, “Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points,” *Journal of Computational Physics*, vol. 207, no. 2, pp. 542 – 567, 2005. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999105000409

[90] F. Juretić and A. D. Gosman, “Error analysis of the finite-volume method with respect to mesh type,” *Numerical Heat Transfer, Part B: Fundamentals*, vol. 57, no. 6, pp. 414–439, 2010. [Online]. Available: https://doi.org/10.1080/10407791003685155

[91] B. Diskin, J. L. Thomas, E. J. Nielsen, H. Nishikawa, and J. A. White, “Comparison of node-centered and cell-centered unstructured finite-volume discretizations: viscous fluxes,” *AIAA Journal*, vol. 48, no. 7, pp. 1326–1338, 2010. [Online]. Available: https://doi.org/10.2514/1.44940

[92] P. Roe, “Error estimates for cell-vertex solutions of the compressible euler equations,” 1987. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19870009101.pdf

[93] D. Bouche, J. Ghidaglia, and F. Pascal, “Error estimate and the geometric corrector for the upwind finite volume method applied to the linear advection equation,” *SIAM Journal on Numerical Analysis*, vol. 43, no. 2, pp. 578–603, 2005. [Online]. Available: https://doi.org/10.1137/040605941

[94] J. Banks, T. Aslam, and W. Rider, “On sub-linear convergence for linearly degenerate waves in capturing schemes,” *Journal of Computational Physics*, vol. 227, no. 14, pp. 6985 – 7002, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999108002088

[95] M. Shashkov, *Conservative finite-difference methods on general grids*. CRC press, 2018.

[96] F. Juretic, “Error analysis in finite volume cfd.” Imperial College London, 12 2004. [Online]. Available: http://powerlab.fsb.hr/ped/kturbo/OpenFOAM/docs/FranjoJureticPhD.pdf

[97] P. Tattersall and J. McGuirk, “Evaluation of numerical diffusion effects in viscous flow calculations,” *Computers and Fluids*, vol. 23, no. 1, pp. 177 – 209, 1994. [Online]. Available: https://doi.org/10.1016/0045-7930(94)90034-5

[98] J. J. Mcguirk and W. Rodi, “A depth-averaged mathematical model for the near field of side discharges into open-channel flow,” *Journal of Fluid Mechanics*, vol. 86, no. 4, p. 761–781, 1978.

[99] M. J. Berger and A. Jameson, *An adaptive multigrid method for the euler equations*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1985, pp. 92–97. [Online]. Available: https://doi.org/10.1007/3-540-13917-6_115

[100] D. C. Haworth, S. H. El Tahry, and M. S. Huebler, “A global approach to error estimation and physical diagnostics in multidimensional computational fluid dynamics,” *International Journal for Numerical Methods in Fluids*, vol. 17, no. 1, pp. 75–97, 1993. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1650170106

[101] P. Dawkins. (2019) Paul’s online math notes and tutorials. [Online]. Available: http://tutorial.math.lamar.edu/Classes/CalcII/TaylorSeries.aspx

[102] K. Hoffmann and S. Chiang, *Computational Fluid Dynamics*, ser. Computational Fluid Dynamics, no. v. 1. [Online]. Available: https://books.google.co.in/books?id=98gjAAAACAAJ

[103] A. A. G. Neelan and M. T. Nair, “Hybrid finite difference-finite volume schemes on non-uniform grid,” in *Applications of Fluid Dynamics*, M. Singh, B. Kushvah, G. Seth, and J. Prakash, Eds. Singapore: Springer Singapore, 2018, pp. 329–340. [Online]. Available: https://link.springer.com/chapter/10.1007/978-981-10-5329-0_24

[104] J. Berland, C. Bogey, and C. Bailly, "Low-dissipation and low-dispersion fourth-order Runge-Kutta algorithm," *Computers and Fluids*, 2006.

[105] A. G. Neelan and M. Nair, "Hyperbolic runge–kutta method using evolutionary algorithm," *Journal of Computational and Nonlinear Dynamics*, vol. 13, no. 11, 2018. [Online]. Available: https://doi.org/10.1115/1.4040708

[106] B. van Leer, "Towards the ultimate conservative difference scheme. ii. monotonicity and conservation combined in a second-order scheme," *Journal of Computational Physics*, vol. 14, no. 4, pp. 361 – 370, 1974. [Online]. Available: https://doi.org/10.1016/0021-9991(74)90019-9

[107] P. K. Sweby, "High resolution schemes using flux limiters for hyperbolic conservation laws," *SIAM Journal on Numerical Analysis*, 1984.

