Entropy-stable finite difference and finite volume schemes for compressible flows

A Thesis

submited to the
Tata Institute of Fundamental Research, Mumbai
for the degree of **Doctor of Philosophy**
in **Mathematics**

by

Deep Ray

Centre for Applicable Mathematics
Tata Institute of Fundamental Research
Bangalore - 560065
India

February, 2017

DECLARATION

This thesis is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions.

The work was done under the guidance of Dr. Praveen Chandrashekar at the Tata Institute of Fundamental Research, Centre For Applicable Mathematics, and Prof. Dr. Siddhartha Mishra at ETH, Zurich.

Deep Ray

In our capacity as supervisors of the candidate's thesis, we certify that the above statements are true to the best of our knowledge.

Dr. Praveen Chandrashekar	**Prof. Dr. Siddhartha Mishra**
Date: March 17, 2018	Date: March 17, 2018

This thesis is dedicated to my grandfather, who showed me that there is no age bar to ignite one's passion for research.

Acknowledgements [1]

I would like to thank all the people who have contributed to making this thesis possible. I begin by thanking my advisors Praveen Chandrashekar and Siddhartha Mishra, whose patient guidance, inspiring discussions and words of encouragement enabled me to complete this crucial leg of my academic journey with a greater passion for research.

I would like to mention and thank the following people: Ulrik Fjordholm for his collaboration and for posing insightful questions that contributed to a large portion of the work presented in this thesis, Jayesh Badwaik, Himanshu Sharma and Andreas Hiltebrand for all their advice which helped me parallelize my solver, and Sandra May for her invaluable inputs to help me present my research work with greater clarity. Furthermore, I would like to thank Saumya Sinha for her critical comments on this thesis and for being a pillar of support through the years of my doctoral research.

I would like to thank my friends and colleagues at TIFR-CAM and ETH-Zurich, for enriching my life as a researcher through hours of academic discussions and all the memorable escapades that kept my spirits high.

Finally, I would like to thank my family for believing in my abilities. I would have never made it this far without their love and support.

[1] This research work benefited from the support of the AIRBUS Group Corporate Foundation Chair in Mathematics of Complex Systems, established in TIFR/ICTS, Bangalore.

Abstract

The solutions of hyperbolic systems of conservation laws are often discontinuous, and must be understood in the weak sense. Imposing additional entropy conditions ensures the uniqueness of solutions for scalar conservation laws, and provides the only non-linear energy estimates available for generic hyperbolic systems. In this thesis, we discuss the construction of high-order finite difference schemes on uniform Cartesian grids for the compressible Euler equations, that are entropy stable i.e., the schemes satisfy a discrete entropy condition. Additionally, the numerical flux is formulated such that the discrete kinetic energy evolves in a manner consistent with the continuous level dynamics. Such schemes are said to be kinetic energy preserving. The construction of high-order entropy stable schemes requires the reconstruction of variables at the cell-interfaces using a method that satisfies a sign-property. Only a handful of methods are known to satisfy this property, and we propose a third-order sign-preserving WENO-type reconstruction, which also satisfies other important monotonicity and stability properties.

Semi-discrete entropy conservative schemes can lose their ability to conserve entropy once the set of ODEs is integrated in time using a suitable time-marching strategy. Based on a Crank-Nicolson type discretization, we propose a fully-discrete entropy conservative scheme for the Euler equations, which is also kinetic energy preserving.

The inclusion of viscous terms in the Euler equations leads to the Navier-Stokes equations. Restricting the entropy framework available for the Euler equations can symmetrize the viscous fluxes of the Navier-Stokes equations. Based on this idea, we propose a suitable discretization of the viscous fluxes of multidimensional Navier-Stokes equations on Cartesian grids, which leads to kinetic energy preserving and entropy stable finite difference schemes. The schemes are used for direct numerical simulations of the viscous Taylor-Green vortex, to test their performance in approximating turbulent flows when the mesh is under-resolved.

We also consider the vector-invariant formulation of the shallow water equations with rotation due to Coriolis forces, which is a popular model in the meteorological community. A high-order energy preserving finite difference scheme is proposed for this model, where the (absolute) vorticity is solved for directly as an independent variable, with the aim to accurately approximate several associated integral invariants.

Finally, finite volume methods can be easily implemented on unstructured grids, which makes them useful for problems involving complex domains. Thus, we propose a vertex-centered finite volume scheme for the Navier-Stokes equations on triangular grids, which is entropy stable at the semi-discrete level. This is achieved by using a high-resolution entropy stable inviscid flux and discretizing the symmetric form of the viscous fluxes written in terms of the entropy variables. Wall boundary conditions are also constructed to be entropy stable and are imposed in a weak manner.

Notations

We use the following notations throughout this thesis.

We primarily consider systems of equations, with vectors and matrices denoted by bold letters. For instance $\mathbf{U} \in \mathbb{R}^m$ and $\mathbf{D} \in \mathbb{R}^{m \times n}$, where $\mathbb{R}$ denotes the real line. Additionally, $\mathbb{R}_+$ is used to denote the non-negative real axis. The components of vectors are denoted by subscripts, for example $\mathbf{x} = (x_1, x_2, ..., x_d)$ or superscripts. The latter is generally used when subscripts are used to represent discretized solution values at mesh points, for instance $V_i^{(3)}$ denotes the value of the third component of the vector $\mathbf{V}$ at the mesh point i. We may also use the notations V^x, V^y to denote the components of a two-dimensional vector $\mathbf{V}$. Scalar product between vectors $\mathbf{U}, \mathbf{V} \in \mathbb{R}^m$ is written using angular brackets as $\langle \mathbf{U}, \mathbf{V} \rangle$, while the induced norm is denoted by $|\mathbf{U}| := \sqrt{\langle \mathbf{U}, \mathbf{U} \rangle}$. The dyadic/outer product of $\mathbf{X} \in \mathbb{R}^n, \mathbf{Y} \in \mathbb{R}^m$ is denoted by $(\mathbf{X} \otimes \mathbf{Y})_{i,j} := \mathbf{X}_i \mathbf{Y}_j$. In certain situations, we consider quantities like $\mathbf{F} = (\mathbf{F}_1, \mathbf{F}_2, ..., \mathbf{F}_d)$, where each $\mathbf{F}_i \in \mathbb{R}^m$ is a state vector while d corresponds to the space-dimension. Then, for a vector $\mathbf{x} \in \mathbb{R}^d$, we define the operator $\mathbf{F} \cdot \mathbf{x} := \sum_{i=1}^d \mathbf{F}_i x_i$ to denote the product corresponding to the spatial dimension, with $\mathbf{F} \cdot \mathbf{x} \in \mathbb{R}^m$.

For a function $f(\mathbf{x}, t)$ depending on space-time variables $(\mathbf{x}, t) \in \mathbb{R}^d \times \mathbb{R}_+$, the partial derivatives are written as $\partial_t f, \partial_{x_1} f$, etc. An equivalent notation $\partial_t f \equiv f_t$ may also be used. Additionally, for $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, ...\alpha_d)$ denoting a multi-index, we define $\mathbf{x}^{\boldsymbol{\alpha}} := x_1^{\alpha_1} x_2^{\alpha_2} ... x_d^{\alpha_d}$ and $\partial^{\boldsymbol{\alpha}} := \partial_{x_1}^{\alpha_1} \partial_{x_2}^{\alpha_2} ... \partial_{x_d}^{\alpha_d}$. The gradient of a scalar function $f(\mathbf{x}, t)$ is denoted as $\nabla_{\mathbf{x}} f$, with the subscript $\mathbf{x}$ omitted whenever it is clear which independent variable is being referred to. The divergence of a vector $\mathbf{U}(\mathbf{x}, t) \in \mathbb{R}^d$ with respect to $\mathbf{x}$ is denoted by $\nabla \cdot \mathbf{U}$. For a scalar function $\eta(\mathbf{U})$ depending on the vector $\mathbf{U} \in \mathbb{R}^d$, we define the gradient and Hessian of η by the equivalent notations $\eta_{\mathbf{U}} \equiv \partial_{\mathbf{U}} \eta \equiv \eta'$ and $\eta_{\mathbf{U}\mathbf{U}} \equiv \partial_{\mathbf{U}\mathbf{U}} \eta \equiv \eta''$ respectively. For vector-valued functions $\mathbf{F}(\mathbf{U})$, we denote the Jacobian as $\mathbf{F}_{\mathbf{U}} \equiv \partial_{\mathbf{U}} \mathbf{F} \equiv \mathbf{F}'$.

Contents

Abstract **v**

Notations **vii**

1 Introduction **1**

2 Hyperbolic conservation laws **7**

2.1 Weak formulation . . . 8

2.2 Entropy conditions . . . 8

2.3 The Riemann problem . . . 10

2.4 Entropy measure-valued solutions . . . 11

2.5 Examples . . . 13

2.5.1 Linear advection equation . . . 13

2.5.2 Burgers' equation . . . 14

2.5.3 Wave equation . . . 14

2.5.4 Shallow water equations . . . 15

3 Compressible flows **17**

3.1 Euler equations . . . 17

3.1.1 Hyperbolicity . . . 18

3.1.2 Entropy framework . . . 18

3.2 Navier-Stokes equations . . . 19

3.2.1 Symmetrization of viscous fluxes . . . 20

3.3 Evolution of kinetic energy . . . 20

3.4 Initial boundary value problem . . . 21

3.4.1 Boundary conditions and entropy stability . . . 22

4 Finite difference and finite volume schemes for conservation laws **25**

4.1 Finite volume scheme . . . 25

4.1.1 Fully discrete scheme . . . 25

4.1.2 Godunov's exact Riemann solver . . . 26

4.1.3 Roe's approximate Riemann solver . . . 27

4.1.4 Reconstruction and limiters . . . 29

4.1.5 Method of lines approach . . . 31

4.1.6 A general reconstruction strategy . . . 32

4.1.7 ENO reconstruction . . . 33

4.1.8 WENO reconstruction . . . 34

4.2 Finite difference scheme 34

5 Entropy stable finite difference schemes 37

5.1 Entropy conservative schemes 37

5.2 Kinetic energy preserving for Euler equations 39

5.3 Numerical fluxes for the Euler equations 40

5.3.1 Roe's entropy conservative flux 40

5.3.2 Jameson's kinetic energy preserving flux 40

5.3.3 Kinetic energy and entropy conservative flux 41

5.4 High-order entropy conservative fluxes 41

5.5 Entropy stable schemes 42

5.5.1 Dissipation operator 43

5.5.2 High-order diffusion operators 44

5.5.3 Reconstruction procedure 46

5.5.4 Reconstruction with sign property 47

5.6 Viscous flux discretization 48

5.6.1 Kinetic energy preservation for one-dimensional Navier-Stokes equations 49

5.6.2 Entropy stability for one-dimensional Navier-Stokes equations . . . 49

5.7 Two-dimensional finite difference scheme 51

5.7.1 Inviscid discretization 53

5.7.2 Viscous discretization 54

5.7.3 Kinetic energy preservation 54

5.7.4 Entropy stability 55

5.8 Numerical results 56

5.8.1 Smooth density wave 58

5.8.2 Modified Sod test case 59

5.8.3 Low density problem 63

5.8.4 Isentropic vortex 64

5.8.5 Shock vortex interaction 70

5.8.6 3D Taylor-Green vortex 72

6 A sign-preserving WENO reconstruction 79

6.1 WENO reconstruction 79

6.2 Properties 81

6.2.1 Consistency 81

6.2.2 Sign property 81

6.2.3 Negation symmetry 82

6.2.4 Mirror property 82

6.2.5 Inner jump condition 82

6.2.6 Accuracy 83

6.2.7 The feasible region 84

6.3 Explicit weights: SP-WENO 87

6.3.1 Stability estimates 88

6.4 Reconstruction accuracy of SP-WENO 90

6.5 Numerical results with SP-WENO for scalar conservation laws 90

6.5.1 Linear advection 91

6.5.2 Burgers' Equation 94

6.6 Numerical results with SP-WENO for Euler equations 98

6.7 SP-WENOc: A fix for systems of conservation laws 100

6.8 Reconstruction accuracy of SP-WENOc 101

6.9 Numerical results with SP-WENOc for Euler equations 102

7 Energy preserving finite difference scheme for the shallow water equations 107

7.1 Governing equations 107

7.1.1 Preserved quantities 109

7.1.2 $\mathbf{u} - h - \vartheta$ model 109

7.2 Finite difference scheme 110

7.2.1 An energy preserving flux 111

7.2.2 WENO-5 flux 112

7.3 Numerical results 113

7.3.1 Advecting vortex 114

7.3.2 Effect of Coriolis force 115

7.3.3 Perturbed geostrophic balance 123

8 A fully-discrete kinetic energy preserving and entropy stable scheme 133

8.1 Notations 133

8.2 Fully discrete finite volume scheme 135

8.2.1 Fully discrete kinetic energy preserving scheme 135

8.2.2 Fully discrete entropy conservative scheme 136

8.2.3 Construction of $\widetilde{\mathbf{V}}$ 137

8.3 Time integration 138

8.3.1 Explicit iteration 139

8.3.2 Newton-GMRES 139

8.4 Numerical results 140

8.4.1 Smooth inviscid solution 140

8.4.2 DNS of 1D Navier-Stokes: Sod test case 141

9 Finite volume schemes for Euler equations 145

9.1 Mesh 145

9.1.1 Cell-centered scheme 145

9.1.2 Vertex-centered scheme 146

9.2 Semi-discrete finite volume scheme 148

9.2.1 Cell-centered semi-discrete scheme 149

9.2.2 Vertex-centered semi-discrete scheme 150

9.3 Entropy conservative and entropy stable schemes 153

9.3.1 Entropy conservative scheme 153

9.3.2 First order entropy stable scheme 154

9.3.3 High-order diffusion operators 155

9.3.4 Reconstruction procedure and the sign-property 157

9.3.5 Computation of gradients 159

9.4 Numerical results 160

9.4.1 Smooth density wave 160

9.4.2 Modified shock tube problem 161
9.4.3 Supersonic flow over wedge 164
9.4.4 Transonic flow past NACA-0012 airfoil 164
9.4.5 Supersonic flow past a cylinder 171
9.4.6 Subsonic flow past a cylinder 173
9.4.7 Step in wind tunnel 174

10 Finite volume scheme for Navier-Stokes equations 177

10.1 Mesh notations and discretization 177
10.1.1 Approximation of gradient and divergence operators 177
10.1.2 Finite volume scheme 180
10.2 Discrete entropy estimates 181
10.3 Code implementation 186
10.3.1 Mesh generation and partitioning 186
10.3.2 Parallelization 186
10.4 Numerical results 188
10.4.1 Shock tube problem 188
10.4.2 Vortex advection 194
10.4.3 Laminar flat-plate boundary layer 194
10.4.4 Lid-driven cavity 196
10.4.5 Transonic flow past NACA-0012 airfoil 197
10.4.6 Flow past a cylinder 204

11 Conclusions 207

11.1 Future scope 209

A Viscous flux symmetrization 211

B Logarithmic average 213

C Numerical flux expression in three-dimensions 215

C.1 Central fluxes 215
C.1.1 KEPEC 215
C.1.2 ROE-EC 215
C.1.3 KEP 216
C.2 Dissipation matrix used in entropy stable schemes 216

D GMRES method 219

E Cell averages of entropy variables and positivity 221

Bibliography 233

1. Introduction

Conservation is an important principle of physics, and forms the basis of mathematical models in numerous fields of science and engineering. The notion of conservation states that *the rate of change of a conserved quantity* $\mathbf{U}$ *in a volume* Ω *depends only on the flux* $\mathbf{F}$ *across the boundary* $\partial\Omega$. Some important examples of systems of conservation laws include the shallow water equations, the Euler equations and the magnetohydrodynamics (MHD) equations.

Solutions of *non-linear hyperbolic conservation laws* can develop discontinuities in finite time, even when the initial data is sufficiently smooth [25]. Thus, one must abandon the notion of classical differentiable solutions and interpret the solutions in a weak (distributional) sense. Whenever a classical solution exists, it coincides with the weak solution. However, weak solutions of conservation laws are in general not unique, and must be supplemented with additional *entropy conditions* to single out a *physically relevant* solution. In the entropy framework, we assume the system of conservation laws is equipped with a family of *entropy-entropy flux pairs*, which are functions of the conserved variable of the system under consideration. If a solution satisfies the entropy condition for every such pair, the solution is called an *entropy solution*. Scalar conservation laws are endowed with a rich class of entropy-entropy flux pairs, and this fact was exploited by Kruzkov [70] to prove the existence and uniqueness of entropy solutions for multi-dimensional scalar conservation laws. However, the situation is quite different for systems of conservation laws. Apart from some partial one-dimensional results [13, 10, 43], no well-posedness results are available for general multi-dimensional systems. Nevertheless, the entropy conditions provide the only generic non-linear estimates for systems of conservation laws available at present. Thus, for a given system of conservation laws, we choose a suitable entropy-entropy flux pair (if available) and prove the corresponding entropy estimates.

Numerical methods for hyperbolic systems of conservation laws have undergone extensive development over the past few decades. In finite difference/volume methods, the computational domain is divided into control volumes and a discrete version of the conservation law imposed on each control volume. In particular, (approximate) Riemann solver based numerical flux functions, non-oscillatory reconstructions which are Total-Variation-Diminishing (TVD), Essentially-Non-Oscillatory (ENO) or of Weighted ENO (WENO) type, along with strong stability preserving Runge-Kutta methods, constitute an attractive and widely used package for the robust approximation of systems of conservation laws. An alternative is the use of Runge-Kutta Discontinuous Galerkin (DG) finite element methods [20] together with limiters to obtain non-oscillatory approximation.

Although many rigorous convergence results for these methods (at least for their first and second order versions) are known for scalar conservation laws (see [69, 68] and references therein), very few rigorous results are available for schemes approximating systems of conservation laws, particularly in several space dimensions. Since obtaining rigorous convergence results of numerical approximation to entropy solutions seems out of reach currently (see [34] for a discussion on this issue) the design of *entropy stable schemes* – numerical schemes that satisfy a discrete form of the entropy inequality – is a reasonable goal. Entropy stable schemes automatically satisfy an L^p estimate and provide the only global stability estimates currently available for numerical methods for multi-dimensional conservation laws.

An important class of entropy stable schemes for systems of conservation laws was proposed by Tadmor in [116], which paved the way for obtaining high-order entropy stable schemes. The construction is based on two ingredients – *(i)* construction of an entropy conservative flux satisfying a discrete entropy equality, and *(ii)* addition of suitable dissipation operators to satisfy a discrete entropy inequality. First-order entropy stable schemes, in which the solution is assumed to be piecewise constant in the cells, have been tested by Fjordholm et al. [125] for the shallow-water equations, and by Roe and Ismail [58] for the Euler equations on Cartesian meshes. High order accurate schemes are constructed by reconstructing the solution in each cell by a polynomial. Arbitrarily high-order entropy conservative fluxes for Cartesian grids were developed in [71]. However, the design of arbitrary-high order entropy stable schemes was only carried out recently by Fjordholm et al. in [35]. These so-called *TeCNO schemes* judiciously combine high-order entropy conservative fluxes with arbitrarily high-order numerical diffusion operators, based on piecewise polynomial reconstruction. The reconstructions have to satisfy a *sign property* at each interface to ensure entropy stability. This means that the jump in the reconstructed values at every cell face must have the same sign as the jump in the corresponding cell values. A second-order reconstruction with the minmod limiter satisfies the sign-property. ENO reconstruction methods, which were first introduced in [54], are used to construct high-order polynomial reconstructions by adaptively choosing the smoothest stencil. It was shown in [36] that the standard ENO reconstruction procedure satisfies the sign property. Recently, a third order sign-preserving reconstruction based on appropriate limiting of quadratic polynomials, was proposed in [18]. To the best of our knowledge, no other known reconstruction satisfies this crucial property.

WENO schemes [73, 62] were proposed as an improvement over ENO schemes. The basic idea of WENO is to take a convex combination of lower order polynomials and obtain an effective high-order approximation of the solution. The convex weights are chosen so as to give the least weight to polynomials whose stencils contain discontinuities. In this thesis, we propose a third-order sign-preserving WENO-type reconstruction procedure. This WENO scheme, termed as SP-WENO, enjoys a few other important monotonicity and stability properties. When used in the TeCNO framework, SP-WENO leads to a third-order entropy stable scheme on uniform Cartesian grids.

The majority of this thesis focuses on a specific system of conservation laws, namely the compressible Euler equations. These are formulated on the basis of the conservation of mass, momentum and energy in fluid flows. The Euler equations describe an idealistic model, where the effects of viscosity are ignored. The inclusion of viscous forces in the balance of momentum and energy leads to the compressible Navier-Stokes equations, which are hyperbolic-parabolic in nature. While a family of entropy-entropy flux pairs are available for the Euler equations [52], a specific choice of entropy-entropy flux pair symmetrizes the viscous flux Jacobians [57, 31], which leads to the formulation of suitable entropy estimates for the viscous flow.

Obtaining high-resolution numerical solvers for the Navier-Stokes equations is of great practical importance, especially for large eddy simulations (LES). At the same time, it is important to be able to control the numerical instabilities triggered by the existence of discontinuities in the numerical solution or under-resolved flow features, where the balance of advection and diffusion plays a delicate role. A popular approach of handling issues of instability is by constructing discrete operators for the Navier-Stokes equations such that they satisfy discrete non-linear entropy stability estimates, analogous to those existing for continuous equations. The symmetric formulation of the viscous fluxes in terms of the entropy variables, has been utilized to construct a finite-difference scheme for the Navier-Stokes equation in [31], and time-discontinuous Galerkin finite-element methods in [103]. An alternate approach of the Summation-by-Parts (SBP) framework has been used to derive provably stable, polynomial-based spectral collocation element methods of arbitrary order [15].

A faithful representation of kinetic energy dynamics is also of key importance for compressible flows. Although a bound on the kinetic energy does not ensure a bound on the numerical solution in a compressible flow, the correct evolution of kinetic energy is a crucial requirement for the accurate simulation of turbulence [105, 78, 89]. A method to construct finite difference/volume schemes for the Euler equations that ensures a consistent evolution of kinetic energy at the discrete level, was proposed in [59]. Such schemes are said to be *kinetic energy preserving*. Recently, a semi-discrete finite volume scheme that is both kinetic energy preserving and entropy conservative was constructed for the Euler equations [16], and is termed as the *KEPEC* flux. With a suitable discretization of the viscous fluxes for the one-dimensional Navier-Stokes equations, the KEPEC flux leads to a kinetic energy preserving and entropy stable semi-discrete scheme for the viscous model.

The high-order TeCNO schemes are only available for Cartesian (structured) grids in several space dimensions. However, many applications of interest, particularly in engineering, involve domains with complex geometry [27, 60] which can be more easily discretized using *unstructured grids*. The construction of high resolution, entropy stables schemes on unstructured grids is not as mature. In [74], a first-order finite volume scheme was constructed in the framework of cell-centered schemes, where the solution is stored at the center of the cells. It does not seem to be possible to extend this approach to high resolution while at the same time maintaining the sign property and the accuracy of the scheme. Recently, an entropy stable space-time DG finite element scheme has been proposed on unstructured meshes [56], with shock-capturing and streamline diffusion terms to handle discontinuities. In this thesis, we propose a vertex-centered finite volume scheme where the solution is stored at the vertices of the mesh, and a dual cell is constructed around each vertex on which the conservation law is satisfied [132, 111, 2, 88, 76, 1]. The high resolution scheme is constructed by using a reconstruction process to obtain the solution values at the faces of the cells. In the literature, there are several approaches to perform this reconstruction [126, 30, 21, 99, 11, 88, 109, 7, 130, 131, 65]. We use a simple approach for reconstruction, (cf. chapter IV - section 5.1 of [45]), but this process is combined with the structure of the dissipation operator so that the sign property can be satisfied. We hence construct a semi-discrete, high resolution scheme which is entropy stable on general triangulations.

Another major hurdle in constructing numerical methods for compressible flows is the prescription of boundary conditions for the initial boundary value problem. Most existing approaches are based on linearizing the Navier-Stokes equation near the boundary, followed by the energy method to derive suitable boundary conditions [50, 80, 55]. Nordström and Svärd [81] have used this idea to analyze the well-posedness of boundary conditions for the linearized Navier-Stokes system in three dimensions on a general domain. Svärd and Mishra [113] have constructed a conservative finite difference scheme using the SBP approach and simultaneous-approximation-term (SAT) penalty technique for the Euler equations on bounded domains, which have been shown to satisfy an appropriate *boundary entropy inequality* [29] numerically. Unfortunately, this methodology cannot be extended to the Navier-Stokes equation, as the specific form of entropy function used does not symmetrize the viscous fluxes. In [31], a normalized entropy function is used to derive a global energy estimate, with boundary conditions prescribed to bound/dissipate the total energy of the Navier-Stokes equations. However, it is not clear how one can consistently choose the various constants introduced to describe the inflow/outflow conditions. Recently, non-linear entropy-stable wall boundary conditions have been proposed in [87] and tested in the framework of discontinuous spectral collocation operators. The slip boundary condition for the Euler equations is imposed using a manufactured boundary state, the boundary viscous heat flux requires the construction of a suitable numerical boundary flux and the no-slip boundary condition is imposed using a standard SAT approach.

In this thesis, a vertex-centered finite volume is proposed for the initial-boundary-valued Navier-Stokes system in two dimensions. The inviscid flux is discretized at each control interface using an entropy stable flux, while the viscous fluxes are evaluated on triangles in terms of the entropy variables to preserve the symmetric structure of the continuous system. The boundary conditions are weakly imposed by constructing suitable inviscid boundary fluxes based on the numerical value at the boundary node and the given boundary data.

The rest of the thesis is organized in the following manner:

Chapter 2 introduces the basic notations to describe hyperbolic systems of conservation laws. The various solution frameworks for conservation laws are briefly described. A few important examples of scalar conservation laws and systems of conservation laws are discussed.

Chapter 3 discusses the equations describing compressible flows. First, the Euler equations are introduced along with the appropriate entropy framework. Then, the viscous and heat conduction terms of the Navier-Stokes equations are described, followed by the extension of the entropy framework to symmetrize the viscous fluxes. Finally, the initial-boundary-value problem is discussed for the Navier-Stokes equations and entropy estimates are obtained for wall boundary conditions.

Chapter 4 describes the formulation of finite difference and finite volume schemes for conservation laws. A few important first-order methods are discussed, followed by a brief summary of reconstruction techniques and limiters used to obtain high-order schemes.

Chapter 5 describes the construction of semi-discrete finite difference schemes on Cartesian meshes which are provably entropy stable. The sufficient conditions for constructing entropy conservative/stable schemes are briefly outlined. A few important numerical fluxes for the Euler equations are highlighted and numerically compared. Multi-dimensional viscous flux approximations satisfying the SBP property are discussed, which leads to kinetic energy preservation and entropy stability for the Navier-Stokes equations.

In **Chapter 6**, we discuss the construction of the new sign-preserving SP-WENO scheme, and detail other crucial properties satisfied by it. The SP-WENO method is tested for both scalar and systems of conservation laws in the TeCNO framework.

In **Chapter 7**, we take a small detour and consider the shallow water equations in the *vector-invariant* form. A semi-discrete finite difference scheme is proposed for the model which preserves the total energy of the system. We test the capabilities of the scheme in preserving other invariants such as the total potential enstrophy.

Chapter 8 details the construction of a fully-discrete scheme for the one-dimensional compressible flow equations, which is both kinetic energy preserving and entropy stable.

Chapter 9 describes the finite volume formulation for the Euler equations. The cell-centered and vertex-centered approaches are compared from the point of view of local truncation errors. A second-order entropy stable flux is proposed by reconstructing the the solution at the cell-interfaces using the minmod limiter. The reconstruction procedure requires nodal gradients of entropy variables. A method to compute gradients which are exact for linear functions, is discussed.

In **Chapter 10** the finite volume scheme introduced in Chapter 9 is extended to incorporate the viscous fluxes. It is shown that the proposed SBP-type discretisation of the viscous fluxes and the prescription of appropriate numerical boundary fluxes lead to consistent discrete entropy estimates.

2. Hyperbolic conservation laws

In this chapter, we give a brief overview of systems of conservation laws. We first introduce the basic notations to describe the Cauchy problem for the generic system, followed by various existing solution frameworks.

Let $\mathbf{U} : \mathbb{R}^d \times \mathbb{R}_+ \mapsto \mathbb{R}^m$ be a vector of conserved quantities. Then, the rate of change of $\mathbf{U}$ in any volume $\Omega_0 \in \mathbb{R}^d$ depends on the flux through the boundary $\partial\Omega_0$. The relation is given by the integral equation

$$\frac{\mathrm{d}}{\mathrm{d}t} \int\limits_{\Omega_0} \mathbf{U}(\mathbf{x}, t) \mathrm{d}\,\mathbf{x} = - \int\limits_{\partial\Omega_0} \sum_{i=1}^{d} \mathbf{f}_i(\mathbf{U}(\mathbf{x}, t)) n_i \mathrm{d}\,S, \tag{2.1}$$

where $\mathbf{f}_i$ are the Cartesian components of the (smooth) flux function and $\mathbf{n} = (n_1, n_2, ...n_d)$ is the unit outward normal at the boundary. When $\mathbf{U}$ is smooth enough, we can use the divergence theorem in (2.1) to obtain the differential formulation

$$\partial_t \mathbf{U} + \sum_{i=1}^{d} \partial_{x_i} \mathbf{f}_i(\mathbf{U}) = 0, \quad (\mathbf{x}, t) \in \mathbb{R}^d \times \mathbb{R}_+. \tag{2.2}$$

We assume that the flux function depends only on the conserved variables $\mathbf{U}$, which is indeed the case in most physical phenomena governed by *hyperbolic* conservation laws.

Definition 2.0.1. *Let $\mathbf{A}_i(\mathbf{U}) := \mathbf{f}_i'(\mathbf{U})$ be the flux Jacobians. Then the system* (2.2) *is said to be hyperbolic if, for any $\mathbf{U} \in \Pi \subset \mathbb{R}^m$ and any $\mathbf{n} \in \mathbb{R}^d$, the matrix*

$$\mathbf{A}(\mathbf{U}, \mathbf{n}) = \sum_{i=1}^{d} \mathbf{A}_i(\mathbf{U}) n_i,$$

has m real eigenvalues $\lambda_1(\mathbf{U}) \leqslant ... \leqslant \lambda_m(\mathbf{U})$ and m linearly independent eigenvectors $\mathbf{r}_1(\mathbf{U}), ..., \mathbf{r}_m(\mathbf{U})$. If, in addition, these eigenvalues are distinct, then the system is said to be "strictly hyperbolic".

Usually Π corresponds to some admissible set dictated by constraints on $\mathbf{U}$, such as the positivity of certain quantities. Each pair $(\lambda_i(\mathbf{U}), \mathbf{r}_i(\mathbf{U}))$ corresponding to $\mathbf{A}(\mathbf{U}, \mathbf{n})$ defines a *characteristic field* call the λ_i-field. A λ_i-field is *linearly degenerate* if $\langle \lambda_i'(\mathbf{U}), \mathbf{r}_i(\mathbf{U}) \rangle = 0$ for all $\mathbf{U} \in \Pi$, and *genuinely nonlinear* if $\langle \lambda_i'(\mathbf{U}), \mathbf{r}_i(\mathbf{U} \rangle \neq 0$ for all $\mathbf{U} \in \Pi$. In this thesis, we will only consider hyperbolic conservation laws which have linearly degenerate or genuinely nonlinear characteristic fields.

The *Cauchy problem* for the above system also requires the prescription of an initial condition

$$\mathbf{U}(\mathbf{x}, 0) = \mathbf{U}_0(\mathbf{x}), \quad \mathbf{x} \in \mathbb{R}^d, \tag{2.3}$$

where $\mathbf{U}_0 : \mathbb{R}^d \mapsto \mathbb{R}^m$.

2.1 Weak formulation

It is well known that due to the non-linearity of the flux function, hyperbolic systems of conservation laws can develop discontinuities in finite time, even when the initial condition is smooth [25]. Thus, we can no longer talk about classical solutions and must interpret the solutions in a weak (distributional) sense.

Definition 2.1.1. *A function* $\mathbf{U} \in (L^1_{loc}(\mathbb{R}^d \times \mathbb{R}_+))^m$ *is called a weak solution of the Cauchy problem for* (2.2) *with* $\mathbf{U}_0 \in (L^1_{loc}(\mathbb{R}^d))^m$ *if*

$$\int_0^\infty \int_{\mathbb{R}^d} \left(\langle \mathbf{U}, \partial_t \boldsymbol{\phi} \rangle + \sum_{i=1}^d \langle \mathbf{f}_i(\mathbf{U}), \partial_{x_i} \boldsymbol{\phi} \rangle \right) dt\, d\mathbf{x} + \int_{\mathbb{R}^d} \langle \mathbf{U}_0(\mathbf{x}), \boldsymbol{\phi}(\mathbf{x}, 0) \rangle \; d\mathbf{x} = 0, \qquad (2.4)$$

for all $\boldsymbol{\phi} \in (C_0^\infty(\mathbb{R}^d \times \mathbb{R}_+))^m$.

Equation (2.4) is formally obtained by integrating the inner product of (2.2) and the test function $\boldsymbol{\phi}$ over space and time, so that all derivatives are transferred to the test function. The advantage of considering the weak formulation is that we can relax the smoothness conditions on the solution. However, not every discontinuity is admissible. Let Γ be a surface of discontinuity in the $t - \mathbf{x}$-plane for the solution $\mathbf{U}$, and $\widetilde{\mathbf{n}} = (n_t, n_1, ..., n_d) \neq 0$ be the normal vector to Γ. Let us denote by $\mathbf{U}_+$ and $\mathbf{U}_-$ the limits of $\mathbf{U}$ on either side of Γ

$$\mathbf{U}_\pm(\mathbf{x}, t) = \lim_{\epsilon \downarrow 0} \mathbf{U}((\mathbf{x}, t) \pm \epsilon \widetilde{\mathbf{n}}).$$

Theorem 2.1.1 (**Rankine-Hugoniot (RH) condition**). *The weak solution* $\mathbf{U}$ *of* (2.2) *must satisfy the jump condition*

$$(\mathbf{U}_+ - \mathbf{U}_-) n_t + \sum_{i=1}^d \big(\mathbf{f}_i(\mathbf{U}_+) - \mathbf{f}_i(\mathbf{U}_-)\big) n_i = 0, \qquad (2.5)$$

across the surface of discontinuity Γ.

When $d = 1$, we assume that Γ is parametrized as $(t, \xi(t))$, in which case $\widetilde{\mathbf{n}} = (-s, 1)$ with $s = \mathrm{d}\xi/\mathrm{d}t$ being the speed of the discontinuity. The corresponding RH condition reads as

$$s(\mathbf{U}_+ - \mathbf{U}_-) = \mathbf{f}(\mathbf{U}_+) - \mathbf{f}(\mathbf{U}_-). \qquad (2.6)$$

Clearly, every classical solution is a weak solution. However, weak solutions need not be unique, and must be supplemented with additional *entropy conditions* to single out a physically relevant solution.

2.2 Entropy conditions

Assume that the system (2.2) is equipped with a strictly convex function $\eta : \mathbb{R}^m \mapsto \mathbb{R}$ and functions $q_i : \mathbb{R}^m \mapsto \mathbb{R}$ such that

$$q_i'(\mathbf{U}) = \eta'(\mathbf{U})^\top \mathbf{f}_i'(\mathbf{U}) \qquad i = 1, 2, .., d. \qquad (2.7)$$

The function η is known as an *entropy function*, while $(q_1, q_2, ..., q_d)$ is the *entropy flux*. Additionally, $\mathbf{V} = \eta'(\mathbf{U})$ is called the (vector of) *entropy variables*. Taking the scalar product of (2.2) with $\mathbf{V}$ results in the following additional conservation law

$$\partial_t \eta(\mathbf{U}) + \sum_{i=1}^{d} \partial_{x_i} q_i(\mathbf{U}) = 0, \tag{2.8}$$

which is satisfied for smooth solutions. The entropy condition states that weak solutions should satisfy the entropy inequality

$$\partial_t \eta(\mathbf{U}) + \sum_{i=1}^{d} \partial_{x_i} q_i(\mathbf{U}) \leqslant 0, \tag{2.9}$$

which is understood in the sense of distributions, i.e.,

$$\int_0^\infty \int_{\mathbb{R}^d} \left(\eta(\mathbf{U}) \partial_t \phi + \sum_{i=1}^{d} q_i(\mathbf{U}) \partial_{x_i} \phi \right) \mathrm{d}t \, \mathrm{d}\mathbf{x} + \int_{\mathbb{R}^d} \eta\big(\mathbf{U}_0(\mathbf{x})\big) \phi(\mathbf{x}, 0) \, \mathrm{d}\mathbf{x} \geqslant 0, \tag{2.10}$$

for all $\phi \in C_0^\infty(\mathbb{R}^d \times \mathbb{R}_+)$, with $\phi(\mathbf{x}, t) \geqslant 0$. The solution $\mathbf{U}$ is called an *entropy solution* if it satisfies (2.10) for every convex entropy.

If $\eta(\mathbf{U})$ is strictly convex, then there exists a one-to-one mapping between $\mathbf{U}$ and $\mathbf{V}$, thus allowing the change of variables $\mathbf{U} = \mathbf{U}(\mathbf{V})$. The following theorem based on the work of Godunov [47] and Mock [77] links the existence of convex entropy functions for the system (2.2) with the *symmetrization* of the system under the change of variable. The transformed system

$$\partial_{\mathbf{V}} \mathbf{U} \partial_t \mathbf{V} + \sum_{i=1}^{d} \partial_{\mathbf{V}} \mathbf{f}_i \partial_{x_i} \mathbf{V} = 0,$$

is said to be symmetrized by the change of variable $\mathbf{U} = \mathbf{U}(\mathbf{V})$, if the Jacobian $\partial_{\mathbf{V}} \mathbf{U}$ is symmetric positive definite and $\partial_{\mathbf{V}} \mathbf{f}_i$ are symmetric.

Theorem 2.2.1. *A necessary and sufficient condition for the hyperbolic system* (2.2) *to possess a strictly convex entropy* η *is that there exists a change of variable* $\mathbf{U} = \mathbf{U}(\mathbf{V})$ *that symmetrizes* (2.2).

For the case of scalar conservation laws (m=1), every convex function serves as an entropy function. This idea was exploited by Kruzkov [70] to prove the existence and uniqueness of entropy solutions in the class of functions of bounded variation. The existence result was proved by considering the solution U^ϵ of the *parabolic regularized* problem

$$\partial_t U^\epsilon + \sum_{i=1}^{d} \partial_{x_i} f_i(U^\epsilon) = \epsilon \Delta U^\epsilon, \tag{2.11}$$

for $\epsilon > 0$, and then showing that the the sequence $\{U^\epsilon\}_\epsilon$ converges to an entropy solution U of the original conservation law as the coefficient ϵ goes to zero. The proof relies heavily on local bounds on the total variation of the sequence $\{U^\epsilon\}_\epsilon$. An alternate proof of existence of entropy solutions for scalar conservation laws can be obtained by relaxing the bounded variation conditions on the perturbed solutions $\{U^\epsilon\}_\epsilon$, and using the method of *compensated compactness* by Murat and Tartar [79, 120].

For systems of conservation laws, the situation is quite different. Apart from some partial results for one-dimensional systems [13, 10, 43], no well-posedness results are available for general multi-dimensional systems. However, the entropy conditions do play an important role in providing global stability estimates. Formally integrating (2.9) in space and assuming suitable decay conditions on the entropy flux or periodic boundary conditions, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\mathbb{R}^d}\eta(\mathbf{U})\mathrm{d}\mathbf{x}\leqslant 0 \qquad \Longrightarrow \qquad \int\limits_{\mathbb{R}^d}\eta(\mathbf{U}(\mathbf{x},t))\mathrm{d}\mathbf{x}\leqslant\int\limits_{\mathbb{R}^d}\eta(\mathbf{U}_0(\mathbf{x}))\mathrm{d}\mathbf{x}. \tag{2.12}$$

The above bound on total entropy along with the convexity of η gives rise to an a priori estimate on the solution of (2.2) in suitable L^p spaces [25]. This is the only generic non-linear estimate for systems of conservation laws available at present.

2.3 The Riemann problem

Since the solution of conservation laws can develop discontinuities in finite time, it becomes important to understand the structure of the solution after the appearance of a discontinuity. With this in mind, we consider the following special type of Cauchy problem for a one-dimensional system of hyperbolic conservation laws

$$\begin{gathered}\partial_t\mathbf{U}+\partial_x\mathbf{f}(\mathbf{U})=0,\\ \mathbf{U}_0(x)=\begin{cases}\mathbf{U}_L, & \text{if } x<0\\ \mathbf{U}_R, & \text{if } x>0\end{cases},\end{gathered} \tag{2.13}$$

where $\mathbf{U}_L$ and $\mathbf{U}_R$ are constant states. The Cauchy problem of the type (2.13) is also known as a *Riemann problem*, and is the simplest initial value problem that can be posed for conservation laws. Note that the initial condition is discontinuous if $\mathbf{U}_L\neq\mathbf{U}_R$. We assume that the system is hyperbolic, with the flux Jacobian $\mathbf{A}(\mathbf{U})=\mathbf{f}'(\mathbf{U})$ having distinct eigenvalues $\lambda_1(\mathbf{U})<...<\lambda_m(\mathbf{U})$. As shown in Figure 2.1, the solution to this problem consists of m waves emanating from the origin, corresponding to each eigenvalue. Furthermore, the solutions of (2.13) are *self-similar* [45]. In other words, the solutions are of the form $\mathbf{U}(x,t)=\mathbf{W}(x/t)$, and consist of $m+1$ constant states separated by the m waves. These $m+1$ states are connected by the following waves:

- **Shock wave:** The λ_i-wave is a shock wave, if it corresponds to a genuinely nonlinear field and connects two states $\mathbf{U}_-$ and $\mathbf{U}_+$ through a single jump discontinuity. The discontinuity moves with a speed S_i given by the RH condition (2.6). Furthermore, the (Lax) entropy condition holds, i.e.,

  $$\lambda_i(\mathbf{U}_-)>S_i>\lambda_i(\mathbf{U}_+),$$

  which can be deduced from the entropy condition (2.9) for a convex flux. The *characteristic* lines $\mathrm{d}x/\mathrm{d}t=\lambda_i$ on both sides of the shock line $\mathrm{d}x/\mathrm{d}t=S_i$ run into the shock wave. This is depicted in Figure 2.2(a).

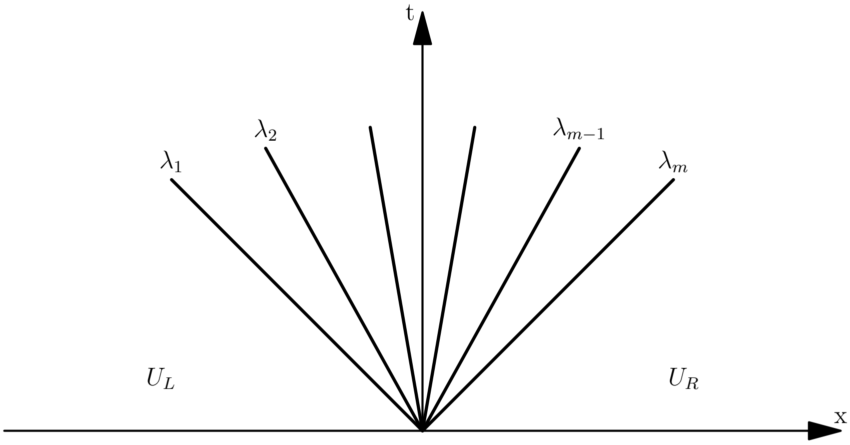


Figure 2.1: Solution structure for the Riemann problem of a system of conservation laws.

- **Contact wave:** The λ_i-wave is a contact wave, if it corresponds to a linearly degenerate field and connects two states $\mathbf{U}_-$ and $\mathbf{U}_+$ through a single jump discontinuity. As in the case of the shock wave, the discontinuity moves with a speed S_i given by the RH condition (2.6). It additionally satisfies the *parallel characteristic condition*

$$\lambda_i(\mathbf{U}_-) = S_i = \lambda_i(\mathbf{U}_+).$$

In other words, the characteristic lines on either side of the contact line $\mathrm{d}x/\mathrm{d}t = S_i$ run parallel to it. This is depicted in Figure 2.2(b).

- **Rarefaction:** The λ_i-wave corresponds to a rarefaction, if it connects two states $\mathbf{U}_-$ and $\mathbf{U}_+$ through a smooth transition in a genuinely nonlinear field. The characteristic lines corresponding to a rarefaction diverge from each other, i.e.,

$$\lambda_i(\mathbf{U}_-) < \lambda_i(\mathbf{U}_+),$$

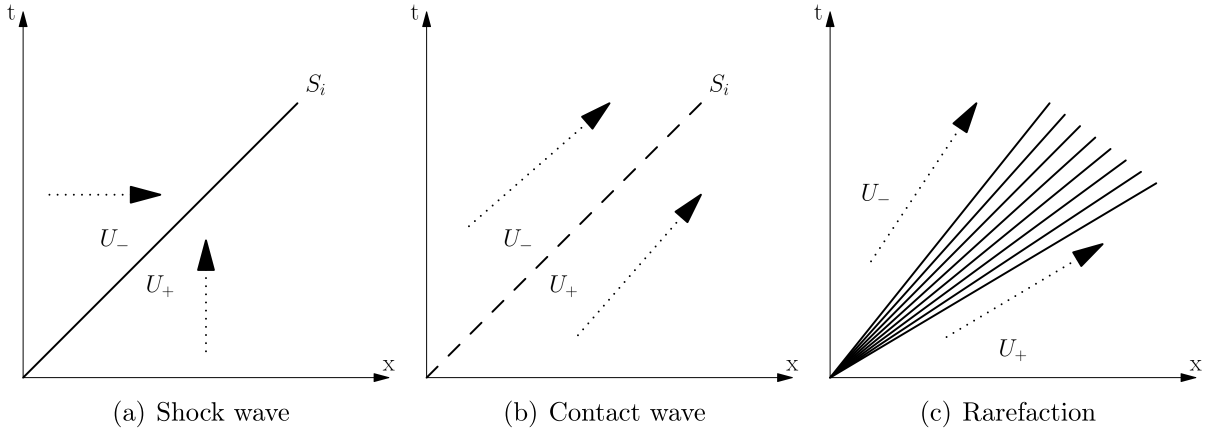
as is shown in Figure 2.2(c).

A more thorough discussion on various concepts associated with Riemann solutions for systems of conservation laws can be found in [45].

2.4 Entropy measure-valued solutions

As discussed towards the end of Section 2.2, there are no well-posedness results available for general multi-dimensional systems of conservation laws. The non-uniqueness of entropy solutions for some specific multi-dimensional systems has recently been shown in [19]. This suggests that the classical notion of entropy solutions may not be adequate to establish the existence and uniqueness of solutions for a general system. Thus, one needs to consider a more general notion of solutions for (2.2). One such suitable framework is based on the notion of *measure-valued solutions* formulated by DiPerna [28]. Consider the mapping

$$\nu : \mathbb{R}^d \times \mathbb{R}_+ \mapsto \mathcal{P}(\mathbb{R}^m),$$



(a) Shock wave (b) Contact wave (c) Rarefaction

Figure 2.2: Characteristic lines for simple waves forming the solution to a Riemann problem

where $\mathcal{P}(\mathbb{R}^m)$ is the space of probability measures over $\mathbb{R}^m$. Thus, ν assigns a probability measure $\nu_{\mathbf{x},t}$ for each $(\mathbf{x},t)$. The function ν is called a *Young measure*. Furthermore, a Young measure can be composed with a continuous function $g : \mathbb{R}^m \mapsto \mathbb{R}$ by defining

$$\langle \nu_{\mathbf{x},t}, g \rangle_{\mathcal{M}} := \int_{\mathbb{R}^m} g(\boldsymbol{\zeta}) \mathrm{d}\nu_{\mathbf{x},t}(\boldsymbol{\zeta}),$$

which is precisely the expectation of g with respect to the measure $\nu_{\mathbf{x},t}$. Note that the operator $\langle .,. \rangle_{\mathcal{M}}$ is different from the scalar product operator $\langle .,. \rangle$, and that $\langle \nu_{\mathbf{x},t}, g \rangle_{\mathcal{M}}$ is a real-valued function of space-time. Additionally, if $\mathbf{g} : \mathbb{R}^m \mapsto \mathbb{R}^m$ is vector of continuous functions, then $\langle \nu_{\mathbf{x},t}, \mathbf{g} \rangle_{\mathcal{M}}$ will be a vector of real-valued function formed by composing ν with each component of $\mathbf{g}$.

Every measurable function $\mathbf{U} : \mathbb{R}^d \times \mathbb{R}_+ \mapsto \mathbb{R}^m$ gives rise to a Young measure given by

$$\nu_{\mathbf{x},t} := \delta_{\mathbf{U}(\mathbf{x},t)},$$

where $\delta_{\boldsymbol{\zeta}}$ is the Dirac measure centered at $\boldsymbol{\zeta} \in \mathbb{R}^m$. Such Young measures are termed as *atomic*. For an atomic Young measure, we have

$$\langle \nu_{\mathbf{x},t}, g \rangle_{\mathcal{M}} = g(\mathbf{U}(\mathbf{x},t)).$$

Based on the above notations, we consider the following generalization of the Cauchy problem corresponding to (2.2)

$$\begin{aligned} \partial_t \langle \nu_{\mathbf{x},t}, id \rangle_{\mathcal{M}} + \sum_{i=1}^{m} \partial_{x_i} \langle \nu_{\mathbf{x},t}, \mathbf{f}_i \rangle_{\mathcal{M}} = 0, \quad (\mathbf{x},t) \in \mathbb{R}^d \times \mathbb{R}_+, \\ \nu_{\mathbf{x},0} = \sigma_{\mathbf{x}}, \quad \mathbf{x} \in \mathbb{R}^d, \end{aligned} \tag{2.14}$$

where $id(\boldsymbol{\zeta}) = \boldsymbol{\zeta}$ is the identity function and σ is the Young measure denoting the measure-valued initial condition.

Definition 2.4.1. *A Young measure ν is called a measure-valued solution of the system* (2.14) *if it satisfies*

$$\begin{aligned}\int_0^\infty \int_{\mathbb{R}^d} \left(\left\langle \langle \nu_{\mathbf{x},t}, id \rangle_{\mathcal{M}}, \partial_t \boldsymbol{\phi} \right\rangle + \sum_{i=1}^d \left\langle \langle \nu_{\mathbf{x},t}, \mathbf{f}_i \rangle_{\mathcal{M}}, \partial_{x_i} \boldsymbol{\phi} \right\rangle \right) dt\, d\mathbf{x} \\ + \int_{\mathbb{R}^d} \left\langle \langle \sigma_{\mathbf{x}}, id \rangle_{\mathcal{M}}, \boldsymbol{\phi}(\mathbf{x}, 0) \right\rangle \, d\mathbf{x} = 0,\end{aligned} \tag{2.15}$$

for all $\boldsymbol{\phi} \in (C_0^\infty(\mathbb{R}^d \times \mathbb{R}_+))^m$.

In a similar manner, one can extend (2.10) to define ν to be an *entropy measure-valued (EMV) solution* of (2.14) for a given entropy-entropy flux pair, if it satisfies

$$\begin{aligned}\int_0^\infty \int_{\mathbb{R}^d} \left(\langle \nu_{\mathbf{x},t}, \eta \rangle_{\mathcal{M}} \, \partial_t \phi + \sum_{i=1}^d \langle \nu_{\mathbf{x},t}, \mathbf{q}_i \rangle_{\mathcal{M}} \, \partial_{x_i} \phi \right) \mathrm{d}t \, \mathrm{d}\mathbf{x} \\ + \int_{\mathbb{R}^d} \langle \sigma_{\mathbf{x}}, \eta \rangle_{\mathcal{M}} \, \phi(\mathbf{x}, 0) \, \mathrm{d}\mathbf{x} \geqslant 0,\end{aligned} \tag{2.16}$$

for all $\phi \in C_0^\infty(\mathbb{R}^d \times \mathbb{R}_+)$, with $\phi(\mathbf{x}, t) \geqslant 0$.

DiPerna [28] has shown that if $U(\mathbf{x}, t)$ is the entropy solution for a scalar conservation law corresponding to the initial condition $U_0(x)$, then the EMV solution corresponding to the initial Dirac measure $\delta_{U_0(\mathbf{x})}$ coincides with $\delta_{U(\mathbf{x},t)}$. This result has been generalized to non-atomic initial measures for scalar conservation laws in [34], where it has been shown that an EMV solution exists for a bounded initial measure $\sigma_{\mathbf{x}}$, i.e., there exists a compact set $K \in \mathbb{R}^m$ such that $\text{supp}(\sigma_{\mathbf{x}}) \in K$ for all $\mathbf{x} \in \mathbb{R}^d$. Although EMV solutions need not be unique for non-atomic initial measures, it is possible to show that by restricting to a relevant class of atomic initial data, the EMV solutions for scalar conservation laws are stable [34]. For a system of conservation laws, if a classical solution is known to exist for a given initial condition, then a weak-strong uniqueness can be proved for EMV solutions under certain boundedness assumptions. Extensive numerical experiments in [34] indicate the stability of EMV solutions for selected systems of conservation laws.

2.5 Examples

We now briefly describe a few important examples of conservation laws.

2.5.1 Linear advection equation

The linear advection equation is the simplest example of conservation laws, and is given by

$$\partial_t U + \sum_{i=1}^d a_i \partial_{x_i} U = 0, \tag{2.17}$$

where U is a scalar conserved variable and $\mathbf{a} = (a_1, ..., a_d)$ determines the constant advection velocity. Given the initial condition $U(\mathbf{x}, 0) = U_0(\mathbf{x})$ for the Cauchy problem, the exact solution has the expression

$$U(\mathbf{x}, t) = U_0(\mathbf{x} - \mathbf{a}t),$$

which corresponds to advection of the initial condition by the velocity $\mathbf{a}$. As mentioned earlier, any convex function can serve as an entropy function for scalar equations. In particular, we can choose the quadratic function $\eta(U) = U^2/2$. The corresponding entropy flux functions for the linear advection equation can be obtained using the compatibility relation (2.7) as

$$q_i(U) = \int\limits_U \eta'(w) f_i'(w) \mathrm{d}w = \frac{a_i U^2}{2}. \tag{2.18}$$

Since the flux is a linear function, discontinuities in the solution can only be connected by contact waves.

For the linear advection equation, the weak solution is already unique and in principle does not require additional entropy conditions. Even with discontinuous initial conditions, the total entropy is preserved by assuming the initial condition has suitable decay conditions as $|\mathbf{x}| \to \infty$. With the choice of the quadratic entropy function, we can conclude that the L^2 norm of the solution is preserved in time.

2.5.2 Burgers' equation

The Burgers' equation is the simplest example of a non-linear scalar conservation law, and is given by

$$\partial_t U + \partial_x \left(\frac{U^2}{2} \right) = 0. \tag{2.19}$$

The eigenvalue corresponding to the scalar flux Jacobian is $\lambda = U$. The λ-field is genuinely nonlinear, and thus, discontinuous solution states are connected by either shocks or rarefaction waves. Choosing the quadratic entropy function leads to the entropy flux function

$$q(U) = \frac{U^3}{3}. \tag{2.20}$$

Unlike the linear advection equation, the appearance of a shock dissipates the total entropy.

2.5.3 Wave equation

The wave equation in one-dimension is given by

$$\partial_{tt} u - c^2 \partial_{xx} u = 0, \tag{2.21}$$

with the initial conditions

$$u(x,0) = u_0(x), \quad \partial_t u(x,0) = u_1(x),$$

where c is a constant speed of wave propagation. By introducing new variables $v = \partial_x u$ and $w = \partial_t u$, (2.21) can be re-formulated as the following linear system of conservation laws

$$\begin{pmatrix} v \\ w \end{pmatrix}_t + \begin{pmatrix} -w \\ -c^2 v \end{pmatrix}_x = 0,$$

with initial conditions $\big(v(x,0),\ w(x,0)\big) = \big(u_0'(x),\ u_1(x)\big)$. Furthermore, the flux Jacobian

$$\mathbf{A} = \begin{pmatrix} 0 & -1 \\ -c^2 & 0 \end{pmatrix},$$

has eigenvalues and eigenvectors

$$\lambda_1 = c, \quad \lambda_2 = -c, \quad \mathbf{r}_1 = \begin{pmatrix} 1 \\ -c \end{pmatrix}, \quad \mathbf{r}_2 = \begin{pmatrix} 1 \\ c \end{pmatrix},$$

and is thus hyperbolic provided $c \neq 0$.

The wave equation is a specific case of the following more general framework of linear symmetric systems

$$\partial_t \mathbf{U} + \sum_{i=1}^{d} \mathbf{A}_i \partial_{x_i} \mathbf{U} = 0, \tag{2.22}$$

where $\mathbf{A}_i$ are constant matrices. Assume there exists a positive definite matrix $\mathbf{B}$ such that $\mathbf{B}\mathbf{A}_i$ are symmetric. Then, (2.22) is equipped with the following entropy-entropy flux functions

$$\eta(\mathbf{U}) = \frac{1}{2}\langle \mathbf{U}, \mathbf{B}\mathbf{U}\rangle, \quad q_i(\mathbf{U}) = \frac{1}{2}\langle \mathbf{U}, \mathbf{B}\mathbf{A}_i\mathbf{U}\rangle, \quad i = 1, ..., d$$

with entropy variables $\mathbf{V} = \mathbf{B}\mathbf{U}$. For the wave equation (2.21), $d = 1$ and $\mathbf{B}$ is the identity operator.

2.5.4 Shallow water equations

The shallow water equations are an example of a non-linear system of conservation laws, that describe the motion in a thin layer of fluid with a constant density, bounded below by a flat bottom topography, and from above by a free surface. This system is given by the equations

$$\begin{pmatrix} h \\ hu_1 \\ hu_2 \end{pmatrix}_t + \begin{pmatrix} hu_1 \\ hu_1^2 + \frac{1}{2}gh^2 \\ hu_1u_2 \end{pmatrix}_x + \begin{pmatrix} hu_2 \\ hu_1u_2 \\ hu_2^2 + \frac{1}{2}gh^2 \end{pmatrix}_y = 0, \tag{2.23}$$

where h is the fluid height, $\mathbf{u} = (u_1, u_2)$ is the horizontal velocity and g is the gravity constant. Referring to Definition 2.0.1, we have for $\mathbf{n} \in \mathbb{R}^2$

$$\mathbf{A}(\mathbf{U}, \mathbf{n}) = \begin{pmatrix} 0 & n_1 & n_2 \\ -u_1u_n + n_1gh & n_1u_1 + u_n & n_2u_1 \\ -u_2u_n + n_2gh & n_1u_2 & n_2u_2 + u_n \end{pmatrix}, \quad u_n = \mathbf{u} \cdot \mathbf{n},$$

with the eigenvalues and eigenvectors of $\mathbf{A}(\mathbf{U}, \mathbf{n})$ given by

$$\lambda_1 = u_n - c,\ \lambda_2 = u_n,\ \lambda_3 = u_n + c, \quad \mathbf{r}_1 = \begin{pmatrix} 1 \\ u_1 - n_1c \\ u_2 - n_2c \end{pmatrix}, \ \mathbf{r}_2 = \begin{pmatrix} 0 \\ -n_2 \\ n_1 \end{pmatrix}, \ \mathbf{r}_3 = \begin{pmatrix} 1 \\ u_1 + n_1c \\ u_2 + n_2c \end{pmatrix}$$

where $c = \sqrt{gh}$ is the speed of gravity waves. Assuming $h > 0$, the matrix $\mathbf{A}(\mathbf{U}, \mathbf{n})$ has real distinct eigenvalues, and thus ensures the two-dimensional system shallow water equations is strictly hyperbolic.

Furthermore, the system (2.23) is equipped with the following entropy-entropy flux functions

$$\eta(\mathbf{U}) = \frac{h}{2}(u_1^2 + u_2^2) + \frac{1}{2}gh^2, \quad q_i(\mathbf{U}) = \frac{h}{2}\sum_{j=1}^{2} u_j^2 u_i + g u_i h^2, \quad i = 1, 2$$

where the entropy function is nothing but the energy of the flow.

3. Compressible flows

In this chapter, we introduce the equations that describe the motion of compressible flows. We begin by considering the Euler equations which describe the flow mechanics in the absence of viscous forces. The Euler equations are formulated based on fundamental principles of conservation of mass, momentum and energy, and can be shown to be hyperbolic in nature. We discuss the entropy framework for this system, which will play a crucial role in constructing suitable numerical schemes.

3.1 Euler equations

The three dimensional Euler equations are given by

$$\begin{aligned}
\partial_t\rho + \partial_{x_1}(\rho u_1) + \partial_{x_2}(\rho u_2) + \partial_{x_3}(\rho u_3) &= 0,\\
\partial_t(\rho u_1) + \partial_{x_1}(\rho u_1^2 + p) + \partial_{x_2}(\rho u_1 u_2) + \partial_{x_3}(\rho u_1 u_3) &= 0,\\
\partial_t(\rho u_2) + \partial_{x_1}(\rho u_2 u_1) + \partial_{x_2}(\rho u_2^2 + p) + \partial_{x_3}(\rho u_2 u_3) &= 0,\\
\partial_t(\rho u_3) + \partial_{x_1}(\rho u_3 u_1) + \partial_{x_2}(\rho u_3 u_2) + \partial_{x_3}(\rho u_3^2 + p) &= 0,\\
\partial_t E + \partial_{x_1}\big(u_1(E+p)\big) + \partial_{x_2}\big(u_2(E+p)\big) + \partial_{x_3}\big(u_3(E+p)\big) &= 0,
\end{aligned} \tag{3.1}$$

where ρ, $\mathbf{u} = (u_1, u_2, u_3)^\top$ and p denote the fluid density, velocity and pressure, respectively. The quantity E is the total energy per unit volume

$$E = \rho\left(\frac{1}{2}|\mathbf{u}|^2 + e\right), \tag{3.2}$$

where e is the specific internal energy given by a caloric equation of state, $e = e(\rho, p)$. For the remainder of this thesis, we take the equation of state to be that of the ideal gas, given by

$$e = \frac{p}{(\gamma - 1)\rho}, \tag{3.3}$$

with $\gamma = c_p/c_v$ denoting the ratio of specific heats.

The first equation of (3.1) describes the conservation of mass, the next three describe the conservation of the three components of momentum, and the final equation describes the conservation of energy. The Euler equations can be re-formulated as the following system of conservation laws

$$\partial_t\mathbf{U} + \sum_{i=1}^{3}\partial_{x_i}\mathbf{f}_i(\mathbf{U}) = 0,$$

where the vector of conserved variables $\mathbf{U}$ and the flux components $\mathbf{f}_1$, $\mathbf{f}_2$, $\mathbf{f}_3$ are given by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u_1 \\ \rho u_2 \\ \rho u_3 \\ E \end{pmatrix}, \quad \mathbf{f}_i(\mathbf{U}) = \begin{pmatrix} \rho u_i \\ \rho u_i u_1 + p\delta_{i1} \\ \rho u_i u_2 + p\delta_{i2} \\ \rho u_i u_3 + p\delta_{i3} \\ u_i(E+p) \end{pmatrix}, \quad \delta_{ij} = \begin{cases} 1 & \text{if i=j} \\ 0 & \text{otherwise} \end{cases}, \quad i = 1,2,3. \tag{3.4}$$

3.1.1 Hyperbolicity

Define the flux Jacobian $\mathbf{A}_i(\mathbf{U}) = \mathbf{f}_i(\mathbf{U})$ with $i = 1,2,3$ and consider the matrix

$$\mathbf{A}(\mathbf{U},\mathbf{n}) = \mathbf{A}_1 n_1 + \mathbf{A}_2 n_2 + \mathbf{A}_2 n_3, \quad \mathbf{n} = (n_1, n_2, n_3) \in \mathbb{R}^3. \tag{3.5}$$

The eigenvalue and corresponding matrix of eigenvectors for $\mathbf{A}(\mathbf{U},\mathbf{n})$ are

$$\begin{aligned} \lambda_1 = u_n - a, \quad \lambda_2 = \lambda_3 = \lambda_4 &= u_n, \quad \lambda_5 = u_n + a \\ \mathbf{R}(\mathbf{U},\mathbf{n}) &= \begin{pmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 & \mathbf{r}_4 & \mathbf{r}_5 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ u_1 - an_1 & u_1 & n_2 & -n_3 & u_1 + an_1 \\ u_2 - an_2 & u_2 & -n_1 & 0 & u_2 + an_2 \\ u_3 - an_3 & u_3 & 0 & n_1 & u_3 + an_3 \\ H - au_n & \frac{1}{2}|\mathbf{u}|^2 & u_1n_2 - u_2n_1 & u_3n_1 - u_1n_3 & H + au_n \end{pmatrix} \end{aligned} \tag{3.6}$$

where $u_n = \mathbf{u}\cdot\mathbf{n}$, $a = \sqrt{\gamma p/\rho}$ is the speed of sound in air and $H = (\gamma-1)^{-1}a^2 + |\mathbf{u}|^2/2$ is the specific enthalpy. Assuming the positivity of density and pressure, the eigenvalues are real and the corresponding eigenvectors are linearly-independent, thus making the system hyperbolic.

Additionally, we define the Mach number of the flow as $M = |\mathbf{u}|/a$. The Mach number is used to describe various flow regimes: the flow is *subsonic* for $M < 1$, *supersonic* for $M > 1$ and *transonic* if the flow has both supersonic and subsonic regions.

3.1.2 Entropy framework

Harten [52] has shown that the Euler equations are equipped with a family of entropy-entropy flux functions of the form

$$\eta(\mathbf{U}) = -\frac{\rho h(s)}{\gamma - 1}, \quad q_i(\mathbf{U}) = -\frac{\rho u_i h(s)}{\gamma - 1}, \quad i = 1,2,3 \tag{3.7}$$

with an additional constraint $h''/h' < \gamma^{-1}$ to enforce strict convexity of η. Here the quantity $s = \ln(p) - \gamma\ln(\rho)$ is obtained from the thermodynamic specific entropy $\tilde{s}$ by $s = (\tilde{s} - \tilde{s_0})/c_v$ where $\tilde{s_0}$ is an arbitrary constant (see [44] for a detailed derivation). A convenient choice which we adhere to for the rest of this dissertation is

$$\eta(\mathbf{U}) = -\frac{\rho s}{\gamma - 1}, \quad q_i(\mathbf{U}) = -\frac{\rho u_i s}{\gamma - 1}, \quad i = 1,2,3 \tag{3.8}$$

where we have made the affine choice $h(s) = s$. The corresponding entropy variables $\mathbf{V}$ are given by

$$\mathbf{V} = \begin{pmatrix} V^{(1)} \\ V^{(2)} \\ V^{(3)} \\ V^{(4)} \\ V^{(5)} \end{pmatrix} = \begin{pmatrix} \frac{\gamma-s}{\gamma-1} - \beta|\mathbf{u}|^2 \\ 2\beta u_1 \\ 2\beta u_2 \\ 2\beta u_3 \\ -2\beta \end{pmatrix}, \quad \beta = \frac{\rho}{2p}. \tag{3.9}$$

3.2 Navier-Stokes equations

The Euler equations are a good approximation for several flow scenarios. However, viscous effects become important for studying flows with boundary layers near solid walls and the behaviour of fluids in turbulent regimes. If viscous effects are included in the balance of momentum and energy, it leads to the formulation of the more general compressible Navier-Stokes equations, which are hyperbolic-parabolic in nature. The three-dimensional Navier-Stokes equations can be written as

$$\partial_t \mathbf{U} + \sum_{i=1}^{3} \partial_{x_i} \mathbf{f}_i(\mathbf{U}) = \sum_{i=1}^{3} \partial_{x_i} \mathbf{g}_i(\mathbf{U}, \nabla \mathbf{U}), \tag{3.10}$$

where the vector of variables $\mathbf{U}$ and the *inviscid fluxes* $\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$ are given by (3.4). The *viscous fluxes* $\mathbf{g}_1, \mathbf{g}_2$ and $\mathbf{g}_3$ can be expressed as

$$\mathbf{g}_i(\mathbf{U}) = \begin{pmatrix} 0 \\ \tau_{i1} \\ \tau_{i2} \\ \tau_{i3} \\ u_1\tau_{i1} + u_2\tau_{i2} + u_3\tau_{i3} - Q_i \end{pmatrix}, \quad i = 1, 2, 3 \tag{3.11}$$

with the shear stress tensor $\boldsymbol{\tau}$ and the heat flux $\mathbf{Q}$ given by Newtonian and Fourier constitutive relations respectively,

$$\boldsymbol{\tau} = \mu(\nabla \mathbf{u} + (\nabla \mathbf{u})^\top)) - \frac{2}{3}\mu(\nabla \cdot \mathbf{u})\mathcal{I}, \quad \mathbf{Q} = (Q_1, Q_2, Q_3) = -\kappa \nabla \theta.$$

Here, $\mathcal{I}$ is the unit tensor, μ is the coefficient of dynamic viscosity and κ is the coefficient of heat conductance. Furthermore, θ denotes the temperature of the flow, which is obtained using the *ideal gas law* given by $p = \rho R \theta$, where R is the gas constant with $R = c_p - c_v$. Note that the shear stress tensor is symmetric. The coefficient of heat conductance can be determined from μ using the relation

$$\kappa = \frac{\mu c_p}{Pr}, \tag{3.12}$$

where Pr is the Prandtl number, which is assumed to be constant for a given gas. The Euler equations can be recovered from (3.10) by setting $\mu = 0$.

An important non-dimensional number defined for viscous flows is the *Reynolds number*, which is given by

$$Re = \frac{LU}{\nu},$$

where L and U are the characteristic length and velocity scales of the flow respectively, while $\nu = \mu/\rho_0$ is the coefficient of kinematic viscosity. The Reynolds number can be seen as measure of the ratio of inertial forces to viscous forces of the flow. For low Reynolds numbers, the viscous forces dominate and the flow is *laminar*. As Reynolds number is increased, the flow transitions from a laminar regime into a *turbulent* regime, which is associated with the formation of eddies at several length scales [39].

We introduce the notations

$$\mathbf{F}(\mathbf{U}, \mathbf{n}) = \sum_{i=1}^{d} \mathbf{f}_i(\mathbf{U}) n_i, \quad \mathbf{G}(\mathbf{U}, \nabla\mathbf{U}, \mathbf{n}) = \sum_{i=1}^{d} \mathbf{g}_i(\mathbf{U}, \nabla\mathbf{U}) n_i, \quad q(\mathbf{U}, \mathbf{n}) = \sum_{i=1}^{d} q_i(\mathbf{U}) n_i, \tag{3.13}$$

for $d = 1, 2, 3$, which will be useful to represent the flux in the direction $\mathbf{n} \in \mathbb{R}^d$. In most cases, $\mathbf{n}$ will correspond to the outward normal to a domain boundary. The viscous fluxes may at times be written in terms of the entropy variables, in which case we have the alternate notation

$$\mathbf{G}(\mathbf{V}, \nabla\mathbf{V}, \mathbf{n}) = \sum_{i=1}^{d} \mathbf{g}_i(\mathbf{V}, \nabla\mathbf{V}) n_i. \tag{3.14}$$

3.2.1 Symmetrization of viscous fluxes

The system of Euler equations is symmetrized when formulated in terms of $\mathbf{V}$, as was discussed in Section 2.2. Hughes et al. [57] have shown that the entropy pairs of the form (3.7) also symmetrize the viscous fluxes if $h(s)$ is restricted to be at most affine. In particular, we can work with the choice (3.8). To see this, we reformulate the viscous fluxes (3.11) in terms of the entropy variable and its first order spatial derivatives

$$\mathbf{g}_i(\mathbf{V}, \nabla\mathbf{V}) = \sum_{j=1}^{3} \mathbf{K}_{ij}(\mathbf{V}) \partial_{x_j} \mathbf{V}, \quad i = 1, 2, 3. \tag{3.15}$$

It can be shown that the matrix

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{K}_{13} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \mathbf{K}_{23} \\ \mathbf{K}_{31} & \mathbf{K}_{32} & \mathbf{K}_{33} \end{pmatrix} \in \mathbb{R}^{15\times 15},$$

is symmetric and positive semi-definite [31]. This is what is meant by the symmetrization of the viscous fluxes. The explicit expression for $\mathbf{K}$ is given in Appendix A.

3.3 Evolution of kinetic energy

The kinetic energy is an important quantity in fluid flows, especially in turbulent regimes where kinetic energy is transferred from large scales to small scales, and finally dissipated by viscous forces. Let us consider a general formulation for the dimension d, where $d = 1, 2, 3$. The kinetic energy density $\mathcal{K} = \rho|\mathbf{u}|^2/2$ for compressible flow evolves according to the equation

$$\begin{aligned}\partial_t\mathcal{K} &= -\frac{1}{2}|\mathbf{u}|^2\partial_t\rho + \langle\mathbf{u},\partial_t(\rho\mathbf{u})\rangle\\ &= \frac{1}{2}|\mathbf{u}|^2\sum_{j=1}^{d}\partial_{x_j}(\rho u_j) - \sum_{i=1}^{d}\sum_{j=1}^{d}u_i\partial_{x_j}(\rho u_i u_j + p\delta_{ij} - \tau_{ij})\\ &= \sum_{j=1}^{d}\left[\partial_{x_j}\left(u_j\left(p-\rho\frac{1}{2}|\mathbf{u}|^2\right)+\sum_{i=1}^{d}u_i\tau_{ij}\right)\right] + p\nabla\cdot\mathbf{u} - \sum_{i=1}^{d}\sum_{j=1}^{d}\tau_{ij}\partial_{x_j}u_i.\end{aligned}\tag{3.16}$$

Splitting $\partial_{x_j}u_i$ into its symmetric and anti-symmetric part i.e., $\partial_{x_j}u_i = S_{ij} + A_{ij}$, and using the fact that

$$\tau_{ij} = \mu(\partial_{x_j}u_i + \partial_{x_i}u_j) - \frac{2}{3}\mu\delta_{ij}\nabla\cdot\mathbf{u}$$

we get

$$\sum_{i=1}^{d}\sum_{j=1}^{d}\tau_{ij}\partial_{x_j}u_i = \sum_{i=1}^{d}\sum_{j=1}^{d}\tau_{ij}S_{ij} = \sum_{i=1}^{d}\sum_{j=1}^{d}\frac{1}{2}\mu\left(\partial_{x_j}u_i + \partial_{x_i}u_j\right)^2 - \frac{2}{3}\mu\left(\nabla\cdot\mathbf{u}\right)^2.\tag{3.17}$$

Integrating (3.16) over a domain $\Omega \in \mathbb{R}^d$ and assuming periodic or no-flow boundary conditions leads to the following equation of evolution of total kinetic energy

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \mathcal{K}\mathrm{d}\mathbf{x} = \int_\Omega p\nabla\cdot\mathbf{u}\mathrm{d}\mathbf{x} - \int_\Omega\left[\sum_{i=1}^{d}\sum_{j=1}^{d}\frac{1}{2}\mu\left(\partial_{x_j}u_i + \partial_{x_i}u_j\right)^2 - \frac{2}{3}\mu\left(\nabla\cdot\mathbf{u}\right)^2\right]\mathrm{d}\mathbf{x}.\tag{3.18}$$

The first term on the right of (3.18) describes the rate at which work is done by pressure forces, and is present only for compressible flows, i.e., if $\nabla\cdot\mathbf{u} \neq 0$. The second term represents the destruction of kinetic energy by viscous forces, which is converted to internal energy. Note that (3.17) can be rewritten as

$$\sum_{i=1}^{d}\sum_{j=1}^{d}\tau_{ij}\partial_{x_j}u_i = \sum_{i=1}^{d}\sum_{j=1}^{d}\frac{1}{2}\mu\left[\partial_{x_j}u_i + \partial_{x_i}u_j - \frac{2}{3}\delta_{ij}w(d)\nabla\cdot\mathbf{u}\right]^2 \geqslant 0,\ w(d) = \frac{3+\sqrt{9-3d}}{d}.$$

Thus, the viscous forces clearly dissipate the kinetic energy. For the one-dimensional Navier-Stokes equations, (3.18) reduces to the following simpler equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_\Omega \mathcal{K}\mathrm{d}x = \int_\Omega p\partial_x u\mathrm{d}x - \int_\Omega \frac{4}{3}\mu(\partial_x u)^2\mathrm{d}x.\tag{3.19}$$

3.4 Initial boundary value problem

Thus far, we have been considering the Cauchy problem. We can also consider the *initial boundary value problem* (IBVP) for the Navier-Stokes equations. Let $\Omega \in \mathbb{R}^3$ be a domain with boundary given by $\partial\Omega$. Then the IBVP on this domain can be written as

$$\begin{aligned}\partial_t\mathbf{U} + \sum_{i=1}^{3}\partial_{x_i}\mathbf{f}_i(\mathbf{U}) &= \sum_{i=1}^{3}\partial_{x_i}\mathbf{g}_i(\mathbf{V},\nabla\mathbf{V}) \quad (\mathbf{x},t)\in\Omega\times\mathbb{R}^+,\\ \mathbf{U}(\mathbf{x},0) &= \mathbf{U}_0(\mathbf{x}) \quad \mathbf{x}\in\Omega,\\ + &\quad \textit{Boundary Conditions},\end{aligned}\tag{3.20}$$

where the viscous flux is written in terms of the entropy variables. Taking the scalar product of the Navier-Stokes equations with $\mathbf{V}$ and integrating over Ω gives us

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\Omega}\eta\mathrm{d}\mathbf{x} = -\int\limits_{\partial\Omega} q(\mathbf{U},\mathbf{n})\mathrm{d}S - \int\limits_{\Omega}\sum_{i=1}^{3}\left\langle\partial_{x_i}\mathbf{V},\mathbf{g}_i\right\rangle\mathrm{d}\mathbf{x} + \int\limits_{\partial\Omega}\left\langle\mathbf{V},\mathbf{G}(\mathbf{V},\nabla\mathbf{V},\mathbf{n})\right\rangle\mathrm{d}S, \tag{3.21}$$

where $\mathbf{n}$ is the outward normal at the domain boundary. We introduce the notation

$$\widetilde{\nabla}\mathbf{V} = \begin{pmatrix}\partial_{x_1}\mathbf{V}\\ \partial_{x_2}\mathbf{V}\\ \partial_{x_3}\mathbf{V}\end{pmatrix} \in \mathbb{R}^{15},$$

which we distinguish from the usual gradient notation $\nabla\mathbf{V} = \left(\partial_{x_1}\mathbf{V},\ \partial_{x_2}\mathbf{V},\ \partial_{x_3}\mathbf{V}\right) \in \mathbb{R}^{5\times 3}$. Using (3.15) and the fact that $\mathbf{K}$ is symmetric and positive semi-definite, we have

$$-\sum_{i=1}^{3}\left\langle\partial_{x_i}\mathbf{V},\mathbf{g}_i\right\rangle = -\left\langle\mathbf{K}\widetilde{\nabla}\mathbf{V},\widetilde{\nabla}\mathbf{V}\right\rangle \leqslant 0.$$

Thus, (3.21) results in the following entropy relation for the Navier-Stokes equations,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\Omega}\eta\mathrm{d}\mathbf{x} \leqslant -\int\limits_{\partial\Omega} q(\mathbf{U},\mathbf{n})\mathrm{d}S + \int\limits_{\partial\Omega}\left\langle\mathbf{V},\mathbf{G}(\mathbf{V},\nabla\mathbf{V},\mathbf{n})\right\rangle\mathrm{d}S. \tag{3.22}$$

The relation (3.22) shows that if there is no net entropy flux through the boundaries, then the total entropy is non-increasing in time.

3.4.1 Boundary conditions and entropy stability

We focus on solid wall boundary conditions for the Navier-Stokes equations, i.e., $\mathbf{u}\Big|_{\partial\Omega} = 0$, which leads to the entropy relation

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\Omega}\eta\mathrm{d}\mathbf{x} \leqslant \int\limits_{\partial\Omega}\mathbf{V}^{\top}\mathbf{G}(\mathbf{V},\nabla\mathbf{V},\mathbf{n})\mathrm{d}S = -\int\limits_{\partial\Omega}\frac{\kappa}{R\theta}\partial_{\mathbf{n}}\theta\mathrm{d}S, \tag{3.23}$$

where $\partial_{\mathbf{n}}\theta$ denotes the directional derivative of the temperature in the direction of the outward normal. In addition, we must either specify the heat flux or the temperature at the boundary. If we assume the heat flux is prescribed

$$\mathbf{Q}\cdot\mathbf{n}\Big|_{\partial\Omega} = -\kappa\partial_{\mathbf{n}}\theta\Big|_{\partial\Omega} = h^b,$$

we obtain the estimate

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\Omega}\eta\mathrm{d}\mathbf{x} \leqslant \int\limits_{\partial\Omega}\frac{h^b}{R\theta}\mathrm{d}S.$$

The inviscid and viscous boundary fluxes take the form

$$\mathbf{F}(\mathbf{U},\mathbf{n}) = \left(0,\ p\mathbf{n},\ 0\right)^{\top},\quad \mathbf{G}(\mathbf{V},\nabla\mathbf{V},\mathbf{n}) = \left(0,\ \boldsymbol{\tau}\cdot\mathbf{n},\ -h^b\right)^{\top}, \tag{3.24}$$

respectively.

If on the other hand we have an isothermal solid wall with $\theta\Big|_{\partial\Omega} = \theta^b$ prescribed, then the following estimate holds

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\Omega} \eta \mathrm{d}\mathbf{x} \leqslant -\int\limits_{\partial\Omega} \frac{\kappa}{R\theta^b}\partial_{\mathbf{n}}\theta \mathrm{d}S, \tag{3.25}$$

which can also be written in terms of the entropy variable as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\Omega} \eta \mathrm{d}\mathbf{x} \leqslant \int\limits_{\partial\Omega} \frac{\kappa}{RV^{b,(5)}}\partial_{\mathbf{n}}V^{(5)} \mathrm{d}S, \tag{3.26}$$

where $V^{b,(5)} = -2\beta = -1/R\theta^b$.

If there is an external forcing term active at the boundary, we cannot expect to obtain an "entropy stability" estimate in general. However, if the forcing term enforces adiabatic solid wall conditions i.e., $h^b = 0$, or extracts heat from the system i.e., $h^b < 0$, then we can obtain the entropy stability estimate

$$\frac{\mathrm{d}}{\mathrm{d}t}\int\limits_{\Omega} \eta \mathrm{d}\mathbf{x} \leqslant 0. \tag{3.27}$$

Isothermal solid wall conditions lead to the estimate (3.25) (or (3.26)), which is only considered to be an entropy estimate and not a stability relation as it may not be possible to a priori bound the heat flux with the prescribed boundary data.

4. Finite difference and finite volume schemes for conservation laws

In this chapter we describe the formulation of finite difference and finite volume schemes for systems of conservation laws. For these methods, the computational domain is discretized using non-overlapping control volume, following which a discrete version of the conservation law is posed on each control volume. Finite difference methods approximate the differential form of the conservation law and are more suited for *Cartesian grids*. High-order finite difference schemes are obtained by a suitable polynomial reconstruction of variables in each cell using the *point values* of the variables. On the other hand, finite volume methods are obtained by integrating the conservation law over a control volume, and evolving *cell-averages* of the solution. For two and higher-dimensional problems ($d > 1$), the integral of the flux on the boundary of the control volumes also needs to be approximated using suitable quadratures. High-order finite volume methods are obtained by using a high-order quadrature formula, followed by the reconstruction of solution values at the boundary quadrature points using neighbouring cell-average values.

4.1 Finite volume scheme

We present details for finite volume schemes approximating one-dimensional conservation laws on Cartesian grids. Finite volume schemes on two-dimensional unstructured grids will be discussed in Chapter 9. Consider the Cauchy problem for a one-dimensional system of conservation laws

$$\begin{aligned} \partial_t \mathbf{U} + \partial_x \mathbf{f}(\mathbf{U}) &= 0, \quad (x,t) \in \mathbb{R} \times \mathbb{R}_+, \\ \mathbf{U}(x,0) &= \mathbf{U}_0(x), \quad x \in \mathbb{R}. \end{aligned} \tag{4.1}$$

We discretize the domain using disjoint intervals $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}})$ of uniform length $\Delta x = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$. We use the notation x_i to denote the center of the interval I_i.

4.1.1 Fully discrete scheme

Consider the time interval $[t^n, t^{n+1})$ with the time-step $\Delta t = t^{n+1} - t^n$. We integrate the conservation law over the space-time control volume $I_i \times [t^n, t^{n+1})$ to get

$$\int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \left(\mathbf{U}(x,t^{n+1}) - \mathbf{U}(x,t^n)\right) \mathrm{d}x = \int\limits_{t_n}^{t_{n+1}} \Big(\mathbf{f}(\mathbf{U}(x_{i+\frac{1}{2}},t)) - \mathbf{f}(\mathbf{U}(x_{i-\frac{1}{2}},t))\Big) \mathrm{d}t, \tag{4.2}$$

which is exact in both space and time. We use the notation $\mathbf{U}_i^n$ to denote the space average of the conserved variables in the cell I_i at the time level t^n

$$\mathbf{U}_i^n = \frac{1}{\Delta x} \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \mathbf{U}(x, t^n)\mathrm{d}x.$$

Time integral of the flux along the interface $x_{i+\frac{1}{2}}$ is approximated by $\Delta t \mathbf{F}^n_{i+\frac{1}{2}}$, which gives the fully discrete finite volume scheme

$$\mathbf{U}_i^{n+1} = \mathbf{U}_i^n - \frac{\Delta t}{\Delta x}\left(\mathbf{F}^n_{i+\frac{1}{2}} - \mathbf{F}^n_{i-\frac{1}{2}}\right). \tag{4.3}$$

The numerical flux is generally chosen to be a two-point flux of the form $\mathbf{F}^n_{i+\frac{1}{2}} := \mathbf{F}(\mathbf{U}_i^n, \mathbf{U}_{i+1}^n)$. Note that the scheme (4.3) describes the evolution of cell-average values of $\mathbf{U}$. We are interested in numerical methods which satisfy a discrete version of (2.9).

Definition 4.1.1. *The fully-discrete scheme* (4.3) *is said to entropy stable if it satisfies a discrete cell entropy estimate of the form*

$$\eta(\mathbf{U}_i^{n+1}) \leqslant \eta(\mathbf{U}_i^n) - \frac{\Delta t}{\Delta x}\left(q^n_{i+\frac{1}{2}} - q^n_{i-\frac{1}{2}}\right), \tag{4.4}$$

where $q^n_{i+\frac{1}{2}}$ is a consistent numerical entropy flux.

For scalar conservation laws in one dimension, *monotone schemes* have been shown to be total variation diminishing (TVD) and satisfy the entropy condition [24]. The total variation of a function measures the amount of oscillations in the solution. It can be evaluated at the discrete level for a function $v(x)$ as

$$TV(v) = \sum_i |v_i - v_{i-1}|.$$

For scalar conservation laws, the TVD property is essential to prove the convergence of numerical schemes [44]. Additionally, *E-schemes* have been designed in [83] to preserve a discrete version of the entropy condition. However, E-schemes – and in particular monotone schemes – are at most first-order accurate.

4.1.2 Godunov's exact Riemann solver

Godunov [46] proposed a method for constructing finite volume schemes which are entropy stable [44, 72]. At each cell interface $x_{i+\frac{1}{2}}$, a local Riemann problem is solved *exactly* with the left and right initial states given by the cell average values $\mathbf{U}_i^n$ and $\mathbf{U}_{i+1}^n$ respectively. The solution is evolved for a time-step Δt to obtain $\mathbf{U}(x, t^{n+1})$, which is then averaged in each cell to obtain the new cell-average values $\mathbf{U}_i^{n+1}$ at the time level t^{n+1}. For the Godunov scheme, the flux integrals on the right of (4.3) can be evaluated exactly by using the solution to the local Riemann problem centered at the interface $x_{i+\frac{1}{2}}$. Recall from Section 2.3 that the solution to the Riemann problem has a self-similar structure, which is constant along the lines $(x - x_{i+\frac{1}{2}})/(t - t^n) = c$ for $t^n < t < t^{n+1}$, where c is some constant. In particular, the solution is constant along the line $x = x_{i+\frac{1}{2}}$, which we denote as $\widetilde{\mathbf{U}}_{i+\frac{1}{2}}$. This leads to the exact expression

$$\frac{1}{\Delta t} \int\limits_{t_n}^{t_{n+1}} \mathbf{f}(\mathbf{U}(x_{i+\frac{1}{2}}, t))\mathrm{d}t = \mathbf{f}(\widetilde{\mathbf{U}}_{i+\frac{1}{2}}),$$

with the numerical flux chosen as $\mathbf{F}^n_{i+\frac{1}{2}} = \mathbf{f}(\widetilde{\mathbf{U}}_{i+\frac{1}{2}})$.

The time-step Δt must be small enough so that the Riemann solution at $x_{i+\frac{1}{2}}$ is not disturbed by the neighbouring local Riemann solutions. This can be ensured if the distance travelled by the fastest wave of the Riemann solution in time Δt is less than Δx, i.e,

$$CFL = \max_j \left(\lambda_j(\mathbf{U}^n_i)\right) \frac{\Delta t}{\Delta x} < 1. \tag{4.5}$$

A condition of the type (4.5) is known as the *CFL condition* for the numerical method, named after Courant, Friedrich and Lewy [23]. Hyperbolic conservations laws have a finite speed of propagation, and thus the solution at a point (x, t) has a finite *domain of dependence* obtained by back-tracing the characteristic lines passing through this point. The CFL condition ensures that the numerical domain of dependence contains the true domain of dependence. This is a crucial (necessary) condition used to prove the stability of numerical schemes. Rigorous estimates of CFL conditions are generally proved for the linearized problem, which act as guiding principles to choose the time-step for the non-linear problem.

Remark 4.1.1. *The Godunov scheme can be shown to be monotone [44], and is thus entropy stable and first-order accurate.*

4.1.3 Roe's approximate Riemann solver

The Godunov method requires local Riemann problem to be solved exactly. This can be computationally expensive to do for a non-linear problem. We can instead consider a linear approximation to the problem at each interface, while ensuring that some of the essential hyperbolic properties of the non-linear problem are still retained. The linear system of conservation laws is much simpler to solve. Such methods are called *approximate Riemann solvers.* Amongst these, the approximate Riemann solver of Roe [95] is quite popular.

The basis of Roe's approach is to replace the flux Jacobian matrix $\mathbf{f}'(\mathbf{U}) = \mathbf{A}(\mathbf{U})$ at each cell-interface by the constant matrix $\tilde{\mathbf{A}} = \tilde{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R)$, where $\mathbf{U}_L$ and $\mathbf{U}_R$ are respectively the left and right solution states at the interface. The corresponding *approximate* linear Riemann problems are then solved *exactly.* The Roe Jacobian matrix $\tilde{\mathbf{A}}$ needs to satisfy the following conditions

1. $\tilde{\mathbf{A}}$ should have real eigenvalues and a complete set of linearly independent right eigenvectors. This ensures the problem is still hyperbolic.

2. $\tilde{\mathbf{A}}$ should be consistent with the exact Jacobian matrix, i.e., $\tilde{\mathbf{A}}(\mathbf{U}, \mathbf{U}) = \mathbf{A}(\mathbf{U})$.

3. Conservation should be ensured across discontinuities, i.e., $\mathbf{F}(\mathbf{U}_R) - \mathbf{F}(\mathbf{U}_L) = \tilde{\mathbf{A}}(\mathbf{U}_L, \mathbf{U}_R)(\mathbf{U}_R - \mathbf{U}_L)$.

For the one-dimensional Euler equations, i.e., (3.4) with d=1, the Roe matrix is obtained in terms of an average parameter vector

$$\widetilde{Q} = \begin{pmatrix} \widetilde{Q}_1 \\ \widetilde{Q}_3 \\ \widetilde{Q}_3 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} \sqrt{\rho_L} + \sqrt{\rho_R} \\ \sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R \\ \sqrt{\rho_L} H_L + \sqrt{\rho_R} H_R \end{pmatrix},$$

as

$$\tilde{\mathbf{A}} = \begin{pmatrix} 0 & 1 & 0 \\ \frac{\gamma-3}{2}\left(\frac{\widetilde{Q}_2}{\widetilde{Q}_1}\right)^2 & (3-\gamma)\left(\frac{\widetilde{Q}_2}{\widetilde{Q}_1}\right) & \gamma - 1 \\ \frac{\gamma-1}{2}\left(\frac{\widetilde{Q}_2}{\widetilde{Q}_1}\right)^2 - \frac{\widetilde{Q}_2\widetilde{Q}_3}{\widetilde{Q}_1^2} & \frac{\widetilde{Q}_3}{\widetilde{Q}_1} - (\gamma-1)\left(\frac{\widetilde{Q}_2}{\widetilde{Q}_1}\right)^2 & \gamma \end{pmatrix}.$$

The Roe flux for the linearised problem can be written as a combination of a central flux, and a dissipation term based on the jump of the conserved variables across the interface

$$\mathbf{F}_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{f}(\mathbf{U}_L) + \mathbf{f}(\mathbf{U}_R)\right) - \frac{1}{2}\widetilde{\mathbf{R}}\widetilde{\mathbf{\Lambda}}\widetilde{\mathbf{R}}^{-1}(\mathbf{U}_R - \mathbf{U}_L). \tag{4.6}$$

In the expression (4.6), $\widetilde{\mathbf{R}}$ is the matrix of eigenvectors of $\tilde{\mathbf{A}}$ while $\mathbf{\Lambda}$ is the diagonal matrix consisting of the absolute values of the eigenvalues of $\tilde{\mathbf{A}}$

$$\widetilde{\mathbf{R}} = \begin{pmatrix} 1 & 1 & 1 \\ \widetilde{u} - \widetilde{a} & \widetilde{u} & \widetilde{u} + \widetilde{a} \\ \widetilde{H} - \widetilde{u}\,\widetilde{a} & \frac{1}{2}\widetilde{u}^2 & \widetilde{H} + \widetilde{u}\,\widetilde{a} \end{pmatrix}, \quad \widetilde{\mathbf{\Lambda}} = \operatorname{diag}\left(|\widetilde{u} - \widetilde{a}|, \quad |\widetilde{u}|, \quad |\widetilde{u} + \widetilde{a}|\right),$$

which are evaluated at the averaged states

$$\widetilde{u} = \frac{\widetilde{Q}_2}{\widetilde{Q}_1}, \quad \widetilde{H} = \frac{\widetilde{Q}_3}{\widetilde{Q}_1}, \quad \widetilde{a} = \sqrt{(\gamma-1)\left(\widetilde{H} - \frac{1}{2}\widetilde{u}^2\right)}.$$

For further details and extension to higher-dimensions, we refer to [123].

Recall that for a linear system of conservation laws, all the characteristic fields are linearly degenerate. Thus, the solution of a linearised Riemann problem can contain (contact) discontinuous jumps, but not shocks or rarefaction waves. Since rarefactions are continuous expansion waves, the linearised approximation of rarefactions via discontinuous jumps is a terrible approximation. However, in practice, it is only in the case of a *transonic* rarefaction wave that one experiences difficulties. For a Riemann problem centered at $x_{i+\frac{1}{2}}$, a rarefaction wave whose fan is spread on both sides of $x = x_{i+\frac{1}{2}}$ is said to be transonic (see Figure 4.1). In other words, the rarefaction fan contains a characteristic line for which an eigenvalue of the flux Jacobian vanishes. The original Roe scheme gives an *entropy violating jump* in transonic rarefactions (this will be demonstrated in Chapter 5). Several *entropy fixes* have been proposed for the Roe solver, which essentially ensure that the eigenvalues remain non-zero near sonic points [53, 51, 97, 96].

The Roe flux (4.6) is just one important example of approximate Riemann solvers. There are several other approximate Riemann solvers available in literature such as HLL, HLLC, etc. We refer to [123] for an overview of these methods.

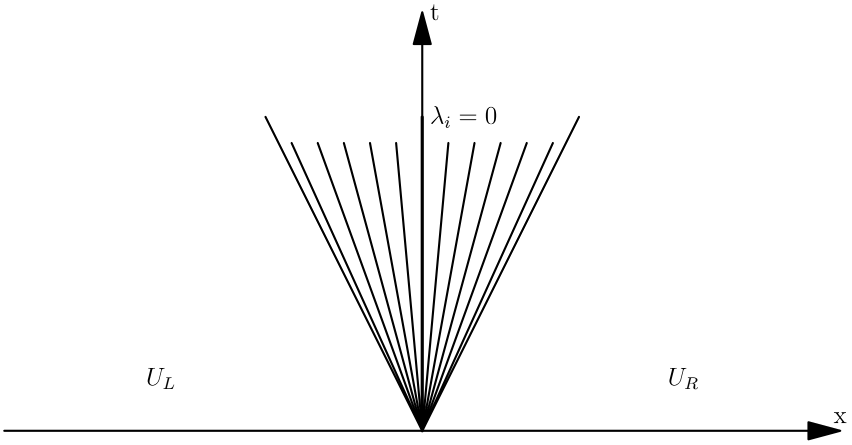


Figure 4.1: A transonic rarefaction wave

4.1.4 Reconstruction and limiters

The numerical fluxes of the aforementioned finite volume methods are evaluated at the cell-interfaces using constant cell averages. Such schemes are only first-order accurate. To obtain a higher order scheme, we reconstruct the solution inside each cell via a polynomial of suitable order. Consider a scalar valued function v, whose cell averages v_i are given. Let $p_i(x)$ represent the polynomial reconstruction of the solution in each cell, as shown in Figure 4.2. In order to develop a conservative scheme, the reconstruction needs to satisfy

$$v_i = \frac{1}{\Delta x} \int\limits_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} p_i(x) \mathrm{d}x. \tag{4.7}$$

The numerical flux is evaluated at the reconstructed states

$$v^-_{i+\frac{1}{2}} = p_i(x_{i+\frac{1}{2}}), \quad v^+_{i+\frac{1}{2}} = p_{i+1}(x_{i+\frac{1}{2}}),$$

instead of the left and right cell-average values. For systems of conservation laws, the state vector is reconstructed componentwise.

A simple approach is to use a linear polynomial of the form

$$p_i(x) = v_i + \sigma_i(x - x_i), \quad x_{i-\frac{1}{2}} < x < x_{i+\frac{1}{2}}, \tag{4.8}$$

where σ_i is the slope of the linear polynomial. Note that (4.8) already satisfies the property given by (4.7). There are several ways of choosing the slope:

- $\sigma_i^c = \frac{v_{i+1} - v_{i-1}}{2\Delta x}$ (Centered slope).
- $\sigma_i^b = \frac{v_i - v_{i-1}}{\Delta x}$ (Backward slope).
- $\sigma_i^f = \frac{v_{i+1} - v_i}{\Delta x}$ (Forward slope).

If v is smooth, then the centered slope would give the best approximation. However, *Gibbs oscillations* may appear if v has discontinuities. Thus, we need to choose the slope

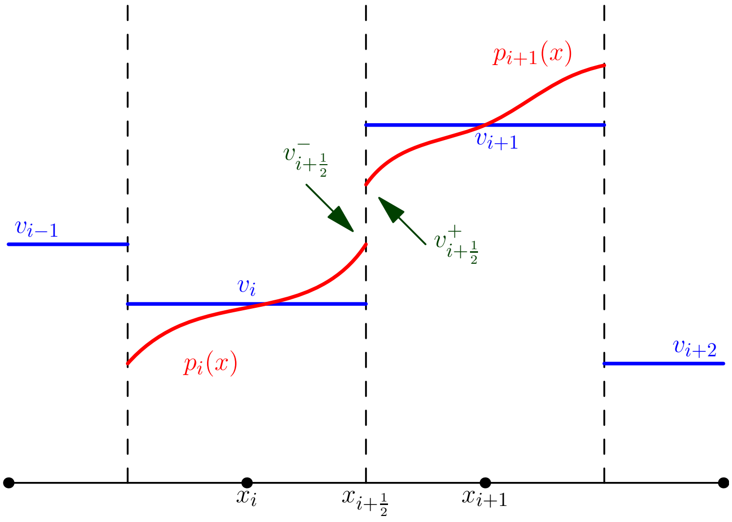


Figure 4.2: Piecewise polynomial reconstruction of the function v in each cell.

carefully, by selecting the smoothest possible stencil. One way to accomplish this is to use the *minmod slope* $\sigma_i = \mathcal{M}(\sigma_i^b, \sigma_i^f)$ where the minmod function is given by

$$\mathcal{M}(a_1, ..., a_k) = \begin{cases} \text{sign}(a_1)\min(|a_1|, ..., |a_k|), & \text{if } \text{sign}(a_1) = ... = \text{ sign}(a_k) \\ 0, & \text{otherwise} \end{cases} . \tag{4.9}$$

The advantage of choosing the minmod slope is that the reconstructed solution has the TVD property. Thus, the reconstructed solution does not introduce any additional oscillations.