[108] B. van Leer, "Towards the ultimate conservative difference scheme. v. a second-order sequel to godunov's method," *Journal of Computational Physics*, vol. 32, no. 1, pp. 101 – 136, 1979. [Online]. Available: https://doi.org/10.1016/0021-9991(79)90145-1

[109] P. Colella and P. R. Woodward, "The Piecewise Parabolic Method (PPM) for gas-dynamical simulations," 1984.

[110] A. Suresh and H. T. Huynh, "Accurate monotonicity-preserving schemes with Runge-Kutta time stepping," *Journal of Computational Physics*, 1997.

[111] A. Marquina, "Local piecewise hyperbolic reconstruction of numerical fluxes for nonlinear scalar conservation laws," *SIAM J. ScI. Ccompt.*, 1994.

[112] N. P. Waterson and H. Deconinck, "Design principles for bounded higher-order convection schemes - a unified approach," *Journal of Computational Physics*, 2007.

[113] B. V. Leer, "Upwind and high-resolution methods for compressible flow: from donor cell to residual-distribution schemes," *Communications in Computational Physics*, 2006. [Online]. Available: https://doi.org/10.2514/6.2003-3559

[114] R. F. Warming and R. M. Beam, "Upwind second-order difference schemes and applications in aerodynamic flows," *AIAA Journal*, 1976. [Online]. Available: https://doi.org/10.2514/3.61457

[115] J. E. Fromm. [Online]. Available: https://doi.org/10.1016/0021-9991(68)90015-6

[116] R. Agarwal, "A third-order-accurate upwind scheme for navier–stokes solutions at high reynolds numbers," 1981. [Online]. Available: https://doi.org/10.2514/6.1981-112

[117] B. P. Leonard, "The QUICK algorithm - A uniformly third-order finite-difference method for highly convective flows," pp. 159–195, 1980.

[118] K. Friedemann, "A comparative study of tvd-limiters—well-known limiters and an introduction of new ones," *International Journal for Numerical Methods in Fluids*, vol. 67, no. 4, pp. 404–440, 2011. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.2357

[119] P. Roe and M. Baines, "Algorithms for advection and shock problems," in *Numerical Methods in Fluid Mechanics*, 1982, pp. 281–290.

[120] P. L. Roe, "Some contributions to the modelling of discontinuous flows," in *Large-scale computations in fluid mechanics*, 1985, pp. 163–193.

[121] B. V. Leer, "Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection," *Journal of Computational Physics*, vol. 23, no. 3, pp. 276 – 299, 1977. [Online]. Available: https://doi.org/10.1016/0021-9991(77)90095-X

[122] G. D. van Albada, B. van Leer, and W. W. Roberts, "A comparative study of computational methods in cosmic gas dynamics," pp. 95–103, 1997. [Online]. Available: https://doi.org/10.1007/978-3-642-60543-7_6

[123] G. Zhou, *Numerical simulations of physical discontinuities in single and multi-fluid flows for arbitrary Mach numbers*. Chalmers University of Technology, 1995.

[124] N. Waterson and H. Deconinck, “Design principles for bounded higher-order convection schemes – a unified approach,” *Journal of Computational Physics*, vol. 224, no. 1, pp. 182 – 207, 2007, special Issue Dedicated to Professor Piet Wesseling on the occasion of his retirement from Delft University of Technology. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199910700040X

[125] S. Jerez and M. Arciga, “Switch flux limiter method for viscous and nonviscous conservation laws,” *Applied Mathematics and Computation*, vol. 246, pp. 292 – 305, 2014. [Online]. Available: https://doi.org/10.1016/j.amc.2014.08.011

[126] J. Wu, Y.-y. He, G.-h. Ding, and Y.-y. Han, “Hybrid optimized low-dissipation and adaptive muscl reconstruction technique for hyperbolic conservation laws,” *Journal of Scientific Computing*, Apr 2018. [Online]. Available: https://doi.org/10.1007/s10915-018-0717-7