An important reconstruction method called the *MUSCL approach* was proposed by Van Leer [127]. The main idea of the scheme is to reconstruct using high-order polynomials, and ensure the reconstruction is TVD by introducing *limiters* based on the local smoothness of the solution. Consider the Taylor series expansion of the exact solution

$$v(x) = v(x_i) + (x - x_i)v_x(x_i) + (x - x_i)^2 v_{xx}(x_i) + \mathcal{O}(\Delta x^3), \tag{4.10}$$

where $v(x_i)$ need not be equal to the cell average v_i. Integrating (4.10) over the cell I_i, we get

$$v(x_i) = v_i - \frac{\Delta x^2}{24} v_{xx}(x_i) + \mathcal{O}(\Delta x^3).$$

Thus, (4.10) can be written as

$$v(x) \quad = \quad v_i + (x - x_i)v_x(x_i) + \frac{1}{2}\left[(x - x_i)^2 - \frac{\Delta x^2}{12}\right] v_{xx}(x_i) + \mathcal{O}(\Delta x^3).$$

This is used to construct the following second degree polynomial

$$p_i(x) \quad = \quad v_i + (x - x_i)\frac{v_{i+1} - v_{i-1}}{2\Delta x} + \frac{3\kappa}{2}\left[(x - x_i)^2 - \frac{\Delta x^2}{12}\right] \frac{v_{i+1} - 2v_i + v_{i-1}}{\Delta x^2},$$

in each cell. Note that p_i satisfies the condition (4.7), and depends on the parameter κ. The reconstruction is second-order accurate for all values of κ, except for $\kappa = 1/3$ for which it is third-order accurate. The cell interface values are given by

$$\begin{aligned} v^+_{i-\frac{1}{2}} &= v_i - \frac{1}{4}\left[(1+\kappa)\Delta v_{i-\frac{1}{2}} + (1-\kappa)\Delta v_{i+\frac{1}{2}}\right], && (4.11)\\ v^-_{i+\frac{1}{2}} &= v_i + \frac{1}{4}\left[(1-\kappa)\Delta v_{i-\frac{1}{2}} + (1+\kappa)\Delta v_{i+\frac{1}{2}}\right]. && (4.12)\end{aligned}$$

In order to make the scheme TVD, the reconstruction is restricted by introducing a limiter function ψ

$$\begin{aligned} v^+_{i-\frac{1}{2}} &= v_i - \frac{1}{4}\left[(1+\kappa)\psi\left(\theta_i^+\right)\Delta v_{i+\frac{1}{2}} + (1-\kappa)\psi(\theta_i^-)\Delta v_{i-\frac{1}{2}}\right], && (4.13)\\ v^-_{i+\frac{1}{2}} &= v_i + \frac{1}{4}\left[(1-\kappa)\psi\left(\theta_i^+\right)\Delta v_{i+\frac{1}{2}} + (1+\kappa)\psi(\theta_i^-)\Delta v_{i-\frac{1}{2}}\right], && (4.14)\end{aligned}$$

where

$$\theta_i^- = \frac{\Delta v_{i+\frac{1}{2}}}{\Delta v_{i-\frac{1}{2}}}, \qquad \theta_i^+ = \frac{1}{\theta_i^-} = \frac{\Delta v_{i-\frac{1}{2}}}{\Delta v_{i+\frac{1}{2}}}, \tag{4.15}$$

are the jump ratios across interfaces, which are a good measure of the local smoothness of the solution. In smooth regions we expect $\theta_i^- \approx 1$ (except at extrema), while near a discontinuity we expect that θ_i^- may be far from 1. Thus, the limiter should take values near 1 in smooth regions, to obtain second order accuracy. We would also like the limiter to be 0 if $\theta_i^- \leqslant 0$, which indicates an extremum. Two important limiters for TVD reconstructions (taking $\kappa = -1$) are

- Minmod limiter

$$\psi_{MM}(R) = \max\left(0, \min\left(R, 1\right)\right).$$

- Van Albada limiter

$$\psi_{VA}(R) = \begin{cases} \frac{R(R+1)}{(1+R^2)}, & \text{if} \quad R \geqslant 0 \\ 0, & \text{if} \quad R < 0 \end{cases}.$$

For other suitable limiters, refer to [44, 72].

4.1.5 Method of lines approach

The fully-discrete schemes discussed in the previous sections are not easy to extend beyond second-order accuracy. It is more useful to adopt the *method of lines* approach, which decouples the space and time discretizations. Integrating the conservation law (4.1) over the cell I_i leads to the semi-discrete conservation law

$$\Delta x \frac{\mathrm{d}\mathbf{U}_i(t)}{\mathrm{d}t} + \mathbf{f}(\mathbf{U}(x_{i+\frac{1}{2}}, t)) - \mathbf{f}(\mathbf{U}(x_{i-\frac{1}{2}}, t)) = 0, \tag{4.16}$$

which is still continuous in time. The flux at the interface $x_{i+\frac{1}{2}}$ is approximated by $\mathbf{F}_{i+\frac{1}{2}}(t) = (\mathbf{U}^-_{i+\frac{1}{2}}(t), \mathbf{U}^+_{i+\frac{1}{2}}(t))$ where the left and right states $\mathbf{U}^-_{i+\frac{1}{2}}(t), \mathbf{U}^+_{i+\frac{1}{2}}(t)$ are obtained from suitable high-order polynomial approximations in each cell. This leads to a scheme which is high-order accurate in space. Finally, the set of ODEs describing the semi-discrete scheme is integrated in time using a high-order time-marching scheme, such as Runge-Kutta schemes.

4.1.6 A general reconstruction strategy

The recontructed values to be used in the semi-discerete scheme, may be obtained using the MUSCL approach described in Section 4.1.4. However, TVD reconstructions can lead to diffusion of shocks and clipping of smooth local extrema [84]. We describe an alternate and more general method of obtaining high-order polynomial approximations of the solutions in each cell, without the use of limiters [106]. The reconstruction problem for finite volume schemes is formulated as follows.

Problem. *Given the cell average values v_i of a function $v(x)$, find a polynomial $p_i(x)$of degree at most $(k-1)$ in each cell I_i such that the following properties hold*

- *The polynomial is a k-th order approximation of the exact function*

$$p_i(x) = v(x) + \mathcal{O}(\Delta x^k), \quad x_{i-\frac{1}{2}} < x < x_{i+\frac{1}{2}}.$$

- *If cell averaged values corresponding to the stencil $S_i = \{x_{i-r}, ..., x_{i+s}\}$ are used to construct the $p_i(x)$, where $r+s+1 = k$, then the polynomial must preserve the cell averages for each cell in the stencil*

$$\frac{1}{\Delta x}\int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} p_i(x)dx = v_j, \quad j = i-1, ..., i+s.$$

This ensures the uniqueness of the polynomial of degree at most $k-1$ on the stencil S_i.

The coefficients of the polynomials $p_i(x)$ can be obtained by first considering the primitive of $v(x)$

$$V(x) = \int\limits_{-\infty}^{x} v(y)\mathrm{d}y,$$

where the lower limit $-\infty$ is not important. Note that the value of the primitive at the cell-interfaces can be written in terms of the cell averages v_i as follows

$$V(x_{i+\frac{1}{2}}) = \sum_{j=-\infty}^{i} \int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} v(y)\mathrm{d}y = \sum_{j=-\infty}^{i} v_j \Delta x.$$

Let $\mathcal{P}_i(x)$ corresponds to the polynomial of degree at most k, which interpolates the primitive at the $k+1$ points $x_{i-r-\frac{1}{2}}, ..., x_{i+s+\frac{1}{2}}$. Then, the required polynomial is obtained as

$$p_i(x) = \mathcal{P}_i'(x)$$

which can be shown to satisfy the conservation property on the stencil $S_i = \{x_{i-r}, ..., x_{i+s}\}$ (see [106] for details). Thus, the coefficients of $p_i(x)$ can be written in terms of the cell-averages v_i.

4.1.7 ENO reconstruction

Constructing high-order polynomials using the technique described in Section 4.1.6 would lead to Gibbs oscillations when the function $v(x)$ contains discontinuities. In the MUSCL approach, these oscillations were avoided with the use of a limiter. Alternately, one could choose the stencil of reconstruction S_i *adaptively*, so as to select the smoothest stencil. This is the idea behind the essentially non-oscillatory (ENO) schemes which where first introduced by Harten et al. [54]. We briefly explain the procedure below, and refer to [106] for further details. We first recall the *Newton divided differences* which will be used to choose the appropriate stencil. The 0-th degree divided difference for the primitive $V(x)$ is given by

$$V[x_{i-\frac{1}{2}}] \equiv V(x_{i-\frac{1}{2}}),$$

while the j-th degree divided difference is obtained in an inductive manner by

$$V[x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}] \equiv \frac{V[x_{i+\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}] - V[x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{3}{2}}]}{x_{i+j-\frac{1}{2}} - x_{i-\frac{1}{2}}}, \quad j \geqslant 1.$$

In a similar manner, we can define the divided difference of the cell averages of $v(x)$ by

$$v[x_i] \equiv v_i, \quad v[x_i, ..., x_{i+j}] \equiv \frac{v[x_{i+1}, ..., x_{i+j}] - v[x_i, ..., x_{i+j-1}]}{x_{i+j} - x_i}, \quad j \geqslant 1.$$

Note that $V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] = v_i$. Thus, the first and higher degree divided differences of $V(x)$ can be written in terms of divided differences of the cell averages of $v(x)$.

The Newton divided differences have the property that

$$V[x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}] = \frac{1}{j!}\frac{\mathrm{d}^j V(\zeta)}{\mathrm{d}x^j},$$

for some $\zeta \in (x_{i-\frac{1}{2}}, x_{i+j-\frac{1}{2}})$, if the function $V(x)$ is smooth in the stencil $\widetilde{S}_i = \{x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}\}$. A proof of this result can be found in any standard numerical analysis text book (for instance see [5]). However, if the stencil contains a point of discontinuity, then it is easy to show that

$$V[x_{i-\frac{1}{2}}, ..., x_{i+j-\frac{1}{2}}] = \mathcal{O}\left(\Delta x^{-j}\right).$$

Thus, the Newton divided difference is a good measure of regularity of the function $V(x)$.

The objective of the ENO algorithm is to find the smoothest stencil of $k+1$ consecutive cell-interface points containing the interfaces $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$, so that a polynomial $V_i(x)$ of degree at most k can be constructed. We begin with the stencil $\widetilde{S}_i = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}$ for $V(x)$. Note that this will correspond to the stencil $S_i = \{x_i\}$ for the cell averages of $v(x)$. Next, we compare the magnitude of the two divided differences $V[x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$ and $V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}}]$. If

$$|V[x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]| < |V[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i-\frac{3}{2}}]|,$$

then we extend the stencil for $V(x)$ to the left to get $\widetilde{S}_i = \{x_{i-\frac{3}{2}}, x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}\}$ (correspondingly $S_i = \{x_{i-1}, x_i\}$). Otherwise, we extend the stencil to the right leading to

$\widetilde{S}_i = \{x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}, x_{i+\frac{3}{2}}\}$ (with $S_i = \{x_i, x_{i+1}\}$). This procedure is continued till $\widetilde{S}_i$ has $k+1$ points, following which the k-th degree polynomial $\mathcal{P}_i(x)$ is constructed using any preferred interpolation technique (for instance using Newton interpolation). Note that the final stencil $\widetilde{S}_i$ for constructing $\mathcal{P}_i(x)$ is one of the k possible stencils that could be chosen, each containing the points $x_{i-\frac{1}{2}}$ and $x_{i+\frac{1}{2}}$. Thus, there are k possible polynomials approximations of degree k (correspondingly degree $k-1$ for $v(x)$) that can be obtained for the cell I_i.

ENO schemes have been quite successful in practice, but can show deterioration in accuracy due to selection of unstable stencils [98]. A modification was proposed by Shu [108], which was able to recover the loss of accuracy. Furthermore, ENO reconstructions have been shown to satisfy an important *sign property*, which is crucial in the construction of high-order entropy stable schemes [36]. The sign-property will be discussed in detail in Chapter 5.

4.1.8 WENO reconstruction

Weighted ENO (WENO) schemes [73, 62] were proposed as an improvement over ENO schemes. The basic idea of WENO is to take a convex combination of all k polynomials involved in the k-th order ENO (ENO-k) approximation in a cell, and obtain a $(2k-1)$-th order approximation. The weights are chosen so as to give the least weight to stencils containing discontinuities. It has been shown in [106] that WENO schemes do not suffer from the deterioration of accuracy faced by ENO schemes. For further details on the implementation of WENO methods for finite volume schemes, we refer to [106].

4.2 Finite difference scheme

Having discussed finite volume schemes, we now briefly describe the formulation of finite difference schemes on uniform Cartesian meshes. For ease of notation, we restrict our discussion to one-dimensional systems of conservation laws. These techniques can be easily extended to the higher-dimensional setting by a dimension-by-dimension treatment. Based on the discussion in Section 4.1.5 for finite volume schemes, we consider the following semi-discrete finite difference scheme approximating (4.1)

$$\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} + \frac{1}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}\right) = 0, \tag{4.17}$$

where $\mathbf{U}_i(t) = \mathbf{U}(x_i, t)$ is the point values of the solution at the cell center x_i, while $\mathbf{F}_{i+\frac{1}{2}} := \mathbf{F}(\mathbf{U}_{i-m+1}, ...\mathbf{U}_{i+m})$ is a $2m$-point numerical flux at the cell interface $x_{i+\frac{1}{2}}$ which is consistent, i.e., $\mathbf{F}(\mathbf{U}, ..., \mathbf{U}) = \mathbf{f}(\mathbf{U})$, for all $\mathbf{U}$. The initial condition for the discrete set-up are taken to be constants $\mathbf{U}_i(0)$ in each cell, with $\mathbf{U}_i(0) = \mathbf{U}_0(x_i)$. For finite difference schemes, the notion of accuracy is understood in terms of the approximation of conservative flux differences

$$\frac{1}{\Delta x}\left(\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}\right) = \partial_x \mathbf{f}(\mathbf{U}_i) + \mathcal{O}(\Delta x^p). \tag{4.18}$$

Higher-order conservative differences can be obtained by following the approach described in [106]. Assume there exists a function $\mathbf{h}(x)$ such that

$$\mathbf{f}(\mathbf{U}(x)) = \frac{1}{\Delta x} \int\limits_{x-\frac{\Delta x}{2}}^{x+\frac{\Delta x}{2}} \mathbf{h}(y) \mathrm{d}y,$$

where we have dropped the notation for the time variable for convenience. Then, the spatial derivative of the flux is given by the exact relation

$$\partial_x \mathbf{f}(\mathbf{U}(x)) = \frac{1}{\Delta x} \left(\mathbf{h} \left(x + \frac{\Delta x}{2} \right) - \mathbf{h} \left(x - \frac{\Delta x}{2} \right) \right).$$

Thus, the numerical flux can be chosen such that

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{h}(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^p). \tag{4.19}$$

Remark 4.2.1. *At first glance, it seems that the approximation error in* (4.19) *needs to be $\mathcal{O}(\Delta x^{p+1})$ to ensure the error in* (4.18) *is $\mathcal{O}(\Delta x^p)$. However, in practice, the $\mathcal{O}(\Delta x^p)$ term in* (4.19) *is found to be smooth, and thus an additional $\mathcal{O}(\Delta x)$ would appear in the differencing of* (4.18) *[106].*

The obvious issue with the approximation (4.19) is that it may not be possible to find the explicit expression for $\mathbf{h}(x)$, even if it exists. This problem can be circumvented by considering the primitive of $\mathbf{h}(x)$

$$\mathbf{H}(x) = \int\limits_{-\infty}^{x} \mathbf{h}(y) \mathrm{d}y,$$

and noting that

$$\mathbf{H}(x_{i+\frac{1}{2}}) = \sum_{j=-\infty}^{i} \int\limits_{x_{j-\frac{1}{2}}}^{x_{j+\frac{1}{2}}} \mathbf{h}(y) \mathrm{d}y = \sum_{j=-\infty}^{i} \mathbf{f}(\mathbf{U}_j) \Delta x.$$

Note that this is identical to the reconstruction problem discussed for the finite volume set-up, where we can consider the the values $\mathbf{f}(\mathbf{U}_i)$ to be the cell-averages of $\mathbf{h}(x)$, whose primitive $\mathbf{H}(x)$ can be expressed at the cell-interfaces in terms of $\mathbf{f}(\mathbf{U}_i)$. Thus, the ENO or WENO reconstruction can be used (component-wise) to obtain the approximation (4.19), without actually requiring the expression for $\mathbf{h}(x)$.

Recall that ENO and WENO methods give rise to two reconstructed values at the cell interface $x_{i+\frac{1}{2}}$, which are obtained from the polynomial approximations in the cell I_i and I_{i+1}. It is unclear a priori which reconstructed value should be chosen. This question can be resolved by introducing *upwinding* into the flux, which is also important to ensure stability of the flux. Upwinding can be achieved by a *flux-splitting* strategy where the flux function is written as

$$\mathbf{f}(\mathbf{U}) = \mathbf{f}^{\mathscr{U}}(\mathbf{U}) + \mathbf{f}^{\mathscr{D}}(\mathbf{U}), \qquad \frac{\mathrm{d}\mathbf{f}^{\mathscr{U}}(\mathbf{U})}{\mathrm{d}\mathbf{U}} \geqslant 0, \qquad \frac{\mathrm{d}\mathbf{f}^{\mathscr{D}}(\mathbf{U})}{\mathrm{d}\mathbf{U}} \leqslant 0,$$

with $\mathbf{f}^{\mathscr{U}}$ being an upwind flux while $\mathbf{f}^{\mathscr{D}}$ is a downwind flux. At each cell-interface, both $\mathbf{f}^{\mathscr{U}}$ and $\mathbf{f}^{\mathscr{D}}$ are approximated via ENO/WENO reconstruction methods using the values $\mathbf{f}^{\mathscr{U}}(\mathbf{U}_i)$ and $\mathbf{f}^{\mathscr{D}}(\mathbf{U}_i)$ respectively. The left reconstruction value is chosen to obtain $\mathbf{F}^{\mathscr{U}}_{i+\frac{1}{2}}$, while the right reconstruction value is used to obtain $\mathbf{F}^{\mathscr{D}}_{i+\frac{1}{2}}$. The final numerical flux is then simply

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}^{\mathscr{U}}_{i+\frac{1}{2}} + \mathbf{F}^{\mathscr{D}}_{i+\frac{1}{2}}.$$

The simplest smooth flux splitting is given by the *Lax-Friedrichs splitting*

$$\mathbf{F}^{\mathscr{U}}(\mathbf{U}) = \frac{1}{2}(\mathbf{f}(\mathbf{U}) + \alpha\mathbf{U}), \quad \mathbf{F}^{\mathscr{D}}(\mathbf{U}) = \frac{1}{2}(\mathbf{f}(\mathbf{U}) - \alpha\mathbf{U}), \quad \alpha = \max_{\mathbf{U}} \max_{j} |\lambda_j(\mathbf{U})|.$$

Other types of splittings are also possible, the details of which can be found in [106].

Remark 4.2.2. *On uniform Cartesian grids, first and second-order finite volume schemes can also be interpreted as finite difference schemes, by replacing the cell-average values by point values, without losing accuracy for smooth solutions. Thus, in literature, finite difference and finite volume methods are at times used interchangeably.*

5. Entropy stable finite difference schemes

The high-order finite difference methods described in Chapter 4 are known to perform well in practice. However, no theoretical results are available about the entropy stability of such schemes. An alternate method of constructing finite difference schemes which are provably entropy stable was proposed by Tadmor [116]. The approach consists of two components: i) constructing a second-order *entropy conservative* scheme that preserves entropy, ii) adding artificial dissipation to get entropy stability. Higher-order entropy conservative finite difference schemes have been constructed in [71]. The design of arbitrarily-high order entropy stable schemes was proposed recently by Fjordholm et al. [35].

5.1 Entropy conservative schemes

Consider the semi-discrete finite difference scheme (4.17) for a one-dimensional system of conservation laws, on a uniform Cartesian mesh. Following the approach described by Tadmor [116], the first step towards constructing an entropy stable scheme is to construct a scheme that conserves entropy.

Definition 5.1.1. *The numerical scheme* (4.17) *is said to be* entropy conservative *if it satisfies the discrete cell entropy relation*

$$\frac{\mathrm{d}\eta(\mathbf{U}_i)}{\mathrm{d}t} + \frac{1}{\Delta x}\big(q^*_{i+\frac{1}{2}} - q^*_{i-\frac{1}{2}}\big) = 0, \tag{5.1}$$

where $q^*_{i+\frac{1}{2}} := q^*(\mathbf{U}_{i-m+1}, ...\mathbf{U}_{i+m})$ *is a consistent numerical entropy flux.*

We introduce the following notations

$$\Delta(\cdot\,)_{i+\frac{1}{2}} = (\cdot\,)_{i+1} - (\cdot\,)_i, \quad \overline{(\cdot\,)}_{i+\frac{1}{2}} = \frac{(\cdot\,)_{i+1} + (\cdot\,)_i}{2},$$

which denote the undivided jump and the average across the interface $x_{i+\frac{1}{2}}$ respectively. Moreover, we introduce the *entropy potential*

$$\Psi(\mathbf{U}) := \langle \mathbf{V}(\mathbf{U}), \mathbf{f}(\mathbf{U})\rangle - q(\mathbf{U}), \tag{5.2}$$

for a given entropy-entropy flux pair (η, q), where $\mathbf{V}$ is the corresponding vector of entropy variables. The following theorem gives a sufficient condition for constructing entropy conservative fluxes.

Theorem 5.1.1 (Tadmor [115]). *The numerical scheme* (4.17) *with the flux* $\mathbf{F}^*_{i+\frac{1}{2}}$ *is entropy conservative if*

$$\left\langle \Delta\mathbf{V}_{i+\frac{1}{2}}, \mathbf{F}^*_{i+\frac{1}{2}} \right\rangle = \Delta\Psi_{i+\frac{1}{2}}, \tag{5.3}$$

where Ψ *is defined by* (5.2)*. Specifically, it satisfies* (5.1) *with the consistent numerical entropy flux given by*

$$q^*_{i+\frac{1}{2}} = \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2}}, \mathbf{F}^*_{i+\frac{1}{2}} \right\rangle - \overline{\Psi}_{i+\frac{1}{2}}.$$

Proof. Taking the scalar product of the finite difference scheme (4.17) with $\mathbf{V}_i$, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{U}_i) = -\frac{1}{\Delta x}\left(\left\langle \mathbf{V}_i, \mathbf{F}_{i+\frac{1}{2}} \right\rangle - \left\langle \mathbf{V}_i, \mathbf{F}_{i-\frac{1}{2}} \right\rangle \right).$$

Assuming (5.3) hold, we have

$$\begin{aligned}\left\langle \mathbf{V}_i, \mathbf{F}_{i+\frac{1}{2}} \right\rangle &= \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2}}, \mathbf{F}_{i+\frac{1}{2}} \right\rangle - \frac{1}{2}\left\langle \Delta\mathbf{V}_{i+\frac{1}{2}}, \mathbf{F}_{i+\frac{1}{2}} \right\rangle \\ &= \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2}}, \mathbf{F}_{i+\frac{1}{2}} \right\rangle - \frac{1}{2}\Delta\Psi_{i+\frac{1}{2}} \\ &= q^*_{i+\frac{1}{2}} + \Psi_i,\end{aligned}$$

and similarly, $\left\langle \mathbf{V}_i, \mathbf{F}_{i-\frac{1}{2}} \right\rangle = q^*_{i-\frac{1}{2}} + \Psi_i$. Thus, the numerical scheme satisfies (5.1). □

For scalar conservation laws, given an entropy function $\eta(U)$, a two-point entropy conservative flux is uniquely determined by the relation

$$F^*_{i+\frac{1}{2}} = \frac{\Delta\Psi_{i+\frac{1}{2}}}{\Delta V_{i+\frac{1}{2}}}.$$

Example 5.1.1. *Consider the linear advection equation with flux* $f(U) = cU$*, where* c *is a constant. Choosing the square entropy* $\eta(U) = U^2/2$ *and the corresponding entropy flux as* $q(U) = cU^2/2$*, we get the entropy conservative flux*

$$F^*_{i+\frac{1}{2}} = c\frac{(U_i + U_{i+1})}{2}. \tag{5.4}$$

Example 5.1.2. *Consider the Burgers equation with flux* $f(U) = U^2/2$*. The square entropy* $\eta(U) = U^2/2$ *and the corresponding entropy flux as* $q(U) = U^3/3$ *results in the entropy conservative flux*

$$F^*_{i+\frac{1}{2}} = \frac{\left(U_i^2 + U_{i+1}^2 + U_iU_{i+1}\right)}{6}. \tag{5.5}$$

For a system of conservation laws with $\mathbf{U} \in \mathbb{R}^m$, the algebraic relation (5.3) is a single equation for the m unknown components of the numerical flux $\mathbf{F}^*_{i+\frac{1}{2}}$, and thus need not have a unique solution. Tadmor [117] proposed the following entropy conservative flux

$$\mathbf{F}^*_{i+\frac{1}{2}} = \int_0^1 \mathbf{f}(\mathbf{V}_{i+\frac{1}{2}}(s))\mathrm{d}s, \quad \mathbf{V}_{i+\frac{1}{2}}(s) = \mathbf{V}_i + s(\mathbf{V}_{i+1} - \mathbf{V}_i). \tag{5.6}$$

However, the integral (5.6) need not admit a closed form expression, and thus needs to be approximated by suitable quadratures. An alternate entropy conservative flux was proposed by Tadmor in [116], in which the straight line path joining the states $\mathbf{V}_i$ and $\mathbf{V}_{i+1}$ is replaced by a piecewise linear path along a set of linearly independent vectors. Let $\{\mathbf{r}^k\}$ be an arbitrary set of m linearly independent vectors, and let $\{\boldsymbol{\ell}^k\}$ be the corresponding orthogonal set satisfying $\left\langle \boldsymbol{\ell}^j, \mathbf{r}^k \right\rangle = \delta_{jk}$. Define

$$\mathbf{V}^0 = \mathbf{V}_i, \qquad \mathbf{V}^j = \mathbf{V}^{j-1} + \left\langle \Delta \mathbf{V}_{i+\frac{1}{2}}, \boldsymbol{\ell}^j \right\rangle \mathbf{r}^j, \quad j = 1, ..., m-1, \qquad \mathbf{V}^m = \mathbf{V}_{i+1}.$$

Then, the entropy conservative flux is obtained as

$$\mathbf{F}^*_{i+\frac{1}{2}} = \sum_{k=1}^{m} \frac{\Psi(\mathbf{V}^k) - \Psi(\mathbf{V}^{k-1})}{\left\langle \Delta \mathbf{V}_{i+\frac{1}{2}}, \boldsymbol{\ell}^k \right\rangle} \boldsymbol{\ell}^k, \tag{5.7}$$

which satisfies (5.3) [116]. The advantage of the flux (5.7) over (5.6) is that it is explicit and does not require any additional quadrature approximations. However, (5.7) may be computationally expensive and numerically unstable [125, 35]. Thus, we consider numerical fluxes constructed directly to satisfy the algebraic relation (5.3), for a given system of conservation laws. In particular, we focus on the one-dimensional Euler equations, for which the conserved variables and flux function are given by

$$\mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \quad \mathbf{f}(\mathbf{U}) = \begin{pmatrix} f^\rho \\ f^m \\ f^e \end{pmatrix} = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}. \tag{5.8}$$

5.2 Kinetic energy preserving for Euler equations

In addition to entropy stability, an important property that is desirable for a numerical scheme for the Euler equations is kinetic energy preservation. In other words, the numerical scheme should evolve the kinetic energy at the discrete level in a manner that is consistent with (3.18) (or (3.19) in one-dimension), ignoring the viscous terms. The evolution of total kinetic energy at the discrete level for the scheme (4.17), with numerical flux $\mathbf{F} = (F^\rho,\ F^m,\ F^e)^\top$, is given by

$$\begin{aligned}
\sum_i \Delta x \frac{\mathrm{d}\mathcal{K}_i}{\mathrm{d}t} &= \sum_i \left[-\frac{1}{2} u_i^2 \frac{\mathrm{d}\rho_i}{\mathrm{d}t} + u_i \frac{\mathrm{d}}{\mathrm{d}t} (\rho u)_i \right] \Delta x \\
&= \sum_i \left[\frac{1}{2} u_i^2 (F^\rho_{i+\frac{1}{2}} - F^\rho_{i-\frac{1}{2}}) - u_i (F^m_{i+\frac{1}{2}} - F^m_{i-\frac{1}{2}}) \right] \\
&= \sum_i \left[\frac{1}{2} (u_i^2 - u_{i+1}^2) F^\rho_{i+\frac{1}{2}} - (u_i - u_{i+1}) F^m_{i+\frac{1}{2}} \right] \\
&= \sum_i \frac{\Delta u_{i+\frac{1}{2}}}{\Delta x} \left[\overline{u}_{i+\frac{1}{2}} F^\rho_{i+\frac{1}{2}} - F^m_{i+\frac{1}{2}} \right] \Delta x.
\end{aligned}$$

Thus, if we choose

$$F^m = \widetilde{p} + \overline{u} F^\rho, \tag{5.9}$$

for some consistent approximations for $\widetilde{p}$ and F^ρ, then we get the discrete evolution equation

$$\sum_i \Delta x \frac{\mathrm{d}\mathcal{K}_i}{\mathrm{d}t} \quad = \quad \sum_i \frac{\Delta u_{i+\frac{1}{2}}}{\Delta x} \widetilde{p}_{i+\frac{1}{2}} \Delta x,$$

which is consistent with (3.19). The sufficient condition (5.9) was proposed by Jameson [59] to construct fluxes for the Euler equations that preserve kinetic energy.

5.3 Numerical fluxes for the Euler equations

We now describe a few important two-point numerical fluxes for the Euler equations, that are kinetic energy preserving and/or entropy conservative.

5.3.1 Roe's entropy conservative flux

Ismail and Roe [58] have constructed a numerical flux by introducing the parameter vector

$$\mathbf{Z} = \begin{pmatrix} Z_1, & Z_2, & Z_3 \end{pmatrix}^\top = \sqrt{\frac{\rho}{p}} \begin{pmatrix} 1, & u, & p \end{pmatrix}^\top,$$

and writing the flux in terms of $\mathbf{Z}$ in order to satisfy (5.3). The flux has the expression

$$\mathbf{F} = \begin{pmatrix} F^\rho \\ F^m \\ F^e \end{pmatrix} = \begin{pmatrix} \overline{Z_2}\widehat{Z_3} \\ \frac{\overline{Z_3}}{\overline{Z_1}} + \frac{\overline{Z_2}}{\overline{Z_1}} F^\rho \\ F^e \end{pmatrix}, \quad F^e = \frac{1}{2\overline{Z_1}} \left[\frac{(\gamma+1)}{(\gamma-1)} \frac{F^\rho}{\widehat{Z_1}} + \overline{Z_2} F^m \right], \tag{5.10}$$

where

$$\widehat{\phi}_{i+\frac{1}{2}} = \frac{\phi_{i+1} - \phi_i}{\ln(\phi_{i+1}) - \ln(\phi_i)},$$

is the *logarithmic average*, and is well defined for strictly positive quantities ϕ. A numerically stable procedure to evaluate the log average when ϕ_i and ϕ_{i+1} are almost equal is given in [58], and is also mentioned in Appendix B. The flux (5.10) will be referred to as the ROE-EC flux. Note that the ROE-EC flux is not kinetic energy preserving as it does not satisfy the condition (5.9).

5.3.2 Jameson's kinetic energy preserving flux

Jameson [59] proposed the following simple central flux

$$\mathbf{F} = \begin{pmatrix} F^\rho \\ F^m \\ F^e \end{pmatrix} = \begin{pmatrix} \overline{\rho}\,\overline{u} \\ \overline{p} + \overline{u} F^\rho \\ \overline{\rho} \overline{H} \overline{u} \end{pmatrix}, \tag{5.11}$$

which clearly satisfies the condition (5.9), and is thus kinetic energy preserving. However, the flux does not satisfy sufficient condition (5.3) required to ensure entropy conservation. We demonstrate through numerical experiments in Section 5.8.4 that the flux (5.11) need not conserve entropy for smooth solutions. The flux (5.11) will be referred to as the KEP flux.

5.3.3 Kinetic energy and entropy conservative flux

An entropy conservative flux satisfying the condition (5.3) was proposed in [16], whose expression is given by

$$\mathbf{F} = \begin{pmatrix} F^\rho \\ F^m \\ F^e \end{pmatrix} = \begin{pmatrix} \widehat{\rho}\overline{u} \\ \widetilde{p} + \overline{u}F^\rho \\ F^e \end{pmatrix}, \; F^e = \left[\frac{1}{2(\gamma-1)\widehat{\beta}} - \frac{1}{2}\overline{|u|^2} \right] F^\rho + \overline{u}F^m, \tag{5.12}$$

where $\widetilde{p} = \overline{\rho}/(2\overline{\beta})$, while $\widehat{\rho}, \widehat{\beta}$ are the logarithmic averages of the respective quantities. Note that the flux (5.12) satisfies the condition (5.9), and is thus kinetic energy preserving as well. This flux will be referred to as the KEPEC flux.

5.4 High-order entropy conservative fluxes

The numerical fluxes described in Section 5.3 are only second-order accurate [116]. Lefloch et al. [71] proposed a method to construct high-order entropy conservative fluxes, by using a suitable linear combination of two-point second-order entropy conservative fluxes. This approach is outlined in the following theorem.

Theorem 5.4.1 ([71])**.** *For $p \in \mathbb{N}$, assume that $\alpha_1^p, \alpha_2^p, ..., \alpha_p^p$ solve the linear equations*

$$2\sum_{r=1}^{p} r\alpha_r^p = 1, \qquad \sum_{l=1}^{p} l^{2s-1}\alpha_r^p = 0 \quad (s = 2, ..., p), \tag{5.13}$$

and define the flux

$$\mathbf{F}^{*,2p}_{i+\frac{1}{2}} = \mathbf{F}^{*,2p}(\mathbf{U}_{i-p+1}..., \mathbf{U}_{i+p}) = \sum_{r=1}^{p} \alpha_r^p \sum_{s=0}^{r-1} \mathbf{F}^*(\mathbf{U}_{i-s}, \mathbf{U}_{i-s+r}), \tag{5.14}$$

where $\mathbf{F}^*$ is a two-point second-order entropy conservative flux satisfying (5.3)*. Then the semi-discrete scheme* (4.17) *with flux $\mathbf{F}^{*,2p}$ is*

1. *2pth order accurate, i.e., for sufficiently smooth solutions $\mathbf{U}$ we have*

$$\frac{\mathbf{F}^{*,2p}_{i+\frac{1}{2}} - \mathbf{F}^{*,2p}_{i-\frac{1}{2}}}{h} = \partial_x \mathbf{F}(\mathbf{U}_i) + \mathcal{O}(h^{2p}).$$

2. *Entropy conservative, i.e., it satisfies the discrete entropy identity*

$$\frac{d}{dt}\eta_i + \frac{1}{h}\left(q^{*,2p}_{i+\frac{1}{2}} - q^{*,2p}_{i-\frac{1}{2}} \right) = 0,$$

where

$$q^{*,2p}_{i+\frac{1}{2}} = \sum_{r=1}^{p} \alpha_r^p \sum_{s=0}^{r-1} q^*(\mathbf{U}_{i-s}, \mathbf{U}_{i-s+r}),$$

with q^* being the consistent numerical entropy flux corresponding to $\mathbf{F}^*$.

As an example, the expression for the fourth-order entropy conservative flux is given by

$$\mathbf{F}^{*,4}_{i+\frac{1}{2}} = \frac{4}{3}\mathbf{F}^{*}(\mathbf{U}_i, \mathbf{U}_{i+1}) - \frac{1}{6}\left(\mathbf{F}^{*}(\mathbf{U}_{i-1}, \mathbf{U}_{i+1}) + \mathbf{F}^{*}(\mathbf{U}_i, \mathbf{U}_{i+2})\right). \tag{5.15}$$

Remark 5.4.1. *For any* $2p$*-th order accurate entropy conservative flux of the form* (5.14)*, the following condition holds*

$$\left\langle \mathbf{V}_i, \mathbf{F}^{*,2p}_{i+\frac{1}{2}} - \mathbf{F}^{*,2p}_{i-\frac{1}{2}} \right\rangle = q^{*,2p}_{i+\frac{1}{2}} - q^{*,2p}_{i-\frac{1}{2}}. \tag{5.16}$$

Remark 5.4.2. *In principle, any second-order numerical flux can be used in* (5.14) *to obtain a* $2p$*-th order accurate flux. However, the high-order flux obtained cannot be guaranteed to be entropy conservative unless the base flux is entropy conservative.*

5.5 Entropy stable schemes

The fluxes discussed in Sections 5.3 and 5.4 perform well while approximating smooth solutions of conservation laws. However, they lead to high-frequency Gibbs oscillations near discontinuities. Thus, we need to introduce additional dissipation terms to control the oscillations. Furthermore, while entropy is conserved for smooth solutions, it must be dissipated near discontinuities in accordance to the entropy condition (2.9). Hence, the artificial diffusion terms need to be carefully chosen to ensure that entropy condition is satisfied at the discrete level.

Definition 5.5.1. *The numerical scheme* (4.17) *is said to be entropy stable if it satisfies the discrete cell entropy relation*

$$\frac{d\eta(\mathbf{U}_i)}{dt} + \frac{1}{\Delta x}\left(q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}\right) \leqslant 0, \tag{5.17}$$

where $q_{i+\frac{1}{2}} := q(\mathbf{U}_{i-m+1}, ...\mathbf{U}_{i+m})$ *is a consistent numerical entropy flux.*

An important class of entropy stable schemes has been proposed by Tadmor [116, 35], where an entropy variable based numerical dissipation term is augmented to the entropy conservative numerical flux $\mathbf{F}^{*,2p}_{i+\frac{1}{2}}$, to obtain

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}^{*,2p}_{i+\frac{1}{2}} - \frac{1}{2}\mathbf{D}_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}. \tag{5.18}$$

The dissipation matrix $\mathbf{D}_{i+\frac{1}{2}}$ in (5.18) must be symmetric and positive semi-definite, i.e., $\mathbf{D}_{i+\frac{1}{2}} = \mathbf{D}^{\top}_{i+\frac{1}{2}} \geqslant 0$.

Theorem 5.5.1. *The semi-discrete numerical scheme* (4.17) *with numerical flux* (5.18) *is entropy stable. Specifically, it satisfies the discrete entropy inequality* (5.17)*, with a consistent numerical entropy flux given by*

$$q_{i+\frac{1}{2}} = q^{*,2p}_{i+\frac{1}{2}} - \frac{1}{2}\overline{\mathbf{V}}_{i+\frac{1}{2}}\mathbf{D}_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}.$$

Proof. Taking the scalar product of the finite difference scheme (4.17) with $\mathbf{V}_i$ and using (5.16), we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{U}_i) = -\frac{1}{\Delta x}\left(q^{*,2p}_{i+\frac{1}{2}} - q^{*,2p}_{i-\frac{1}{2}}\right) + \frac{1}{2\Delta x}\left\langle \mathbf{V}_i, \mathbf{D}_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}\right\rangle - \frac{1}{2\Delta x}\left\langle \mathbf{V}_i, \mathbf{D}_{i-\frac{1}{2}}\Delta\mathbf{V}_{i-\frac{1}{2}}\right\rangle.$$

Since $\mathbf{D}_{i+\frac{1}{2}} \geqslant 0$, we have

$$\begin{aligned}\left\langle \mathbf{V}_i, \mathbf{D}_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}\right\rangle &= \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2}}, \mathbf{D}_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}\right\rangle - \frac{1}{2}\left\langle \Delta\mathbf{V}_{i+\frac{1}{2}}, \mathbf{D}_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}\right\rangle \\ &\leqslant \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2}}, \mathbf{D}_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}\right\rangle,\end{aligned}$$

and

$$\begin{aligned}\left\langle \mathbf{V}_i, \mathbf{D}_{i-\frac{1}{2}}\Delta\mathbf{V}_{i-\frac{1}{2}}\right\rangle &= \left\langle \overline{\mathbf{V}}_{i-\frac{1}{2}}, \mathbf{D}_{i-\frac{1}{2}}\Delta\mathbf{V}_{i-\frac{1}{2}}\right\rangle + \frac{1}{2}\left\langle \Delta\mathbf{V}_{i-\frac{1}{2}}, \mathbf{D}_{i-\frac{1}{2}}\Delta\mathbf{V}_{i-\frac{1}{2}}\right\rangle \\ &\geqslant \left\langle \overline{\mathbf{V}}_{i-\frac{1}{2}}, \mathbf{D}_{i-\frac{1}{2}}\Delta\mathbf{V}_{i-\frac{1}{2}}\right\rangle.\end{aligned}$$

Thus, we get the required result (5.17). □

5.5.1 Dissipation operator

To construct the dissipation matrix $\mathbf{D}_{i+\frac{1}{2}}$, we take inspiration from Roe's approximate Riemann solver [95], which is based on the linearization of the conservation law about some average state. The numerical flux of the Roe scheme has the form

$$\mathbf{F}_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{f}(\mathbf{U}_{i+1}) + \mathbf{f}(\mathbf{U}_i)\right) - \frac{1}{2}\mathbf{R}_{i+\frac{1}{2}}\mathbf{\Lambda}_{i+\frac{1}{2}}\mathbf{R}^{-1}_{i+\frac{1}{2}}\Delta\mathbf{U}_{i+\frac{1}{2}}, \tag{5.19}$$

where $\mathbf{R}$ is the matrix of right eigenvectors of the flux Jacobian $\partial_{\mathbf{U}}\mathbf{f}(\mathbf{U})$ and $\mathbf{\Lambda} = \mathbf{\Lambda}(\mathbf{U})$ is the non-negative diagonal matrix

$$\mathbf{\Lambda}^{Roe} = \operatorname{diag}\Big(|\lambda_1|, \ldots, |\lambda_n|\Big), \tag{5.20}$$

with λ_k being the eigenvalues of the flux Jacobian. These matrices and eigenvalues are evaluated at some average state depending on the state $\mathbf{U}_i$ and $\mathbf{U}_{i+1}$.

The dissipation in (5.19) can be written approximately in terms of the jump in the entropy variables, by linearizing the jump in the conserved variables as $\Delta\mathbf{U} = \partial_{\mathbf{V}}\mathbf{U}\Delta\mathbf{V}$, where the Jacobian $\partial_{\mathbf{V}}\mathbf{U}$ is symmetric positive definite [116]. The eigenvector rescaling theorem [8] ensures the existence of a scaling of the eigenvectors $\mathbf{R} \to \widetilde{\mathbf{R}}$, such that $\partial_{\mathbf{V}}\mathbf{U} = \widetilde{\mathbf{R}}\widetilde{\mathbf{R}}^\top$. The Roe-type flux can thus be rewritten as

$$\mathbf{F}_{i+\frac{1}{2}} = \frac{1}{2}\left(\mathbf{f}(\mathbf{U}_{i+1}) + \mathbf{f}(\mathbf{U}_i)\right) - \frac{1}{2}\widetilde{\mathbf{R}}_{i+\frac{1}{2}}\mathbf{\Lambda}_{i+\frac{1}{2}}\widetilde{\mathbf{R}}^\top_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}.$$

This motivates us to choose the *Roe-type* diffusion operator [35]

$$\mathbf{D}_{i+\frac{1}{2}} = \widetilde{\mathbf{R}}_{i+\frac{1}{2}}\mathbf{\Lambda}_{i+\frac{1}{2}}\widetilde{\mathbf{R}}^\top_{i+\frac{1}{2}}, \tag{5.21}$$

which is clearly symmetric and positive semi-definite. For the one-dimensional Euler equations, these matrices are given by

$$\mathbf{R} = \begin{pmatrix} 1 & 1 & 1 \\ u-a & u & u+a \\ H-ua & \frac{1}{2}u^2 & H+ua \end{pmatrix}, \quad \mathbf{\Lambda} = \mathbf{\Lambda}^{Roe} = \operatorname{diag}\Big(|u-a|,\ |u|,\ |u+a|\Big),$$
$$\mathbf{S} = \operatorname{diag}\Big(\frac{\rho}{2\gamma},\ \frac{(\gamma-1)\rho}{\gamma},\ \frac{\rho}{2\gamma}\Big), \quad \widetilde{\mathbf{R}} = \mathbf{R}\mathbf{S}^{\frac{1}{2}}. \tag{5.22}$$

Note that the matrix $\mathbf{R}$ can be obtained from (3.6) by choosing $\mathbf{n} = (1,0,0)$ and removing the third and fourth rows and columns. For convenience we will drop the $\widetilde{(.)}$ notation, where it will be understood that $\mathbf{R}_{i+\frac{1}{2}}$ denotes the scaled eigenvectors.

Remark 5.5.1. *It has been shown in [16] that if the KEPEC flux is used with dissipation matrix of the form* (5.22)*, then stationary contact waves are resolved exactly provided the enthalpy* $H_{i+\frac{1}{2}}$ *and speed of sound* $a_{i+\frac{1}{2}}$ *in* (5.22) *are evaluated as*

$$H_{i+\frac{1}{2}} = \frac{1}{\gamma-1}a^2_{i+\frac{1}{2}} + \frac{1}{2}u^2_{i+\frac{1}{2}}, \quad a_{i+\frac{1}{2}} = \sqrt{\frac{\gamma}{2\widehat{\beta}_{i+\frac{1}{2}}}}, \tag{5.23}$$

with the remaining averages $u_{i+\frac{1}{2}}$ *and* $\rho_{i+\frac{1}{2}}$ *chosen in any consistent manner.*

Remark 5.5.2. *The Roe type dissipation operator, as chosen above, is just one of a whole host of options available. In particular, we can choose* $\mathbf{\Lambda}^{Rus} = \max_i |\lambda_i|\mathbf{I}$ *to obtain a Rusanov-type diffusion operator, which is much more dissipative as compared to the Roe-type diffusion operator. Further examples of polynomial viscosity operators are provided in [35].*

In [58], Ismail and Roe attempted to answer the question about "how much" entropy should actually be dissipated across shocks. They claimed that the entropy contained within a shock should neither be too great nor too small, or else the shock profile will be oscillatory or smeared out respectively. With a *weak shock* assumption, it is possible to show that the entropy is dissipated as $\mathcal{O}(|\Delta \mathbf{U}|^3)$. Based on this fact, they recommended the following modification to the dissipation operator for the Euler equations, to ensure that the scheme is *entropy consistent* – it dissipates entropy at the correct rate

$$\mathbf{\Lambda}^{mod}_{i+\frac{1}{2}} = \mathbf{\Lambda}_{i+\frac{1}{2}} + \alpha\tilde{\mathbf{\Lambda}}_{i+\frac{1}{2}}, \qquad \tilde{\mathbf{\Lambda}}_{i+\frac{1}{2}} = \operatorname{diag}\Big(\Delta(u-a)_{i+\frac{1}{2}},\ 0,\ \Delta(u+a)_{i+\frac{1}{2}}\Big). \tag{5.24}$$

The value $\alpha = \frac{1}{6}$ has been suggested in [58] based on the weak shock assumption. While it is not possible to make an analytic connection between entropy dissipation and the shock quality, they justify their claims through numerical experiments.

5.5.2 High-order diffusion operators

For smooth solutions, $\Delta \mathbf{V}_{i+\frac{1}{2}} = \mathcal{O}(\Delta x)$, thus making the diffusion term in the flux (5.18) (and hence the flux itself) only first-order accurate. The first-order scheme is a consequence of taking the solution to be constant in each cell, and equal to the value at the cell-center. In order to obtain a higher order scheme, we need to appropriately reconstruct the solution to the cell interfaces. Let $\mathcal{V}_i(x)$ and $\mathcal{V}_{i+1}(x)$ be polynomial reconstructions in the cells I_i and I_{i+1} respectively, as shown in Figure 5.1. Define the reconstructed values at the interface $x_{i+\frac{1}{2}}$ and the reconstructed jump by

$$\mathbf{V}^-_{i+\frac{1}{2}} = \mathcal{V}_i(x_{i+\frac{1}{2}}), \quad \mathbf{V}^+_{i+\frac{1}{2}} = \mathcal{V}_{i+1}(x_{i+\frac{1}{2}}), \quad [\![\mathbf{V}]\!]_{i+\frac{1}{2}} := \mathbf{V}^+_{i+\frac{1}{2}} - \mathbf{V}^-_{i+\frac{1}{2}}.$$

We use the above high-order jump in the numerical flux (5.18) instead of $\Delta\mathbf{V}_{i+\frac{1}{2}}$. For instance, if the reconstruction is exact for polynomials of degree $2p-1$, then $[\![\mathbf{V}]\!]_{i+\frac{1}{2}} = \mathcal{O}(\Delta\mathbf{x}_{i+\frac{1}{2}})^{2p}$ for smooth functions, thus making the flux $2p$-th accurate (provided the entropy conservative flux is also $2p$-th order accurate). The following lemma gives a sufficient condition to ensure that the reconstruction does not lead to the violation of entropy stability.

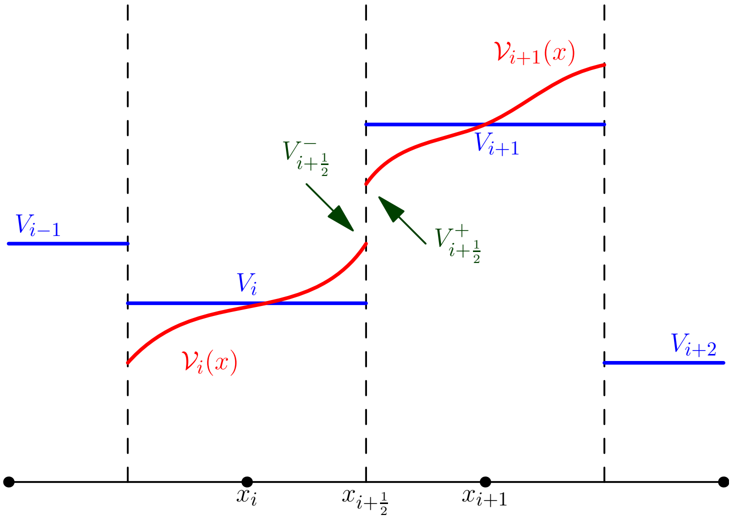


Figure 5.1: Piecewise polynomial reconstruction of the entropy variables V in each cell.

Lemma 5.5.1 (Fjordholm et al.[35]). *For each interface $x_{i+\frac{1}{2}}$, let $\mathbf{R}_{i+\frac{1}{2}}$ be non-singular and $\mathbf{\Lambda}_{i+\frac{1}{2}}$ be any non-negative diagonal matrix. Define the numerical diffusion matrix*

$$\mathbf{D}_{i+\frac{1}{2}} = \mathbf{R}_{i+\frac{1}{2}}\mathbf{\Lambda}_{i+\frac{1}{2}}\mathbf{R}^\top_{i+\frac{1}{2}}.$$

Let $\mathbf{V}^-_{i+\frac{1}{2}}$ and $\mathbf{V}^+_{i+\frac{1}{2}}$ be the reconstructed values of the entropy variables at the interface, from cells I_i and I_{i+1} respectively. Assume that the reconstruction ensures the existence of a diagonal matrix $\mathbf{B}_{i+\frac{1}{2}} \geqslant 0$ such that

$$[\![\mathbf{V}]\!]_{i+\frac{1}{2}} = \left(\mathbf{R}^\top_{i+\frac{1}{2}}\right)^{-1}\mathbf{B}_{i+\frac{1}{2}}\mathbf{R}^\top_{i+\frac{1}{2}}\Delta\mathbf{V}_{i+\frac{1}{2}}. \tag{5.25}$$

Then the scheme with the numerical flux

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}^{*,2p}_{i+\frac{1}{2}} - \frac{1}{2}\mathbf{D}_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}}, \tag{5.26}$$

is entropy stable, with the consistent numerical entropy flux

$$q_{i+\frac{1}{2}} := q^{*,2p}_{i+\frac{1}{2}} - \frac{1}{2}\overline{\mathbf{V}}^\top_{i+\frac{1}{2}}\mathbf{D}_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}}.$$

Proof. As in the proof of Lemma 5.5.1, consider (4.17) with the flux defined by (5.26), and take the scalar product with the entropy variables $\mathbf{V}_i$ to get

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{U}_i) = -\frac{1}{\Delta x}\left(q^{*,2p}_{i+\frac{1}{2}} - q^{*,2p}_{i-\frac{1}{2}}\right) + \frac{1}{2}\left\langle \mathbf{V}_i, \mathbf{D}_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}}\right\rangle - \frac{1}{2}\left\langle \mathbf{V}_i, \mathbf{D}_{i-\frac{1}{2}}[\![\mathbf{V}]\!]_{i-\frac{1}{2}}\right\rangle.$$

Now,

$$\left\langle \mathbf{V}_i, \mathbf{D}_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}}\right\rangle = \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2}}, \mathbf{D}_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}}\right\rangle - \frac{1}{2}\left\langle \Delta\mathbf{V}_{i+\frac{1}{2}}, \mathbf{R}_{i+\frac{1}{2}}\mathbf{\Lambda}_{i+\frac{1}{2}}\mathbf{B}_{i+\frac{1}{2}}\mathbf{R}^\top_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}}\right\rangle.$$

Since $\mathbf{R}_{i+\frac{1}{2}}\mathbf{\Lambda}_{i+\frac{1}{2}}\mathbf{B}_{i+\frac{1}{2}}\mathbf{R}^\top_{i+\frac{1}{2}}$ is symmetric positive semi-definite, we get

$$\left\langle \mathbf{V}_i, \mathbf{D}_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}}\right\rangle \leqslant \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2}}, \mathbf{D}_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}}\right\rangle.$$

Similarly,

$$\left\langle \mathbf{V}_i, \mathbf{D}_{i-\frac{1}{2}}[\![\mathbf{V}]\!]_{i-\frac{1}{2}}\right\rangle \geqslant \left\langle \overline{\mathbf{V}}_{i-\frac{1}{2}}, \mathbf{D}_{i-\frac{1}{2}}[\![\mathbf{V}]\!]_{i-\frac{1}{2}}\right\rangle.$$

Thus, the entropy inequality (5.17) is satisfied. □

5.5.3 Reconstruction procedure

In order to use Lemma 5.5.1, we describe a reconstruction procedure that satisfies (5.25). For each cell interface $x_{i+\frac{1}{2}}$, define the *scaled entropy variables* $\mathbf{Z} = \mathbf{R}^\top_{i+\frac{1}{2}}\mathbf{V}$. Let $\mathbf{Z}^-_{i+\frac{1}{2}}$, $\mathbf{Z}^+_{i+\frac{1}{2}}$ be the reconstructed values of $\mathbf{Z}$ at the interface from cells I_i and I_{i+1} respectively. Define

$$\mathbf{V}^-_{i+\frac{1}{2}} = (\mathbf{R}^\top_{i+\frac{1}{2}})^{-1}\mathbf{Z}^-_{i+\frac{1}{2}}, \quad \mathbf{V}^+_{i+\frac{1}{2}} = (\mathbf{R}^\top_{i+\frac{1}{2}})^{-1}\mathbf{Z}^+_{i+\frac{1}{2}} \qquad \Longrightarrow \qquad [\![\mathbf{V}]\!]_{i+\frac{1}{2}} = (\mathbf{R}^\top_{i+\frac{1}{2}})^{-1}[\![\mathbf{Z}]\!]_{i+\frac{1}{2}}.$$

Thus, the dissipation terms in the flux given by (5.26) can be written as

$$\mathbf{D}_{i+\frac{1}{2}}[\![\mathbf{V}]\!]_{i+\frac{1}{2}} = \mathbf{R}_{i+\frac{1}{2}}\mathbf{\Lambda}_{i+\frac{1}{2}}[\![\mathbf{Z}]\!]_{i+\frac{1}{2}}.$$

The condition given by (5.25) can now be interpreted in terms of the scaled variables as

$$[\![\mathbf{Z}]\!]_{i+\frac{1}{2}} = \mathbf{B}_{i+\frac{1}{2}}\Delta\mathbf{Z}_{i+\frac{1}{2}},$$

for some diagonal matrix $\mathbf{B}_{i+\frac{1}{2}}$ with non-negative entries. This further reduces to a *sign property* on $\mathbf{Z}$ which holds component-wise:

$$\operatorname{sign}\left([\![\mathbf{Z}]\!]_{i+\frac{1}{2}}\right) = \operatorname{sign}\left(\Delta\mathbf{Z}_{i+\frac{1}{2}}\right). \tag{5.27}$$

Figure 5.2(a) shows an example of a reconstruction (in $\mathbf{Z}$) satisfying the sign property, while Figure 5.2(b) gives an example of a reconstruction violating the sign property.

High-order entropy stable schemes, based on a combination of high-order entropy conservative fluxes and high-order dissipation operators obtained by using a sign-preserving reconstruction of the scaled entropy variables, are termed as *TeCNO schemes* [35].

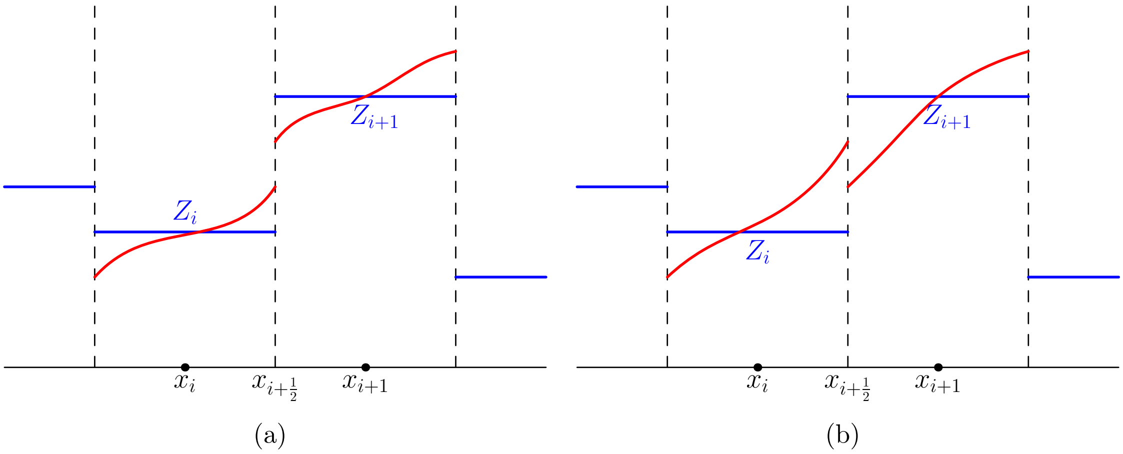


Figure 5.2: Demonstration of a reconstruction a) satisfying the sign property, b) violating the sign property.

5.5.4 Reconstruction with sign property

Only a small class of reconstructions are known to satisfy the sign-property.

Minmod reconstruction

A linear reconstruction of $\mathbf{Z}$ (component-wise) with the minmod limiter (4.9), leads to second-order (limited) reconstructed states

$$\mathbf{Z}^-_{i+\frac{1}{2}} = \mathbf{Z}_i + \frac{1}{2}\mathcal{M}\left(\Delta\mathbf{Z}_{i+\frac{1}{2}}, \Delta\mathbf{Z}_{i-\frac{1}{2}}\right), \quad \mathbf{Z}^+_{i+\frac{1}{2}} = \mathbf{Z}_{i+1} - \frac{1}{2}\mathcal{M}\left(\Delta\mathbf{Z}_{i+\frac{3}{2}}, \Delta\mathbf{Z}_{i+\frac{1}{2}},\right).$$

The fact that the minmod limiter satisfies the sign-property has already been proved on Cartesian grids [35]. This is also proved on unstructured grids in Chapter 9.

ENO interpolation

In Section 4.1.7, we described the ENO reconstruction procedure using cell averages, which additionally satisfies the sign-property [36]. The procedure can be used to get high-order finite difference schemes by treating the value of the flux $\mathbf{f}(\mathbf{U})$ at the cell-centers to be cell-averages of another function (See Section 4.2). However, we cannot prove the resultant scheme to be entropy stable.

We can instead consider the *ENO interpolation* [36] technique, which makes use of point values of the solution. We briefly describe the algorithm for ENO interpolation. Let $v(x)$ be a function, with $v_i = v(x_i)$ denoting its cell-center values. We wish to construct a polynomial $p_i(x)$ in the cell I_i such that $p_i(x_i) = v_i$ and

$$p_i(x_{i-\frac{1}{2}}) = v(x_{i-\frac{1}{2}}) + \mathcal{O}(\Delta x^k), \quad p_i(x_{i+\frac{1}{2}}) = v(x_{i+\frac{1}{2}}) + \mathcal{O}(\Delta x^k).$$

In order to find such polynomial of degree at most $k-1$, which leads to a k-th order approximation of $v(x)$ in the cell I_i, we need a stencil S_i with k points. We make use of the Newton divided differences

$$v[x_i] \equiv v_i, \quad v[x_i, ..., x_{i+j}] \equiv \frac{v[x_{i+1}, ..., x_{i+j}] - v[x_i, ..., x_{i+j-1}]}{x_{i+j} - x_i}.$$

to adaptively select the smoothest stencil of k consecutive cell-centers. We begin with the stencil $S_i = \{x_i\}$ and compare the magnitude of the two divided differences $v[x_{i-1}, x_i]$ and $v[x_i, x_{i+1}]$. If

$$|v[x_{i-1}, x_i]| < |v[x_i, x_{i+1}]|,$$

then we extend the stencil to the left and get $S_i = \{x_{i-1}, x_i\}$. Otherwise, we extend the stencil to the right, leading to $S_i = \{x_i, x_{i+1}\}$. This procedure is continued till S_i has k points, following which the polynomial $p_i(x)$ is constructed using any preferred interpolation technique. As was the case with ENO reconstruction, there are k possible polynomial approximations of degree $k-1$ that can be constructed in the cell I_i. The ENO interpolation algorithm picks the polynomial corresponding to the smoothest stencil. ENO interpolation has been shown to satisfy the sign property in [36], and can thus be used to obtain high-order accurate entropy stable schemes.

A third-order sign-preserving reconstruction based on appropriate limiting of quadratic polynomials, has also been recently proposed in [18]. To the best of our knowledge, no other known reconstruction satisfies this crucial property. In Chapter 6, we propose a third-order WENO-type interpolation procedure that satisfies the sign-property.

5.6 Viscous flux discretization

In this section we discretize the viscous flux of the Navier-Stokes equations, with the aim to satisfy a discrete version of entropy stability (3.27), and to ensure that the discrete kinetic energy evolution is consistent with (3.19).

Consider the Navier-Stokes equations in one-dimension

$$\partial_t \mathbf{U} + \partial_x \mathbf{f}(\mathbf{U}) = \partial_x \mathbf{g}(\mathbf{U}), \tag{5.28}$$

where $\mathbf{U}$ and $\mathbf{f}$ are given by (5.8), while the viscous flux $\mathbf{g}$ is given by

$$\mathbf{g} = \begin{pmatrix} g^\rho \\ g^m \\ g^e \end{pmatrix} = \begin{pmatrix} 0 \\ \tau \\ u\tau - Q \end{pmatrix}, \quad \tau = \frac{4}{3}\mu \partial_x u, \quad Q = -\kappa \partial_x \theta. \tag{5.29}$$

Consider the following semi discrete scheme for (5.28)

$$\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} + \frac{1}{\Delta x}(\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}) = \frac{1}{\Delta x}(\mathbf{G}_{i+\frac{1}{2}} - \mathbf{G}_{i-\frac{1}{2}}), \tag{5.30}$$

with the numerical fluxes given by

$$\mathbf{F}_{i+\frac{1}{2}} = \begin{pmatrix} F^\rho \\ F^m \\ F^e \end{pmatrix}_{i+\frac{1}{2}}, \quad \mathbf{G}_{i+\frac{1}{2}} = \begin{pmatrix} G^\rho \\ G^m \\ G^e \end{pmatrix}_{i+\frac{1}{2}} = \begin{pmatrix} 0 \\ \widetilde{\tau} \\ \widetilde{u\tau} - \tilde{Q} \end{pmatrix}_{i+\frac{1}{2}} \tag{5.31}$$

where $F^\rho, F^m, F^e, \widetilde{\tau}, \widetilde{u}$ and $\tilde{Q}$ are any consistent approximations of the corresponding quantities. We choose the following approximation for the terms in the viscous flux

$$\widetilde{\tau}_{i+\frac{1}{2}} = \frac{4}{3}\mu\frac{\Delta u_{i+\frac{1}{2}}}{\Delta x}, \quad \widetilde{u}_{i+\frac{1}{2}} = \overline{u}_{i+\frac{1}{2}}, \quad \tilde{Q}_{i+\frac{1}{2}} = -\kappa\frac{\Delta \theta_{i+\frac{1}{2}}}{\Delta x}, \tag{5.32}$$

and show that the discretization (5.32) leads to the formulation of kinetic energy preserving and entropy stable schemes for the Navier-Stokes equations [16], provided the numerical inviscid fluxes are chosen appropriately.

5.6.1 Kinetic energy preservation for one-dimensional Navier-Stokes equations

The evolution of total kinetic energy corresponding to the scheme (5.30) can be derived as

$$\begin{aligned}\sum_i \Delta x \frac{\mathrm{d}\mathcal{K}_i}{\mathrm{d}t} &= \sum_i \left[-\frac{1}{2}u_i^2\frac{\mathrm{d}\rho_i}{\mathrm{d}t} + u_i\frac{\mathrm{d}}{\mathrm{d}t}(\rho u)_i\right]\Delta x \\ &= \sum_i \left[\frac{1}{2}u_i^2(F^\rho_{i+\frac{1}{2}} - F^\rho_{i-\frac{1}{2}}) - u_i(F^m_{i+\frac{1}{2}} - F^m_{i-\frac{1}{2}})\right] + \sum_i \left[u_i(G^m_{i+\frac{1}{2}} - G^m_{i-\frac{1}{2}})\right].\end{aligned}$$

In Section 5.2, we have already shown that if the inviscid momentum flux is of the form (5.9), then

$$\sum_i \left[\frac{1}{2}u_i^2(F^\rho_{i+\frac{1}{2}} - F^\rho_{i-\frac{1}{2}}) - u_i(F^m_{i+\frac{1}{2}} - F^m_{i-\frac{1}{2}})\right] = \sum_i \frac{\Delta u_{i+\frac{1}{2}}}{\Delta x}\widetilde{p}_{i+\frac{1}{2}}\Delta x,$$

where $\widetilde{p}$ is any consistent approximation for pressure. Using (5.32), the viscous contribution to the evolution of kinetic energy can be reformulated as

$$\sum_i \left[u_i(G^m_{i+\frac{1}{2}} - G^m_{i-\frac{1}{2}})\right] = -\sum_i \left[(u_i - u_{i+1})G^m_{i+\frac{1}{2}}\right] = -\frac{4}{3}\mu\sum_i\left(\frac{\Delta u_{i+\frac{1}{2}}}{\Delta x}\right)^2\Delta x.$$

Thus, for schemes with a kinetic energy preserving inviscid flux satisfying (5.9), we have

$$\sum_i \Delta x\frac{\mathrm{d}\mathcal{K}_i}{\mathrm{d}t} = \sum_i \frac{\Delta u_{i+\frac{1}{2}}}{\Delta x}\widetilde{p}_{i+\frac{1}{2}}\Delta x - \frac{4}{3}\mu\sum_i\left(\frac{\Delta u_{i+\frac{1}{2}}}{\Delta x}\right)^2\Delta x,$$

which is consistent with (3.19). Note that the viscous forces dissipate the total kinetic energy at the discrete level, as is the case at the continuous level.

Both KEPEC and KEP fluxes with a viscous flux dicretization given by (5.32), lead to consistent formulations for the discrete evolution of kinetic energy.

5.6.2 Entropy stability for one-dimensional Navier-Stokes equations

As discussed in Section 3.4, if we ignore the boundary contributions by assuming periodic or adiabatic no-slip boundary conditions, then the total entropy for the Navier-Stokes system decays in time. A scheme for the Navier-Stokes equations which is consistent with this behaviour will be called entropy stable. Taking the scalar product of the scheme (5.30) and summing over all cells, we get

$$\sum_i \left\langle \mathbf{V}_i, \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} \right\rangle \Delta x + \sum_i \left\langle \mathbf{V}_i, (\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}) \right\rangle = \sum_i \left\langle \mathbf{V}_i, (\mathbf{G}_{i+\frac{1}{2}} - \mathbf{G}_{i-\frac{1}{2}}) \right\rangle . \tag{5.33}$$

If $\mathbf{F}_{i+\frac{1}{2}}$ is an entropy conservative flux as discussed in Section 5.1, then the following relation holds

$$\left\langle \mathbf{V}_i, (\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}) \right\rangle = q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}, \tag{5.34}$$

where $q_{i+\frac{1}{2}}$ is a consistent numerical entropy flux. The viscous contribution in (5.33) can be rewritten as

$$\sum_i \left\langle \mathbf{V}_i, (\mathbf{G}_{i+\frac{1}{2}} - \mathbf{G}_{i-\frac{1}{2}}) \right\rangle = -\sum_i \left\langle \Delta\mathbf{V}_{i+\frac{1}{2}}, \mathbf{G}_{i+\frac{1}{2}} \right\rangle . \tag{5.35}$$

For the viscous discretization (5.32), the summand on the right of (5.35) evaluates out to be

$$\left\langle \Delta\mathbf{V}_{i+\frac{1}{2}}, \mathbf{G}_{i+\frac{1}{2}} \right\rangle = \frac{4}{3}\mu\Delta(2\beta u)_{i+\frac{1}{2}} \frac{\Delta u_{i+\frac{1}{2}}}{\Delta x} - \frac{4}{3}\mu\overline{u}_{i+\frac{1}{2}}\Delta(2\beta)_{i+\frac{1}{2}} \frac{\Delta u_{i+\frac{1}{2}}}{\Delta x} - \kappa\Delta(2\beta)_{i+\frac{1}{2}} \frac{\Delta\theta_{i+\frac{1}{2}}}{\Delta x}.$$

Using the relation $\Delta(ab) = \overline{a}\Delta b + \overline{b}\Delta a$ and the fact that $\beta = 1/(R\theta)$, we get

$$\left\langle \Delta\mathbf{V}_{i+\frac{1}{2}}, \mathbf{G}_{i+\frac{1}{2}} \right\rangle = \frac{8}{3}\mu\overline{\beta}_{i+\frac{1}{2}} \left(\frac{\Delta u_{i+\frac{1}{2}}}{\Delta x} \right)^2 + \frac{\kappa}{R\theta_i\theta_{i+1}} \left(\frac{\Delta\theta_{i+\frac{1}{2}}}{\Delta x} \right)^2 \geqslant 0. \tag{5.36}$$

Thus, (5.33), (5.34), (5.35) and (5.36) lead to the relation

$$\sum_i \left\langle \mathbf{V}_i, \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} \right\rangle \Delta x = \sum_i \frac{\mathrm{d}\eta_i}{\mathrm{d}t}\Delta x \leqslant -\sum_i \left(q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}} \right) = 0, \tag{5.37}$$

since the flux terms $q_{i+\frac{1}{2}}$ cancel one another when summed over all cells. The relation (5.37) proves that the total entropy for the discrete system decays in time, which is consistent with the behaviour at the continuous level.

The KEPEC and ROE-EC fluxes, being entropy conservative, will lead to entropy stability provided the viscous flux is discretized according to (5.32)

Remark 5.6.1. *The discrete entropy estimate* (5.37) *can also be obtained if we replace the condition* (5.34) *with*

$$\left\langle \mathbf{V}_i, (\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}) \right\rangle \leqslant q_{i+\frac{1}{2}} - q_{i-\frac{1}{2}}$$

which holds true for any entropy stable flux $\mathbf{F}_{i+\frac{1}{2}}$.

Remark 5.6.2. *The KEPEC flux satisfies both* (5.9) *and* (5.34)*, thus can be used to construct a kinetic energy preserving and entropy stable finite difference scheme, provided the viscous fluxes are discretized according to* (5.32)*.*

Remark 5.6.3. *Viscous flux approximations of the form* (5.32) *cannot be use for the Navier-Stokes equations in higher-dimensions, due to the existence of cross derivative terms in the viscous fluxes. An alternate formulation for the multi-dimensional Navier-Stokes equations is described in the next section.*

5.7 Two-dimensional finite difference scheme

We briefly describe the extension of the various notions described previously in this chapter, to the two-dimensional setting. The extension to three-dimensions can be done in a similar manner.

Consider a uniform two-dimensional Cartesian mesh, with mesh point $\mathbf{x}_{i,j} = (x_i, y_j) = (i\Delta x, j\Delta y)$ forming the cell centers of cells $I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}) \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}})$, for $(i,j) \in \mathbb{Z}^2$. We call the points $(x_{i+\frac{1}{2}}, y_{j+\frac{1}{2}})$ as the *nodes* of the mesh. Define the jumps and arithmetic averages across cell interfaces by

$$\begin{aligned}
\Delta\phi_{i+\frac{1}{2},j} &= \phi_{i+1,j} - \phi_{i,j}, \quad \overline{\phi}_{i+\frac{1}{2},j} = \frac{\phi_{i+1,j} + \phi_{i,j}}{2}, \\
\Delta\phi_{i,j+\frac{1}{2}} &= \phi_{i,j+1} - \phi_{i,j}, \quad \overline{\phi}_{i,j+\frac{1}{2}} = \frac{\phi_{i,j+1} + \phi_{i,j}}{2}.
\end{aligned}$$

Additionally, we define two discrete multi-dimensional operators $\mathcal{T}^{xy}, \mathcal{T}^{yx}$, which act at nodes as

$$\begin{aligned}
[\mathcal{T}^{xy}\phi]_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2}\left(\frac{\phi_{i+1,j+1} - \phi_{i-1,j+1}}{\Delta x} + \frac{\phi_{i+1,j-1} - \phi_{i-1,j-1}}{\Delta x}\right), \\
[\mathcal{T}^{yx}\phi]_{i+\frac{1}{2},j+\frac{1}{2}} &= \frac{1}{2}\left(\frac{\phi_{i+1,j+1} - \phi_{i+1,j-1}}{\Delta y} + \frac{\phi_{i-1,j+1} - \phi_{i-1,j-1}}{\Delta y}\right),
\end{aligned}$$

and at cell-centers as

$$\begin{aligned}
[\mathcal{T}^{xy}\phi]_{i,j} &= \frac{1}{2}\left(\frac{\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}}}{\Delta x} + \frac{\phi_{i+\frac{1}{2},j-\frac{1}{2}} - \phi_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x}\right), \\
[\mathcal{T}^{yx}\phi]_{i,j} &= \frac{1}{2}\left(\frac{\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i+\frac{1}{2},j-\frac{1}{2}}}{\Delta y} + \frac{\phi_{i-\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y}\right).
\end{aligned}$$

If ϕ is not defined at the nodes, then the nodal values are obtained by averaging its values over the neighbouring cell-centers

$$\phi_{i+\frac{1}{2},j+\frac{1}{2}} = \frac{1}{4}(\phi_{i+1,j+1} + \phi_{i,j+1} + \phi_{i+1,j} + \phi_{i,j}). \tag{5.38}$$

Note that $\mathcal{T}^{xy}$ approximates the partial derivative in the x-direction, by a second-order divided difference in the x-direction, followed by an averaging in the y-direction. Similarly, $\mathcal{T}^{yx}$ approximates the partial derivative in the y-direction, by a second-order divided difference in the y-direction, followed by an averaging in the x-direction. The following lemmas describe important summation-by-parts formulae for these two operators.

Lemma 5.7.1. *Consider two grid function ϕ and χ, whose values are given at cell-centers. Then the following relation holds*

$$\sum_{i,j} \phi_{i,j}[\mathcal{T}^{xy}\chi]_{i,j} = -\sum_{i,j} \chi_{i,j}[\mathcal{T}^{xy}\phi]_{i,j}, \qquad \sum_{i,j} \phi_{i,j}[\mathcal{T}^{yx}\chi]_{i,j} = -\sum_{i,j} \chi_{i,j}[\mathcal{T}^{yx}\phi]_{i,j}, \tag{5.39}$$

assuming periodic boundary conditions. The required nodal values of ϕ and χ are obtained using (5.38).

Proof. We prove the relation for the $\mathcal{T}^{xy}$ operator, with the proof for $\mathcal{T}^{yx}$ following on the same lines. We have

$$\sum_{i,j} \phi_{i,j}[\mathcal{T}^{xy}\chi]_{i,j} = \sum_{i,j} \phi_{i,j} \frac{1}{2\Delta x}\left(\chi_{i+\frac{1}{2},j+\frac{1}{2}} - \chi_{i-\frac{1}{2},j+\frac{1}{2}} + \chi_{i+\frac{1}{2},j-\frac{1}{2}} - \chi_{i-\frac{1}{2},j-\frac{1}{2}}\right). \tag{5.40}$$

Using (5.38) to obtain the nodal values of ϕ and χ, we get

$$\begin{aligned} \sum_{i,j} \phi_{i,j}\chi_{i+\frac{1}{2},j+\frac{1}{2}} &= \sum_{i,j} \phi_{i,j}\left(\chi_{i+1,j+1} + \chi_{i,j+1} + \chi_{i+1,j} + \chi_{i,j}\right) \\ &= \sum_{i,j} \left(\phi_{i-1,j-1} + \phi_{i,j-1} + \phi_{i-1,j} + \phi_{i,j}\right)\chi_{i,j} \\ &= \sum_{i,j} \phi_{i-\frac{1}{2},j-\frac{1}{2}}\phi_{i,j}. \end{aligned}$$

Handling the remaining terms on the right of (5.40) in a similar manner, we can write

$$\begin{aligned} \sum_{i,j} \phi_{i,j}[\mathcal{T}^{xy}\chi]_{i,j} &= -\sum_{i,j} \chi_{i,j}\frac{1}{2\Delta x}\left(\phi_{i+\frac{1}{2},j+\frac{1}{2}} - \phi_{i-\frac{1}{2},j+\frac{1}{2}} + \phi_{i+\frac{1}{2},j-\frac{1}{2}} - \phi_{i-\frac{1}{2},j-\frac{1}{2}}\right) \\ &= -\sum_{i,j} \chi_{i,j}[\mathcal{T}^{xy}\phi]_{i,j}, \end{aligned}$$

which is the required relation. □

Lemma 5.7.2. *Consider two grid function ϕ and χ such that ϕ is defined at cell-centers, while χ is defined at nodes. Then the following relation holds*

$$\begin{aligned} \sum_{i,j} \phi_{i,j}[\mathcal{T}^{xy}\chi]_{i,j} &= -\sum_{i,j} \chi_{i+\frac{1}{2},j+\frac{1}{2}}[\mathcal{T}^{xy}\phi]_{i+\frac{1}{2},j+\frac{1}{2}}, \\ \sum_{i,j} \phi_{i,j}[\mathcal{T}^{yx}\chi]_{i,j} &= -\sum_{i,j} \chi_{i+\frac{1}{2},j+\frac{1}{2}}[\mathcal{T}^{yx}\phi]_{i+\frac{1}{2},j+\frac{1}{2}}, \end{aligned} \tag{5.41}$$

assuming periodic boundary conditions.

Proof. As done in Lemma 5.7.1, we prove the relation for the $\mathcal{T}^{xy}$ operator. Since χ is defined at nodes and ϕ is defined at cell-centers, we have

$$\begin{aligned} \sum_{i,j} \phi_{i,j}[\mathcal{T}^{xy}\chi]_{i,j} &= \sum_{i,j} \phi_{i,j}\frac{1}{2\Delta x}\left(\chi_{i+\frac{1}{2},j+\frac{1}{2}} - \chi_{i-\frac{1}{2},j+\frac{1}{2}} + \chi_{i+\frac{1}{2},j-\frac{1}{2}} - \chi_{i-\frac{1}{2},j-\frac{1}{2}}\right) \\ &= \sum_{i,j} \left(\phi_{i,j} - \phi_{i+1,j} + \phi_{i,j+1} - \phi_{i+1,j+1}\right)\frac{1}{2\Delta x}\chi_{i+\frac{1}{2},j+\frac{1}{2}} \\ &= -\sum_{i,j} \chi_{i+\frac{1}{2},j+\frac{1}{2}}[\mathcal{T}^{xy}\phi]_{i+\frac{1}{2},j+\frac{1}{2}}, \end{aligned}$$

which proves the result. □

5.7.1 Inviscid discretization

Consider the following semi-discrete scheme for the two-dimensional Euler equations, i.e., (3.4) with d=2:

$$\frac{\mathrm{d}\mathbf{U}_{i,j}}{\mathrm{d}t} + \frac{\left(\mathbf{F}^x_{i+\frac{1}{2},j} - \mathbf{F}^x_{i-\frac{1}{2},j}\right)}{\Delta x} + \frac{\left(\mathbf{F}^y_{i,j+\frac{1}{2}} - \mathbf{F}^y_{i,j-\frac{1}{2}}\right)}{\Delta y} = 0. \tag{5.42}$$

Here, $\mathbf{U}_{i,j}(t) = \mathbf{U}(x_i, y_j, t)$ is the value of the solution at the cell-center $\mathbf{x}_{i,j}$, and $\mathbf{F}^x_{i+\frac{1}{2},j}$, $\mathbf{F}^y_{i,j+\frac{1}{2}}$ are consistent approximations of the inviscid flux components $\mathbf{f}_1$, $\mathbf{f}_2$ respectively. The numerical scheme (5.42) is entropy conservative if it satisfies

$$\frac{\mathrm{d}\eta(\mathbf{U}_i)}{\mathrm{d}t} + \frac{\left(q^{x,*}_{i+\frac{1}{2},j} - q^{x,*}_{i-\frac{1}{2},j}\right)}{\Delta x} + \frac{\left(q^{y,*}_{i,j+\frac{1}{2}} - q^{y,*}_{i,j-\frac{1}{2}}\right)}{\Delta y} = 0, \tag{5.43}$$

where $q^{x,*}_{i+\frac{1}{2},j}, q^{y,*}_{i,j+\frac{1}{2}}$ are numerical entropy fluxes consistent with q_1, q_2. The entropy potential functions corresponding to the entropy-entropy flux pair $(\eta(\mathbf{U}), \mathbf{q}(\mathbf{U}))$ are given by

$$\Psi^x(\mathbf{U}) := \langle \mathbf{V}(\mathbf{U}), \mathbf{f}_1(\mathbf{U})\rangle - q_1(\mathbf{U}), \quad \Psi^y(\mathbf{U}) := \langle \mathbf{V}(\mathbf{U}), \mathbf{f}_2(\mathbf{U})\rangle - q_2(\mathbf{U}).$$

The numerical scheme (5.42) with the flux $\mathbf{F}^{x,*}_{i+\frac{1}{2},j}, \mathbf{F}^{y,*}_{i,j+\frac{1}{2}}$ is entropy conservative if the sufficient conditions

$$\left\langle \Delta\mathbf{V}_{i+\frac{1}{2},j}, \mathbf{F}^{x,*}_{i+\frac{1}{2},j} \right\rangle = \Delta\Psi^x_{i+\frac{1}{2},j}, \quad \left\langle \Delta\mathbf{V}_{i,j+\frac{1}{2}}, \mathbf{F}^{y,*}_{i,j+\frac{1}{2}} \right\rangle = \Delta\Psi^y_{i,j+\frac{1}{2}}, \tag{5.44}$$

hold [35], with the consistent numerical entropy fluxes given by

$$q^{x,*}_{i+\frac{1}{2},j} = \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2},j}, \mathbf{F}^{x,*}_{i+\frac{1}{2},j} \right\rangle - \overline{\Psi^x}_{i+\frac{1}{2},j}, \quad q^{y,*}_{i,j+\frac{1}{2}} = \left\langle \overline{\mathbf{V}}_{i,j+\frac{1}{2}}, \mathbf{F}^{y,*}_{i,j+\frac{1}{2}} \right\rangle - \overline{\Psi^y}_{i,j+\frac{1}{2}}.$$

Entropy stable schemes can be constructed by augmenting the entropy conservative fluxes with entropy variable based dissipation

$$\mathbf{F}^x_{i+\frac{1}{2},j} = \mathbf{F}^{x,*}_{i+\frac{1}{2},j} - \frac{1}{2}\mathbf{D}^x_{i+1,j}\Delta\mathbf{V}_{i+\frac{1}{2},j}, \quad \mathbf{F}^y_{i,j+\frac{1}{2}} = \mathbf{F}^{y,*}_{i,j+\frac{1}{2}} - \frac{1}{2}\mathbf{D}^y_{i,j+\frac{1}{2}}\Delta\mathbf{V}_{i,j+\frac{1}{2}},$$

where $\mathbf{D}^x_{i+1,j}, \mathbf{D}^y_{i,j+\frac{1}{2}}$ are symmetric and positive semi-definite matrices. Higher-order entropy stable fluxes can be obtained by a dimension-by-dimension application of the interpolation formula (5.14), and a sign-preserving reconstruction of scaled entropy variables in the dissipation terms.

Let the inviscid numerical fluxes be of the form $\mathbf{F}^x = \left(F^{x,\rho},\ F^{x,m_1},\ F^{x,m_2},\ F^{x,e}\right)^\top$ and $\mathbf{F}^y = \left(F^{y,\rho},\ F^{y,m_1},\ F^{y,m_2},\ F^{y,e}\right)^\top$. Then the condition (5.9), required to construct kinetic energy preserving schemes for the Euler equation, can be extended to two-dimensions [59] as

$$\mathbf{F}^{x,m} = \widetilde{p}\mathbf{e}_1 + \overline{\mathbf{u}}F^{x,\rho}, \quad \mathbf{F}^{y,m} = \widetilde{p}\mathbf{e}_2 + \overline{\mathbf{u}}F^{y,\rho}, \tag{5.45}$$

where $\mathbf{e}_1 = (1,0)^\top$ and $\mathbf{e}_2 = (0,1)^\top$, while $\widetilde{p}, F^{x,\rho}, F^{y,\rho}$ are any consistent approximations of the pressure and the mass flux components.

5.7.2 Viscous discretization

We consider the following semi-discrete finite difference scheme for the two-dimensional Navier-Stokes equations, i.e., (3.10) with d=2, by including viscous discretization terms in the scheme (5.42)

$$\frac{\mathrm{d}\mathbf{U}_{i,j}}{\mathrm{d}t} + \frac{\left(\mathbf{F}^x_{i+\frac{1}{2},j} - \mathbf{F}^x_{i-\frac{1}{2},j}\right)}{\Delta x} + \frac{\left(\mathbf{F}^y_{i,j+\frac{1}{2}} - \mathbf{F}^y_{i,j-\frac{1}{2}}\right)}{\Delta y} = [\mathcal{T}^{xy}\mathbf{G}^x]_{i,j} + [\mathcal{T}^{yx}\mathbf{G}^y]_{i,j}. \tag{5.46}$$

The numerical viscous fluxes are evaluated in a consistent manner at the nodes of the mesh, i.e., $\mathbf{G}^x_{i+\frac{1}{2},j+\frac{1}{2}}$, $\mathbf{G}^y_{i+\frac{1}{2},j+\frac{1}{2}}$, thus the viscous terms on the right of (5.46) make sense. In the following two sections, we describe the approximation of the viscous flux from the point of view of obtaining kinetic energy preservation or entropy stability.

5.7.3 Kinetic energy preservation

Consider the viscous fluxes in the component-wise form $\mathbf{G}^x = (G^{x,\rho},\ G^{x,m_1},\ G^{x,m_2},\ G^{x,e})^\top$ and $\mathbf{G}^y = (G^{y,\rho},\ G^{y,m_1},\ G^{y,m_2},\ G^{y,e})^\top$. The discrete evolution of total kinetic energy corresponding to the scheme (5.46) is given by

$$\begin{aligned}
\sum_{i,j}\Delta x\Delta y\frac{\mathrm{d}\mathcal{K}_{i,j}}{\mathrm{d}t} &= \sum_{i,j}\left[-\frac{1}{2}|\mathbf{u}|^2_{i,j}\frac{\mathrm{d}\rho_{i,j}}{\mathrm{d}t} + \left\langle \mathbf{u}_{i,j}, \frac{\mathrm{d}}{\mathrm{d}t}(\rho\mathbf{u})_{i,j}\right\rangle\right]\Delta x\Delta y \\
&= \underbrace{\sum_{i,j}\left[\frac{1}{2}|\mathbf{u}|^2_{i,j}\frac{\left(F^{x,\rho}_{i+\frac{1}{2},j} - F^{x,\rho}_{i-\frac{1}{2},j}\right)}{\Delta x} + \frac{1}{2}|\mathbf{u}|^2_{i,j}\frac{\left(F^{y,\rho}_{i,j+\frac{1}{2}} - F^{y,\rho}_{i,j-\frac{1}{2}}\right)}{\Delta y}\right]\Delta x\Delta y}_{I} \\
&\quad \underbrace{-\sum_{i,j}\left[\frac{\left\langle \mathbf{u}_{i,j}, \mathbf{F}^{x,m}_{i+\frac{1}{2},j} - \mathbf{F}^{x,m}_{i-\frac{1}{2},j}\right\rangle}{\Delta x} + \frac{\left\langle \mathbf{u}_{i,j}, \mathbf{F}^{y,m}_{i,j+\frac{1}{2}} - \mathbf{F}^{y,m}_{i,j-\frac{1}{2}}\right\rangle}{\Delta y}\right]\Delta x\Delta y}_{II} \\
&\quad \underbrace{+\sum_{i,j}\left[\langle \mathbf{u}_{i,j}, [\mathcal{T}^{xy}\mathbf{G}^{x,m}]_{i,j}\rangle + \langle \mathbf{u}_{i,j}, [\mathcal{T}^{yx}\mathbf{G}^{y,m}]_{i,j}\rangle\right]\Delta x\Delta y,}_{III}
\end{aligned}$$

where $I + II$ is the inviscid contribution and III is the contribution due to viscous forces. Assuming the inviscid flux satisfies the condition (5.45), then after a few steps of manipulation similar to those done in Section 5.2, we can rewrite the inviscid contribution as

$$I + II = \sum_{i,j}\left[\frac{\Delta(u_1)_{i+\frac{1}{2},j}}{\Delta x}\widetilde{p}_{i+\frac{1}{2},j} + \frac{\Delta(u_2)_{i,j+\frac{1}{2}}}{\Delta y}\widetilde{p}_{i,j+\frac{1}{2}}\right]\Delta x\Delta y,$$

which is consistent with the work done by the pressure forces on the right hand side of (3.18).

We approximate the momentum viscous flux components at the mesh nodes by

$$\begin{aligned}
\mathbf{G}^{x,m}_{i+\frac{1}{2},j+\frac{1}{2}} &= \mu \begin{pmatrix} \frac{4}{3}[\mathcal{T}^{xy}u_1] - \frac{2}{3}[\mathcal{T}^{yx}u_2] \\ \\ [\mathcal{T}^{xy}u_2] + [\mathcal{T}^{yx}u_1] \end{pmatrix}_{i+\frac{1}{2},j+\frac{1}{2}}, \\
\mathbf{G}^{y,m}_{i+\frac{1}{2},j+\frac{1}{2}} &= \mu \begin{pmatrix} [\mathcal{T}^{xy}u_2] + [\mathcal{T}^{yx}u_1] \\ \\ \frac{4}{3}[\mathcal{T}^{yx}u_2] - \frac{2}{3}[\mathcal{T}^{xy}u_1] \end{pmatrix}_{i+\frac{1}{2},j+\frac{1}{2}},
\end{aligned} \tag{5.47}$$

which are well defined since $\mathbf{u}$ is defined at the cell-centers. Using Lemma 5.7.2, the viscous contribution III can be rewritten as

$$III = -\sum_{i,j} \left[\left\langle [\mathcal{T}^{xy}\mathbf{u}]_{i+\frac{1}{2},j+\frac{1}{2}}, \mathbf{G}^{x,m}_{i+\frac{1}{2},j+\frac{1}{2}} \right\rangle + \left\langle [\mathcal{T}^{yx}\mathbf{u}]_{i+\frac{1}{2},j+\frac{1}{2}}, \mathbf{G}^{y,m}_{i+\frac{1}{2},j+\frac{1}{2}} \right\rangle \right] \Delta x \Delta y. \tag{5.48}$$

Note that the summand on the left hand side of (5.48) is exactly the viscous contribution in the last equality of the continuous equation (3.16), with the partial derivative operators replaced by the discrete derivative operators $\mathcal{T}^{xy}$ and $\mathcal{T}^{yx}$. Following the same steps as those outlined in Section 3.3, we can show that the contribution III is consistent with the viscous contribution at the continuous level.

Thus, the evolution of total kinetic energy by a numerical scheme with the inviscid fluxes satisfying (5.45), and the viscous fluxes discretized according to (5.47), is consistent with (3.18).

Remark 5.7.1. *The condition* (5.47) *only describes the discretization of the momentum terms of the viscous fluxes. In order to be consistent, we choose* $G^{x,\rho} = G^{y,\rho} = 0$*, while we are free to choose* $G^{x,e}, G^{y,e}$ *in any consistent manner. In particular, we make the choice*

$$\begin{aligned}
G^{x,e}_{i+\frac{1}{2},j+\frac{1}{2}} &= \left\langle \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}, \mathbf{G}^{x,m}_{i+\frac{1}{2},j+\frac{1}{2}} \right\rangle + \kappa[\mathcal{T}^{xy}\theta]_{i+\frac{1}{2},j+\frac{1}{2}}, \\
G^{y,e}_{i+\frac{1}{2},j+\frac{1}{2}} &= \left\langle \mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}, \mathbf{G}^{y,m}_{i+\frac{1}{2},j+\frac{1}{2}} \right\rangle + \kappa[\mathcal{T}^{yx}\theta]_{i+\frac{1}{2},j+\frac{1}{2}},
\end{aligned} \tag{5.49}$$

where $\mathbf{u}_{i+\frac{1}{2},j+\frac{1}{2}}$ *is obtained using* (5.38)*, while* $[\mathcal{T}^{xy}\theta]_{i+\frac{1}{2},j+\frac{1}{2}}, [\mathcal{T}^{yx}\theta]_{i+\frac{1}{2},j+\frac{1}{2}}$ *use the values of temperature available at cell-centers.*

5.7.4 Entropy stability

While the discretization (5.47) leads to kinetic energy preservation, we are unable to find a consistent approximation for $G^{x,e}$ and $G^{y,e}$ that would also lead to an entropy stable scheme for the Navier-Stokes equations. We thus propose an alternate approximation of the viscous fluxes that gives us the desired discrete estimate.

Taking the scalar product of the scheme (5.46) with $\mathbf{V}_{i,j}$ and summing over all cells leads to

$$\begin{aligned}
\sum_{i,j} \frac{\mathrm{d}\eta_{ij}}{\mathrm{d}t} \Delta x \Delta y &+ \sum_{i,j} \left\langle \mathbf{V}_{i,j}, \left(\mathbf{F}^x_{i+\frac{1}{2},j} - \mathbf{F}^x_{i-\frac{1}{2},j} \right) \right\rangle \Delta x + \sum_{i,j} \left\langle \mathbf{V}_{i,j}, \left(\mathbf{F}^y_{i,j+\frac{1}{2}} - \mathbf{F}^y_{i,j-\frac{1}{2}} \right) \right\rangle \Delta y \\
&= \underbrace{\sum_{i,j} \left[\langle \mathbf{V}_{i,j}, [\mathcal{T}^{xy}\mathbf{G}^{x,m}]_{i,j} \rangle + \langle \mathbf{V}_{i,j}, [\mathcal{T}^{yx}\mathbf{G}^{y,m}]_{i,j} \rangle \right] \Delta x \Delta y}_{IV} .
\end{aligned} \tag{5.50}$$

As done in Section 5.6.2, the inviscid flux terms in (5.50) drop out by choosing an entropy conservative/stable flux, satisfying the cell entropy relations

$$\left\langle \mathbf{V}_{i,j}, \left(\mathbf{F}^x_{i+\frac{1}{2},j} - \mathbf{F}^x_{i-\frac{1}{2},j}\right)\right\rangle \leqslant q^x_{i+\frac{1}{2},j} - q^x_{i-\frac{1}{2},j}, \quad \left\langle \mathbf{V}_{i,j}, \left(\mathbf{F}^y_{i,j+\frac{1}{2}} - \mathbf{F}^y_{i,j-\frac{1}{2}}\right)\right\rangle \leqslant q^y_{i,j+\frac{1}{2}} - q^y_{i,i-\frac{1}{2}}, \tag{5.51}$$

with strict equality holding for entropy conservative fluxes.

In order to approximate the viscous fluxes at the cell-centers, we make use of its symmetric formulation when written in terms of the entropy variables (see Section 3.2.1). We choose

$$\begin{aligned} \mathbf{G}^x_{i+\frac{1}{2},j+\frac{1}{2}} &= \mathbf{K}_{11}(\mathbf{V}_{i+\frac{1}{2},j+\frac{1}{2}})[\mathcal{T}^{xy}\mathbf{V}]_{i+\frac{1}{2},j+\frac{1}{2}} + \mathbf{K}_{12}(\mathbf{V}_{i+\frac{1}{2},j+\frac{1}{2}})[\mathcal{T}^{yx}\mathbf{V}]_{i+\frac{1}{2},j+\frac{1}{2}}, \\ \mathbf{G}^y_{i+\frac{1}{2},j+\frac{1}{2}} &= \mathbf{K}_{21}(\mathbf{V}_{i+\frac{1}{2},j+\frac{1}{2}})[\mathcal{T}^{xy}\mathbf{V}]_{i+\frac{1}{2},j+\frac{1}{2}} + \mathbf{K}_{22}(\mathbf{V}_{i+\frac{1}{2},j+\frac{1}{2}})[\mathcal{T}^{yx}\mathbf{V}]_{i+\frac{1}{2},j+\frac{1}{2}}, \end{aligned} \tag{5.52}$$

where the matrices $\mathbf{K}_{11}, \mathbf{K}_{12}, \mathbf{K}_{21}, \mathbf{K}_{22}$ were introduced in see Section 3.2.1. The matrices are evaluated with the nodal value of the entropy variables, obtained using (5.38). Recall that the matrix

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} \\ \mathbf{K}_{21} & \mathbf{K}_{22} \end{pmatrix},$$

is symmetric and positive semi-definite. With an application of the results of Lemma 5.7.2, the right hand side of (5.50) can be rewritten as

$$\begin{aligned} IV &= -\sum_{i,j} \left[\left\langle [\mathcal{T}^{xy}\mathbf{V}]_{i+\frac{1}{2},j+\frac{1}{2}}, \mathbf{G}^x_{i+\frac{1}{2},j+\frac{1}{2}} \right\rangle + \left\langle [\mathcal{T}^{yx}\mathbf{V}]_{i+\frac{1}{2},j+\frac{1}{2}}, \mathbf{G}^y_{i+\frac{1}{2},j+\frac{1}{2}} \right\rangle \right] \Delta x \Delta y \\ &= -\sum_{i,j} \left\langle \mathbf{W}_{i+\frac{1}{2},j+\frac{1}{2}}, \mathbf{K}(\mathbf{V}_{i+\frac{1}{2},j+\frac{1}{2}})\mathbf{W}_{i+\frac{1}{2},j+\frac{1}{2}} \right\rangle \Delta x \Delta y \leqslant 0, \end{aligned} \tag{5.53}$$

where $\mathbf{W}^\top_{i+\frac{1}{2},j+\frac{1}{2}} = \left([\mathcal{T}^{xy}\mathbf{V}]^\top_{i+\frac{1}{2},j+\frac{1}{2}}, \quad [\mathcal{T}^{yx}\mathbf{V}]^\top_{i+\frac{1}{2},j+\frac{1}{2}} \right) \in \mathbb{R}^8$.

Thus, choosing an entropy conservative/stable flux approximation for the inviscid flux, in tandem with the viscous flux discretization (5.53), gives the desired consistent entropy stability estimate

$$\sum_{i,j} \frac{\mathrm{d}\eta_{ij}}{\mathrm{d}t} \Delta x \Delta y \leqslant 0.$$

Remark 5.7.2. *The above formulations can be extend to three-dimensions by constructing operators similar to $\mathcal{T}^{xy}$ and $\mathcal{T}^{yx}$, such that they take a central divided difference in one coordinate direction (depending on the partial derivative being approximated) and perform a central averaging in the remaining two coordinate directions.*

5.8 Numerical results

We now present numerical results with the schemes discussed in this chapter, on several standard test cases for compressible flows. We introduce the following flux nomenclature:

- **KEPEC**: The kinetic energy preserving and entropy conservative flux (5.12).