[127] H. Lin and C. Chieng, “Characteristic based flux limiters of an essentially third-order flux splitting method for hyperbolic conservation laws,” *International Journal for Numerical Methods in Fluids*, vol. 13, no. 3, pp. 287–307. [Online]. Available: onlinelibrary.wiley.com/doi/abs/10.1002/fld.1650130303

[128] D. Ray, P. Chandrashekar, U. S. Fjordholm, and S. Mishra, “Entropy stable scheme on two-dimensional unstructured grids for euler equations,” *Communications in Computational Physics*, vol. 19, no. 5, p. 1111–1140, 2016. [Online]. Available: https://doi.org/10.4208/cicp.scpde14.43s

[129] B. Koren, *A robust upwind discretization method for advection, diffusion and source terms*. Centrum voor Wiskunde en Informatica Amsterdam, 1993.

[130] S. Chatkravathy, “High resolution applications of the osher upwind scheme for the euler equations, aiaa paper 83-1943,” in *Proc. AIAA 6th Computational Fluid Dynamics Conference*, 1983, pp. 363–373. [Online]. Available: doi.org/10.2514/6.1983-1943

[131] P. H. Gaskell and A. K. C. Lau, “Curvature-compensated convective transport: Smart, a new boundedness- preserving transport algorithm,” *International Journal*

for Numerical Methods in Fluids, vol. 8, no. 6, pp. 617–641. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1650080602

[132] F. S. Lien and M. A. Leschziner, "Upstream monotonic interpolation for scalar transport with application to complex turbulent flows," *International Journal for Numerical Methods in Fluids*, 1994. [Online]. Available: https://doi.org/10.1002/fld.1650190606

[133] V. Venkatakrishnan, "Convergence to steady state solutions of the euler equations on unstructured grids with limiters," *Journal of Computational Physics*, vol. 118, no. 1, pp. 120 – 130, 1995. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999185710844

[134] J. Burguete and P. García-Navarro, "Efficient construction of high-resolution TVD conservative schemes for equations with source terms: Application to shallow water flows," *International Journal for Numerical Methods in Fluids*, 2001. [Online]. Available: https://doi.org/10.1002/fld.175

[135] J. B. Goodman and R. J. LeVeque, "On the accuracy of stable schemes for 2d scalar conservation laws," *Mathematics of Computation*, 1985. [Online]. Available: https://www.jstor.org/stable/2008046

[136] N. M. W. Poe and D. Keith Walters, "A nonlocal convective flux limiter for upwind-biased finite volume simulations," *International Journal for Numerical Methods in Fluids*, 2012. [Online]. Available: https://doi.org/10.1002/fld.2733

[137] E. S. de Góes Maciel and C. R. de Andrade, "Comparison among unstructured tvd, eno and uno schemes in two- and three-dimensions," *Applied Mathematics and Computation*, vol. 321, pp. 130 – 175, 2018. [Online]. Available: https://doi.org/10.1016/j.amc.2017.10.026

[138] R. Kumar and M. K. Kadalbajoo, "Efficient high-resolution relaxation schemes for hyperbolic systems of conservation laws," *International Journal for Numerical Methods in Fluids*, vol. 55, no. 5, pp. 483–507. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.1479

[139] M. Seaıd, A. Klar, and B. Dubroca, “Flux limiters in the coupling of radiation and hydrodynamic models,” *Journal of computational and applied mathematics*, vol. 168, no. 1-2, pp. 425–435, 2004. [Online]. Available: https://doi.org/10.1016/j.cam.2003.06.014

[140] F. Safarzadeh Maleki and A. A. Khan, “Effect of channel shape on selection of time marching scheme in the discontinuous galerkin method for 1-d open channel flow,” *Journal of Hydrodynamics*, vol. 27, no. 3, pp. 413–426, Jun 2015. [Online]. Available: https://doi.org/10.1016/S1001-6058(15)60499-1

[141] M. Kadalbajoo and R. Kumar, “A high resolution total variation diminishing scheme for hyperbolic conservation law and related problems,” *Applied Mathematics and Computation*, vol. 175, no. 2, pp. 1556 – 1573, 2006. [Online]. Available: https://doi.org/10.1016/j.amc.2005.09.006

[142] S. Zalesak, “A preliminary comparison of modern shock-capturing schemes: linear advection,” *Advances in Computer Methods for PDEs. Publ. IMACS*, pp. 15–22, 1987.