- **KEPEC4**: The fourth-order version of the KEPEC flux, obtained using the interpolation formula (5.15). KEPEC4 is also entropy conservative, however it does not satisfy the condition (5.9) required to ensure kinetic energy preservation.
- **KEPES**: The KEPEC flux augmented with a first-order entropy variable based dissipation operator, with the dissipation matrix given by (5.22).
- **KEPES-TeCNO**: The KEPES flux with the scaled entropy variables in the dissipation operator reconstructed using the minmod limiter.
- **ROE-EC**: Roe's entropy conservative flux (5.10). Recall that ROE-EC does not preserve total kinetic energy.
- **ROE-EC4**: The fourth-order version of the ROE-EC flux obtained using the interpolation formula (5.15). ROE-EC4 is also entropy conservative.
- **ROE-ES**: The ROE-EC flux with a first-order entropy variable based dissipation operator, with the dissipation matrix given by (5.22).
- **ROE-ES-TeCNO**: The ROE-ES flux with the scaled entropy variables in the dissipation operator reconstructed using the minmod limiter.
- **KEP**: The kinetic energy preserving flux (5.11) proposed by Jameson.
- **KEP4**: The fourth-order version of the KEP flux, obtained using the interpolation formula (5.15). KEP4 no longer satisfies the condition (5.9) required to ensure kinetic energy preservation.
- **Roe**: The original approximate Riemann solver proposed by Roe, for which the flux is given by (4.6). Without an entropy fix, the Roe scheme can lead to entropy violating stationary shocks near sonic points.

The one-dimensional flux expressions have been described in this chapter, while their higher-dimensional extensions are available in Appendix C.

In practice we work with a finite computational domain, and thus need to apply suitable boundary conditions. Consider a one-dimensional domain $[a, b]$, which is discretized using N cells with a mesh size $\Delta x = (b-a)/N$, such that the cell-interfaces and cell-centers are given by

$$a = x_{\frac{1}{2}} < ... < x_{N+\frac{1}{2}} = b, \qquad a + \frac{\Delta x}{2} = x_1 < ... < x_N = b - \frac{\Delta x}{2},$$

respectively. We describe the procedure to set the boundary conditions on the left boundary, with an analogous procedure holding for the right boundary. We need additional ghost cells at x_0, x_{-1}, etc on the left of x_1, to evaluate the numerical flux at $x_{\frac{1}{2}}$. The solution values in these ghost cells are determine by the type of boundary condition being implemented. We consider the following two types of boundary conditions:

- **Periodic boundary conditions** ensure that the waves exiting the domain from one end, re-enter the domain from the other end. This type of boundary condition is imposed by choosing the ghost cell values as $\mathbf{U}_{-i} = \mathbf{U}_{N-i}$ for $i \geqslant 0$.

- **Transmissive boundary conditions** are artificial boundary conditions, designed to ensure that wave pass out through the boundary without any reflection. This type of boundary condition is imposed by choosing the ghost cell values as $\mathbf{U}_{-i} = \mathbf{U}_1$ for $i \geqslant 0$.

The above discussed boundary conditions can be implemented in a dimension-by-dimension manner for multi-dimensional systems on Cartesian grids.

Finally, the semi-discrete scheme (4.17) is integrated in time using the explicit Strong Stability Preserving Runge-Kutta 3-stage scheme (SSP-RK3) method [48], with the time step depending on the convective and viscous contributions [12]:

$$\Delta t = \frac{\text{CFL} \times \Delta x}{\lambda}, \qquad \lambda = \max_i \left\{ (|\mathbf{u}_i| + a_i) + \frac{\mu}{\Delta x \rho_i} \right\},$$

where the CFL is chosen to be a positive number less than 1. In all test problems, we consider an ideal gas with $\gamma = 1.4$, except when indicated otherwise.

5.8.1 Smooth density wave

We consider a smooth one-dimensional problem for the Euler equations, in which a smooth periodic density wave is advected with a constant velocity. The initial condition defined on the domain $[0, 2]$ is given by

$$\rho = 1 + \frac{1}{2}\sin^4(x), \quad u = 0.5, \quad p = 1,$$

with periodic boundary conditions. The final time is $t_f = 0.5$ with CFL $= 0.5$. The aim of this test case is to validate the convergence rates with various fluxes for the Euler equations. We evaluate the discrete L^1 and L^∞ errors of the solution, with the discrete error norms given by

$$\|(.)\|_{L^p_h} = \left(\sum_{i=1}^{N} |(.)_i|^p h \right)^{\frac{1}{p}} \quad \text{for } p < \infty, \qquad \|(.)\|_{L^\infty_h} = \max_i |(.)_i|.$$

If $\Theta^{\Delta x}$ corresponds to the solution error on a mesh with mesh size Δx, then the scheme is said to be k-th order accurate corresponding to a norm $\|.\|$ if

$$\|\Theta^{\Delta x}\| = C\Delta x^k + \mathcal{O}(\Delta x^{k+1}), \tag{5.54}$$

where C is a constant that depends on the problem being solved, as well as on the final time t_f. Note that the higher-order terms in (5.54) decay to zero as $\Delta x \to 0$ at a much faster rate, as compared to the lower-order error term $C\Delta x^k$. Thus, we make the approximation $\|\Theta^{\Delta x}\| = C\Delta x^k$. Taking a log on both sides, we get

$$\log(\|\Theta^{\Delta x}\|) = \log C + k \log(\Delta x). \tag{5.55}$$

Note that the convergence rate of the scheme is nothing but the slope of the line (5.55).

The errors and convergence rates with different fluxes are given in Tables 5.1-5.3. The second-order KEPEC, ROE-EC and KEP fluxes and their fourth-order version give the expected order of convergence. The order of accuracy drops to first-order for KEPES and ROE-ES, since the dissipation operators are only first-order accurate. The minmod limiter is TVD which can lead to clipping of smooth extrema [84, 85]. Thus, the convergence rate is not completely recovered when the scaled entropy variables are reconstructed with the minmod limiter in KEPES-TeCNO and ROE-ES-TeCNO. This is more pronounced when looking at convergence in the L^∞ norm.

Furthermore, we note that the magnitude of the errors for a given version of the KEPEC flux are comparable to the errors of the corresponding version of the ROE-EC flux and the KEP flux on any fixed mesh. Thus, we might conclude at this stage, that all three schemes (without artificial dissipation) perform equally well in approximating smooth solutions for the Euler equations. However, the results for the long-term simulation of an isentropic vortex discussed in Section 5.8.4 paints a very different picture.

Remark 5.8.1. *It is not appropriate to judge the accuracy of a scheme solely on the basis of the rate of convergence. The value of the error* $\|\Theta^{\Delta x}\|$ *is equally important. For a particular mesh, the contribution of the higher-order terms in* (5.54) *may not be small enough to ignore. We shall return to this point in Section 6.4.*

	KEPEC				KEPEC4			
N	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
100	2.70e-03	-	2.68e-01	-	3.15e-05	-	2.56e-03	-
200	6.80e-04	1.99	6.73e-02	1.99	1.99e-06	3.98	1.61e-04	3.98
400	1.70e-04	1.99	1.68e-02	1.99	1.24e-07	3.99	1.01e-05	3.99
600	7.56e-05	2.00	7.48e-03	1.99	2.46e-08	3.99	2.00e-06	3.99
800	4.25e-05	2.00	4.21e-03	2.00	7.79e-09	3.99	6.36e-07	3.99
1000	2.72e-05	2.00	2.69e-03	2.00	3.19e-09	3.99	2.61e-07	3.99
	KEPES				KEPES-TeCNO			
N	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
100	3.71e-02	-	4.37e-00	-	5.35e-03	-	1.30e-00	-
200	1.94e-02	0.93	2.32e-00	0.91	1.50e-03	1.83	5.64e-01	1.20
400	9.95e-03	0.96	1.19e-00	0.95	4.02e-04	1.90	2.35e-01	1.26
600	6.69e-03	0.98	8.05e-01	0.97	1.84e-04	1.92	1.40e-01	1.28
800	5.03e-03	0.98	6.07e-01	0.98	1.06e-04	1.92	9.67e-02	1.28
1000	4.04e-03	0.98	4.87e-01	0.98	6.89e-05	1.92	7.24e-02	1.29

Table 5.1: Order of convergence for the advecting density wave problem with fluxes involving various versions of KEPEC.

5.8.2 Modified Sod test case

This test case describes a shock tube problem of the Sod type [110]. The initial condition on the domain $[0,1]$ has an initial discontinuity at $x_0 = 0.3$, with the left state $(\rho_L, u_L, p_L) = (1.0, 0.75, 1.0)$ and the right state $(\rho_R, u_R, p_R) = (0.125, 0.0, 0.1)$. The computations are made on a mesh with $N = 100$ cells and transmissive boundary conditions,

	ROE-EC				ROE-EC4			
N	L^1_h		L^∞_h		L^1_h		L^∞_h	
	error	rate	error	rate	error	rate	error	rate
100	2.70e-03	-	2.68e-01	-	3.15e-05	-	2.56e-03	-
200	6.79e-04	1.99	6.72e-02	1.99	1.99e-06	3.98	1.61e-04	3.98
400	1.70e-04	1.99	1.68e-02	1.99	1.24e-07	3.99	1.01e-05	3.99
600	7.55e-05	1.99	7.47e-03	1.99	2.46e-08	3.99	2.00e-06	3.99
800	4.25e-05	1.99	4.20e-03	2.00	7.79e-09	3.99	6.36e-07	3.99
1000	2.72e-05	2.00	2.69e-03	2.00	3.19e-09	3.99	2.61e-07	3.99

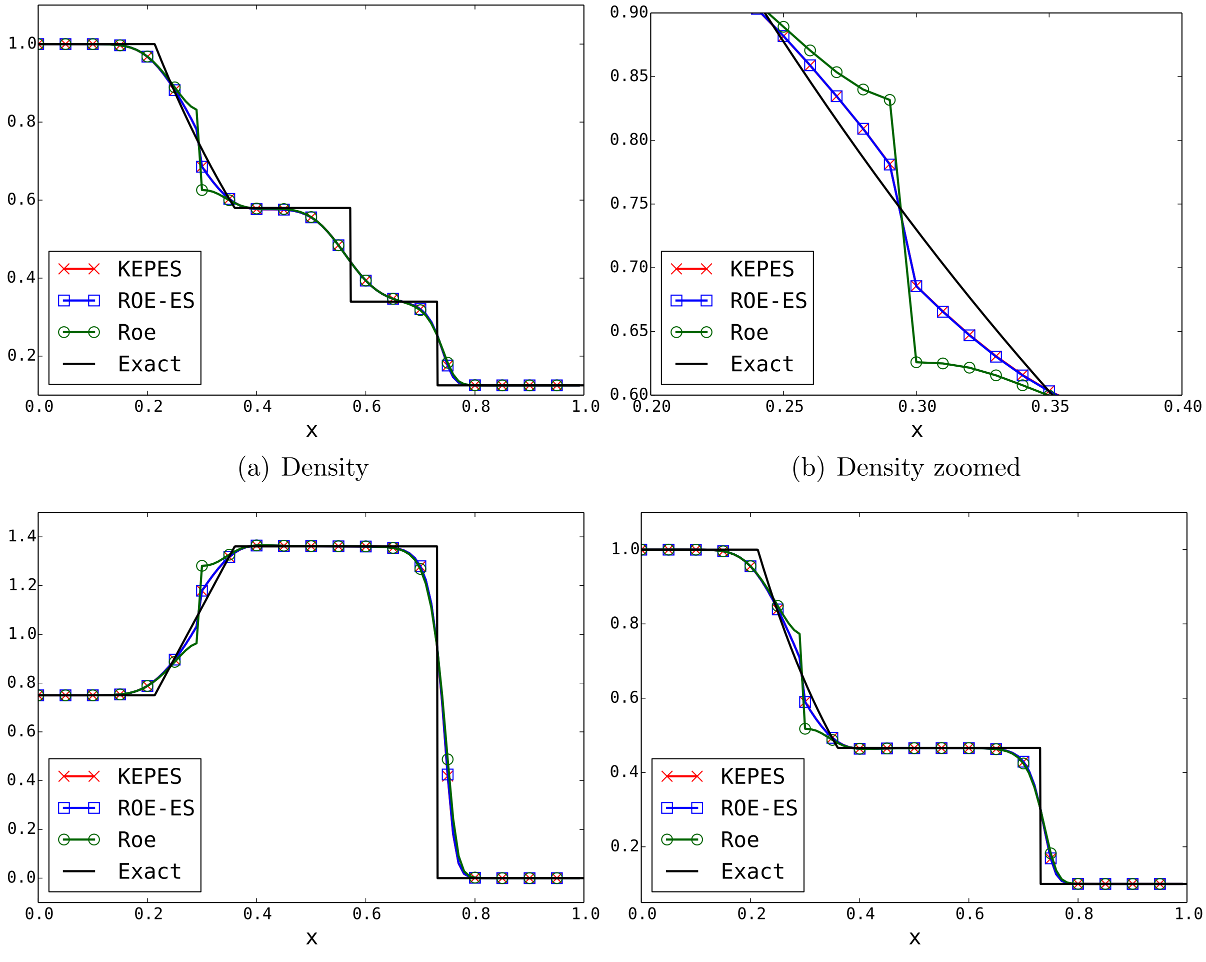
	ROE-ES				ROE-ES-TeCNO			
N	L^1_h		L^∞_h		L^1_h		L^∞_h	
	error	rate	error	rate	error	rate	error	rate
100	3.71e-02	-	4.37e-00	-	5.35e-03	-	1.30e-00	-
200	1.94e-02	0.93	2.32e-00	0.91	1.50e-03	1.83	5.64e-01	1.20
400	9.95e-03	0.96	1.19e-00	0.95	4.02e-04	1.90	2.35e-01	1.26
600	6.69e-03	0.98	8.05e-01	0.97	1.84e-04	1.92	1.40e-01	1.28
800	5.03e-03	0.99	6.07e-01	0.98	1.06e-04	1.92	9.67e-02	1.28
1000	4.04e-03	0.99	4.87e-01	0.99	6.89e-05	1.93	7.24e-02	1.29

Table 5.2: Order of convergence for the advecting density wave problem with fluxes involving various versions of ROE-EC.

	KEP				KEP4			
N	L^1_h		L^∞_h		L^1_h		L^∞_h	
	error	rate	error	rate	error	rate	error	rate
100	2.58e-03	-	2.57e-01	-	2.98e-05	-	3.03e-03	-
200	6.49e-04	1.99	6.41e-02	2.00	1.89e-06	3.97	1.91e-04	3.99
400	1.62e-04	1.99	1.60e-02	2.00	1.18e-07	3.99	1.20e-05	3.99
600	7.22e-05	1.99	7.11e-03	2.00	2.34e-08	3.99	2.38e-06	3.99
800	4.06e-05	1.99	4.00e-03	2.00	7.44e-09	3.99	7.54e-07	3.99
1000	2.59e-05	1.99	2.56e-03	2.00	3.05e-09	3.98	3.09e-07	3.99

Table 5.3: Order of convergence for the advecting density wave problem with fluxes involving various versions of KEP.

up to a final time of $t_f = 0.2$ with CFL=0.4. The exact solution is obtained using the exact Riemann solver available in [123]. The Roe scheme gives an entropy violating jump in the expansion region where the flow becomes sonic, as shown in Figure 5.3. This is not surprising as we have not added any entropy fix. However, both the ROE-ES and KEPES schemes, being entropy stable, are able to remedy this issue to a large extent. In fact, the numerical solutions of ROE-ES and KEPES are indistinguishable. The comparison in Figure 5.4 shows that the high-resolution KEPES-TeCNO scheme is significantly more accurate, as compared to KEPES.



(a) Density

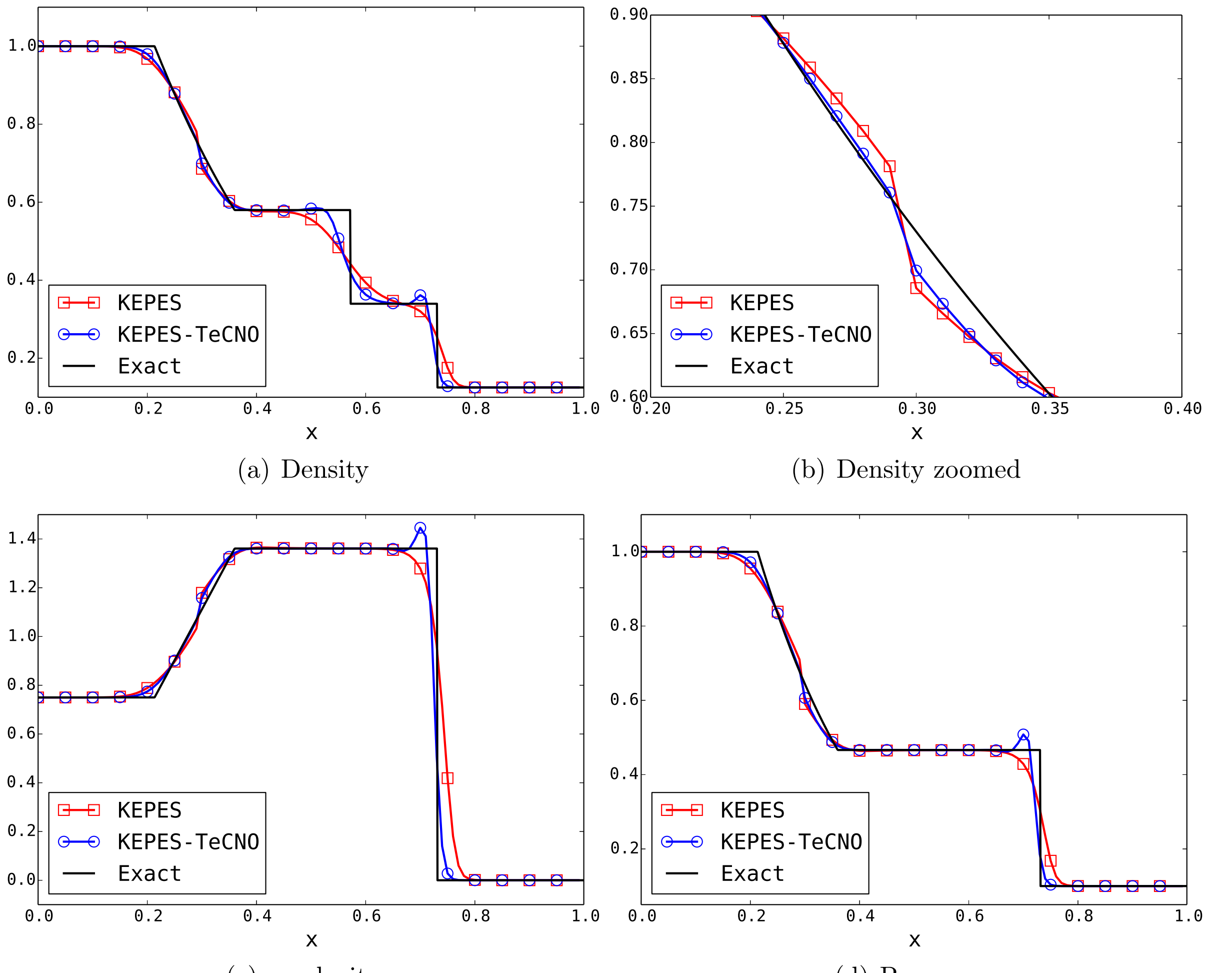
(b) Density zoomed

(c) x-velocity

(d) Pressure

Figure 5.3: Modified shock tube problem: first-order schemes.

Note that the entropy stable schemes also give rise to a small jump near the sonic point, as can be seen in Figure 5.3(b) and 5.4(b). This is because entropy stable schemes also have a vanishing eigenvalue in the expansion fan. However, due to the entropy conservative form of the central part of the flux, they do not give rise to an entropy violating shock. For the Roe scheme, on the other hand, the central part of the flux is a simple arithmetic average $(\mathbf{f}(\mathbf{U}_i) + \mathbf{f}(\mathbf{U}_{i+1}))/2$, which is not entropy conservative. Convergence for the KEPES-TeCNO scheme with mesh refinement is demonstrated in Figure 5.5. Clearly, the jump near the sonic point reduces with mesh refinement. However, this is not the case



(a) Density

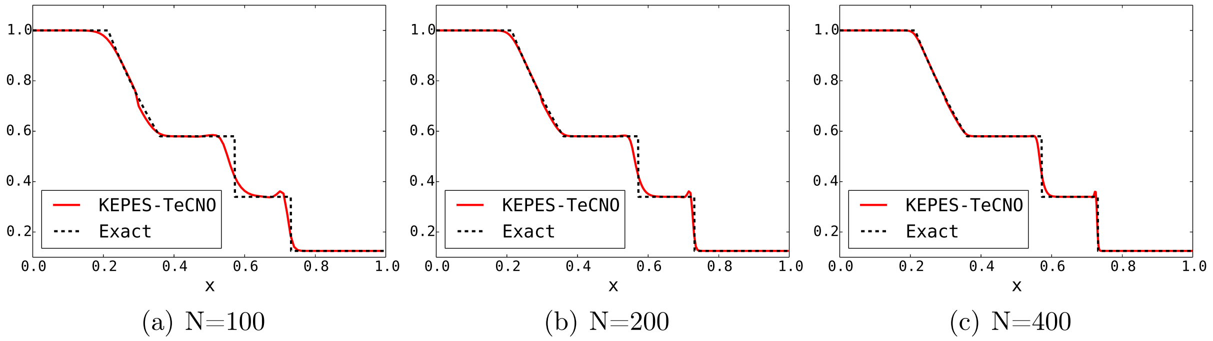
(b) Density zoomed

(c) x-velocity

(d) Pressure

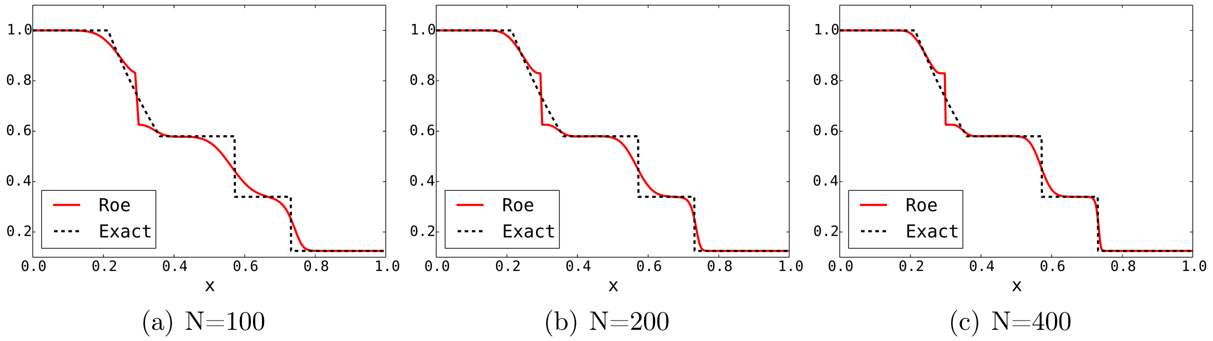
Figure 5.4: Modified shock tube problem: comparison of first-order and high-order scheme.

for the Roe scheme, as shown in Figure 5.6, with the entropy violating jump persisting on all meshes.



(a) N=100 (b) N=200 (c) N=400

Figure 5.5: Modified shock tube problem (Density): mesh refinement study with KEPES-TeCNO.



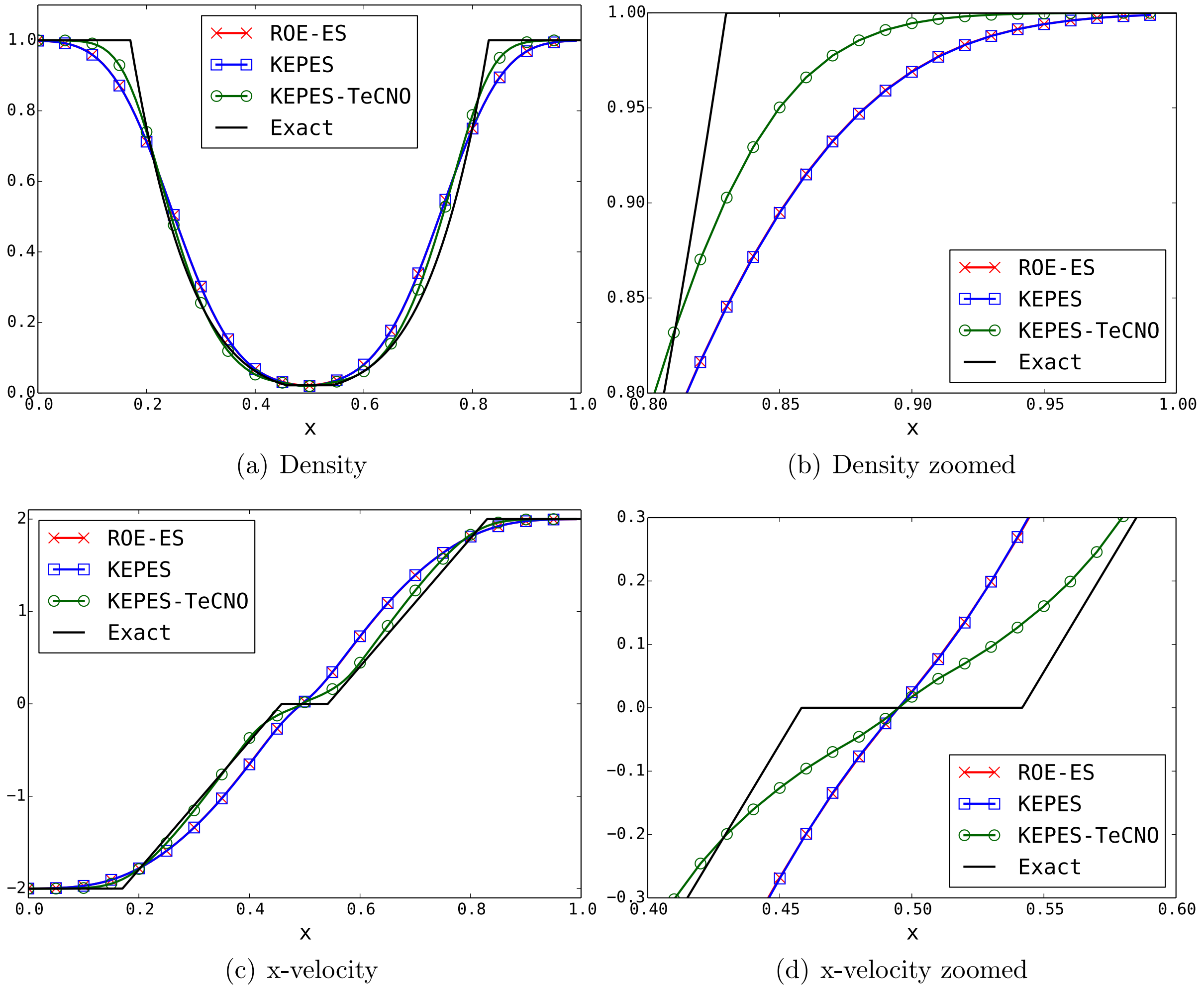
(a) N=100 (b) N=200 (c) N=400

Figure 5.6: Modified shock tube problem (Density): mesh refinement study with Roe scheme.

The KEPES-TeCNO scheme can give rise to minor overshoots near shocks. This can be attributed to the presence of insufficient numerical dissipation in the scheme. However, with mesh refinement, the overshoot does not grow in size and reduces to a Dirac mass. In other words, the numerical solution converges to the exact one in the sense of L^1.

5.8.3 Low density problem

This problem also corresponds to a one-dimensional shock-tube problem, which is used to test the ability of schemes to preserve positivity of density and pressure. It has initial conditions with the left state $(\rho_L, u_L, p_L) = (1.0, -2.0, 0.4)$ and the right state $(\rho_R, u_R, p_R) = (1.0, 2.0, 0.4)$. The domain is $[0, 1]$ with the initial discontinuity at $x_0 = 0.5$. The computations are made on a mesh with $N = 100$ cells and transmissive boundary conditions, up to a final time of $t_f = 0.12$ with CFL=0.4. The exact solution consists of two symmetric rarefaction waves, with the region between the two non-linear waves being close to vacuum. The original Roe scheme fails for this test case. ROE-ES, KEP-ES schemes give almost identical results, and are able to preserve the positivity of density and pressure, as seen in Figure 5.7. The solution with KEPES-TecNO is significantly better resolved, as compared to the first-order KEPES scheme. Convergence of the KEPES-TeCNO scheme with mesh refinement is demonstrated in Figure 5.8.



(a) Density

(b) Density zoomed

(c) x-velocity

(d) x-velocity zoomed

Figure 5.7: Low density problem.

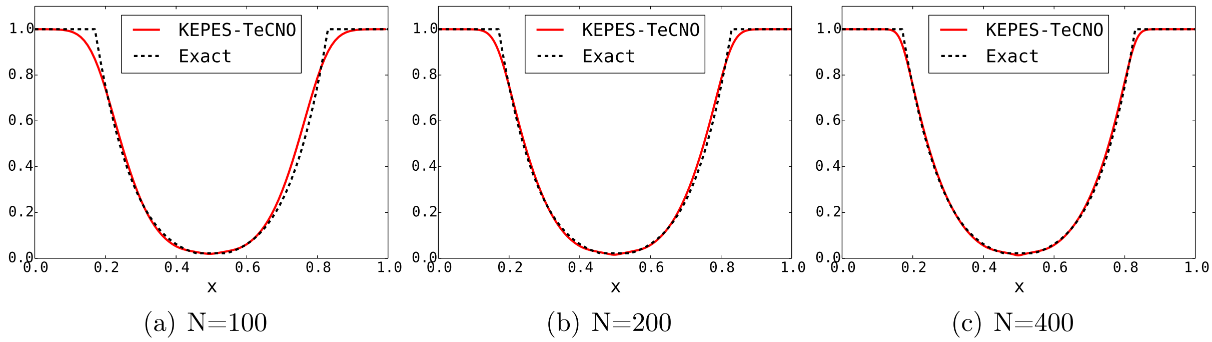
5.8.4 Isentropic vortex

This test, proposed by Yee et al. [136], describes the advection of a smooth isentropic vortex. The initial conditions of the flow on the domain $[-5,5]\times[-5,5]$ are prescribed as

$$\rho = \left[1 - \frac{\beta^2(\gamma-1)}{8\gamma\pi^2}\exp\left(1-r^2\right)\right]^{\frac{1}{(\gamma-1)}}, \quad u_1 = M\cos(\alpha) - \frac{\beta(y-y_c)}{2\pi}\exp\left(\frac{1-r^2}{2}\right),$$

$$u_2 = M\sin(\alpha) + \frac{\beta(x-x_c)}{2\pi}\exp\left(\frac{1-r^2}{2}\right), \quad r = \sqrt{(x-x_c)^2+(y-y_c)^2}.$$

The pressure is initialized by $p = \rho^\gamma$ and β determines the vortex strength. The initial vortex, centered at (x_c, y_c), is passively advected in a direction determined by the angle α, and a velocity determined by the free-stream Mach number M.



(a) N=100 (b) N=200 (c) N=400

Figure 5.8: Low density tube problem (Density): mesh refinement study with KEPES-TeCNO.

We choose $(x_c, y_c) = (0, 0)$, $\alpha = 0°$, $\beta = 5$ and impose periodic boundary conditions. Since this problem has a smooth solution, it is ideal to test the long-time simulation capabilities of the central schemes discussed so far. In particular, we consider the KEP, KEPEC and ROE-EC fluxes, as well as their fourth-order accurate versions obtained using the interpolation formula (5.15). We advect the vortex till the final time $t_f = 100$ with CFL =0.5, with two values of the free-stream Mach number: i) $M = 0.5$, for which the solution completes 5 periodic cycles at final time, ii) $M = 1$, for which the solution completes 10 periodic cycles at final time. The domain is discretized using a mesh with 50 cells in each direction.

The density contour plots for $M = 0.5$ are shown in Figure 5.9. The solution with the KEP scheme blows up well before reaching the final time. The ROE-EC scheme is stable, but the solution profile at the final time is polluted by noise. The KEPEC scheme, on the other hand, succeeds in preserving the vortex structure till the end of the simulation, although the vortex has drifted away from the origin due to dispersive errors. If we consider the fourth-order fluxes, then the solution with KEP4 does not blow-up, but vortex structure is completely destroyed due to accumulation of errors. The ROE-EC4 and KEPEC4 perform almost identically, both succeeding in preserving the vortex.

The evolution of relative total kinetic energy, i.e., the total kinetic energy scaled by its initial value, is shown in Figure 5.10. The exact value of relative total kinetic energy for the isentropic vortex is unity for all time. Although the KEP flux is kinetic energy preserving, it shows a sudden increase in kinetic energy close to time $t = 40$, and blows up soon after. The fourth-order KEP4 flux preserves kinetic energy for a longer time as compared to KEP, but eventually deviates from its expected behaviour. The ROE-EC flux performs much better than both the KEP and KEP4 fluxes, despite not satisfying the form (5.9) required to prove kinetic energy preservation. The KEPEC flux is the best performer among the second-order fluxes, with the value of relative total kinetic energy oscillating at close proximity to unity, as time evolves. The evolution of kinetic energy with ROE-EC4 and KEPEC4 is indistinguishable, with the oscillatory behaviour observed with their second-order counterparts no longer observable.

The total entropy is conserved in time for smooth solutions (assuming periodic boundary conditions). For the exact solution of the isentropic vortex, the entropy function η chosen in accordance to (3.8) is identically zero, since $s \equiv 0$. Thus, the total entropy is

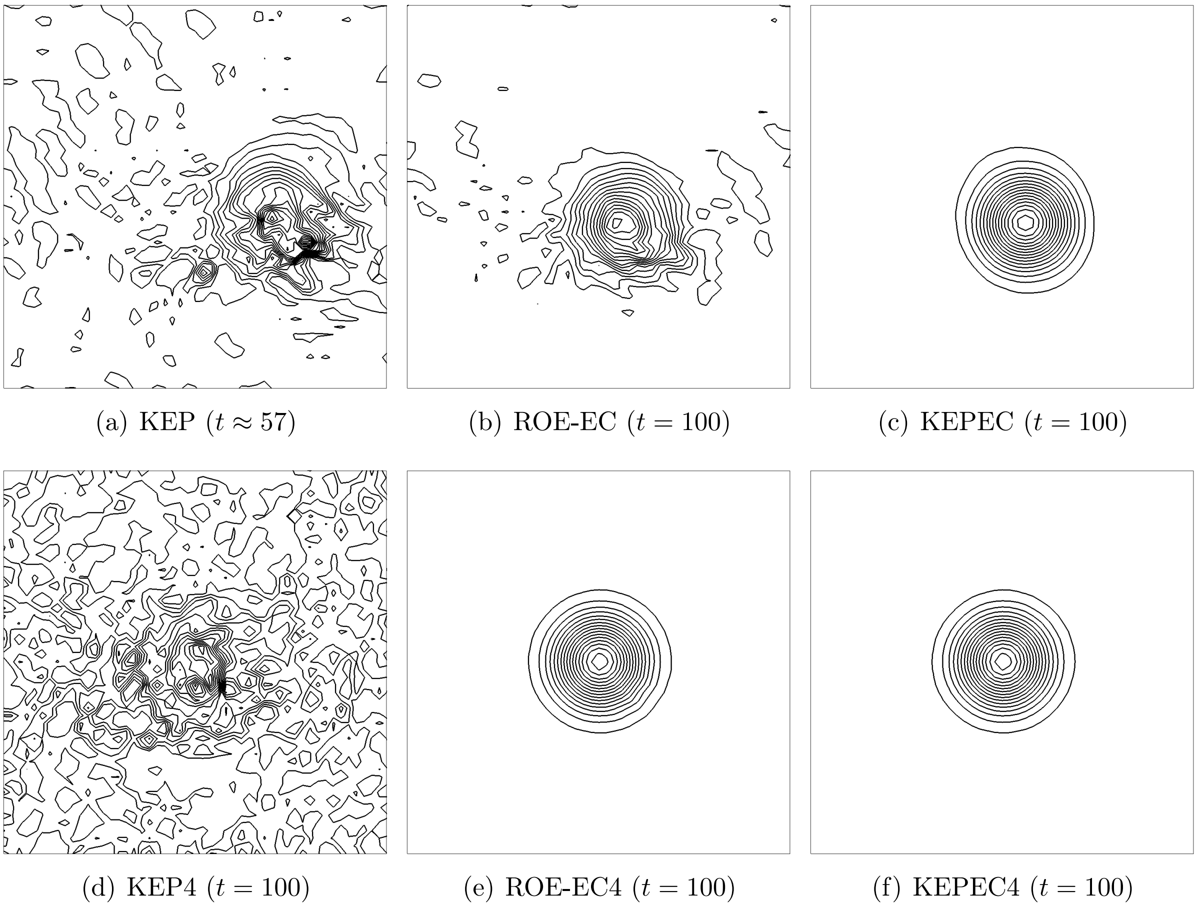
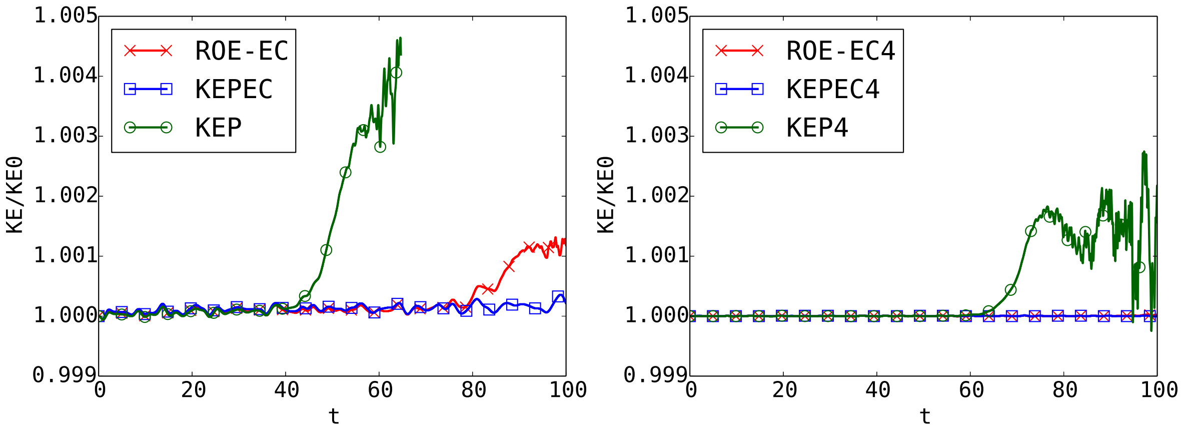

(a) KEP ($t \approx 57$)
(b) ROE-EC ($t = 100$)
(c) KEPEC ($t = 100$)
(d) KEP4 ($t = 100$)
(e) ROE-EC4 ($t = 100$)
(f) KEPEC4 ($t = 100$)

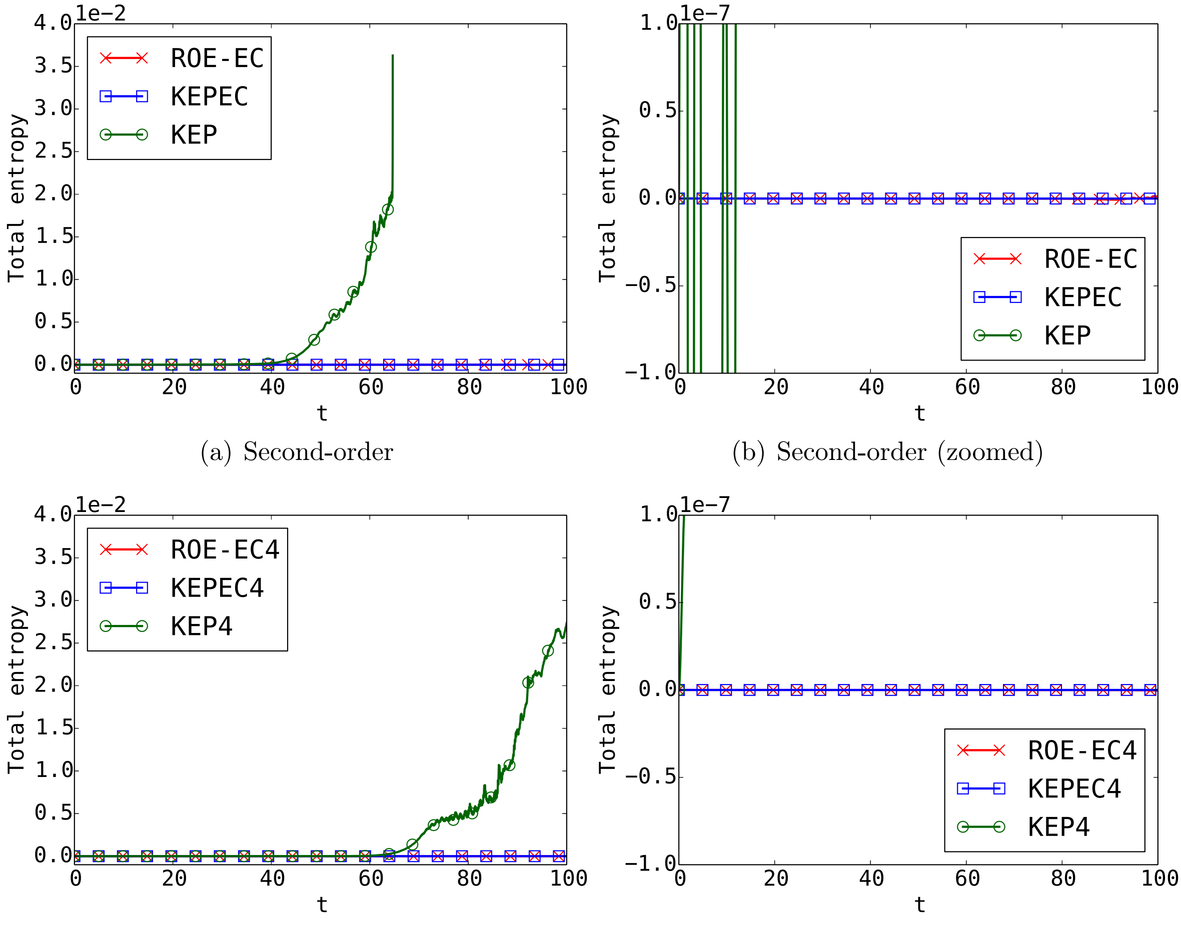
Figure 5.9: Isentropic vortex with $M = 0.5$, 50×50 cells: density contours.



(a) Second-order
(b) Fourth-order

Figure 5.10: Evolution of relative total kinetic energy for isentropic vortex with $M = 0.5$, 50×50 cells.

identically zero for all time. The evolution of total entropy with the various numerical fluxes is shown in Figure 5.11. While the entropy increases for KEP and KEP4, the entropy conservative fluxes ROE-EC, KEPEC and their fourth-order versions are able to conserve the total entropy till the final time.



(a) Second-order

(b) Second-order (zoomed)

(c) Fourth-order

(d) Fourth-order (zoomed)

Figure 5.11: Evolution of total entropy for isentropic vortex with $M = 0.5$, 50×50 cells.

We perform a similar analysis with the Mach number chosen as $M = 1$. The density contour plots for this set-up are shown in Figure 5.12. The solution with KEP and KEP4 schemes blow-up before reaching the final time. The solution profiles with ROE-EC and ROE-EC4 are much worse compared to the previous set-up, with the vortex structure with ROE-EC severely polluted by noise. However, both KEPEC and KEPEC4 are still able to preserve the vortex structure, till the end of the simulation. The evolution of relative total kinetic energy, shown in Figures 5.13, indicate that the ability of the ROE-EC and ROE-EC4 schemes to preserve kinetic energy has degraded. However, both KEPEC and KEPEC4 perform as well as they did for $M = 0.5$. At first glance, the evolution of total entropy shown in Figure 5.14 indicates that ROE-EC, KEPEC and their fourth-order versions are still able to conserve entropy equally well till final time. However, zooming in further shows that this is true for KEPEC and KEPEC4, but not for ROE-EC and ROE-EC4.

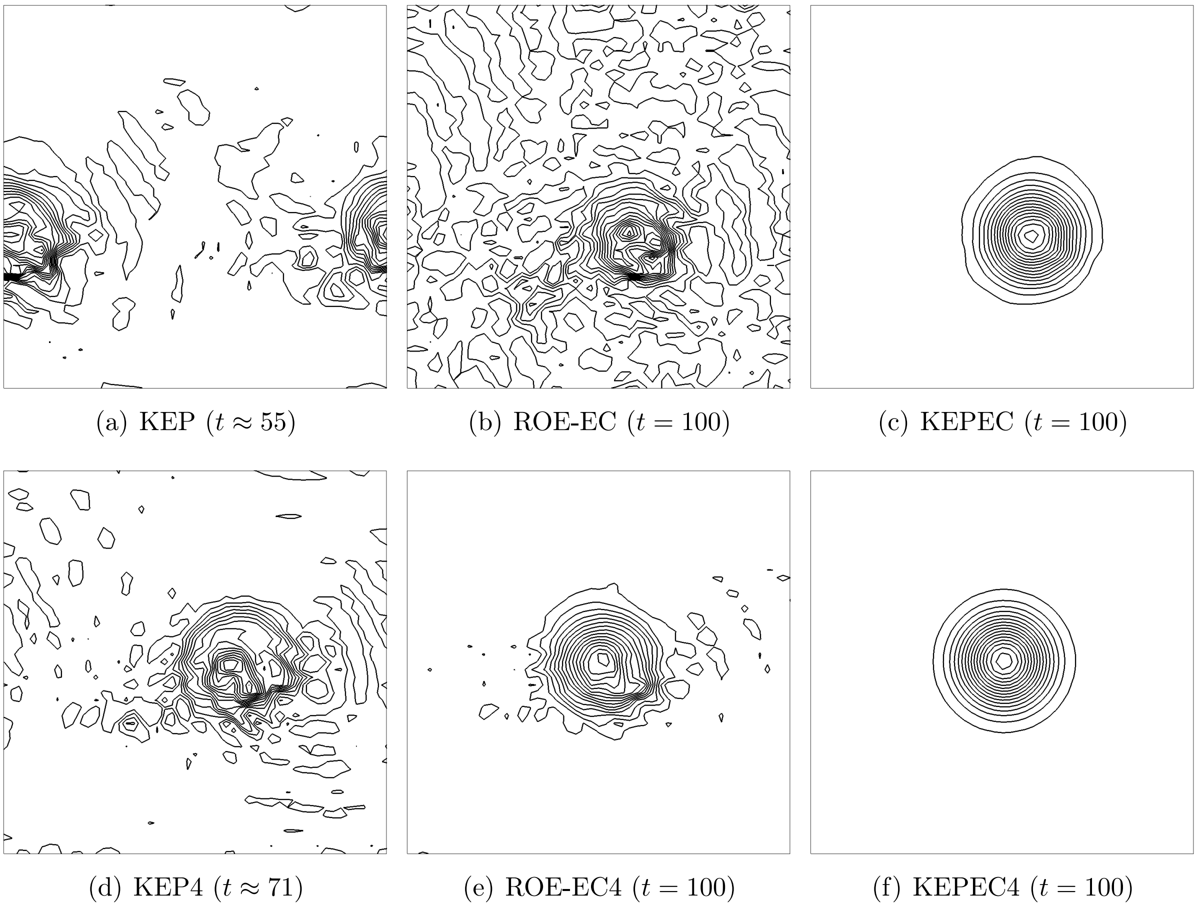
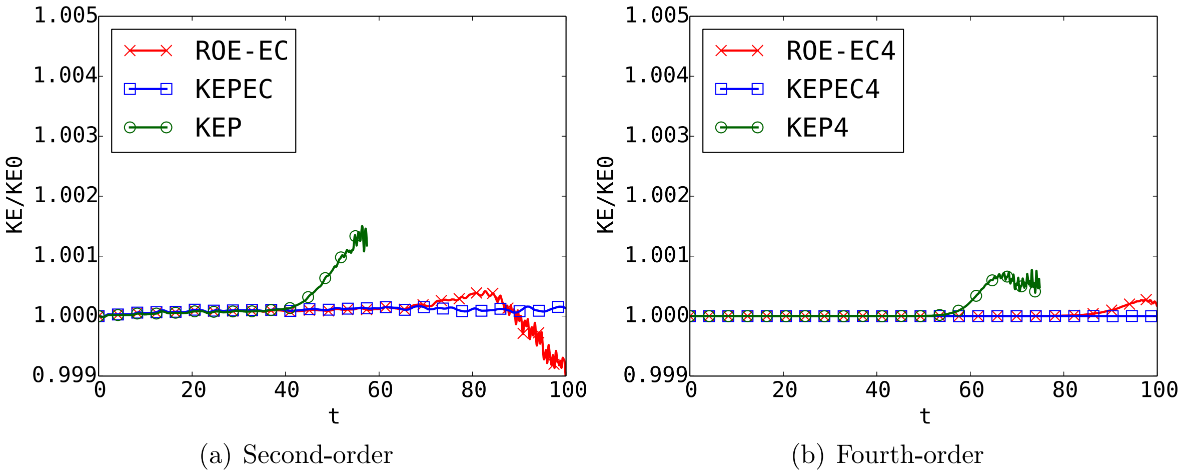
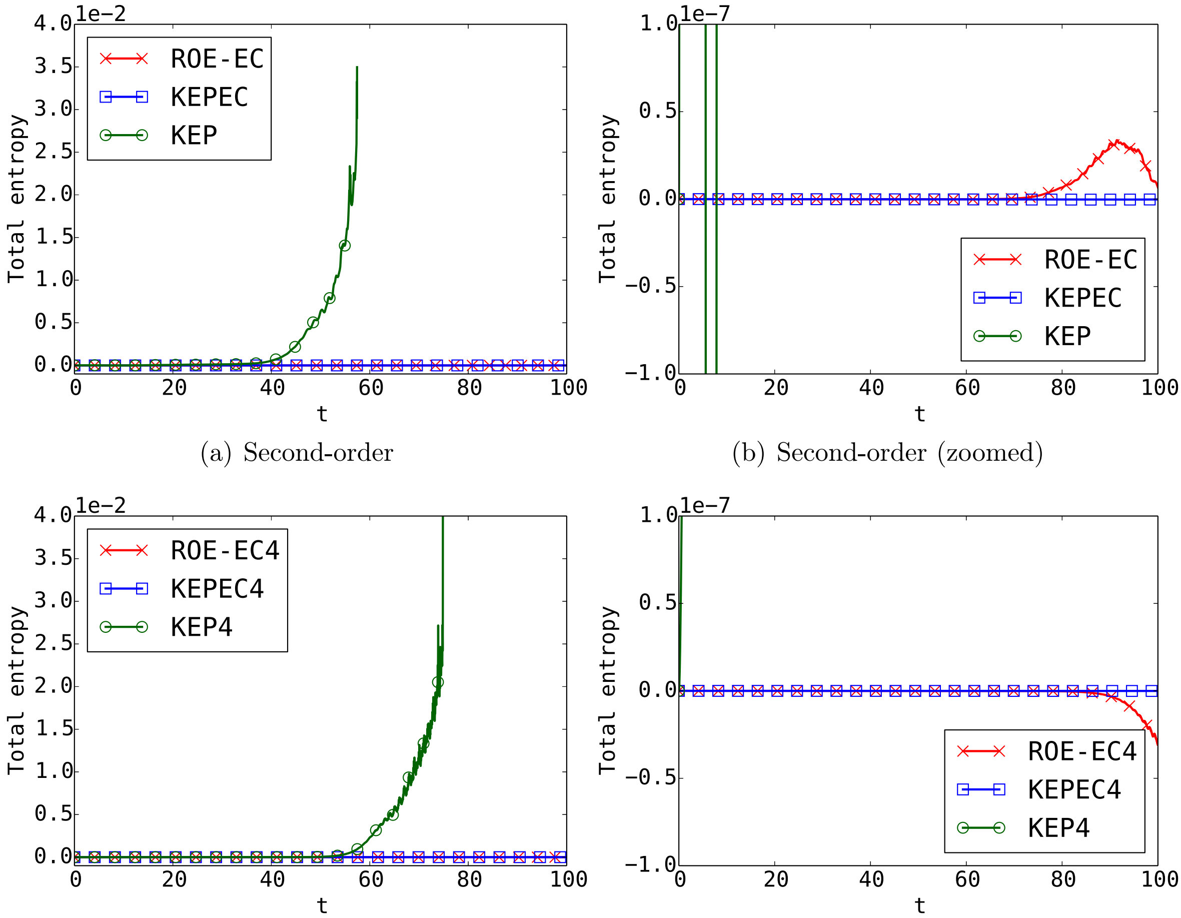

(a) KEP ($t \approx 55$)
(b) ROE-EC ($t = 100$)
(c) KEPEC ($t = 100$)
(d) KEP4 ($t \approx 71$)
(e) ROE-EC4 ($t = 100$)
(f) KEPEC4 ($t = 100$)

Figure 5.12: Isentropic vortex with $M = 1$, 50×50 cells: density contours.