[143] S. F. Davis, “Flux difference splittings and limiters for the resolution of contact discontinuities,” *Applied Mathematics and Computation*, vol. 65, no. 1, pp. 3 – 18, 1994. [Online]. Available: https://doi.org/10.1016/0096-3003(94)90162-7

[144] S. J. Galiano and M. U. Zapata, “A new tvd flux-limiter method for solving nonlinear hyperbolic equations,” *Journal of Computational and Applied Mathematics*, vol. 234, no. 5, pp. 1395 – 1403, 2010. [Online]. Available: https://doi.org/10.1016/j.cam.2010.02.015

[145] R. K. Dubey, “Flux limited schemes: Their classification and accuracy based on total variation stability regions,” *Applied Mathematics and Computation*, vol. 224, pp. 325 – 336, 2013. [Online]. Available: https://doi.org/10.1016/j.amc.2013.08.027

[146] M. N. Koleva and L. G. Vulkov, “A positive flux limited difference scheme for the uncertain correlation 2d black–scholes problem,” *Journal of*

Computational and Applied Mathematics, vol. 293, pp. 112 – 127, 2016, efficient Numerical Methods for Large-scale Scientific Computations. [Online]. Available: https://doi.org/10.1016/j.cam.2015.02.054

[147] D. Kuzmin, "A vertex-based hierarchical slope limiter for p-adaptive discontinuous galerkin methods," *Journal of Computational and Applied Mathematics*, vol. 233, no. 12, pp. 3077 – 3085, 2010, finite Element Methods in Engineering and Science (FEMTEC 2009). [Online]. Available: https://doi.org/10.1016/j.cam.2009.05.028

[148] D. Kuzmin and M. Möller, "Goal-oriented mesh adaptation for flux-limited approximations to steady hyperbolic problems," *Journal of Computational and Applied Mathematics*, vol. 233, no. 12, pp. 3113 – 3120, 2010, finite Element Methods in Engineering and Science (FEMTEC 2009). [Online]. Available: https://doi.org/10.1016/j.cam.2009.07.026

[149] X.-D. Liu and S. Osher, "Convex eno high order multi-dimensional schemes without field by field decomposition or staggered grids," *Journal of Computational Physics*, vol. 142, no. 2, pp. 304 – 330, 1998. [Online]. Available: https://doi.org/10.1006/jcph.1998.5937

[150] M. Čada and M. Torrilhon, "Compact third-order limiter functions for finite volume methods," *Journal of Computational Physics*, vol. 228, no. 11, pp. 4118 – 4145, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999109000953

[151] P. Roe, "Approximate riemann solvers, parameter vectors, and difference schemes," *Journal of Computational Physics*, vol. 43, no. 2, pp. 357 – 372, 1981. [Online]. Available: http://www.sciencedirect.com/science/article/pii/0021999181901285

[152] A. E. Bryson Jr, "An experimental investigation of transonic flow past two-dimensional wedge and circular-arc sections using a mach-zehnder interferometer," National Aeronautics and Space Administration, Langley Research Center, Tech. Rep., 1952. [Online]. Available: https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930092136.pdf

[153] P. Roe, "Characteristic-Based Schemes for the Euler Equations," *Annual Review of Fluid Mechanics*, 1986. [Online]. Available: https://doi.org/10.1146/annurev.fl.18.010186.002005

[154] X.-D. Liu, S. Osher, and T. Chan, "Weighted essentially non-oscillatory schemes," *Journal of Computational Physics*, vol. 115, no. 1, pp. 200 – 212, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999184711879

[155] C.-W. Shu and S. Osher, *Efficient Implementation of Essentially Non-oscillatory Shock-Capturing Schemes, II.* Berlin, Heidelberg: Springer Berlin Heidelberg, 1997, pp. 328–374. [Online]. Available: https://doi.org/10.1007/978-3-642-60543-7_14

[156] A. Kundu and S. De, "Navier–stokes simulation of shock-heavy bubble interaction: Comparison of upwind and weno schemes," *Computers & Fluids*, vol. 157, pp. 131–145, 2017. [Online]. Available: https://doi.org/10.1016/j.compfluid.2017.08.025