(a) Second-order
(b) Fourth-order

Figure 5.13: Evolution of relative total kinetic energy for isentropic vortex with $M = 1$, 50×50 cells.



(a) Second-order

(b) Second-order (zoomed)

(c) Fourth-order

(d) Fourth-order (zoomed)

Figure 5.14: Evolution of total entropy for isentropic vortex with $M = 1$, 50×50 cells.

The results with $M = 0.5$ indicate that the entropy conservative nature of the ROE-EC and KEPEC fluxes makes them (and their high-order version) more accurate for long-time simulation, as compared to the KEP flux that is only kinetic energy preserving. However, we can infer from the $M = 1$ results that KEPEC flux, which is also kinetic energy preserving, is more accurate as compared to the ROE-EC flux. Thus we make the following conjecture.

Conjecture. *A numerical flux for the Euler equations which is both entropy conservative and kinetic energy preserving, is more accurate for long-time simulations of smooth solutions, as compared to a flux that is either entropy conservative or kinetic energy preserving, but not both.*

Unfortunately, we can not give a formal proof to this conjecture at present. This conjecture needs to validated with further experiments, which will be the topic of future work.

5.8.5 Shock vortex interaction

This problem consists of the interaction of a left-moving shock wave with a right-moving vortex [35]. The initial shock discontinuity on the domain $[0, 1] \times [0, 1]$ is given by

$$\mathbf{U}_0(x) = \begin{cases} \mathbf{U}_L & \text{if } x < 0.5 \\ \mathbf{U}_R & \text{if } x \geqslant 0.5 \end{cases},$$

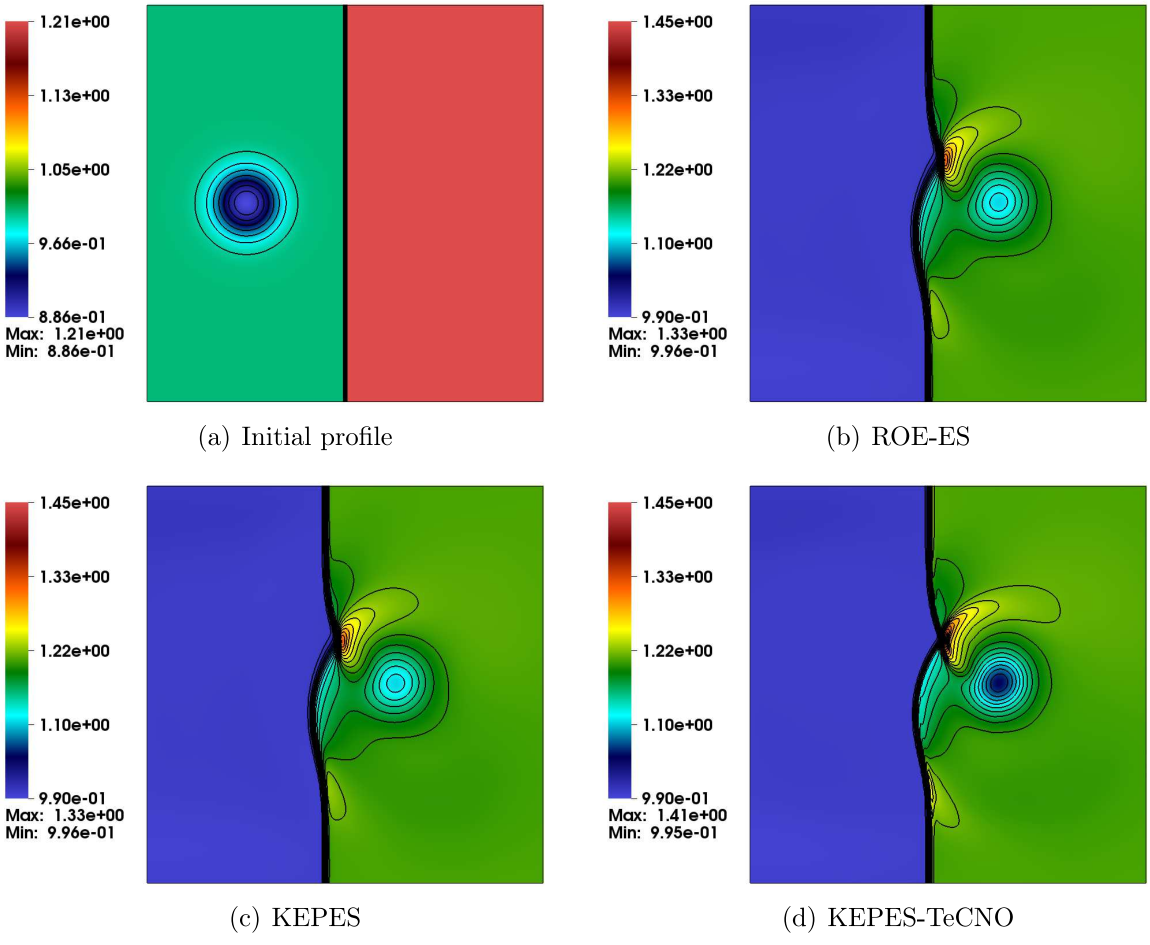
where the left state is given by $(\rho_L, u_{1,L}, u_{2,L}, p_L) = (1, \sqrt{\gamma}, 0, 1)$ while the right state is given by

$$p_R = 1.3, \quad \rho_R = \rho_L \left(\frac{\gamma - 1 + (\gamma + 1)p_R}{\gamma + 1 + (\gamma - 1)p_R} \right)$$
$$u_{1,R} = \sqrt{\gamma} + \sqrt{2} \left(\frac{1 - p_r}{\sqrt{\gamma - 1 + p_R(\gamma + 1)}} \right), \quad u_{2,R} = 0.$$

The left state $\mathbf{U}_L$ is superposed onto a vortex described by the following perturbations in the velocity, temperature and physical entropy respectively

$$\delta u_1 = \epsilon \frac{(y - y_c)}{r_c} \exp\left(\beta(1 - r^2)\right), \quad \delta u_2 = -\epsilon \frac{(x - x_c)}{r_c} \exp\left(\beta(1 - r^2)\right),$$
$$\delta\theta = -\frac{\gamma - 1}{4\beta\gamma} \epsilon^2 \exp\left(2\beta(1 - r^2)\right), \quad \delta s = 0,$$

where $r^2 = ((x - x_c)^2 + (y - y_c)^2)/r_c^2$. Note that the perturbation describes a steady isentropic vortex solution for the two-dimensional Euler equations. The various parameters of the perturbation are chosen as $\epsilon = 0.3$, $r_c = 0.05$, $\beta = 0.204$ and $(x_c, y_c) = (0.25, 0.5)$. The initial profile of solution is shown in Figure 5.15(a). The domain is discretized with 200 cells in each direction with transmissive boundary conditions, and the final time is chosen as $t_f = 0.35$ with CFL=0.5. The final numerical solutions are shown in Figure 5.15(b)-(d). The KEPES and ROE-ES schemes give comparable results, but the shock profile is quite diffused. Sharper profiles are obtained with the higher-order KEPES-TeCNO scheme.



(a) Initial profile

(b) ROE-ES

(c) KEPES

(d) KEPES-TeCNO

Figure 5.15: Density profiles for shock-vortex interaction.

5.8.6 3D Taylor-Green vortex

We now test the DNS capabilities of the KEPEC flux, by considering the three-dimensional viscous Taylor-Green vortex (TGV3D) problem [75]. The initial flow field on the domain $\Omega = [0, 2L\pi]^3$ is given by

$$\begin{aligned}
u &= V_0 \sin\left(\frac{x}{L}\right)\cos\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right),\\
v &= -V_0 \cos\left(\frac{x}{L}\right)\sin\left(\frac{y}{L}\right)\cos\left(\frac{z}{L}\right),\\
w &= 0,\\
p &= p_0 + \frac{\rho_0 V_0^2}{16}\left(\cos\left(\frac{2x}{L}\right)+\cos\left(\frac{2y}{L}\right)\right)\left(\cos\left(\frac{2z}{L}\right)+2\right),
\end{aligned}$$

with periodic boundary conditions. Starting from a single large scale of the initial conditions in terms of sine and cosine waves, the flow rapidly evolves into fully homogenous turbulence. The kinetic energy generated by the mean flow is transferred down the energy cascade by the non-linear interactions, and finally dissipated by the viscous forces of the eddies at the smallest scale, also known as the *Kolmogorov scale* [39]. Thus, accurate representation of the kinetic energy and viscous stresses by the numerical scheme is essential to observe the correct decay of energy.

For the current problem, the Reynolds number of the flow is $Re = 1600$, with a Mach number of $M = 0.1$. The mean free-stream velocity is $V_0 = 1$, the domain length scale is $L = 1$, the initial density is $\rho_0 = 1$, the gas constant is taken as $R = 1$ and the Prandtl number as $Pr = 0.71$. Based on these values, the remaining parameters are chosen as follows

$$p_0 = \frac{V_0^2\rho_0}{M^2\gamma}, \quad \mu = \frac{LV_0\rho_0}{Re}, \quad c_p = \frac{R\gamma}{\gamma-1},$$

with the coefficient of heat conductance κ chosen according to (3.12). The physical duration of the computation is based on the characteristic convective time $t_c = V_0/L$, and is set to $t_f = 20t_c$. We consider two grid sizes for our numerical simulations, namely 128^3 and 256^3. The mesh is under-resolved for these grid sizes, and thus unable to capture the flow features at the smallest scales. We use a reference solution corresponding to an incompressible flow, obtained using a dealiased pseudo-spectral code [22, 128] on a 512^3 grid, on which all scales are well resolved. However, solving the Navier-Stokes equations on a grid of size 512^3 is very demanding on computational resources. Thus, we wish to test the performance of the KEPEC flux in under-resolved scenarios. We use the notations A, B and C to denote the meshes with sizes 128^3, 256^3 and 512^3 respectively.

The first quantity of interest is the temporal evolution of the non-dimensional total kinetic energy

$$E_{\mathcal{K}} = \frac{1}{\rho_0|\Omega|V_0^2}\int_\Omega \rho\frac{|\mathbf{u}|^2}{2}\mathrm{d}\mathbf{x}.$$

Recall from Section 3.3, that the *decay rate* of (non-dimensional) total kinetic energy, which we denote by $\epsilon = -\frac{\mathrm{d}E_{\mathcal{K}}}{\mathrm{d}t}$, is the sum of two components

$$\epsilon_1 = \frac{L}{\rho_0|\Omega|V_0^3}\int_\Omega \sum_{i=1}^{3}\sum_{j=1}^{3}\tau_{ij}\partial_{x_j}u_i \,\mathrm{d}\mathbf{x}, \qquad \epsilon_2 = -\frac{L}{\rho_0|\Omega|V_0^3}\int_\Omega p\nabla\cdot\mathbf{u}\,\mathrm{d}\mathbf{x}.$$

Since the flow is nearly incompressible, the main contribution is expected to be from ϵ_1. We are also interested in the temporal evolution of the *total enstrophy*

$$\mathcal{E} = \frac{L^2}{\rho_0|\Omega|\mathbf{V}_0^2}\int_\Omega \rho\frac{|\boldsymbol{\omega}|^2}{2}\,\mathrm{d}\mathbf{x},$$

which is obtained in terms of the *vorticity* of the flow $\boldsymbol{\omega} = \nabla \times \mathbf{u}$. This is indeed an important diagnostic tool, as it can be shown that $\epsilon = 2\mu\mathcal{E}/\rho_0$ holds exactly for incompressible flows, and approximately for compressible flow at low Mach numbers.

In Section 5.6, we described two ways of discretizing the viscous flux for the multi-dimensional Navier-Stokes equations. The first formulation ensured that the viscous fluxes dissipated the total kinetic energy in a consistent manner. The second formulation ensured the entropy stability of the scheme. We use the notations *NS-KEP* and *NS-ES* to indicate the first and second formulation respectively. The DNS of compressible Navier–Stokes equations seeks to resolve all physically relevant time and length scales associated with turbulence. Resolution of these phenomena requires strict temporal error tolerances. Thus, we use a fourth-order Runge-Kutta scheme for time-integration, with CFL=0.1.

Extending the discussion in Section 5.7.3 to three-dimensions, the decay rates of kinetic energy are approximated directly using the numerical fluxes, in the following manner

$$\begin{aligned}
\epsilon_1 \approx &-\sum_{i,j,k}\left[\frac{1}{2}|\mathbf{u}|^2_{i,j,k}\left(\frac{F^{x,\rho}_{i+\frac{1}{2},j,k}-F^{x,\rho}_{i-\frac{1}{2},j,k}}{\Delta x}+\frac{F^{y,\rho}_{i,j+\frac{1}{2},k}-F^{y,\rho}_{i,j-\frac{1}{2},k}}{\Delta y}+\frac{F^{z,\rho}_{i,j,k+\frac{1}{2}}-F^{z,\rho}_{i,j,k-\frac{1}{2}}}{\Delta z}\right)\right]|\Omega_h| \\
&+\sum_{i,j,k}\left[\frac{\left\langle \mathbf{u}_{i,j,k},\mathbf{F}^{x,m}_{i+\frac{1}{2},j,k}-\mathbf{F}^{x,m}_{i-\frac{1}{2},j,k}\right\rangle}{\Delta x}+\frac{\left\langle \mathbf{u}_{i,j,k},\mathbf{F}^{y,m}_{i,j+\frac{1}{2},k}-\mathbf{F}^{y,m}_{i,j-\frac{1}{2},k}\right\rangle}{\Delta y}\right. \\
&\qquad\qquad\left.+\frac{\left\langle \mathbf{u}_{i,j,k},\mathbf{F}^{z,m}_{i,j,k+\frac{1}{2}}-\mathbf{F}^{z,m}_{i,j,k-\frac{1}{2}}\right\rangle}{\Delta z}\right]|\Omega_h|,
\end{aligned}$$

$$\begin{aligned}
\epsilon_2 \approx \sum_{i,j,k}&\left[\left\langle [\mathcal{T}^{xyz}\mathbf{u}]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}},\mathbf{G}^{x,m}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right\rangle+\left\langle [\mathcal{T}^{yxz}\mathbf{u}]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}},\mathbf{G}^{y,m}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right\rangle\right. \\
&\qquad\left.+\left\langle [\mathcal{T}^{zxy}\mathbf{u}]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}},\mathbf{G}^{z,m}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right\rangle\right]|\Omega_h|,
\end{aligned}$$

where $\mathcal{T}^{xyz},\mathcal{T}^{yxz},\mathcal{T}^{zxy}$ are the three-dimensional extensions of the operators $\mathcal{T}^{xy},\mathcal{T}^{yx}$, while $|\Omega_h| = \Delta x\Delta y\Delta z$ is the volume of each cell. We of course need to multiply the approximations by the factor $L/(\rho_0|\Omega|\mathbf{V}_0^2)$ to non-dimensionalize them. The total enstrophy is approximated by

$$\mathcal{E} \approx \frac{1}{2}\sum_{i,j,k}\left[\rho_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}|\boldsymbol{\omega}|^2_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}}\right]|\Omega_h|,$$

where the nodal values of density is obtained using the averaging (5.38), while the vorticity is approximated at the nodes by

$$\boldsymbol{\omega}_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} = \begin{pmatrix} [\mathcal{T}^{yxz}u_3]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - [\mathcal{T}^{zxy}u_2]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \\ [\mathcal{T}^{zxy}u_1]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - [\mathcal{T}^{xyz}u_3]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \\ [\mathcal{T}^{xyz}u_2]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} - [\mathcal{T}^{yxz}u_1]_{i+\frac{1}{2},j+\frac{1}{2},k+\frac{1}{2}} \end{pmatrix},$$

which requires the value of $\mathbf{u}$ at the cell-centers. Note that the approximation for ε needs to be scaled by the factor $L^2/(\rho_0|\Omega|\mathbf{V}_0^2)$ to non-dimensionalize it.

We first simulate the test case using a simple central average to approximate the inviscid flux, i.e.,

$$\mathbf{F}^x_{i+\frac{1}{2},j,k} = \frac{\mathbf{f}_1(\mathbf{U}_{i,j,k}) + \mathbf{f}_1(\mathbf{U}_{i+1,j,k})}{2},$$

with similar expression for the inviscid flux components in the y and z-directions. However, as shown in Figure 5.16, the numerical solution of the Taylor-Green vortex blows up after a finite time (on mesh B), irrespective of the type of discretization chosen for the viscous fluxes. This demonstrates that the central inviscid numerical fluxes must be carefully constructed to compute turbulent flows, in tandem with suitable viscous discretizations.

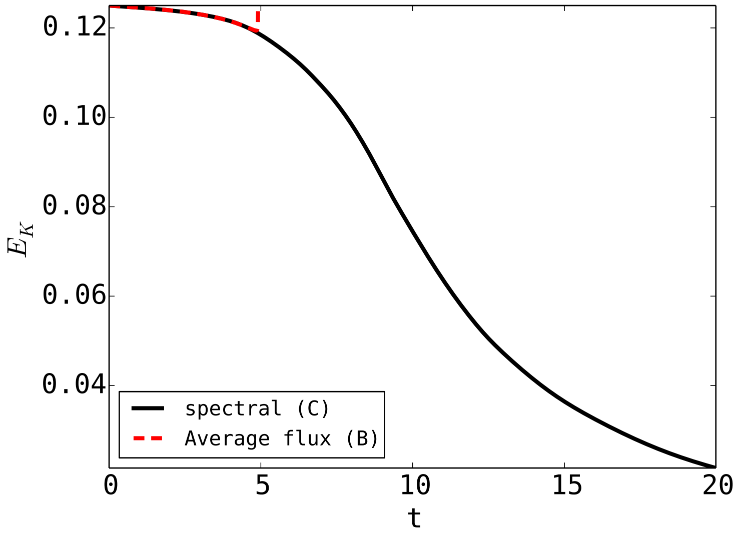
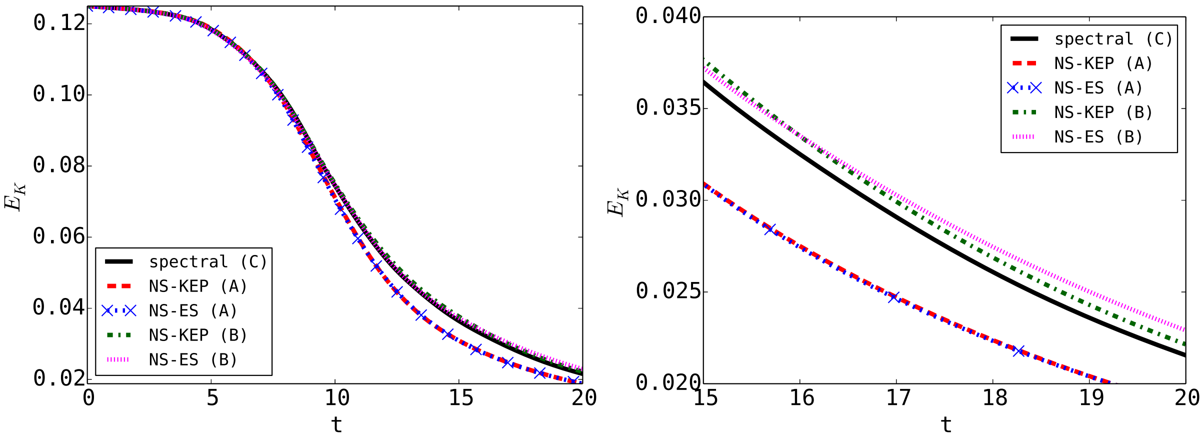


Figure 5.16: Evolution of kinetic energy with a simple central flux on mesh B for the TGV3D problem. The solution blows up in finite time.

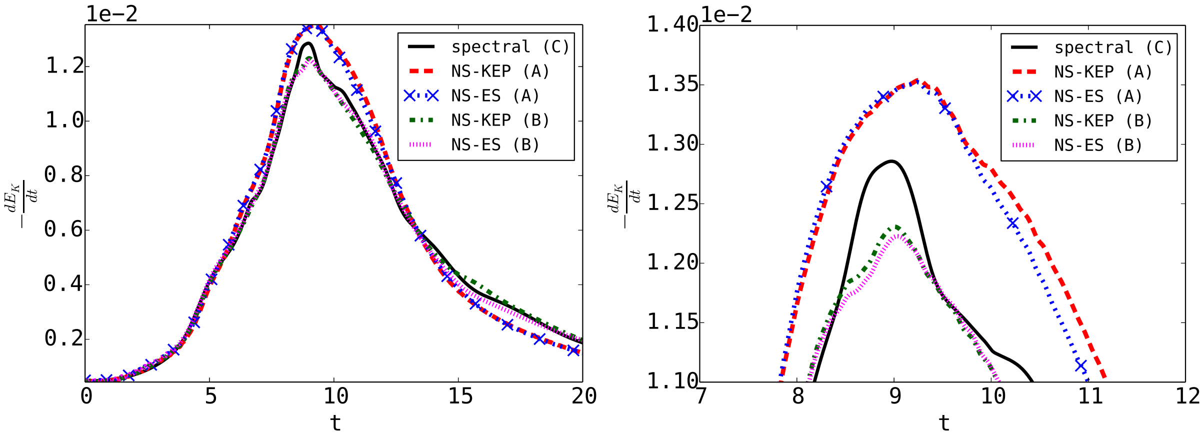
The KEPEC scheme is able to capture the evolution of total kinetic energy and its decay rate fairly well, as compared to the pseudo-spectral results (see Figure 5.17 and 5.18). The solutions improve with mesh refinement, and the evolution of the global quantities with NS-ES and NS-KEP discretizations are comparable. Figure 5.19 reiterates the fact that the contribution of pressure forces to the decay rate kinetic energy is quite small as compared to viscous forces, for low Mach number flows. The numerical solution with the KEPEC flux is unable to capture the peak values of the decay rate predicted by the pseudo-spectral result. This becomes even more evident when we consider the evolution of total enstrophy, as shown in Figure 5.20. A possible explanation for this behaviour, is the fact that the mesh is under-resolved, and thus we are unable to capture the smallest scales which play a key role in the destruction of kinetic energy. Nevertheless, the numerical scheme with the KEPEC flux is able to capture the evolution dynamics reasonably well. We also note from Figure 5.20(b), that the NS-KEP discretization does marginally better in capturing the peak value as compared to the NS-ES formulation.

In Figure 5.21, we plot the iso-contours of the magnitude of dimensionless vorticity $L|\boldsymbol{\omega}|/V_0$, on the periodic face $x = 0$ at time $t = 8t_c$. Although the contour plots with the KEPEC flux seem a bit noisy, they still capture the key features predicted by



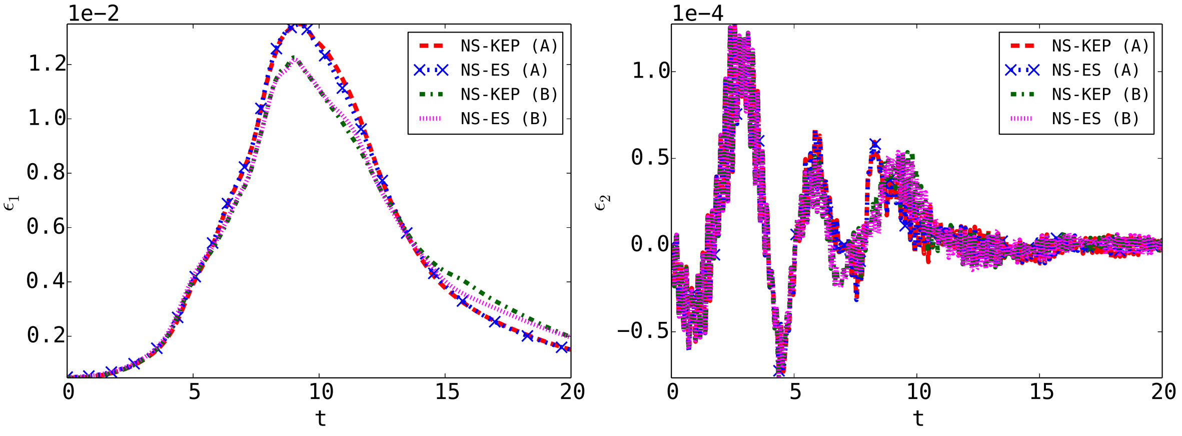
(a) Kinetic energy (b) Zoomed

Figure 5.17: Evolution of kinetic energy with the KEPEC flux for the TGV3D problem



(a) Decay rate (b) Zoomed

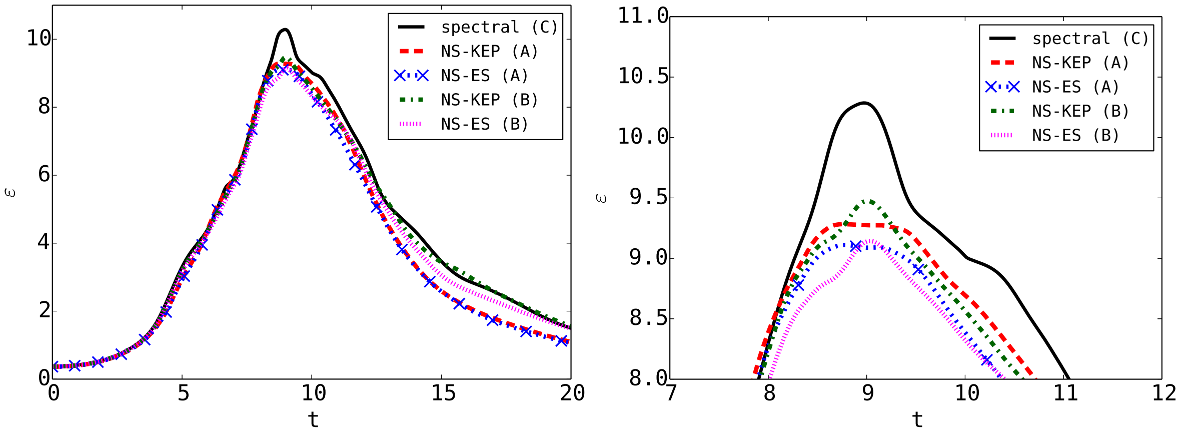
Figure 5.18: Decay rate of kinetic energy with the KEPEC flux for the TGV3D problem.



(a)

(b)

Figure 5.19: Components of kinetic energy decay with the KEPEC flux for the TGV3D problem; (a) contribution due to viscous forces, (b) contribution due to pressure forces.

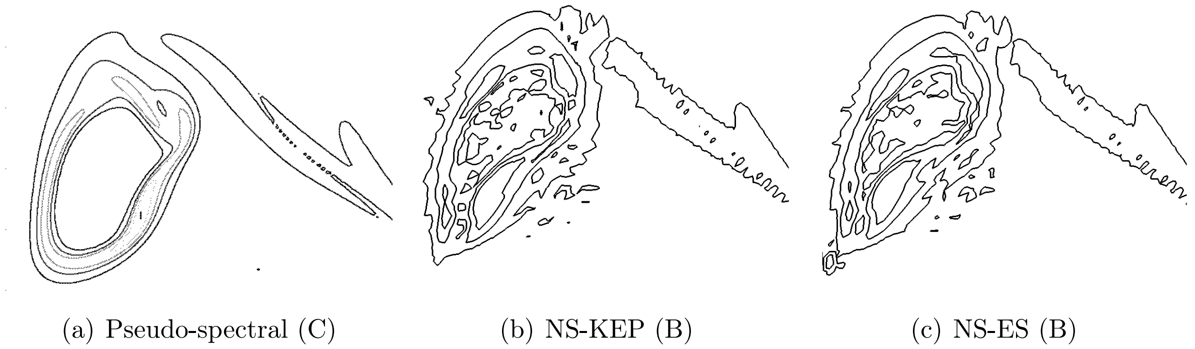


(a) Enstrophy

(b) Enstrophy (zoomed)

Figure 5.20: Evolution of total enstrophy with the KEPEC flux for the TGV3D problem.

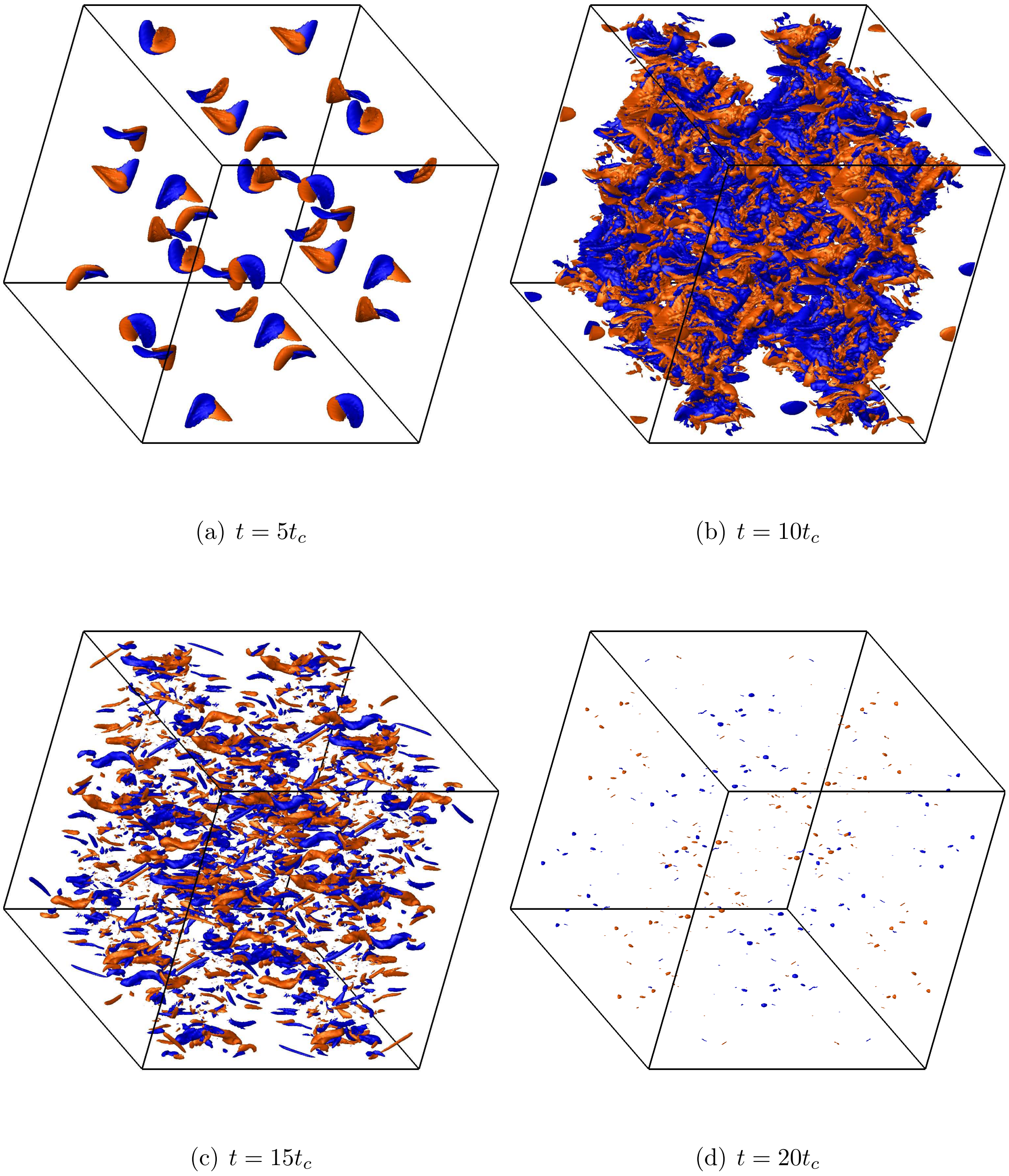
the pseudo-spectral results. We expect the solutions to improve, by either refining the mesh, or applying a suitable filtering technique to the solutions. The iso-contours of the z-component of vorticity are shown in Figure 5.22, to illustrate the vortical motion, transition to turbulence and turbulent decay of the flow.


(a) Pseudo-spectral (C)

(b) NS-KEP (B)

(c) NS-ES (B)

Figure 5.21: Iso-contours of the magnitude of dimensionless vorticity $L|\boldsymbol{\omega}|/V_0$ on the periodic face $x = 0$ at time $t = 8t_c$ with contour levels = 1, 5, 10, 20 ,30.



(a) $t = 5t_c$

(b) $t = 10t_c$

(c) $t = 15t_c$

(d) $t = 20t_c$

Figure 5.22: Iso-surfaces of z-vorticity with NS-KEP on a mesh B.

6. A sign-preserving WENO reconstruction

In Chapter 5, we briefly discussed the sign-preserving ENO interpolation procedure, which can be used to construct high-order entropy stable finite difference schemes. In this chapter, we propose a third-order WENO-type reconstruction which is sign-preserving. The results with the new WENO reconstruction for scalar conservation laws, has been published in [38].

6.1 WENO reconstruction

We begin by describing the basic idea of a third-order WENO procedure, which uses a convex combination of second-order linear reconstructions. For the convenience of notation, we demonstrate the methodology in one-dimension. It can be extended to higher dimensions by using a dimension-by-dimension approach.

The one-dimensional domain is discretized using a uniform mesh, with mesh size $\Delta x = h$. Consider the stencil shown in Figure 6.1 corresponding to the reconstructions at $x_{i+\frac{1}{2}}$. We choose the function $v(x)$ to demonstrate the reconstruction procedure.

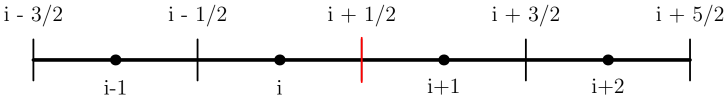


Figure 6.1: Stencil for reconstruction.

Reconstruction from the left: We first consider the reconstruction from the left side of the interface $x_{i+\frac{1}{2}}$. The two stencils considered by ENO-2 to construct linear polynomial approximations in cell I_i, are

$$S_i^0 = \{x_i, x_{i+1}\}, \quad S_i^1 = \{x_{i-1}, x_i\}.$$

The corresponding linear polynomials and their evaluations at $x_{i+\frac{1}{2}}$ are given by

$$p_i^{(0)}(x) = v_i \frac{(x - x_{i+1})}{(x_i - x_{i+1})} + v_{i+1} \frac{(x - x_i)}{(x_{i+1} - x_i)} \qquad \Longrightarrow \qquad v_{i+\frac{1}{2}}^{(0),-} = \frac{v_i}{2} + \frac{v_{i+1}}{2},$$

$$p_i^{(1)}(x) = v_{i-1} \frac{(x - x_i)}{(x_{i-1} - x_i)} + v_i \frac{(x - x_{i-1})}{(x_i - x_{i-1})} \qquad \Longrightarrow \qquad v_{i+\frac{1}{2}}^{(1),-} = -\frac{v_{i-1}}{2} + \frac{3v_i}{2}.$$

Each of the above reconstructions at the interface $x_{i+\frac{1}{2}}$, are second-order accurate. Weighting each of these with non-negative weights $w_{0,i+\frac{1}{2}}$ and $w_{1,i+\frac{1}{2}}$, respectively, we obtain the reconstructed value

$$v^-_{i+\frac{1}{2}} := w_{0,i+\frac{1}{2}}\left(\frac{v_i}{2}+\frac{v_{i+1}}{2}\right) + w_{1,i+\frac{1}{2}}\left(-\frac{v_{i-1}}{2}+\frac{3v_i}{2}\right). \tag{6.1}$$

The weights must be chosen such that third-order accuracy is achieved. Thus, we require that

$$\begin{aligned} v(x_{i+\frac{1}{2}}) + \mathcal{O}(h^3) = v^-_{i+\frac{1}{2}} &= w_{0,i+\frac{1}{2}}\left(\frac{v_i}{2}+\frac{v_{i+1}}{2}\right) + w_{1,i+\frac{1}{2}}\left(-\frac{v_{i-1}}{2}+\frac{3v_i}{2}\right) \\ &= w_{0,i+\frac{1}{2}}\left(v(x_{i+\frac{1}{2}}) + \frac{1}{8}v''(x_{i+\frac{1}{2}})h^2 + \mathcal{O}(h^3)\right) \\ &\quad + w_{1,i+\frac{1}{2}}\left(v(x_{i+\frac{1}{2}}) - \frac{3}{8}v''(x_{i+\frac{1}{2}})h^2 + \mathcal{O}(h^3)\right), \end{aligned}$$

which lead to the following constraints on the weights:

$$w_{0,i+\frac{1}{2}} + w_{1,i+\frac{1}{2}} = 1, \tag{6.2a}$$

$$C_1 := \frac{w_{0,i+\frac{1}{2}}}{8} - \frac{3w_{1,i+\frac{1}{2}}}{8} = \mathcal{O}(h). \tag{6.2b}$$

Reconstruction from the right: We now consider the reconstruction from the right at the interface $x_{i+\frac{1}{2}}$, which requires the stencils

$$\widetilde{S}_0 = \{x_{i+1}, x_{i+2}\}, \quad \widetilde{S}_1 = \{x_i, x_{i+1}\}$$

to obtain linear polynomial approximations in cell I_{i+1}. The corresponding polynomial and their evaluations at $x_{i+\frac{1}{2}}$ are

$$\begin{aligned} \widetilde{p}^{(0)}(x) &= v_{i+1}\frac{(x-x_{i+2})}{(x_{i+1}-x_{i+2})} + v_{i+2}\frac{(x-x_{i+1})}{(x_{i+2}-x_{i+1})} \quad &\Longrightarrow \quad v^{(0),+}_{i+\frac{1}{2}} &= \frac{3v_{i+1}}{2} - \frac{v_{i+2}}{2}, \\ \widetilde{p}^{(1)}(x) &= v_i\frac{(x-x_{i+1})}{(x_i-x_{i+1})} + v_{i+1}\frac{(x-x_i)}{(x_{i+1}-x_i)} \quad &\Longrightarrow \quad v^{(1),+}_{i+\frac{1}{2}} &= \frac{v_i}{2} + \frac{v_{i+1}}{2}. \end{aligned}$$

Let the weights in this case be denoted by $\widetilde{w}_{0,i+\frac{1}{2}}$ and $\widetilde{w}_{1,i+\frac{1}{2}}$. As before, we set

$$v^+_{i+\frac{1}{2}} := \widetilde{w}_{0,i+\frac{1}{2}}\left(-\frac{v_{i+2}}{2}+\frac{3v_{i+1}}{2}\right) + \widetilde{w}_{1,i+\frac{1}{2}}\left(\frac{v_i}{2}+\frac{v_{i+1}}{2}\right), \tag{6.3}$$

and we require

$$\begin{aligned} v(x_{i+\frac{1}{2}}) + \mathcal{O}(h^3) &= \widetilde{w}_{0,i+\frac{1}{2}}\left(v(x_{i+\frac{1}{2}}) - \frac{3}{8}v''(x_{i+\frac{1}{2}})h^2 + \mathcal{O}(h^3)\right) \\ &\quad + \widetilde{w}_{1,i+\frac{1}{2}}\left(v(x_{i+\frac{1}{2}}) + \frac{1}{8}v''(x_{i+\frac{1}{2}})h^2 + \mathcal{O}(h^3)\right). \end{aligned}$$

This enforces the following constraints on the weights:

$$\widetilde{w}_{0,i+\frac{1}{2}} + \widetilde{w}_{1,i+\frac{1}{2}} = 1, \tag{6.4a}$$

$$C_2 := -\frac{3\widetilde{w}_{0,i+\frac{1}{2}}}{8} + \frac{\widetilde{w}_{1,i+\frac{1}{2}}}{8} = \mathcal{O}(h). \tag{6.4b}$$

The weights $w_{1,i+\frac{1}{2}}, w_{1,i+\frac{1}{2}}, \widetilde{w}_{1,i+\frac{1}{2}}, \widetilde{w}_{1,i+\frac{1}{2}}$ must be chosen in accordance to (6.2) and (6.4), to ensure that the desired consistency and accuracy of the reconstruction is achieved. For convenience of notation, we drop the $i + \frac{1}{2}$ subscript in the weights wherever it is clear that we are referring to the interface $x_{i+\frac{1}{2}}$.

6.2 Properties

We list the crucial properties that the reconstruction procedure needs to satisfy, including the sign-property.

6.2.1 Consistency

Using (6.2b) and (6.4b), we rewrite the weights as

$$w_0 = \frac{3}{4} + 2C_1, \quad w_1 = \frac{1}{4} - 2C_1, \quad \widetilde{w}_0 = \frac{1}{4} - 2C_2, \quad \widetilde{w}_1 = \frac{3}{4} + 2C_2.$$

To ensure that the weights are non-negative and that (6.2a) and (6.4a) are satisfied, we require the following consistency condition:

$$0 \leqslant w_0, w_1, \widetilde{w}_0, \widetilde{w}_1 \leqslant 1, \tag{P1}$$

or equivalently,

$$-\frac{3}{8} \leqslant C_1, C_2 \leqslant \frac{1}{8}. \tag{P1'}$$

6.2.2 Sign property

The jump in the reconstructed variables can be written as

$$[\![v]\!]_{i+\frac{1}{2}} = \frac{1}{2}\left[\widetilde{w}_0(1-\theta^-_{i+1}) + w_1(1-\theta^+_i)\right]\Delta v_{i+\frac{1}{2}}, \tag{6.5}$$

where θ^-_{i+1} and θ^+_i are the jump ratios defined by (4.15). Thus, the following is an *equivalent* formulation of the sign property whenever $\Delta v_{i+\frac{1}{2}} \neq 0$:

$$\left[\widetilde{w}_0(1-\theta^-_{i+1}) + w_1(1-\theta^+_i)\right] \geqslant 0. \tag{P2}$$

6.2.3 Negation symmetry

By *negation symmetry*, we mean that the weights are not biased towards positive or negative solution values. In other words, the weights should remain unchanged under the transformation $v \mapsto -v$. The jumps accordingly transform as

$$\Delta v_{i+\frac{1}{2}} \mapsto -\Delta v_{i+\frac{1}{2}} \qquad \forall\ i \in \mathbb{Z}.$$

However, the jump ratios θ_i^- or θ_i^+ remain unchanged. A sufficient condition to enforce negation symmetry is to choose C_1, C_2 as functions of $\theta_i^+, \theta_{i+1}^-$:

$$C_1 = C_1(\theta_i^+, \theta_{i+1}^-), \qquad C_2 = C_2(\theta_i^+, \theta_{i+1}^-). \tag{P3}$$

6.2.4 Mirror property

If we mirror the solution about the interface $x_{i+\frac{1}{2}}$, we would like to ensure that the weights also get mirrored about $x_{i+\frac{1}{2}}$. The mirroring transforms the jump ratios as

$$\theta_{i+1}^- \longleftrightarrow \theta_i^+.$$

It is straightforward to see that the weights must transform as

$$w_0 \longleftrightarrow \widetilde{w}_1, \qquad w_1 \longleftrightarrow \widetilde{w}_0.$$

Assuming that the the form (P3) ensuring negation symmetry holds, the mirror property is true *if and only if*

$$C_1(a,b) = C_2(b,a) \qquad \forall\ a,b \in \mathbb{R}. \tag{P4}$$

Remark 6.2.1. *There are other invariants corresponding to the transformation $v \mapsto -v$, such as $|\Delta v_{i+\frac{1}{2}}|$ or $(|v_i| + |v_{i+1}|)$. Thus, one can also choose*

$$C_1 = C_1(\theta_i^+, \theta_{i+1}^-, |\Delta v_{i+\frac{1}{2}}|, (|v_i| + |v_{i+1}|)),\ C_2 = C_2(\theta_i^+, \theta_{i+1}^-, |\Delta v_{i+\frac{1}{2}}|, (|v_i| + |v_{i+1}|)), \tag{P3$'$}$$

to ensure negation symmetry holds. These terms are also invariant if the solution is mirrored about the interface $x_{i+\frac{1}{2}}$. Assuming that the form described by (P3$'$) *holds, a generalized sufficient condition for mirror symmetry is given by*

$$C_1(a,b,c,d) = C_2(b,a,c,d) \qquad \forall\ a,b,c,d \in \mathbb{R}, \quad c,d \geqslant 0. \tag{P4$'$}$$

6.2.5 Inner jump condition

In addition to the sign property, we would like the reconstructed variables to satisfy the *inner jump condition* in each cell i:

$$\text{sign}(v_{i+\frac{1}{2}}^- - v_{i-\frac{1}{2}}^+) = \text{sign}(\Delta v_{i+\frac{1}{2}}) = \text{sign}(\Delta v_{i-\frac{1}{2}}),$$

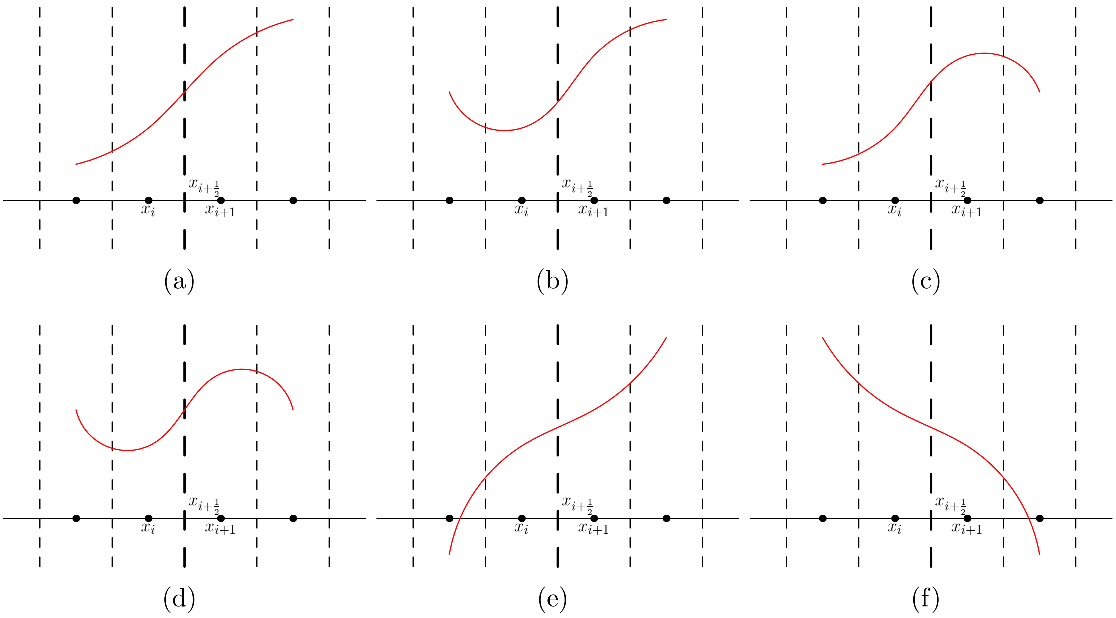
whenever the second equality holds. This property ensures that the monotonicity of the solution is preserved. The second-order ENO reconstruction satisfies this property, while it need not hold true for higher order ENO. Recalling the definitions (6.1), (6.3) of $v^-_{i+\frac{1}{2}}$ and $v^+_{i-\frac{1}{2}}$, we obtain

$$\begin{aligned} v^-_{i+\frac{1}{2}} - v^+_{i-\frac{1}{2}} &= w_{0,i+\frac{1}{2}}\left(\frac{v_i}{2}+\frac{v_{i+1}}{2}\right) + w_{1,i+\frac{1}{2}}\left(-\frac{v_{i-1}}{2}+\frac{3v_i}{2}\right) \\ &\quad - \widetilde{w}_{0,i-\frac{1}{2}}\left(-\frac{v_{i+1}}{2}+\frac{3v_i}{2}\right) - \widetilde{w}_{1,i-\frac{1}{2}}\left(\frac{v_i}{2}+\frac{v_{i-1}}{2}\right) \\ &= \Delta v_{i+\frac{1}{2}}\frac{w_{0,i+\frac{1}{2}}+\widetilde{w}_{0,i-\frac{1}{2}}}{2} + \Delta v_{i-\frac{1}{2}}\frac{w_{1,i+\frac{1}{2}}+\widetilde{w}_{1,i-\frac{1}{2}}}{2}. \end{aligned}$$

By the assumption (P1) of non-negativity of the weights, the coefficients of $\Delta v_{i-\frac{1}{2}}$ and $\Delta v_{i+\frac{1}{2}}$ in the above expression are non-negative, and hence the inner jump condition is *automatically satisfied.*

6.2.6 Accuracy

In general, conditions (6.2b) and (6.4b) require C_1 and C_2 to be $\mathcal{O}(h)$ for smooth solutions. However, this condition can be relaxed in scenarios in which $v''(\hat{x}) = 0$, for some $\hat{x}$ such that $|\hat{x} - x_{i+\frac{1}{2}}| = \mathcal{O}(h)$. Figure 6.2 depicts a few situations in which this can happen. Assuming sufficient regularity on the solution, the following is true in these special scenarios:



(a) (b) (c)

(d) (e) (f)

Figure 6.2: Special cases when $v''(\hat{x}) = 0$.

$$v''(x_{i+\frac{1}{2}}) = v''(\hat{x}) + (x_{i+\frac{1}{2}} - \hat{x})v'''(\hat{x}) + \mathcal{O}(h^2) = \mathcal{O}(h).$$

This in turn implies that each of the linear polynomials used for reconstruction, gives a third-order accurate approximation of the solution at $x_{i+\frac{1}{2}}$. Thus, (6.2b) and (6.4b) become redundant. In other words, the reconstruction is third-order accurate provided

$$C_1, C_2 = \begin{cases} \mathcal{O}(h), & \text{in GC} \\ \text{no order restriction,} & \text{in SC} \end{cases}, \tag{P5}$$

where we use the abbreviations GC and SC to denote general cases and special cases respectively.