[157] S. Rathan *et al.*, "An improved non-linear weights for seventh-order weighted essentially non-oscillatory scheme," *Computers & Fluids*, vol. 156, pp. 496–514, 2017. [Online]. Available: https://doi.org/10.1016/j.compfluid.2017.08.023

[158] V. U. S, B. Zang, and T. New, "Adaptive mapping for high order weno methods," *Journal of Computational Physics*, vol. 381, pp. 162 – 188, 2019. [Online]. Available: https://doi.org/10.1016/j.jcp.2018.12.034

[159] A. Peer, M. Dauhoo, and M. Bhuruth, "A method for improving the performance of the weno5 scheme near discontinuities," *Applied Mathematics Letters*, vol. 22, no. 11, pp. 1730 – 1733, 2009. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0893965909002250

[160] S. Rathan and G. N. Raju, "Improved weighted eno scheme based on parameters involved in nonlinear weights," *Applied Mathematics and Computation*, vol. 331, pp. 120 – 129, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0096300318302078

[161] R. Borges, M. Carmona, B. Costa, and W. S. Don, “An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws,” *Journal of Computational Physics*, vol. 227, no. 6, pp. 3191 – 3211, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999107005232

[162] E. M. Taylor, M. Wu, and M. P. Martín, “Optimization of nonlinear error for weighted essentially non-oscillatory methods in direct numerical simulations of compressible turbulence,” *Journal of Computational Physics*, vol. 223, no. 1, pp. 384 – 397, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999106004426

[163] L. Fu, X. Y. Hu, and N. A. Adams, “A family of high-order targeted eno schemes for compressible-fluid simulations,” *Journal of Computational Physics*, vol. 305, pp. 333 – 359, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999115007147

[164] S. Serna and J. Qian, “Fifth-order weighted power-eno schemes for hamilton-jacobi equations,” *Journal of Scientific Computing*, vol. 29, no. 1, pp. 57–81, Oct 2006. [Online]. Available: https://doi.org/10.1007/s10915-005-9015-2

[165] H. Li, S. Do, and M. Kang, “A wavelet-based adaptive weno algorithm for euler equations,” *Computers & Fluids*, vol. 123, pp. 10–22, 2015. [Online]. Available: https://doi.org/10.1016/j.compfluid.2015.09.005

[166] D. S. Balsara, S. Garain, and C.-W. Shu, “An efficient class of weno schemes with adaptive order,” *Journal of Computational Physics*, vol. 326, pp. 780 – 804, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999116304211

[167] H. Luo, J. D. Baum, and R. Löhner, “A hermite weno-based limiter for discontinuous galerkin method on unstructured grids,” *Journal of Computational Physics*, vol. 225, no. 1, pp. 686 – 713, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999106006164

[168] U. S. Fjordholm and D. Ray, "A sign preserving weno reconstruction method," *Journal of Scientific Computing*, vol. 68, no. 1, pp. 42–63, Jul 2016. [Online]. Available: https://doi.org/10.1007/s10915-015-0128-y

[169] D. Avesani, M. Dumbser, and A. Bellin, "A new class of moving-least-squares weno–sph schemes," *Journal of Computational Physics*, vol. 270, pp. 278 – 299, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999114002289

[170] J. Shi, C. Hu, and C.-W. Shu, "A technique of treating negative weights in weno schemes," *Journal of Computational Physics*, vol. 175, no. 1, pp. 108 – 127, 2002. [Online]. Available: https://doi.org/10.1006/jcph.2001.6892

[171] A. Salih and S. G. Moulic, "A level set formulation for the numerical simulation of impact of surge fronts," *Sadhana*, vol. 31, no. 6, pp. 697–707, Dec 2006. [Online]. Available: https://doi.org/10.1007/BF02716889

[172] R. Wang and R. J. Spiteri, "Linear instability of the fifth-order weno method," *SIAM Journal on Numerical Analysis*, vol. 45, no. 5, pp. 1871–1901, 2007. [Online]. Available: 0.1137/050637868

[173] A. R. Appadu and A. A. I. Peer, "Optimized weighted essentially nonoscillatory third-order schemes for hyperbolic conservation laws," *Journal of Applied Mathematics*, vol. 2013, 2013. [Online]. Available: http://dx.doi.org/10.1155/2013/428681