6.2.7 The feasible region

In order to choose weights satisfying (P1)–(P5), we first analyse how the weights behave under the above constraints. We will look at six different scenarios, depending on the values of $\theta_i^+, \theta_{i+1}^-$. In each scenario we will try to determine the *feasible region*, which corresponds to the region where the weights satisfy (P1) and (P2). The remaining properties will be considered in Section 6.3, while trying to construct explicit weights. We define the quantities

$$\psi^+_{i+\frac{1}{2}} := \frac{(1-\theta^-_{i+1})}{(1-\theta^+_i)}, \qquad \psi^-_{i+\frac{1}{2}} := \frac{1}{\psi^+_{i+\frac{1}{2}}}.$$

The $i+\frac{1}{2}$ subscript will be dropped whenever it is obvious that we are referring to the interface $i+\frac{1}{2}$. We also introduce the notation

$$\mathcal{L} := \begin{cases} \frac{C_1}{\frac{1}{8}(1+\psi^+)} + \frac{C_2}{\frac{1}{8}(1+\psi^-)}, & \text{if } \psi^+ \neq -1 \\ C_1 - C_2 + 1, & \text{if } \psi^+ = \psi^- = -1. \end{cases}$$

Furthermore, we denote the open box $\left(-\frac{3}{8}, \frac{1}{8}\right) \times \left(-\frac{3}{8}, \frac{1}{8}\right)$ by $\mathcal{B}$. Recall that the consistency constraint (P1') requires that $(C_1, C_2) \in \overline{\mathcal{B}}$.

Case 1: $\theta_i^+, \theta_{i+1}^- > 1$

The qualitative nature of the (smooth) solution for this case is indicated in Figure 6.3. The solution is clearly not strictly convex or concave in the stencil under consideration, even if the solution is more oscillatory than that shown in Figure 6.3. Thus, we are in the SC regime, which implies that no order of accuracy restrictions must be imposed on C_1, C_2. To ensure that (P2) holds, we need

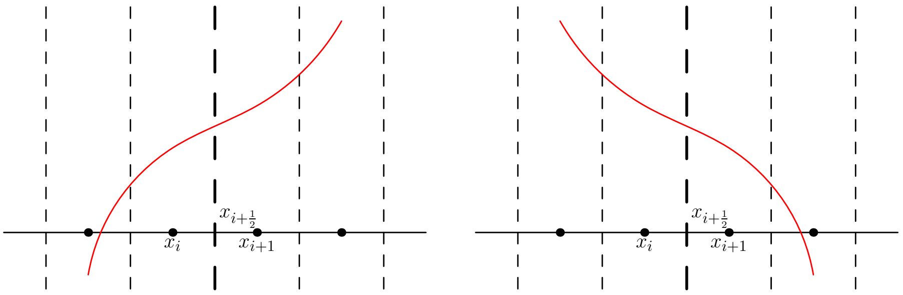
$$\widetilde{w}_0 = w_1 = 0 \qquad \iff \qquad C_1 = C_2 = \frac{1}{8}.$$

Note that this leads to precisely the ENO-2 stencil selection, which is suitable for discontinuous solutions as well.

Case 2: $\theta_i^+ < 1,\ \theta_{i+1}^- > 1$

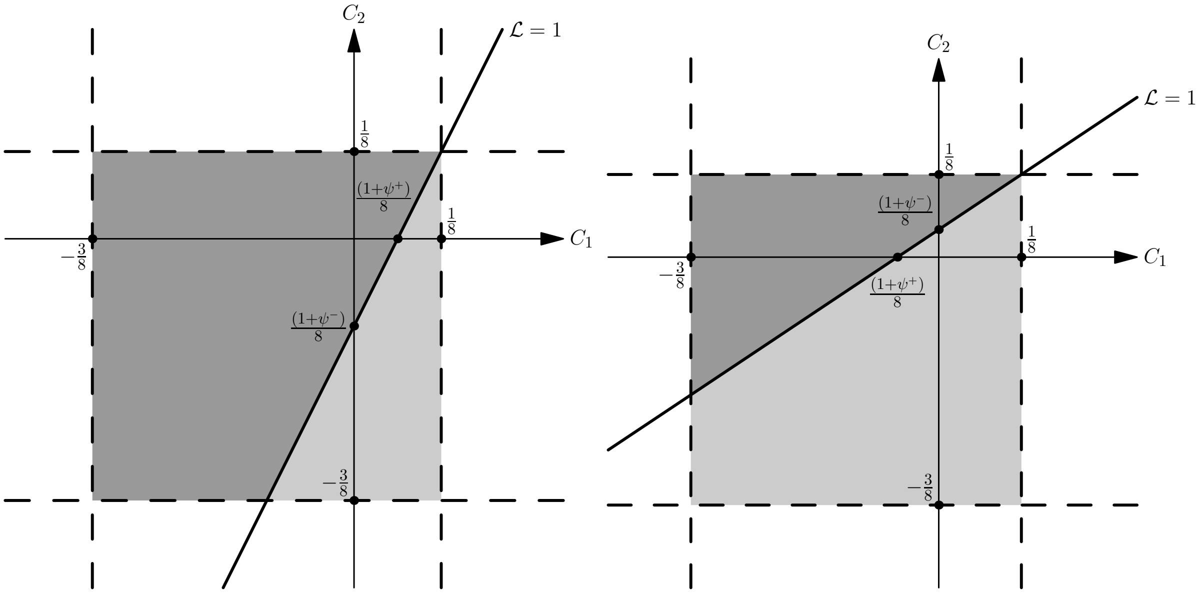
In this case we have

$$\psi^+ < 0, \qquad 1 + \psi^+ < 1.$$



(a) (b)

Figure 6.3: Possible scenarios for Case 1.



(a) $\psi^+ = -0.5 \in (-1, 0)$ (b) $\psi^+ = -1.5 \in (-\infty, -1)$

Figure 6.4: Feasible region for Case 2 (dark grey) and Case 3 (light grey).

Case 2 falls into the GC regime. Thus, we must choose C_1 and C_2 carefully so as not to violate the accuracy condition. The sign property (P2) will hold if

$$w_1 \geqslant -\widetilde{w}_0\psi^+ \iff \left(\frac{1}{4} - 2C_1\right) \geqslant -\left(\frac{1}{4} - 2C_2\right)\psi^+ \iff C_1 + \psi^+ C_2 \leqslant \frac{1}{8}(1+\psi^+).$$

Thus, we have the following constraints on C_1, C_2:

$$\begin{aligned} -\frac{3}{8} < C_1, C_2 < \frac{1}{8}, & \\ \mathcal{L} \leqslant 1 \qquad & \text{if } -1 \leqslant \psi^+ < 0, \\ \mathcal{L} \geqslant 1 \qquad & \text{if } \psi^+ < -1. \end{aligned}$$

The feasible region for C_1, C_2 is shown in Figure 6.4 (in dark grey).

Case 3: $\theta_i^+ > 1,\ \theta_{i+1}^- < 1$

Similiar to case 2, we have

$$\psi^+ < 0, \qquad 1 + \psi^+ < 1,$$

with the solutions falling into the GC regime in general. The sign property holds if

$$\widetilde{w}_0 \geqslant -w_1\psi^- \iff \left(\frac{1}{4} - 2C_2\right) \geqslant -\left(\frac{1}{4} - 2C_1\right)\psi^- \iff C_1 + \psi^+ C_2 \geqslant \frac{1}{8}(1+\psi^+).$$

Comparing the last equivalent condition above with that observed for case 2, we see that the inequality has been flipped. Thus, we have the following constraints on C_1, C_2:

$$\begin{aligned} -\frac{3}{8} < C_1, C_2 < \frac{1}{8}&, \\ \mathcal{L} \geqslant 1 &\qquad \text{if } -1 \leqslant \psi^+ < 0, \\ \mathcal{L} \leqslant 1 &\qquad \text{if } \psi^+ < -1. \end{aligned}$$

The feasible region for C_1, C_2 is shown in Figure 6.4 (in light grey).

Remark 6.2.2. *Any point in $\{(C_1, C_2) : \mathcal{L} = 1\} \cap \mathcal{B}$, satisfies the constraints in cases 2 and 3. This fact will be exploited in constructing explicit weights.*

Case 4: $\theta_{i+1}^- = 1$

In this case the solution either has a linear region, or is oscillatory without being strictly convex or concave. Thus, this case falls in the SC regime. In order to satisfy (P2), we require w_1 to have the same sign as $(1 - \theta_i^+)$.

If $\theta_i^+ \leqslant 1$, then (C_1, C_2) can be chosen as any point in $\overline{\mathcal{B}}$. However, if $\theta_i^+ > 1$, then to satisfy (P1) and (P2), we must take $C_1 = \frac{1}{8}$, while C_2 can be any value in $\left[-\frac{3}{8}, \frac{1}{8}\right]$. This would lead to $w_0 = 1, w_1 = 0$ when $\theta_i^+ > 1$, which is identical to the ENO-2 stencil selection.

Case 5: $\theta_i^+ = 1$

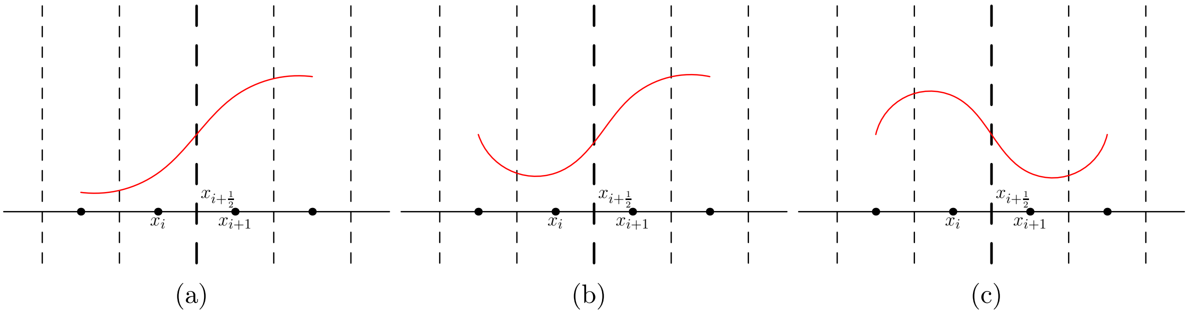
This case is similar to case 4, with the values of θ_i^+ and θ_{i+1}^- interchanged. If $\theta_{i+1}^- \leqslant 1$, then (C_1, C_2) can be chosen as any point in $\overline{\mathcal{B}}$. If $\theta_{i+1}^- > 1$, then we must take $C_2 = \frac{1}{8}$ while C_1 can be any value in $\left[-\frac{3}{8}, \frac{1}{8}\right]$.

Case 6: $\theta_i^+, \theta_{i+1}^- < 1$

In this final case, we have

$$\psi^+ > 0, \qquad 1 + \psi^+ > 1.$$

Note that this was true in case 1 as well. By an argument similar to the one made in case 1, we can show that case 6 falls into the SC regime (see Figure 6.5). Furthermore, the sign property is satisfied as long as the consistency condition (P1) holds true.



(a) (b) (c)

Figure 6.5: Possible scenarios for Case 6.

6.3 Explicit weights: SP-WENO

We now make an explicit choice for the weights, based on the case-by-case analysis done in Section 6.2.7. Recall that from an accuracy point of view, the optimal choice of weights is $(w_0, w_1) = (3/4, 1/4)$ and $(\widetilde{w}_0, \widetilde{w}_1) = (1/4, 3/4)$, or equivalently, $(C_0, C_1) = (0, 0)$. However, the point $(C_0, C_1) = (0, 0)$ in many cases does not lie in the feasible region (see Section 6.2.7).

For cases 2 and 3, we choose (C_0, C_1) to be the point in $\{\mathcal{L} = 1\} \cap \mathcal{B}$, by virtue of Remark 6.2.2. Furthermore, C_0 and C_1 must both be of order $\mathcal{O}(h)$ in these two cases, in order to satisfy (P5). Thus, we choose (C_0, C_1) to be the point on $\mathcal{L} = 1$ closest to the origin, as measured in the Euclidean norm:

$$C_1(\theta_i^+, \theta_{i+1}^-) = \begin{cases} \frac{1}{8}\left(\frac{f^+}{(f^+)^2+(f^-)^2}\right) & \text{if } \psi^+ \neq -1 \\ 0 & \text{otherwise,} \end{cases} \qquad C_2(\theta_i^+, \theta_{i+1}^-) = C_1(\theta_{i+1}^-, \theta_i^+), \tag{6.6}$$

where we have defined

$$f^+(\theta_i^+, \theta_{i+1}^-) := \begin{cases} \frac{1}{1+\psi^+} & \text{if } \theta_i^+ \neq 1, \psi^+ \neq -1 \\ 1 & \text{otherwise,} \end{cases} \qquad f^-(\theta_i^+, \theta_{i+1}^-) := f^+(\theta_{i+1}^-, \theta_i^+).$$

For smooth functions v we can show that $(1 + \psi^+), (1 + \psi^-) = \mathcal{O}(h)$ in the GC regime, so $C_1, C_2 = \mathcal{O}(h)$ for cases 2 and 3, and hence (P5) is satisfied.

For cases 1 and 6, the point on the line $\mathcal{L} = 1$ closest to the origin need not lie in the feasible region. Furthermore, ψ^+ and ψ^- need not be defined for cases 4 and 5, thus the line $\mathcal{L} = 1$ is not defined. Note that for these remainder cases, there is no order restriction on (C_0, C_1). Going through a case-by-case analysis, we propose the following extension of (6.6):

$$C_1(\theta_i^+, \theta_{i+1}^-) = \begin{cases} \frac{1}{8}\left(\frac{f^+}{(f^+)^2+(f^-)^2}\right) & \text{if } \theta_i^+ \neq 1,\ \psi^+ < 0,\ \psi^+ \neq -1 \\ 0 & \text{if } \theta_i^+ \neq 1, \psi^+ = -1 \\ -\frac{3}{8} & \text{if } \theta_i^+ = 1 \text{ or } \psi^+ \geqslant 0,\ |\theta_i^+| \leqslant 1 \\ \frac{1}{8} & \text{if } \psi^+ \geqslant 0,\ |\theta_i^+| > 1 \end{cases}, \tag{6.7}$$

and $C_2(\theta_i^+, \theta_{i+1}^-) = C_1(\theta_{i+1}^-, \theta_i^+)$, as before.

By virtue of lying in the feasible region, the above choices of C_1, C_2 automatically satisfy the consistency and sign properties (P1), (P2). By definition they satisfy the negation and mirror symmetry properties (P3) and (P4), and through a case-by-case analysis, it can be seen that the weights also satisfy the accuracy condition (P5). We refer to this third-order WENO-type reconstruction method as *SP-WENO*.

Remark 6.3.1. *In the above discussion, we have assumed that $\Delta v_{i+\frac{1}{2}} \neq 0$. If $\Delta v_{i+\frac{1}{2}} = 0$, then the weights are chosen to be $w_1 = \widetilde{w}_0 = 0$, leading to $[\![v]\!]_{i+\frac{1}{2}} = 0$.*

Remark 6.3.2. *It can be shown that choosing C_1, C_2 to be any point on the line $\mathcal{L} = 1$, ensures that the reconstructed states are equal, i.e. $v^-_{i+\frac{1}{2}} = v^+_{i+\frac{1}{2}}$. Thus, $[\![v]\!]_{i+\frac{1}{2}} = 0$ for cases 2 and 3.*

6.3.1 Stability estimates

We now show that it is possible to estimate the reconstructed jumps in terms of the original jumps. In case 1, we have $\theta_i^+, \theta_{i+1}^- > 1$ and $w_1 = \widetilde{w}_0 = 0$. Thus, the reconstructed states

$$v^-_{i+\frac{1}{2}} = v^+_{i+\frac{1}{2}} = \frac{1}{2}(v_i + v_{i+1}),$$

have a zero jump. This is also true for cases 2–3, by virtue of Remark 6.3.2. Proceeding in a similar manner for cases 4–6, we find that the jump in reconstructed states is

$$
[\![v]\!]_{i+\frac{1}{2}} = \begin{cases}
0 & \text{if} \quad \left.\begin{array}{ll}
\theta_i^+ > 1 \text{ and } \theta_{i+1}^- > 1 & \text{(case 1)}\\
\theta_i^+ < 1 \text{ and } \theta_{i+1}^- > 1 & \text{(case 2)}\\
\theta_i^+ > 1 \text{ and } \theta_{i+1}^- < 1 & \text{(case 3)}\\
|\theta_i^+| > 1 \text{ and } \theta_{i+1}^- = 1 & \text{(case 4)}\\
\theta_i^+ = 1 \text{ and } |\theta_{i+1}^-| > 1 & \text{(case 5)}\\
\theta_i^+ < -1 \text{ and } \theta_{i+1}^- < -1 & \text{(case 6)}
\end{array}\right\}\Omega_0 \\[2ex]
\frac{1}{2}(\Delta v_{i+\frac{1}{2}} - \Delta v_{i-\frac{1}{2}}) & \text{if} \quad \left.\begin{array}{ll}
|\theta_i^+| \leqslant 1 \text{ and } \theta_{i+1}^- = 1 & \text{(case 4)}\\
-1 \leqslant \theta_i^+ < 1 \text{ and } \theta_{i+1}^- < -1 & \text{(case 6)}
\end{array}\right\}\Omega_1 \\[2ex]
\frac{1}{2}(\Delta v_{i+\frac{1}{2}} - \Delta v_{i+\frac{3}{2}}) & \text{if} \quad \left.\begin{array}{ll}
\theta_i^+ = 1 \text{ and } |\theta_{i+1}^-| \leqslant 1 & \text{(case 5)}\\
\theta_i^+ < -1 \text{ and } -1 \leqslant \theta_{i+1}^- < 1 & \text{(case 6)}
\end{array}\right\}\Omega_2 \\[2ex]
\Delta v_{i+\frac{1}{2}} - \frac{1}{2}(\Delta v_{i-\frac{1}{2}} + \Delta v_{i+\frac{3}{2}}) & \text{if} \quad \left.\begin{array}{ll}
-1 \leqslant \theta_i^+, \theta_{i+1}^- < 1 & \text{(case 6)}
\end{array}\right\}\Omega_3
\end{cases} .
$$

Lemma 6.3.1 (Bounds on jumps)**.** *We have the following estimate on the jump in the SP-WENO reconstruction:*

$$\left|[\![v]\!]_{i+\frac{1}{2}}\right| \leqslant 2\left|\Delta v_{i+\frac{1}{2}}\right| \quad \forall\, i \in \mathbb{Z}. \tag{6.8}$$

Proof. If $\Delta v_{i+\frac{1}{2}} = 0$, then the estimate clearly holds as $[\![v]\!]_{i+\frac{1}{2}} = 0$ by construction of SP-WENO. Thus, we assume $\Delta v_{i+\frac{1}{2}} \neq 0$. Furthermore, the estimate holds trivially for $(\theta_i^+, \theta_{i+1}^-) \in \Omega_0$.

If $(\theta_i^+, \theta_{i+1}^-) \in \Omega_1$, then

$$|\theta_i^+| \leqslant 1 \qquad \Longleftrightarrow \qquad -1 \leqslant \frac{\Delta v_{i-\frac{1}{2}}}{\Delta v_{i+\frac{1}{2}}} \leqslant 1.$$

Thus,

$$\frac{[\![v]\!]_{i+\frac{1}{2}}}{\Delta v_{i+\frac{1}{2}}} = \frac{1}{2} - \frac{\Delta v_{i-\frac{1}{2}}}{2\Delta v_{i+\frac{1}{2}}} \leqslant 1 \qquad \Longrightarrow \qquad \left|[\![v]\!]_{i+\frac{1}{2}}\right| \leqslant \left|\Delta v_{i+\frac{1}{2}}\right| < 2\left|\Delta v_{i+\frac{1}{2}}\right|.$$

Similarly, if $(\theta_i^+, \theta_{i+1}^-) \in \Omega_2$, then

$$|\theta_{i+1}^-| \leqslant 1 \qquad \Longleftrightarrow \qquad -1 \leqslant \frac{\Delta v_{i+\frac{3}{2}}}{\Delta v_{i+\frac{1}{2}}} \leqslant 1.$$

Thus,

$$\frac{[\![v]\!]_{i+\frac{1}{2}}}{\Delta v_{i+\frac{1}{2}}} = \frac{1}{2} - \frac{\Delta v_{i+\frac{3}{2}}}{2\Delta v_{i+\frac{1}{2}}} \leqslant 1 \qquad \Longrightarrow \qquad \left|[\![v]\!]_{i+\frac{1}{2}}\right| \leqslant \left|\Delta v_{i+\frac{1}{2}}\right| < 2\left|\Delta v_{i+\frac{1}{2}}\right|.$$

Finally, if $(\theta_i^+, \theta_{i+1}^-) \in \Omega_3$, then

$$|\theta_i^+|, \ |\theta_{i+1}^-| \leqslant 1.$$

Repeating the above arguments, we once again get

$$\left|[\![v]\!]_{i+\frac{1}{2}}\right| \leqslant 2\left|\Delta v_{i+\frac{1}{2}}\right|.$$

□

Remark 6.3.3. *The bounding constant* 2 *on the right-hand side of* (6.8) *is identical to the one obtained with ENO-2 [35], but smaller than that obtained with ENO-3. Thus, SP-WENO leads to tighter stability bounds for higher order accuracy, as compared to its ENO counterparts.*

Remark 6.3.4. *While attempting to construct SP-WENO, we were able to find several other WENO-3 weights satisfying the above mentioned properties. The weights described by* (6.7) *gave the best numerical results for scalar conservation laws, among all the possible options considered, especially near discontinuous solutions. We were also able to construct weights which ensured the reconstruction to be TVD. However, the reconstruction suffered from loss of accuracy near smooth extrema [84]. Furthermore, the search for TVD property in the set-up of TeCNO schemes would be futile, as the high-order entropy conservative fluxes used do not lead to a TVD scheme.*

6.4 Reconstruction accuracy of SP-WENO

We now demonstrate that SP-WENO does indeed give the desired order of accuracy in approximating the solution at cell-interfaces, assuming the solution is smooth enough. We consider the smooth function

$$u(x) = \sin(10\pi x) + x, \quad x \in [0,1].$$

The domain is discretized using an N-cell uniform mesh with mesh size $h = (b-a)/N$. SP-WENO is used to reconstruct the values at the cell interfaces, based on the point values at the cell centers. We also compare the results with ENO-2, ENO-3 and the existing robust version of WENO-3 proposed in [62] (see also [106]). The error in the interface values are evaluated as

$$\|u^-_{i+\frac{1}{2}} - u(x_{i+\frac{1}{2}})\|_{L^p_h} + \|u^+_{i+\frac{1}{2}} - u(x_{i+\frac{1}{2}})\|_{L^p_h}, \quad p \in [1,\infty].$$

The errors and the corresponding convergence rates with the various reconstruction methods, are shown in Table 6.1.

Let us first analyse the order of convergence for the various methods. If we consider the L^∞_h norm, ENO-2, ENO-3 and SP-WENO give the expected order of convergence, while WENO-3 gives almost fourth-order convergence. This can possibly be explained by the fact that unlike the the first three methods, the weights used in WENO-3 are smooth. For the errors evaluated in the L^1_h norm, the ENO schemes give the expected order of convergence, while both SP-WENO and WENO-3 give more than third-order convergence. In fact, SP-WENO order of convergence seems to be far superior as compared to WENO-3.

Recall the Remark 5.8.1, where we mentioned that the order of convergence is not everything. If we compare the third-order methods, then they can be arranged as ENO-3 $<$ SP-WENO $<$ WENO-3 in the order of increasing L^∞_h errors for any fixed mesh, amongst the meshes considered in Table 6.1. Although WENO-3 has the largest order of convergence and will eventually give the smallest errors if the mesh is refined further, it is not always *practical* to work with such a fine mesh. Similarly, WENO-3 has the largest error measured in the L^1_h norm on all the meshes considered. With SP-WENO and ENO-3, we can clearly see the crossing point: ENO-3 has smaller L^1_h errors on the first two mesh levels, but SP-WENO quickly overtakes ENO-3 and gives the smallest errors on the remaining mesh levels. We are more interested in the L^1_h errors since the convergence theory for conservation laws are generally posed in the L^1 setting.

6.5 Numerical results with SP-WENO for scalar conservation laws

For scalar conservation laws and a given entropy function $\eta(u)$, two-point entropy conservative fluxes are uniquely determined (see Section 5.1). We choose the quadratic entropy function $\eta(U) = U^2/2$, which leads to the entropy variable being equal to the conserved variable, i.e., $V := \partial_U \eta(U) = U$.

We use a *TeCNO4 entropy stable flux*, which is of the form

$$F_{i+\frac{1}{2}} = F^{*,4}_{i+\frac{1}{2}} - \frac{1}{2} D_{i+\frac{1}{2}} [\![V]\!]_{i+\frac{1}{2}},$$

	SP-WENO				ENO-3			
N	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
40	8.59e-02	-	2.24e-01	-	3.95e-02	-	5.60e-02	-
80	6.73e-03	3.67	2.97e-02	2.92	4.90e-03	3.01	7.43e-03	2.92
160	5.01e-04	3.75	3.77e-03	2.98	6.08e-04	3.01	9.42e-04	2.98
320	3.64e-05	3.78	4.73e-04	2.99	7.57e-05	3.01	1.18e-04	2.99
640	2.59e-06	3.81	5.91e-05	3.00	9.47e-06	3.00	1.48e-05	3.00
1280	1.82e-07	3.83	7.39e-06	3.00	1.18e-06	3.01	1.85e-06	3.00
2560	1.26e-08	3.85	9.24e-07	3.00	1.47e-07	3.00	2.31e-07	3.00

	WENO-3				ENO-2			
N	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
40	2.04e-01	-	4.34e-01	-	2.35e-01	-	4.34e-01	-
80	4.03e-02	2.34	1.14e-01	1.93	5.39e-02	2.12	1.14e-01	1.93
160	7.25e-03	2.48	2.88e-02	1.98	1.29e-02	2.07	2.88e-02	1.98
320	1.18e-03	2.62	7.10e-03	2.02	3.14e-03	2.03	7.22e-03	2.00
640	1.77e-04	2.74	1.65e-03	2.10	7.76e-04	2.02	1.81e-03	2.00
1280	2.13e-05	3.05	1.64e-04	3.34	1.93e-04	2.01	4.52e-04	2.00
2560	2.10e-06	3.34	9.08e-06	4.17	4.81e-05	2.00	1.13e-04	2.00

Table 6.1: Inclined sine wave: reconstruction errors.

where $F^{*,4}_{i+\frac{1}{2}}$ is a fourth-order entropy conservative flux given by (5.15), and $D_{i+\frac{1}{2}}$ is a scalar dissipation operator approximating $|F'(u)|$. Note that we can directly reconstruct the entropy variables instead of scaled entropy variable, since the dissipation operator is a scalar. The reconstruction is performed using SP-WENO, ENO-2 or ENO-3, all of which have the sign property. Time integration is performed using SSP-RK3.

6.5.1 Linear advection

Consider the linear advection equation

$$\partial_t U + c\partial_x U = 0,$$

for which the base second-order entropy flux required for TeCNO4 is given by (5.4). The dissipation operator is chosen as $D_{i+\frac{1}{2}} = |c|$. For the following test cases, we take the convective velocity $c = 1$.

Advecting sine-wave

The domain is taken to be $[-\pi, \pi]$, with final time $t_f = 0.5$ and CFL $= 0.4$. The initial profile is given by

$$U_0(x) = \sin(x),$$

with periodic boundary conditions. Table 6.2 shows L_h^1 errors with various reconstructions. SP-WENO gives more than third-order accuracy, while ENO-2 and ENO-3 give expected convergence rates. Furthermore, the magnitude of the errors with SP-WENO and ENO-3 are comparable on each mesh.

N	SP-WENO		ENO-3		ENO-2	
	error	rate	error	rate	error	rate
50	6.22e-04	-	2.58e-04	-	1.61e-02	-
100	6.90e-05	3.17	3.23e-05	3.00	4.36e-03	1.88
200	7.66e-06	3.17	4.04e-06	3.00	1.16e-03	1.91
400	8.29e-07	3.21	5.05e-07	3.00	3.08e-04	1.91
600	2.26e-07	3.20	1.50e-07	3.00	1.41e-04	1.92
800	8.72e-08	3.31	6.31e-08	3.00	8.09e-05	1.93

Table 6.2: L_h^1 errors for linear advection: sine-wave advection test with TeCNO4 flux.

Advecting fourth-power sine-wave

We consider the advection of the following smooth function

$$U_0(x) = \sin^4(x),$$

on a domain $[-\pi, \pi]$ with final time $t_f = 0.5$ and CFL $= 0.5$. For the given test case, the MUSCL scheme using ENO schemes are known to perform poorly. This was first noted by Rogerson and Meiburg [98], who attributed this behaviour to the selection of linearly unstable stencils by the ENO algorithm. A similar behaviour is also observed when the ENO method is used in the TeCNO4 framework, as shown in Table 6.3. Though ENO-2 gives the expected second-order of accuracy, there is a clear deterioration in the convergence rate for ENO-3 with mesh refinement. SP-WENO, on the other hand, does not suffer from such problems and continues to give more than third order accuracy.

N	SP-WENO		ENO-3		ENO-2	
	error	rate	error	rate	error	rate
100	1.32e-03	-	1.48e-03	-	2.13e-02	-
200	1.48e-04	3.16	1.97e-04	2.91	6.12e-03	1.80
400	1.64e-05	3.17	2.57e-05	2.94	1.66e-03	1.89
600	4.61e-06	3.14	8.35e-06	2.77	7.63e-04	1.91
800	1.79e-06	3.29	4.86e-06	1.88	4.41e-04	1.90
1000	8.55e-07	3.31	3.62e-06	1.32	2.87e-04	1.92

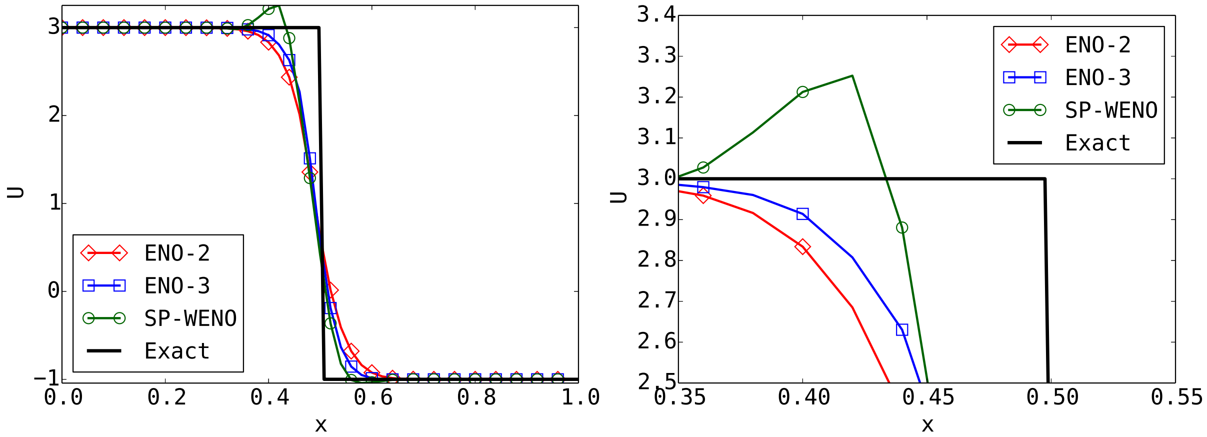
Table 6.3: L_h^1 errors for linear advection: fourth-power sine-wave advection test with TeCNO4 flux.

Advecting discontinuity

In this test case, we demonstrate the performance of SP-WENO in approximating discontinuous solutions. The domain is $[-1, 1]$, with final time $t_f = 0.5$ and CFL $= 0.4$. The initial discontinuous profile is given by

$$\begin{cases} 3 & \text{if } x < 0 \\ -1 & \text{if } x > 0 \end{cases}.$$

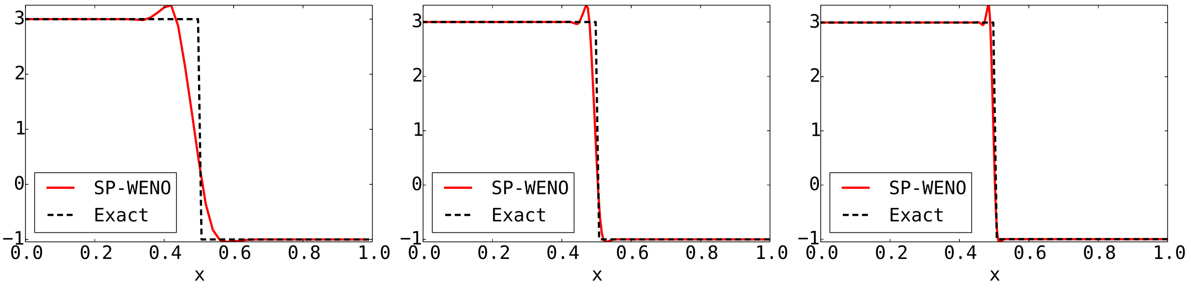
The mesh consists of 100 cells with transmissive boundary conditions. The results with the TeCNO4 scheme are shown in Figure 6.6. While ENO-2 and ENO-3 reconstruction seem to give oscillation-free solutions, SP-WENO leads to minor undershoots near the discontinuity. The solutions with mesh refinement for SP-WENO are shown in Figure 6.7.



(a) Solution with TeCNO4

(b) Zoomed region near contact wave

Figure 6.6: Linear advection of discontinuity: Solution with TeCNO4 at time t=0.5 for $x \in [0, 1]$.



(a) N=100

(b) N=400

(c) N=800

Figure 6.7: Linear advection of discontinuity: mesh refinement study with SP-WENO.

6.5.2 Burgers' Equation

Consider the non-linear Burgers equation

$$\partial_x U + \partial_x \left(\frac{U^2}{2}\right) = 0,$$

whose base second-order entropy flux is chosen as (5.5), while the scalar diffusion matrix is taken to be of the form

$$D_{i+\frac{1}{2}} = \frac{|U_i| + |U_{i+1}|}{2}.$$

Sine-wave

We choose the domain $[-1, 1]$ with a smooth initial profile given by

$$U_0(x) = 1 + \frac{1}{2}\sin(\pi x)$$

and periodic boundary conditions. Being a non-linear problem, the solution develops a discontinuity in finite time, which can be evaluated to be $t = \frac{2}{\pi} \approx 0.636$. We simulate the solution till time $t = 0.3$ with CFL=0.4, at which point the solution is still smooth. The convergence rates with TeCNO4 are shown in Table 6.4. ENO-3 reconstruction shows a clear deterioration in its order of accuracy, while SP-WENO seems to once again give more than third-order accuracy.

In theory, the total entropy for smooth solutions is preserved over time, provided the boundary contributions can be dropped by assuming (say) periodic boundaries. After the appearance of the discontinuity, a sharp decrease in total entropy is expected. To see this, we evaluate the quantity

$$\mathcal{R}(t) = \frac{\mathcal{E}(t) - \mathcal{E}(0)}{\mathcal{E}(0)}, \qquad \text{where} \quad \mathcal{E}(t) := \int_{-1}^{1} \eta(U(x,t))\mathrm{d}x. \tag{6.9}$$

up to time $t = 0.7$. Note that $\mathcal{R}(t)$ is precisely the relative change in total entropy, which is constant (identically zero) for smooth solutions. The quantity $\mathcal{E}(t)$ is approximated by

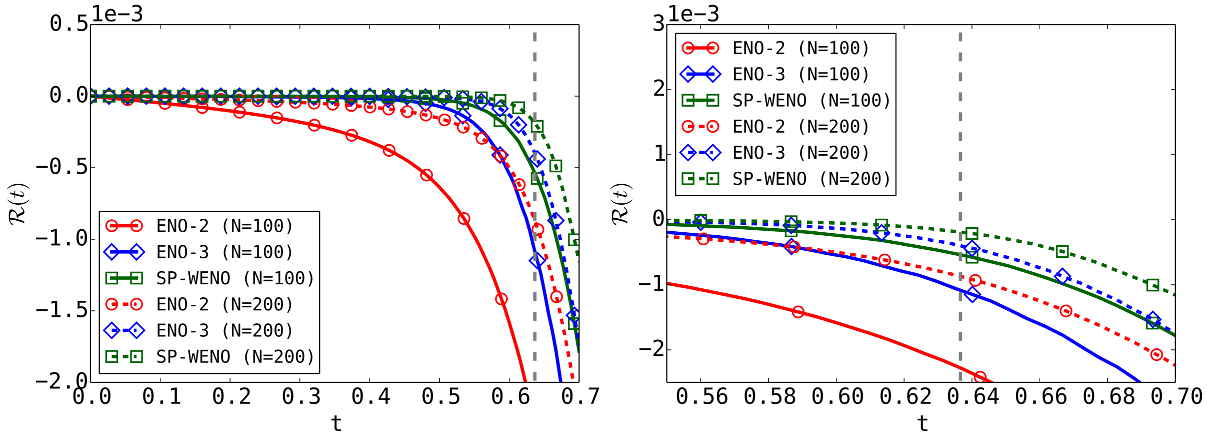
$$\mathcal{E}(t) \approx \mathcal{E}^h(t) = \sum_i \eta_i(t)h. \tag{6.10}$$

The results depicted in Figure 6.8 clearly show that SP-WENO performs the best from the point of view of preservation of total entropy, prior to the shock. The most dissipative solutions are obtained with ENO-2, while ENO-3 lies somewhere in between. Moreover, the performance with all reconstructions improves with mesh refinement.

Remark 6.5.1. *The time-stepping scheme may also introduce a small of amount numerical diffusion. We will return to this point in Chapter 8.*

N	SP-WENO		ENO3		ENO2	
	error	rate	error	rate	error	rate
50	3.41e-04	-	3.07e-04	-	4.73e-03	-
100	4.17e-05	3.03	4.76e-05	2.69	1.35e-03	1.81
200	4.51e-06	3.21	8.44e-06	2.49	3.77e-04	1.84
400	4.98e-07	3.18	1.80e-06	2.23	1.02e-04	1.89
600	1.33e-07	3.26	7.29e-07	2.23	4.71e-05	1.90
800	5.22e-08	3.25	3.91e-07	2.17	2.72e-05	1.92

Table 6.4: L_h^1 errors for burgers equation for sine-wave test.



(a) Evaluation with TeCNO4

(b) Zoomed region near $t = \frac{2}{\pi}$

Figure 6.8: Evolution of $\mathcal{R}(t)$ for the sine-wave test.

Shock wave

Consider the initial condition

$$U_0(x) = \begin{cases} 1, & \text{if } x < 0 \\ 1-x, & \text{if } 0 \leqslant x \leqslant 1 \\ 0, & \text{if } x > 0 \end{cases},$$

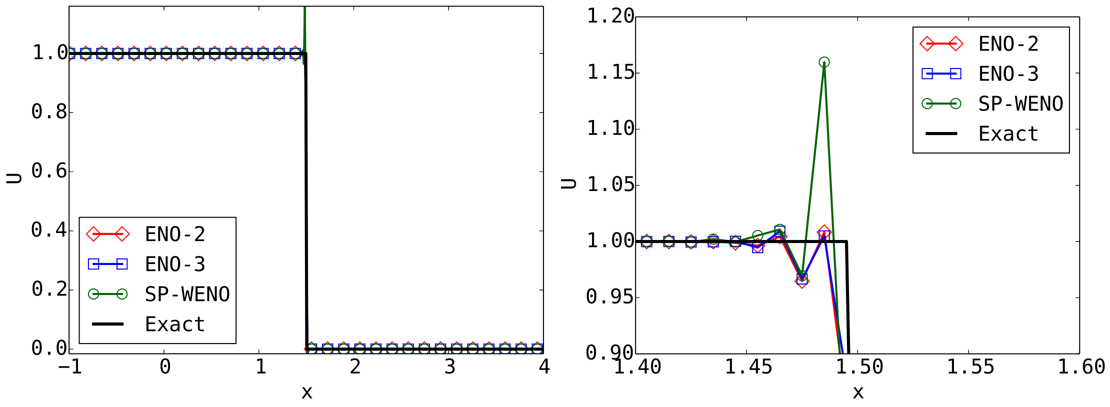
on the domain $[-1, 4]$. The Burgers equations with this initial condition can be solved by the *method of characteristics* (see Chapter 3 of [33]) till time $t < 1$, with the solution given by

$$U(x,t) = \begin{cases} 1, & \text{if } x < t, \quad 0 \leqslant t < 1 \\ \frac{1-x}{1-t}, & \text{if } t \leqslant x \leqslant 1, \quad 0 \leqslant t < 1 \\ 0, & \text{if } x > 1, \quad 0 \leqslant t < 1 \end{cases}. \tag{6.11}$$

However, the method of characteristics breaks down for $t \geqslant 1$ since the characteristics intersect. Beyond $t = 1$, the solution is described by the shock

$$U(x,t) = \begin{cases} 1, & \text{if } x < \frac{1+t}{2}, \quad t \geqslant 1 \\ 0, & \text{if } x \geqslant \frac{1+t}{2}, \quad t \geqslant 1 \end{cases}. \tag{6.12}$$

The solution is evaluated till final time $t_f = 2$ with CFL $= 0.4$. The mesh consists of 500 cells with transmissive boundary conditions. The solutions with the TeCNO4 flux is shown in Figure 6.9. Recall that SP-WENO gave minor overshoots near the discontinuity for the linear advection equation (see Figure 6.6). For the current test case, ENO-2, ENO-3 and SP-WENO all give minor oscillations near the shock, although the overshoot with SP-WENO is comparatively larger. The shock is equally well resolved by each method.



(a) Solution with TeCNO4

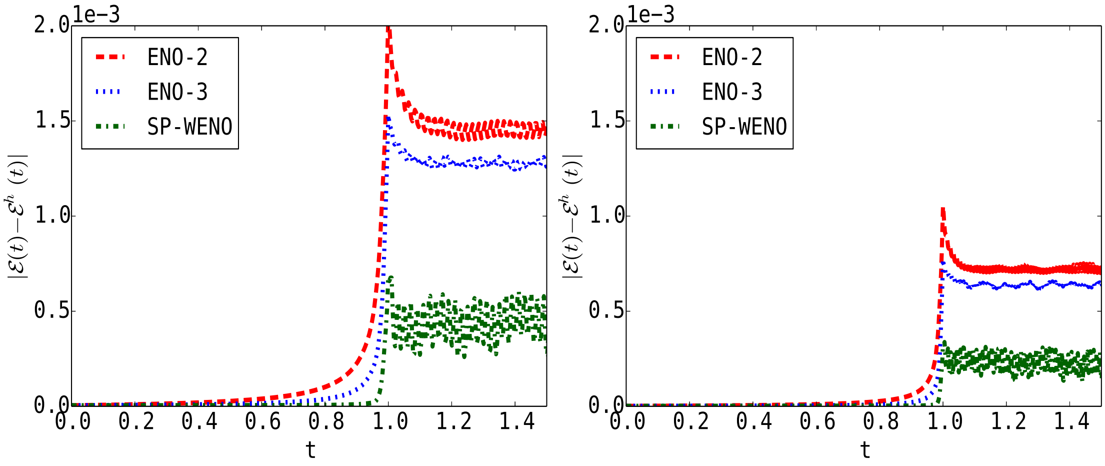
(b) Zoomed pre-shock region

Figure 6.9: Burgers equation with shock solution at $t = 2$.

Corresponding to the exact solution (6.11) and (6.12), the expression for the exact total entropy is given by

$$\mathcal{E}(t) = \int_{-1}^{4} \eta(U(x,t))\mathrm{d}x = \begin{cases} \frac{2+t}{3}, & \text{if } t < 1 \\ \frac{t+3}{4}, & \text{if } t \geqslant 1 \end{cases}, \tag{6.13}$$

while the total entropy with the numerical schemes is approximated by (6.10). We consider the deviation of the numerical total entropy from the exact value, by monitoring the evolution of $|\mathcal{E}(t) - \mathcal{E}^h(t)|$. The results shown in Figure 6.10 indicate that SP-WENO performs the best prior to the appearance of the shock at $t = 1$, while ENO-2 is the most dissipative. After the appearance of the shock, all methods deviate from the exact total entropy, with SP-WENO showing the least deviation. The magnitude of deviation reduces for all methods with mesh refinement.



(a) N=500

(b) N=1000

Figure 6.10: Evolution of deviation in total entropy for the shock wave test case by TeCNO4 schemes.

In the previous test case with the sine-wave, the total entropy was conserved before the appearance of the discontinuity, since periodic boundary conditions were imposed. However, in the present test case we have transmissive boundary conditions. Furthermore, the shock-wave does not reach the boundaries till the final time, ensuring the flux at $x = -1$ and $x = 4$ are constant for the entire simulation time. Thus, the quantity

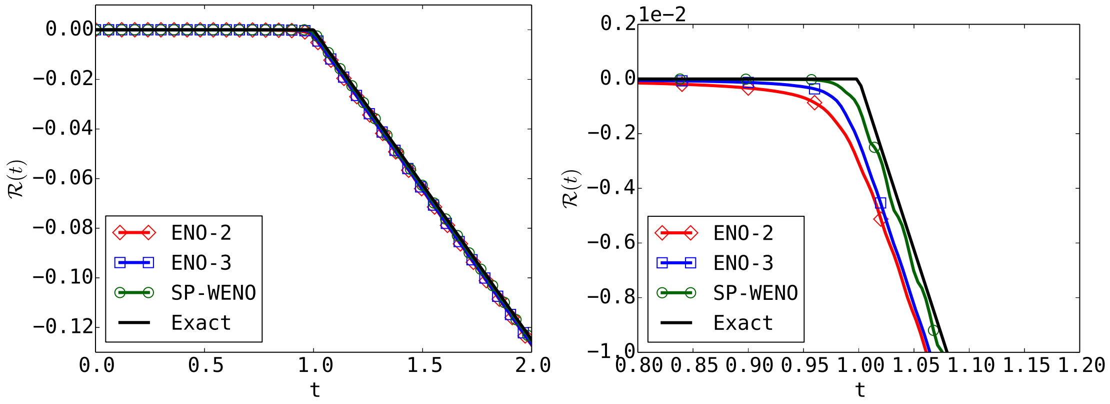
$$\mathcal{R}(t) = \frac{\mathcal{E}(t) - \mathcal{E}(0) + t\left(q(U)\Big|_{x=4} - q(U)\Big|_{x=-1}\right)}{\mathcal{E}(0)}, \tag{6.14}$$

is constant till $t < 1$, and shows a sharp decay after the appearance of the shock. Note that if periodic boundary conditions are imposed, (6.14) reduces to the relative change in total entropy given by (6.9). The entropy flux corresponding to quadratic entropy function is given by (2.20). Thus,

$$q(U)\Big|_{x=4} - q(U)\Big|_{x=-1} = -\frac{1}{3}.$$

Once again, SP-WENO performs the best at preserving $\mathcal{R}(t)$ till $t = 1$ and approximating the decay of $\mathcal{R}(t)$ after the appearance of the shock.

Remark 6.5.2. *The minor oscillations visible in Figure 6.9 can be attributed to insufficient dissipation near shocks. Note that despite the presence of oscillations, the scheme satisfies the entropy condition, and thus guaranteed to converge to the unique entropy solution in the L^1 space.*



(a) Evaluation with TeCNO4

(b) Zoomed region near $t = 1$

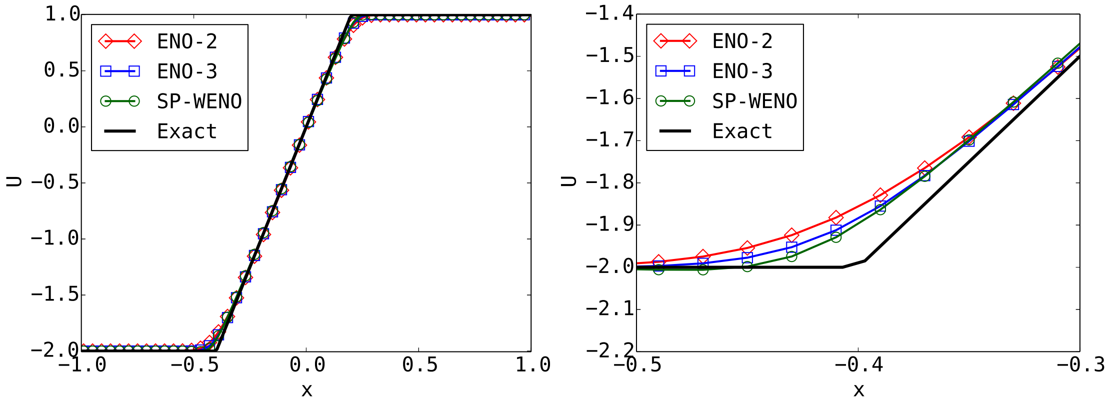
Figure 6.11: Evolution of $\mathcal{R}(t)$ for the shock wave test case.

Rarefaction wave

This test corresponds to a rarefaction wave. The domain is $[-1, 1]$ with the initial profile given by

$$u_0(x) = \begin{cases} -2, & \text{if } x < 0 \\ 1, & \text{if } x > 0 \end{cases} .$$

The solution is evaluated till final time $t_f = 0.2$ with CFL $= 0.4$. The mesh consists of 100 cells with transmissive boundary conditions. The solutions shown in Figure 6.12 indicates that SP-WENO gives the most accurate solution, while ENO-2 is the most dissipative.



(a) Solution with TeCNO4

(b) Zoomed region near x=0.4

Figure 6.12: Burgers equation rarefaction wave: Solution with TeCNO4 at time $t = 0.2$.