[174] V. G. Weirs and G. V. Candler, "Optimization of weighted eno schemes for dns of compressible turbulence," *UMSI research report/University of Minnesota (Minneapolis, Mn). Supercomputer institute*, vol. 97, p. 177, 1997. [Online]. Available: 10.2514/6.1997-1940

[175] G. Kumar, S. Girimaji, and J. Kerimo, "Weno-enhanced gas-kinetic scheme for direct simulations of compressible transition and turbulence," *Journal of Computational Physics*, vol. 234, pp. 499 – 523, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999112005943

[176] X. Hu, Q. Wang, and N. Adams, “An adaptive central-upwind weighted essentially non-oscillatory scheme,” *Journal of Computational Physics*, vol. 229, no. 23, pp. 8952 – 8965, 2010. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999110004560

[177] A. Baeza, R. Bürger, P. Mulet, and D. Zorío, “Central weno schemes through a global average weight,” *Journal of Scientific Computing*, vol. 78, no. 1, pp. 499–530, Jan 2019. [Online]. Available: https://doi.org/10.1007/s10915-018-0773-z

[178] D. Hill and D. Pullin, “Hybrid tuned center-difference-weno method for large eddy simulations in the presence of strong shocks,” *Journal of Computational Physics*, vol. 194, no. 2, pp. 435 – 450, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199910300490X

[179] Z.-S. Sun, Y.-X. Ren, C. Larricq, S. ying Zhang, and Y. cheng Yang, “A class of finite difference schemes with low dispersion and controllable dissipation for dns of compressible turbulence,” *Journal of Computational Physics*, vol. 230, no. 12, pp. 4616 – 4635, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999111001276

[180] P. Castonguay, D. Williams, P. Vincent, and A. Jameson, “Energy stable flux reconstruction schemes for advection–diffusion problems,” *Computer Methods in Applied Mechanics and Engineering*, vol. 267, pp. 400 – 417, 2013. [Online]. Available: https://doi.org/10.1016/j.cma.2013.08.012

[181] T. C. Fisher, M. H. Carpenter, N. K. Yamaleev, and S. H. Frankel, “Boundary closures for fourth-order energy stable weighted essentially non-oscillatory finite-difference schemes,” *Journal of Computational Physics*, vol. 230, no. 10, pp. 3727 – 3752, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999111000684

[182] B. Biswas and R. K. Dubey, “Low dissipative entropy stable schemes using third order weno and tvd reconstructions,” *Advances in Computational Mathematics*, Dec 2017. [Online]. Available: doi.org/10.1007/s10444-017-9576-2

[183] R. K. Dubey and B. Biswas, “Suitable diffusion for constructing non-oscillatory entropy stable schemes,” *Journal of Computational Physics*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999118302626

[184] J. Zhu, J. Qiu, C.-W. Shu, and M. Dumbser, “Runge–kutta discontinuous galerkin method using weno limiters ii: Unstructured meshes,” *Journal of Computational Physics*, vol. 227, no. 9, pp. 4330 – 4353, 2008. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999108000120

[185] R. Abgrall, “On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation,” *Journal of Computational Physics*, vol. 114, no. 1, pp. 45 – 58, 1994. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S002199918471148X

[186] M. Dumbser and M. Käser, “Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems,” *Journal of Computational Physics*, vol. 221, no. 2, pp. 693 – 723, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999106003123

[187] M. P. Ray, B. P. Puranik, and U. V. Bhandakar, “Develpment and assessment of several high-resolution schemes for compressible euler equations,” *International Journal of Computational Methods*, vol. 11, no. 01, p. 1350049, 2014. [Online]. Available: https://www.worldscientific.com/doi/abs/10.1142/S0219876213500497

[188] J. Peng, C. Zhai, G. Ni, H. Yong, and Y. Shen, “An adaptive characteristic-wise reconstruction weno-z scheme for gas dynamic euler equations,” *Computers and Fluids*, vol. 179, pp. 34 – 51, 2019. [Online]. Available: https://doi.org/10.1016/j.compfluid.2018.08.008

[189] J. Peng and Y. Shen, “A novel weighting switch function for uniformly high-order hybrid shock-capturing schemes,” *International Journal for Numerical Methods in Fluids*, vol. 83, no. 9, pp. 681–703, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/fld.4285

[190] R. Bürger and A. Kozakevicius, "Adaptive multiresolution weno schemes for multi-species kinematic flow models," *Journal of Computational Physics*, vol. 224, no. 2, pp. 1190 – 1222, 2007. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999106005729

[191] R. Bürger, R. Donat, P. Mulet, and C. A. Vega, "On the implementation of weno schemes for a class of polydisperse sedimentation models," *Journal of Computational Physics*, vol. 230, no. 6, pp. 2322 – 2344, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999110006832

[192] A. G. Neelan and M. T. Nair, "Order optimized eno and b-eno scheme for aero-acoustic problems," in *Proceedings of the 6th International and 43rd National Conference on Fluid Mechanics and Fluid Power*, 2016. [Online]. Available: https://www.researchgate.net/publication/320034526_Order_Optimized_ENO_and_Bandwidth_Optimized_ENO_Scheme_for_Aero-Acoustic_Problems

[193] Y. Shen and G. Zha, "Improvement of weighted essentially non-oscillatory schemes near discontinuities," *Computers and Fluids*, vol. 96, pp. 1 – 9, 2014. [Online]. Available: https://doi.org/10.1016/j.compfluid.2014.02.010

[194] R. Kumar and P. Chandrashekar, "Simple smoothness indicator and multi-level adaptive order weno scheme for hyperbolic conservation laws," *Journal of Computational Physics*, vol. 375, pp. 1059 – 1090, 2018. [Online]. Available: https://doi.org/10.1016/j.jcp.2018.09.027

[195] B. S. van Lith, J. H. ten Thije Boonkkamp, and W. L. IJzerman, "Embedded weno: A design strategy to improve existing weno schemes," *Journal of Computational Physics*, vol. 330, pp. 529 – 549, 2017. [Online]. Available: https://doi.org/10.1016/j.jcp.2016.11.026

[196] Y. Ha, C. H. Kim, Y. J. Lee, and J. Yoon, "An improved weighted essentially non-oscillatory scheme with a new smoothness indicator," *Journal of Computational Physics*, vol. 232, no. 1, pp. 68 – 86, 2013. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021999112003233

[197] B.-S. Wang, P. Li, Z. Gao, and W. S. Don, "An improved fifth order alternative weno-z finite difference scheme for hyperbolic conservation laws," *Journal of Computational Physics*, 2018. [Online]. Available: https://doi.org/10.1016/j.jcp.2018.07.052

[198] W. Noh, "Errors for calculations of strong shocks using an artificial viscosity and an artificial heat flux," *Journal of Computational Physics*, vol. 72, no. 1, pp. 78 – 120, 1987. [Online]. Available: https://doi.org/10.1016/0021-9991(87)90074-X

[199] F. Kemm, "On the proper setup of the double mach reflection as a test case for the resolution of gas dynamics codes," *Computers and Fluids*, vol. 132, pp. 72 – 75, 2016. [Online]. Available: https://doi.org/10.1016/j.compfluid.2016.04.008

[200] L. I. Sedov, *Similarity and dimensional methods in mechanics*. CRC press, 1993.

[201] M. Arora and P. L. Roe, "A well-behaved TVD limiter for high-resolution calculations of unsteady flow," *Journal of Computational Physics*, 1997. [Online]. Available: https://doi.org/10.1006/jcph.1996.5514

[202] A. Marquina, "Local piecewise hyperbolic reconstruction of numerical fluxes for nonlinear scalar conservation laws," *SIAM Journal on Scientific Computing*, vol. 15, no. 4, pp. 892–915, 1994. [Online]. Available: https://doi.org/10.1137/0915054

[203] X. Deng, S. Inaba, B. Xie, K.-M. Shyue, and F. Xiao, "High fidelity discontinuity-resolving reconstruction for compressible multiphase flows with moving interfaces," *Journal of Computational Physics*, vol. 371, pp. 945 – 966, 2018. [Online]. Available: https://doi.org/10.1016/j.jcp.2018.03.036

[204] V. V. Rusanov, *Calculation of interaction of non-steady shock waves with obstacles*. NRC, Division of Mechanical Engineering, 1962.

[205] P. D. Lax and X.-D. Liu, "Solution of two-dimensional riemann problems of gas dynamics by positive schemes," *SIAM Journal on Scientific Computing*, vol. 19, no. 2, pp. 319–340, 1998. [Online]. Available: https://doi.org/10.1137/S1064827595291819