6.6 Numerical results with SP-WENO for Euler equations

For the Euler equations, we use the KEPEC flux given by (5.12) as the base second-order entropy conservative flux. We choose the Roe-type diffusion operator (5.22) to construct entropy stable schemes. Note that the scaled entropy variables $\mathbf{Z}$ are reconstructed, which leads to the numerical flux

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}^{(4,*)}_{i+\frac{1}{2}} - \frac{1}{2}\mathbf{R}^\top_{i+\frac{1}{2}}\mathbf{\Lambda}_{i+\frac{1}{2}}[\![\mathbf{Z}]\!]_{i+\frac{1}{2}}.$$

Smooth density wave

This test case is in the same spirit as that of linear advection test considered in of Section 6.5.1, for which ENO-3 is shown to perform poorly. The domain is chosen as $[0,2]$ with the initial profile

$$\rho = 1 + 0.5\sin^4(x), \quad u = 0.5, \quad p = 1,$$

and periodic boundary conditions. The final time is $t_f = 0.5$ with CFL $= 0.5$. The exact solution corresponds to the advection of the smooth density profile. The L^1_h errors for density obtained on different meshes are shown in Table 6.5. The solution with ENO-3 loses its expected order of accuracy, which drops well below second-order, as was also observed in Section 6.5.1. SP-WENO on the other hand gives more than third-order accuracy.

N	SP-WENO		ENO-3	
	error	rate	error	rate
100	2.61e-04	-	3.43e-04	-
200	2.91e-05	3.17	4.46e-05	2.94
400	3.21e-06	3.18	6.66e-06	2.74
600	8.91e-07	3.16	2.88e-06	2.05
800	3.56e-07	3.19	1.79e-06	1.66
1000	1.75e-07	3.18	1.25e-06	1.59

Table 6.5: L^1_h error of density for Euler equations: advecting smooth density wave.

Shu-Osher test

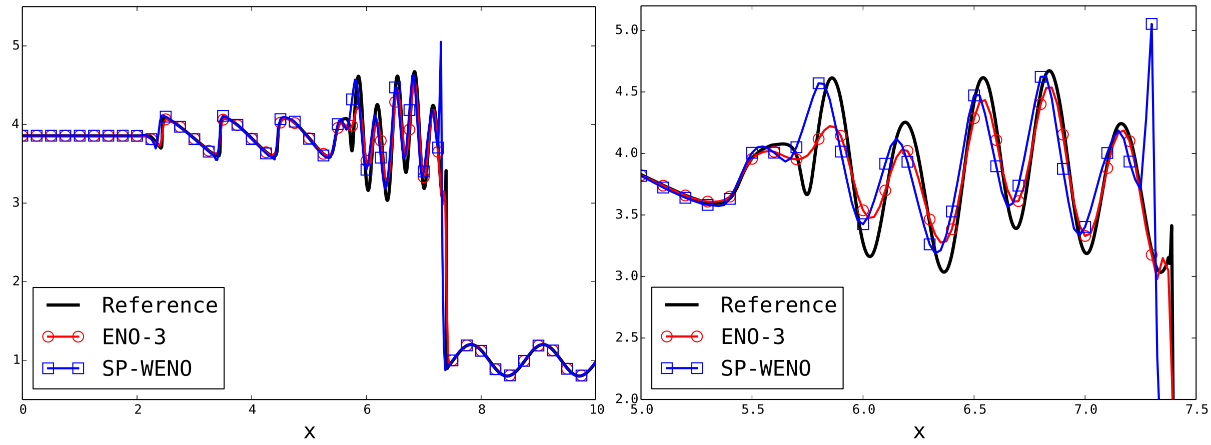
This test case proposed by Shu and Osher [107] involves the interaction of shocks of different strengths and highly oscillatory smooth waves. The domain is chosen as $[-5,5]$ with final time $t_f = 1.8$ and CFL $= 0.4$. The initial condition has a discontinuity at $x = -4$ with

$$\mathbf{U}_0(x) = \begin{cases} \mathbf{U}_L & \text{if } x < -4 \\ \mathbf{U}_R & \text{if } x \geqslant -4 \end{cases},$$

where

$$\begin{bmatrix}\rho_L\\u_L\\p_L\end{bmatrix} = \begin{bmatrix}3.857143\\2.629369\\10.33333\end{bmatrix}, \qquad \begin{bmatrix}\rho_R\\u_R\\p_R\end{bmatrix} = \begin{bmatrix}1.0+0.2\sin(5x)\\0\\1\end{bmatrix}.$$

The solutions with SP-WENO and ENO-3 are shown in Figure 6.13 on a mesh with $N =$ 400 cells. As the expression for an exact solution is not available, a solution with ENO-3 on a mesh with 2000 cells is used for reference. The TeCNO4 with ENO-3 reconstruction does well in approximating the solution, as compared to the reference solution. However SP-WENO gives a fairly large overshoot close to the strong shock. Although the solution will converge in the L^1 sense, large amplitude oscillations for the Euler equations may lead to the violation of positivity of pressure or density.



(a) Density (b) Zoomed profile

Figure 6.13: Euler equations Shu-Osher test: Solutions using TeCNO4 with ENO-3 and SP-WENO at time $t = 1.8$.

6.7 SP-WENOc: A fix for systems of conservation laws

In order to understand why SP-WENO gives unsatisfactory results while approximating discontinuous solutions, we need to take a closer look at the scheme being used. The TeCNO4 flux uses a fourth-order entropy conservative flux, coupled with a dissipation operator where an appropriate reconstruction of the scaled entropy variables leads to the reconstructed jump $[\![\mathbf{Z}]\!]_{i+\frac{1}{2}}$. As shown Section 6.3.1, the reconstructed jump with SP-WENO is zero in a large number of scenarios, i.e., group Ω_0. In other words, there is no numerical dissipation in these regions, and scheme is governed only by the fourth order entropy conservative flux. While the cases covered in Ω_0 need not be satisfied at an interface corresponding to a discontinuity in the solution, it may describe an interface in the close proximity of a shock (or a contact). This could lead to large Gibbs oscillations, as was observed in the numerical results in the previous sections.

A possible fix would be to perturb the reconstruction procedure described by SP-WENO, so that the reconstructed jump is non-zero in key regions. In particular, we focus on cases 2 and 3 of Ω_0, which correspond to a concave or convex solution profile about the interface under consideration. In terms of the jump ratios, these are characterized by either $\theta_i^+ < 1, \theta_{i+1}^- > 1$ or $\theta_i^+ > 1, \theta_{i+1}^- < 1$. Let us collectively call these two scenarios as the *C-region*. Consider the reconstructed jump written in the form (6.5). In the C-region we always have $\left[\widetilde{w}_0(1-\theta_{i+1}^-) + w_1(1-\theta_i^+)\right] = 0$ with SP-WENO reconstruction. Thus, we introduce a small perturbation in terms of the function $\mathcal{G}$, such that

$$[\![v]\!]_{i+\frac{1}{2}} = \frac{1}{2}\left[\widetilde{w}_0(1-\theta_{i+1}^-) + w_1(1-\theta_i^+) + \mathcal{G}\right]\Delta v_{i+\frac{1}{2}}. \tag{6.15}$$

In order to ensure that the perturbation is a consequence of the the appropriate choice of the WENO weights, we propose the following modifications

$$\overline{C}_1 = C_1 - \frac{1}{4}\frac{\mathcal{G}}{(1-\theta_i^+)}, \qquad \overline{C}_2 = C_2 - \frac{1}{4}\frac{\mathcal{G}}{(1-\theta_{i+1}^-)}, \tag{6.16}$$

which is well-defined in the C-region since $\theta_i^+ \neq 1$ and $\theta_{i+1}^- \neq 1$. In order to ensure that the weights are consistent, i.e., (P1') is satisfied, we consider the additional modification

$$C_1^{\#} = \min\left(\max\left(\overline{C}_1, -\frac{3}{8}\right), \frac{1}{8}\right), \qquad C_2^{\#} = \min\left(\max\left(\overline{C}_2, -\frac{3}{8}\right), \frac{1}{8}\right). \tag{6.17}$$

$C_1^{\#}, C_2^{\#}$ are used in place of C_1, C_2 for evaluating the SP-WENO weights in the C-region.

The next task is to choose the function $\mathcal{G}$, with the aim to retain most of the desirable properties of SP-WENO. One possible choice is

$$\mathcal{G} = \left(\frac{|\Delta v_{i+\frac{1}{2}}|}{0.5(|v_i| + |v_{i+1}|)}\right)^3, \tag{6.18}$$

provided $\Delta v_{i+\frac{1}{2}} \neq 0$. Based on the Remark 6.2.1, negation symmetry and the mirror property are preserved for this form of $\mathcal{G}$. Since $\mathcal{G} \geqslant 0$, the constraint (6.17) guarantees that the pair $(C_1^{\#}, C_2^{\#})$ lies in the feasible regions for cases 2 and 3 (refer to Figure 6.4). Furthermore, the perturbation to the initial SP-WENO jump is $\mathcal{O}(|\Delta v_{i+\frac{1}{2}}|^4)$ for smooth solutions (assuming $C_1^{\#} = \overline{C}_1$ and $C_2^{\#} = \overline{C}_2$), which ensures that the superior order of convergence observed with the TeCNO4 scheme is retained.

We refer to the SP-WENO reconstruction with the correction (6.16) (6.17) and (6.18) in the C-region, as *SP-WENOc*.

6.8 Reconstruction accuracy of SP-WENOc

As done for SP-WENO, we first test whether SP-WENOc truly leads to a third-order reconstruction at the cell interfaces. We consider the smooth function

$$u(x) = d_0 + \sin(10\pi x) + x, \tag{6.19}$$

where we have the freedom to choose d_0. The L^1 interface errors and convergence rates for $d_0 = 0$ are shown in Table 6.6. While the original SP-WENO gives more than third-order convergence, SP-WENOc with (6.18) fails to give even third-order convergence. However, SP-WENOc regains the accuracy of SP-WENO if we choose $d_0 = 2$, as shown in Table 6.7. Our explanation for this behaviour of SP-WENOc is that for $d_0 = 0$, the function $u(x)$ can take values very close to zero. In these regions $|u_i|, |u_{i+1}| \sim 0$, which leads to very bad scaling of the correcting function $\mathcal{G}$ given by (6.18). Thus, the numerical dissipation can be quite large causing a drop in accuracy. However, for $d_0 = 2$ the function is translated away from zero, and we find that SP-WENOc performs as well as SP-WENO.

One way to fix this issue is to choose $\mathcal{G}$ as

$$\mathcal{G} = \left(\min\left(\frac{|\Delta v_{i+\frac{1}{2}}|}{0.5(|v_i| + |v_{i+1}|)}, |\Delta v_{i+\frac{1}{2}}|\right)\right)^3, \tag{6.20}$$

which puts a bound in the magnitude of $\mathcal{G}$. If SP-WENOc is used with (6.20), we find that we recover the loss in accuracy for d=0 as shown in Table 6.6. For d=2, the original SP-WENO and both forms of SP-WENOc give almost identical errors and convergence rates. Thus, we use SP-WENOc with (6.20) to reconstruct the scaled entropy variables in the TeCNO4 scheme.

N	SP-WENO		SP-WENOc with (6.18)		SP-WENOc with (6.20)	
	error	rate	error	rate	error	rate
40	8.59e-02	-	1.44e-01	-	8.79e-02	-
80	6.73e-03	3.67	2.20e-02	2.71	7.35e-03	3.58
160	5.01e-04	3.75	3.46e-03	2.66	5.27e-04	3.80
320	3.64e-05	3.78	5.43e-04	2.67	3.78e-05	3.80
640	2.59e-06	3.81	8.49e-05	2.67	2.68e-06	3.82
1280	1.82e-07	3.83	1.33e-05	2.68	1.87e-07	3.84
2560	1.26e-08	3.85	2.08e-06	2.67	1.29e-08	3.85

Table 6.6: L^1_h reconstruction errors with SP-WENO and SP-WENOc for $d_0 = 0$ in (6.19).

N	SP-WENO		SP-WENOc with (6.18)		SP-WENOc with (6.20)	
	error	rate	error	rate	error	rate
40	8.59e-02	-	8.59e-02	-	8.59e-02	-
80	6.73e-03	3.67	6.73e-03	3.67	6.73e-03	3.67
160	5.01e-04	3.75	5.01e-04	3.75	5.01e-04	3.75
320	3.64e-05	3.78	3.64e-05	3.78	3.64e-05	3.78
640	2.59e-06	3.81	2.59e-06	3.81	2.59e-06	3.81
1280	1.82e-07	3.83	1.82e-07	3.83	1.82e-07	3.83
2560	1.26e-08	3.85	1.26e-08	3.85	1.26e-08	3.85

Table 6.7: L^1_h reconstruction errors with SP-WENO and SP-WENOc for $d_0 = 2$ in (6.19).

Remark 6.8.1. *The stability bound* (6.8) *no longer holds for SP-WENOc. However, a careful a case-by-case calculation will lead to an alternate bound of the form*

$$\left|[\![v]\!]_{i+\frac{1}{2}}\right| \leqslant 4\left(\left|\Delta v_{i-\frac{1}{2}}\right| + \left|\Delta v_{i+\frac{1}{2}}\right| + \left|\Delta v_{i+\frac{3}{2}}\right|\right). \tag{6.21}$$

6.9 Numerical results with SP-WENOc for Euler equations

We now present results using the TeCNO4 scheme, with KEPEC as the base flux and the modified SP-WENOc reconstruction.

Smooth density wave

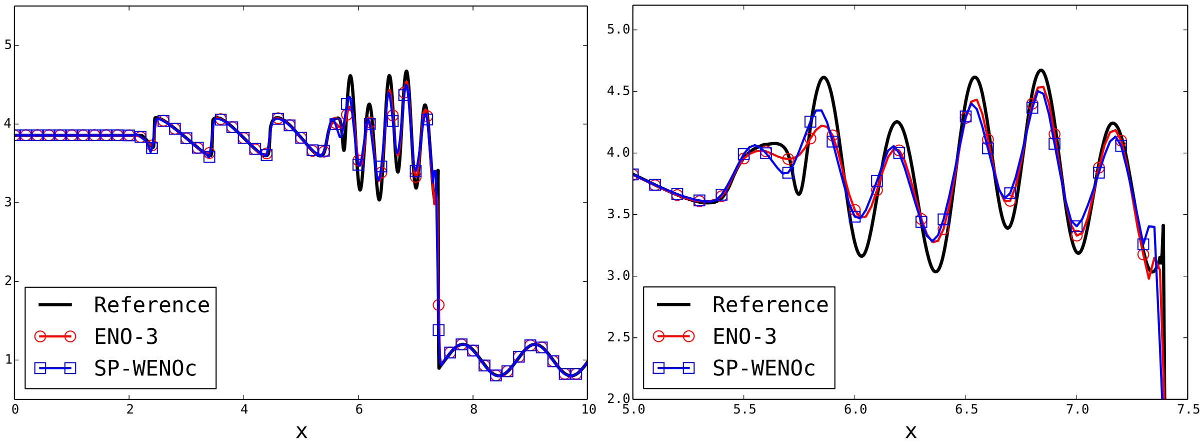
This test corresponds to the one discussed in Section 6.6. The L_h^1 errors for density obtained on different meshes are shown in Table 6.8. Both SP-WENO and SP-WENOc give very similar results, with more than third-order accuracy. Recall that there is a severe loss in accuracy with ENO-3 (see Table 6.5).

N	SP-WENO		SP-WENOc	
	error	rate	error	rate
100	2.61e-04	-	2.60e-04	-
200	2.91e-05	3.17	2.91e-05	3.16
400	3.21e-06	3.18	3.21e-06	3.18
600	8.91e-07	3.16	8.91e-07	3.16
800	3.56e-07	3.19	3.55e-07	3.19
1000	1.75e-07	3.18	1.74e-07	3.18

Table 6.8: L_h^1 error of density for Euler equations: advecting smooth density wave.

Shu-Osher test

We consider the Shu-Osher test, for which SP-WENO resulted in large overshoot near the strong shock (see Figure 6.13). Although the new SP-WENOc reconstruction does not completely remove the overshoot, it definitely gives much better control over the magnitude of oscillation compared to SP-WENO, as can be seen in Figure 6.14. This indicates that the numerical dissipation does not vanish in key regions under the proposed modification.



(a) Density

(b) Zoomed profile

Figure 6.14: Euler equations Shu-Osher test: Solutions using TeCNO4 with ENO-3 and SP-WENOc at time $t = 1.8$.

Isentropic vortex

This corresponds to advection of an isentropic vortex discussed in Section 5.8.4. We run the simulation till $t_f = 20$ with $\alpha = 0°$ and $M = 0.5$, at the end of which the vortex completes one horizontal cycle. The L^1_h errors for density, pressure and both velocity components are shown in Table 6.9. Once again, both SP-WENO and SP-WENOc give almost identical results, with more than third-order accuracy.

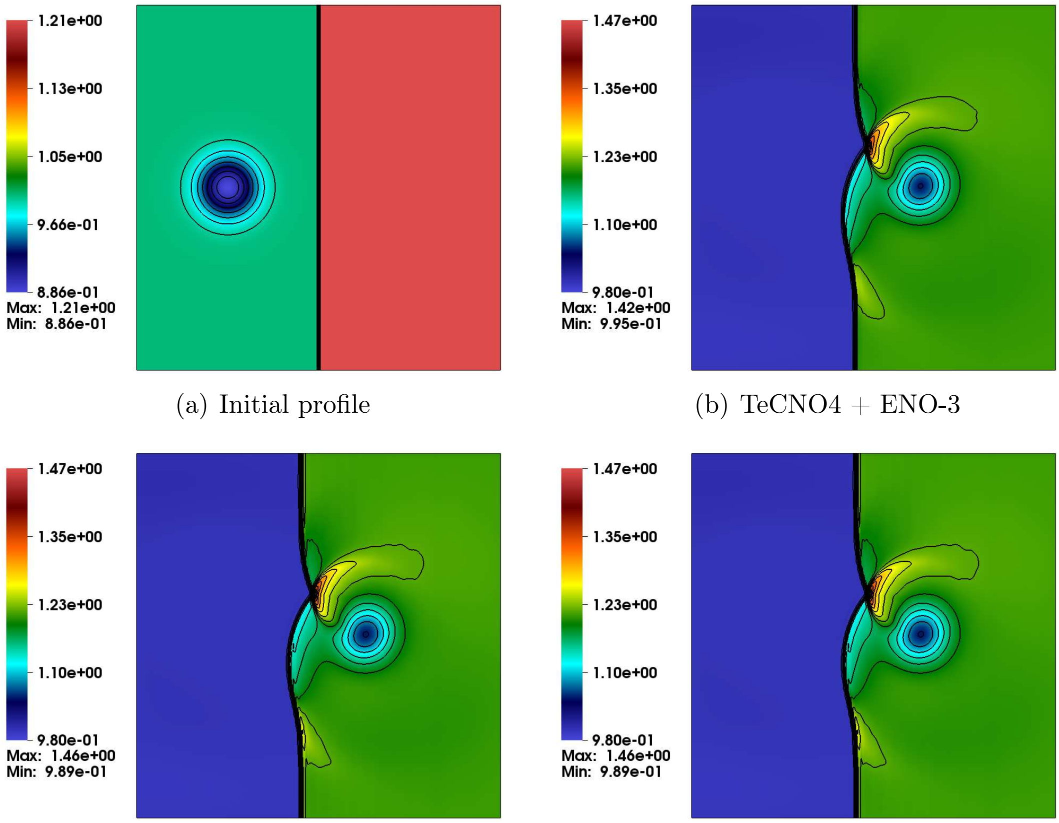
	Density				Pressure			
N	SP-WENO		SP-WENOc		SP-WENO		SP-WENOc	
	error	rate	error	rate	error	rate	error	rate
40	1.43e-01	-	1.46e-01	-	2.01e-01	-	2.09e-01	-
80	1.82e-02	2.97	1.79e-02	3.03	2.54e-02	2.98	2.53e-02	3.05
160	1.33e-03	3.77	1.35e-03	3.72	1.90e-03	3.74	1.95e-03	3.70
320	1.04e-04	3.67	1.06e-04	3.67	1.46e-04	3.70	1.50e-04	3.70

	x-Velocity				y-Velocity			
N	SP-WENO		SP-WENOc		SP-WENO		SP-WENOc	
	error	rate	error	rate	error	rate	error	rate
40	3.68e-01	-	3.66e-01	-	3.66e-01	-	3.69e-01	-
80	6.89e-02	2.41	6.74e-02	2.44	6.81e-02	2.43	6.66e-02	2.47
160	5.73e-03	3.58	5.78e-03	3.54	5.60e-03	3.60	5.60e-03	3.57
320	4.57e-04	3.65	4.60e-04	3.65	4.85e-04	3.53	4.86e-04	3.53

Table 6.9: L^1_h error for advecting isentropic vortex.

Shock vortex interaction

This problem was introduced in Section 5.8.5. The various parameters of the perturbation are chosen as $\epsilon = 0.3$, $r_c = 0.005$, $\beta = 0.204$ and $(x_c, y_c) = (0.25, 0.5)$. The initial profile of solution on a 200 × 200 mesh is shown in Figure 6.15(a). The numerical solutions at $t_f = 0.35$ with ENO-3, SP-WENO and SP-WENOc are shown in Figure 6.15(b)-(d). The solution with all three methods are comparable, with well resolved shock lines.



(a) Initial profile

(b) TeCNO4 + ENO-3

(c) TeCNO4 + SP-WENO

(d) TeCNO4 + SP-WENOc

Figure 6.15: Density profiles for shock-vortex interaction.

7. Energy preserving finite difference scheme for the shallow water equations

The study of fluid dynamics in thin layers, such as the flow in lakes, rivers and the sea near coastal areas, are of great importance in the fields of oceanography and climate-modelling. Such flows are governed by the shallow water equations (2.23), which can be obtained from the incompressible Euler equations by assuming: i) the variations in the vertical direction (along z-direction) are negligible compared to those in the horizontal scales, ii) the pressure is evaluated from hydrostatic balance [72]. Entropy conservative and entropy stable schemes have been constructed and tested for (2.23) in [118, 125]. The shallow water equations (2.23) assume a flat bottom topography. Generalising the model by assuming a non-flat topography, introduces additional source terms in (2.23). Well-balanced energy stable schemes for the shallow water equations with non-flat discontinuous topography have been proposed in [37].

The model (2.23) is written in terms of the conserved variables $(hu_1, hu_2, h)^\top$. However, several meteorological models are actually based on the formulation in terms of the primitive variables $(u_1, u_2, h)^\top$, which is also called the *vector-invariant form* [3, 119, 100, 101, 122]. Furthermore, since these models are used to simulate planetary waves, the *Coriolis force* i.e., the inertial force due the the rotation of the Earth, also needs to be considered. In this chapter, we consider the shallow water equations in the vector-invariant form, with non-flat bottom topography and rotational terms included in the formulation. The total energy of this model is preserved for smooth solutions. Thus, our objective is to construct an energy preserving finite difference scheme for this system.

7.1 Governing equations

Let $\mathbf{u} = (u_1, u_2)^\top$ denote the horizontal velocity, h be the vertical extent of the fluid column above the bottom surface, and $h_s := h_s(x, y)$ be the height of the bottom topography relative to the mean surface of the Earth. Additionally, define the kinetic energy per unit mass $\mathcal{K} = \frac{1}{2}|\mathbf{u}|^2$ and the vorticity $\omega = \mathbf{k} \cdot \nabla \times \mathbf{u}$, where $\mathbf{k}$ is the unit vector normal to the Earth's surface. The shallow water model in primitive form can be written as

$$\begin{aligned} \partial_t \mathbf{u} + \nabla(gH + \mathcal{K}) + (\mathscr{F} + \omega)\, \mathbf{k} \times \mathbf{u} = 0, \\ \partial_t h + \nabla \cdot (h\mathbf{u}) = 0, \end{aligned} \tag{7.1}$$

where $H = h + h_s$ is the height of the fluid column above the mean surface of the earth. The quantity $\mathscr{F} = 2\varpi \sin\theta$ is the Coriolis parameter, with ϖ being the angular velocity of the earth and ϕ corresponding to the geographical latitude of the Earth's surface. We use the notation $\vartheta := \mathscr{F} + \omega$ to denote the *absolute vorticity* of the fluid. Additionally, the quantity ϑ/h is called the *potential vorticity*, while $\vartheta^2/(2h)$ is called the *potential enstrophy.*

We can rewrite the equations (7.1) as

$$\partial_t \mathbf{U} + \mathbf{A}_1(\mathbf{U})\partial_x \mathbf{U} + \mathbf{A}_2(\mathbf{U})\partial_y \mathbf{U} + \widetilde{\boldsymbol{S}} = 0, \tag{7.2}$$

where

$$\mathbf{U} = \begin{pmatrix} u_1 \\ u_2 \\ h \end{pmatrix}, \quad \mathbf{A}_1(\mathbf{U}) = \begin{pmatrix} u_1 & 0 & g \\ 0 & u_1 & 0 \\ h & 0 & u_1 \end{pmatrix}, \quad \mathbf{A}_2(\mathbf{U}) = \begin{pmatrix} u_2 & 0 & 0 \\ 0 & u_2 & g \\ 0 & h & u_2 \end{pmatrix}, \tag{7.3}$$

while the source term is $\widetilde{\boldsymbol{S}} = (-u_2\mathscr{F}, u_1\mathscr{F}, 0)^\top$. Referring to Definition 2.0.1, we consider the matrix $\mathbf{A}(\mathbf{U}, \mathbf{n})$ for $\mathbf{n} \in \mathbb{R}^2$, which has the eigenvalues $\lambda_1 = u_n, \lambda_2 = u_n - \sqrt{gh}, \lambda_3 = u_n + \sqrt{gh}$ where $u_n = \mathbf{u} \cdot \mathbf{n}$. The matrix of eigenvectors corresponding to $\mathbf{A}(\mathbf{U}, \mathbf{n})$ is given by

$$\mathbf{R} = \begin{pmatrix} -n_2 & -\sqrt{g}n_1 & \sqrt{g}n_1 \\ n_1 & -\sqrt{g}n_2 & \sqrt{g}n_2 \\ 0 & \sqrt{h} & \sqrt{h} \end{pmatrix}, \tag{7.4}$$

with the eigenvectors being linearly independent. Note that the eigenvalues are the identical to those evaluated for the shallow water equations written in terms of the conserved variables in Section 2.5.4.

Consider the energy of the shallow water equations,

$$\eta(\mathbf{U}) = h\mathcal{K} + \frac{1}{2}gH^2,$$

which is a sum of the kinetic energy and the potential energy of the fluid. We define a new vector of variables

$$\mathbf{V} = \eta'(\mathbf{U}) = \begin{pmatrix} hu_1 \\ hu_2 \\ \mathcal{K} + gH \end{pmatrix}. \tag{7.5}$$

Taking the scalar product of (7.1) with the $\mathbf{V}$ leads to the following conservation law for energy

$$\partial_t (h\mathcal{K} + \frac{1}{2}gH^2) + \nabla \cdot [(gH + \mathcal{K})h\mathbf{u}] = 0. \tag{7.6}$$

Note that the absolute vorticity term in (7.1) does not affect the energy equation (7.6), since $\mathbf{u} \cdot (\mathbf{k} \times \mathbf{u}) = 0$.

Remark 7.1.1. *Recall from Section 2.5.4, that the energy served as an entropy function for the shallow water equations written in the conservative form. However, $\eta(\mathbf{U})$ is not convex for the vector-invariant formulation in terms of the $\mathbf{u} - h$ variables, and thus it cannot be called an entropy function for the system* (7.2)-(7.3)*. Furthermore, we do not refer to $\mathbf{V}$ as the vector of entropy variables.*

7.1.1 Preserved quantities

Integrating (7.6) (satisfied for smooth solutions) over a domain Ω and assuming periodic or no-flow boundary conditions, leads to the preservation of total energy

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \eta \, \mathrm{d}\mathbf{x} = \frac{\mathrm{d}}{\mathrm{d}t}\int_{\Omega} \left(h\mathcal{K} + \frac{1}{2} g H^2 \right) \mathrm{d}\mathbf{x} = 0.$$

However, total energy is not the only quantity preserved by the the system (7.1). In fact, it is possible to show that any quantity of the form

$$C = \int_{\Omega} h\mathcal{F}\left(\frac{\vartheta}{h}\right) \mathrm{d}\mathbf{x}, \tag{7.7}$$

is conserved, where $\mathcal{F}$ is any arbitrary function of the the potential vorticity [104]. Taking $\mathcal{F} \equiv 1$ leads to the conservation of mass, which can also be obtained by integrating the last equation of (7.1) over the domain Ω. Taking $\mathcal{F}$ as the identity map leads to the conservation of the total absolute vorticity, while the choice $\mathcal{F} = \vartheta^2/(2h^2)$ leads to the conservation of the total potential enstrophy. Using the terminology of Hamiltonian mechanics, the system (7.1) is an example of a *non-canonical Hamiltonian system*, which by Noether's theorem has two types of conserved quantities, namely *Hamiltonians* and *Casimirs* [104]. The total energy is the Hamiltonian of the system, while quantities given by (7.7) are the Casimir invariants for the system.

A second-order energy and potential enstrophy preserving staggered finite-difference scheme for (7.1), was proposed by Arakawa and Lamb [3] on Cartesian and spherical grids. An extension to this scheme was proposed by Takano and Wurtele [119], which ensures the fourth-order accurate advection of potential vorticity in the case of non-divergent mass flux, i.e., $\nabla.(h\mathbf{u}) = 0$. Extending the Arakawa and Lamb scheme to more general grids, while being able to conserve total energy, total potential enstrophy and other invariants, is by no means a trivial task. In [92], a potential enstrophy and energy preserving scheme was constructed on geodesic grids, by suitably discretizing the divergence and curl operators. In a series of papers [122, 91, 121], a family of schemes (referred to as TRiSK) have been developed on non-orthogonal polygonal grids, using the tools of discrete exterior calculus (DEC). However, these schemes are unable to simultaneously preserve both energy and potential enstrophy. Recently, the Arakawa and Lamb scheme has been extended to generalized non-orthogonal quadrilateral grids in [124], where a finite difference scheme has been formulated directly for the shallow water equations cast in generalized curvilinear coordinates. This scheme is able to preserve both invariants.

An alternate approach of constructing energy and potential enstrophy preserving schemes, using the tools of Hamiltonian mechanics along Helmholtz decomposition, was proposed by Salmon [101] on planar grids. Combining the Hamiltonian and DEC approaches, the Arakawa and Lamb scheme has been recently extended to arbitrary spherical grids in [32].

7.1.2 $\mathbf{u} - h - \vartheta$ model

From the above discussions, the (absolute) vorticity ϑ clearly plays an important role in atmospheric flows. Note that ϑ is also a conserved quantity. Assuming that the Coriolis parameter $\mathscr{F}$ is constant in time (which is a valid assumption since the angular velocity of the Earth is independent of time), the evolution equation for ϑ can be obtained by taking the curl of the velocity equations in (7.1). Solving for the velocity $\mathbf{u}$ may not lead to a good description of the vorticity dynamics. Hence, we also solve for ϑ as an independent variable, by considering the larger coupled system

$$\begin{aligned}\partial_t \mathbf{u} + \nabla(gH + \mathcal{K}) + \vartheta(\mathbf{k} \times \mathbf{u}) &= 0,\\ \partial_t h + \nabla \cdot (h\mathbf{u}) &= 0,\\ \partial_t \vartheta + \nabla \cdot (\vartheta \mathbf{u}) &= 0.\end{aligned} \tag{7.8}$$

We can rewrite the equations (7.8) as

$$\begin{aligned}\partial_t \mathbf{U} + \partial_x \mathbf{f}_1(\mathbf{U}) + \partial_y \mathbf{f}_2(\mathbf{U}) &= -\boldsymbol{S},\\ \partial_t \vartheta + \partial_x f_1^\vartheta + \partial_y f_2^\vartheta &= 0,\end{aligned} \tag{7.9}$$

where

$$\mathbf{U} = \begin{pmatrix} u_1 \\ u_2 \\ h \end{pmatrix}, \quad \mathbf{f}_1(\mathbf{U}) = \begin{pmatrix} gH + \mathcal{K} \\ 0 \\ hu_1 \end{pmatrix}, \quad \mathbf{f}_2(\mathbf{U}) = \begin{pmatrix} 0 \\ gH + \mathcal{K} \\ hu_2 \end{pmatrix}, \quad (f_1^\vartheta, f_2^\vartheta) = \vartheta \mathbf{u}^\top, \tag{7.10}$$

while the source term is $\boldsymbol{S} = (-u_2\vartheta, u_1\vartheta, 0)^\top$.

Solving for vorticity directly as an additional variable is not a new approach. The vorticity-velocity formulation for incompressible Navier-Stokes has been quite popular [40, 49, 9], and numerical schemes based on such models have been shown to perform well in practice. In [82], it has been shown that the evaluation of discrete vorticity directly in the finite element set-up, is more suitable to ensure the conservation of *total helicity* for the Navier-Stokes equations. Recently, an energy, enstrophy and vorticity preserving spectral scheme has been proposed for the two-dimensional incompressible Navier-Stokes equations, by solving a coupled vorticity-velocity system [86].

7.2 Finite difference scheme

We propose the following semi-discrete finite difference scheme on a uniform Cartesian grid to solve the system (7.8)

$$\frac{\mathrm{d}\mathbf{U}_{i,j}}{\mathrm{d}t} + \frac{\left(\mathbf{F}^x_{i+\frac{1}{2},j} - \mathbf{F}^x_{i-\frac{1}{2},j}\right)}{\Delta x} + \frac{\left(\mathbf{F}^y_{i,j+\frac{1}{2}} - \mathbf{F}^y_{i,j-\frac{1}{2}}\right)}{\Delta y} = -\boldsymbol{S}_{i,j}, \tag{7.11a}$$

$$\frac{\mathrm{d}\vartheta_{i,j}}{\mathrm{d}t} + \frac{\left(F^{x,\vartheta}_{i+\frac{1}{2},j} - F^{x,\vartheta}_{i-\frac{1}{2},j}\right)}{\Delta x} + \frac{\left(F^{y,\vartheta}_{i,j+\frac{1}{2}} - F^{y,\vartheta}_{i,j-\frac{1}{2}}\right)}{\Delta y} = 0, \tag{7.11b}$$

where the numerical fluxes $\mathbf{F}^x, \mathbf{F}^y$ are consistent with $\mathbf{f}_1, \mathbf{f}_2$ respectively, while $(F^{x,\vartheta}, F^{y,\vartheta})$ is consistent with $(f_1^\vartheta, f_2^\vartheta)$. The following theorem gives a sufficient condition to ensure that the scheme (7.11a) conserves energy, which is in the same spirit as the condition required to construct entropy conservative schemes for conservation laws.

Theorem 7.2.1. *Consider the energy $\eta(\mathbf{U})$ and the corresponding energy flux components*

$$q_1(\mathbf{U}) = (gH + \mathcal{K})hu_1, \quad q_2(\mathbf{U}) = (gH + \mathcal{K})hu_2,$$

appearing in the evolution equation (7.6) *for $\eta(\mathbf{U})$. Define the potential functions*

$$\Psi^x(\mathbf{U}) := \langle \mathbf{V}(\mathbf{U}), \mathbf{f}_1(\mathbf{U})\rangle - q_1(\mathbf{U}), \quad \Psi^y(\mathbf{U}) := \langle \mathbf{V}(\mathbf{U}), \mathbf{f}_2(\mathbf{U})\rangle - q_2(\mathbf{U}), \tag{7.12}$$

where the vector $\mathbf{V}$ is defined by (7.5)*. If the numerical flux functions $\mathbf{F}^x, \mathbf{F}^y$ satisfy the conditions*

$$\left\langle \Delta\mathbf{V}_{i+\frac{1}{2},j}, \mathbf{F}^x_{i+\frac{1}{2},j}\right\rangle = \Delta\Psi^x_{i+\frac{1}{2},j}, \quad \left\langle \Delta\mathbf{V}_{i,j+\frac{1}{2}}, \mathbf{F}^y_{i,j+\frac{1}{2}}\right\rangle = \Delta\Psi^y_{i,j+\frac{1}{2}}, \tag{7.13}$$

then, the numerical scheme (7.11a) *preserves energy. In other words, it satisfies*

$$\frac{d\eta(\mathbf{U}_i)}{dt} + \frac{\left(q^x_{i+\frac{1}{2},j} - q^x_{i-\frac{1}{2},j}\right)}{\Delta x} + \frac{\left(q^y_{i,j+\frac{1}{2}} - q^y_{i,j-\frac{1}{2}}\right)}{\Delta y} = 0 \tag{7.14}$$

where $q^x_{i+\frac{1}{2},j} = \left\langle \overline{\mathbf{V}}_{i+\frac{1}{2},j}, \mathbf{F}^x_{i+\frac{1}{2},j}\right\rangle - \overline{\Psi^x}_{i+\frac{1}{2},j}$ and $q^y_{i,j+\frac{1}{2}} = \left\langle \overline{\mathbf{V}}_{i,j+\frac{1}{2}}, \mathbf{F}^y_{i,j+\frac{1}{2}}\right\rangle - \overline{\Psi^y}_{i,j+\frac{1}{2}}$.

Proof. Noting that $\langle \mathbf{V}_{i,j}, \boldsymbol{S}_{i,j}\rangle = 0$, the proof of theorem follows on the same line as that of Theorem 5.1.1, by taking the scalar product of (7.11a) with $\mathbf{V}_{i,j}$. □

7.2.1 An energy preserving flux

We construct a numerical flux which satisfies the algebraic relations (7.13). We give details for the flux in the x-direction, while similar arguments can be used to construct the y-flux. Furthermore, we omit the indices $i + \frac{1}{2}, j$ for ease of notation.

Let the x-flux be denoted by $\mathbf{F}^x := (F^{x,u_1}, F^{x,u_2}, F^{x,h})^\top$ and consider the algebraic relation $\langle \Delta\mathbf{V}, \mathbf{F}^x\rangle = \Delta\Psi^x$. The potential function given in (7.12) evaluates out to be $\Psi^x(\mathbf{U}) = (gH + \mathcal{K})hu_1$. We compare the coefficients of jumps in the variables $\mathbf{U}$ in the algebraic relation, to find the expression for the components of $\mathbf{F}^x$. We also consider the jump in h_s since the height of the bottom topography need not be constant in space. We have

$$\Delta\mathbf{V} = \begin{pmatrix} \overline{h}\Delta u_1 + \overline{u_1}\Delta h \\ \overline{h}\Delta u_2 + \overline{u_2}\Delta h \\ \Delta\mathcal{K} + g\Delta H \end{pmatrix}, \qquad \Delta\mathcal{K} = \overline{u_1}\Delta u_1 + \overline{u_2}\Delta u_2, \qquad \Delta H = \Delta h + \Delta h_s. \tag{7.15}$$

Thus,

$$\begin{aligned} \langle \Delta\mathbf{V}, \mathbf{F}^x\rangle = (\overline{h}F^{x,u_1} + \overline{u_1}F^{x,h})\Delta u_1 + (\overline{h}F^{x,u_2} + \overline{u_2}F^{x,h})\Delta v_2 \\ + (\overline{u_1}F^{x,u_1} + \overline{u_2}F^{x,u_2} + gF^{x,h})\Delta h + (gF^{x,h})\Delta h_s. \end{aligned} \tag{7.16}$$

Also,

$$\begin{aligned} \Delta\Psi^x &= (\overline{H}g + \overline{\mathcal{K}})(\overline{h}\Delta u_1 + \overline{u_1}\Delta h) + \overline{hu_1}(g\Delta H + \Delta\mathcal{K}) \\ &= [(\overline{H}g + \overline{\mathcal{K}})\overline{h} + \overline{hu_1}\overline{u_1}]\Delta u_1 + (\overline{hu_1}\overline{u_2})\Delta u_2 \\ &\quad + [(\overline{H}g + \overline{\mathcal{K}})\overline{u_1} + \overline{hu_1}g]\Delta h + [\overline{hu_1}g]\Delta h_s. \end{aligned} \tag{7.17}$$

Comparing jump coefficients in (7.16) and (7.17), we get the following set of equations

$$\begin{aligned}
\overline{h}F^{x,u_1} + \overline{u_1}F^{x,h} &= (\overline{H}g + \overline{\mathcal{K}})\overline{h} + \overline{hu_1}\overline{u_1},\\
\overline{h}F^{x,u_2} + \overline{u_2}F^{x,h} &= \overline{hu_1}\overline{u_2},\\
\overline{u_1}F^{x,u_1} + \overline{u_2}F^{x,u_2} + gF^{x,h} &= (\overline{H}g + \overline{\mathcal{K}})\overline{u_1} + \overline{hu_1}g,\\
gF^{x,h} &= \overline{hu_1}g.
\end{aligned}$$

The solution of this system leads to the following expression for an energy preserving flux

$$\mathbf{F}^x_{i+\frac{1}{2},j} = \begin{pmatrix} \overline{\mathcal{K}} + g\overline{H} \\ 0 \\ \overline{hu_1} \end{pmatrix}_{i+\frac{1}{2},j}. \tag{7.18}$$

A similar argument leads to the expression

$$\mathbf{F}^y_{i,j+\frac{1}{2}} = \begin{pmatrix} 0 \\ \overline{\mathcal{K}} + g\overline{H} \\ \overline{hu_2} \end{pmatrix}_{i,j+\frac{1}{2}}. \tag{7.19}$$

Note that (7.18) and (7.19) are the arithmetic average of the exact flux across the cell-interfaces, and thus second-order accurate. High-order energy preserving fluxes can be obtained using the interpolation formula (5.14). In particular, the formula (5.15) leads to a fourth-order accurate energy preserving scheme.

7.2.2 WENO-5 flux

Since $\langle \mathbf{V}_{i,j}, \boldsymbol{S}_{i,j}\rangle = 0$, ϑ does not directly effect the evolution of total energy. However, $\vartheta_{i,j}$ appears in the source term (7.11a). If $\vartheta_{i,j}$ is not evaluated with high-order accuracy, the overall accuracy in the evolution of $\mathbf{U}_{i,j}$ will deteriorate, which will in-turn influence the evolution of energy. Thus, a finite difference WENO-5 approach with flux splitting (see Section 4.2) is used to approximate the conservative flux differences in the scheme (7.11b) for evolving ϑ.

We treat $\mathbf{u}$ as a given constant advection velocity in each cell for ϑ, and consider the following Lax-Friedrichs flux splitting

$$\begin{aligned}
\left(f_1^\vartheta\right)^{\mathscr{U}}(\vartheta, \mathbf{u}) &= \frac{1}{2}\left(f_1^\vartheta(\vartheta) + \alpha^x\vartheta\right), \quad \left(f_1^\vartheta\right)^{\mathscr{D}}(\vartheta, \mathbf{u}) = \frac{1}{2}\left(f_1^\vartheta(\vartheta) - \alpha^x\vartheta\right),\\
\left(f_2^\vartheta\right)^{\mathscr{U}}(\vartheta, \mathbf{u}) &= \frac{1}{2}\left(f_2^\vartheta(\vartheta) + \alpha^y\vartheta\right), \quad \left(f_2^\vartheta\right)^{\mathscr{D}}(\vartheta, \mathbf{u}) = \frac{1}{2}\left(f_2^\vartheta(\vartheta) - \alpha^y\vartheta\right),
\end{aligned} \tag{7.20}$$

where we choose $\alpha^x = \max_{i,j}\{|(u_1)_{i,j}|\}$, $\alpha^y = \max_{i,j}\{|(u_2)_{i,j}|\}$.

We briefly describe the WENO-5 reconstruction algorithm used to approximate the flux in the x-direction, while the flux in the y-direction can be approximated in a similar manner. We begin by considering the upwind flux along the x-direction. For simplicity of notation, for a fixed j we denote $\phi_i = (f_1^\vartheta)^{\mathscr{U}}(\vartheta_i, \mathbf{u}_i)$, where the j index has been suppressed. We wish to find the left approximation $\phi^-_{i+\frac{1}{2}}$ at the interface $x_{i+\frac{1}{2}}$, which will require the cell values $\{\phi_{i-2}, ..., \phi_{i+2}\}$. The three third-order approximations at the interface $i+\frac{1}{2}$, obtained from quadratic polynomial reconstruction in the cell centered at x_i, are given by

$$\begin{aligned}
\phi^{(0),-}_{i+\frac{1}{2}} &= \frac{1}{3}\phi_i + \frac{5}{6}\phi_{i+1} - \frac{1}{6}\phi_{i+2},\\
\phi^{(1),-}_{i+\frac{1}{2}} &= -\frac{1}{6}\phi_{i-1} + \frac{5}{6}\phi_i + \frac{1}{3}\phi_{i+1},\\
\phi^{(2),-}_{i+\frac{1}{2}} &= \frac{1}{3}\phi_{i-2} - \frac{7}{6}\phi_{i-1} + \frac{11}{6}\phi_i.
\end{aligned}$$

Then the fifth-order WENO approximation is given by

$$(F^{x,\vartheta})^{\mathscr{U}}_{i+\frac{1}{2}} = \phi^-_{i+\frac{1}{2}} = w_0\phi^{(0),-}_{i+\frac{1}{2}} + w_1\phi^{(1),-}_{i+\frac{1}{2}} + w_2\phi^{(2),-}_{i+\frac{1}{2}},$$

where the WENO weights are evaluated as described in [106]

$$w_r = \frac{\widetilde{w}_r}{\widetilde{w}_0 + \widetilde{w}_1 + \widetilde{w}_2}, \quad \widetilde{w}_r = \frac{d_r}{(\beta_r + \epsilon)^2}, \quad r = 0, 1, 2,$$

with $d_0 = 3/10$, $d_1 = 3/5$, $d_2 = 1/10$. We take $\epsilon = 10^{-6}$ to ensure the denominator in the expression for $\widetilde{w}_r$ does not vanish. The smoothness indicators β_r are given by

$$\begin{aligned}
\beta_0 &= \frac{13}{12}\left(\phi_i - 2\phi_{i+1} + \phi_{i+2}\right)^2 + \frac{1}{4}\left(3\phi_i - 4\phi_{i+1} + \phi_{i+2}\right)^2,\\
\beta_1 &= \frac{13}{12}\left(\phi_{i-1} - 2\phi_i + \phi_{i+1}\right)^2 + \frac{1}{4}\left(\phi_{i-1} - \phi_{i+1}\right)^2,\\
\beta_2 &= \frac{13}{12}\left(\phi_{i-2} - 2\phi_{i-1} + \phi_i\right)^2 + \frac{1}{4}\left(\phi_{i-2} - 4\phi_{i-1} + 3\phi_{i+1}\right)^2.
\end{aligned}$$

The downward flux needs to be approximated from the right at the interface $x_{i+\frac{1}{2}}$. This is achieved by mirroring the solution about the interface $x_{i+\frac{1}{2}}$, and finding the approximation from the left. In other words, we set $\phi_{i+k} = (f_1^{\vartheta})^{\mathscr{D}}(\vartheta_{i-k+1}, \mathbf{u}_{i-k+1})$ and repeat the above algorithm to get $(F^{x,\vartheta})^{\mathscr{D}}_{i+\frac{1}{2}} = \phi^-_{i+\frac{1}{2}}$. The final flux at the interface $x_{i+\frac{1}{2}}$ is given by

$$(F^{x,\vartheta})_{i+\frac{1}{2}} = (F^{x,\vartheta})^{\mathscr{U}}_{i+\frac{1}{2}} + (F^{x,\vartheta})^{\mathscr{D}}_{i+\frac{1}{2}}.$$

7.3 Numerical results

We now present some numerical results for the shallow water equations. We use a fourth-order energy preserving flux formed using (7.18), (7.19) and the interpolation formula (5.15). The flux for absolute vorticity is approximated with the WENO-5 flux splitting approach described in Section 7.2.2. The combined scheme is termed as the *VI-EP4 scheme*. The time integration is performed using a fourth-order Runge-Kutta scheme, with the time step evaluated based on the convection speedem

$$\Delta t = \frac{\text{CFL} \times \Delta x}{\lambda}, \qquad \lambda = \max_{i,j}\left\{|\mathbf{u}_{i,j}| + \sqrt{gh_{i,j}}\right\}.$$

with CFL chosen as a non-negative number less than unity. We may alternatively choose a constant Δt, while ensuring that the effective CFL < 1. Furthermore, we impose periodic boundary conditions at the domain boundaries.

7.3.1 Advecting vortex

We consider a vortex advection problem obtained by adapting the isentropic vortex advection problem for the Euler equations to the shallow water equations [125]. The initial conditions of the flow on the domain $[-40, 40] \times [-40, 40]$ are prescribed as

$$u_1 = M\cos(\alpha) + c_1(y - y_c)\exp\left(-c_2 r^2\right)\big), \qquad u_2 = M\sin(\alpha) - c_1(x - x_c)\exp\left(-c_2 r^2\right)\big),$$

$$H = 1 - \frac{c_1^2}{4c_2 g}\exp\left(-2c_2 r^2\right)\big), \qquad h = H - h_s, \qquad r = \sqrt{(x - x_c)^2 + (y - y_c)^2},$$

with the parameters $c_1 = 0.04$, $c_2 = 0.02$, $g = 1$ and $\alpha = 0°$. The remaining parameters, i.e., the initial center of the vortex (x_c, y_c), the free-stream velocity parameter M, the bottom topography h_s and the Coriolis term $\mathscr{F}$, are varied depending on the test case being considered.

Accuracy of EP4 scheme

We test the accuracy of the VI-EP4 scheme by setting $(x_c, y_c) = (0, 0)$, $M = 1$, $h_s = 0$ and $\mathscr{F} = 0$. The solution is simulated till final time $t_f = 80$ with CFL=0.5, at the end of which the vortex completes one complete cycle and returns to its original position. Note that the the absolute vorticity for this problem is equal to the vorticity, since the Coriolis terms are not present. Table 7.1 shows that the convergence rate of ϑ is fifth-order, which is expected since ϑ is evolved using a WENO-5 method. Reaping the benefits of a fifth-order approximation of ϑ, which appears in the source term of (7.11a), the components of $\mathbf{U}$ converge at a rate between fourth and fifth-order, despite being evolved by a fourth-order accurate scheme.

	u_1				u_2			
N	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
50	1.43e-00	-	5.31e-03	-	1.87e-00	-	9.59e-03	-
100	5.05e-02	4.83	3.30e-04	4.01	6.84e-02	4.77	4.69e-04	4.35
200	1.63e-03	4.95	1.11e-05	4.90	2.27e-03	4.91	1.60e-05	4.87
300	2.23e-04	4.91	1.40e-06	5.11	3.14e-04	4.88	2.11e-06	4.99
400	5.62e-05	4.79	3.12e-07	5.21	7.96e-05	4.77	4.99e-07	5.01

	h				ϑ			
N	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
50	7.59e-01	-	2.11e-03	-	6.77e-01	-	6.14e-03	-
100	2.78e-02	4.77	9.14e-05	4.53	3.55e-02	4.25	3.04e-04	4.33
200	1.13e-03	4.62	3.55e-06	4.68	1.14e-03	4.96	1.18e-05	4.68
300	1.88e-04	4.42	5.57e-07	4.57	1.34e-04	5.27	1.65e-06	4.85
400	5.47e-05	4.29	1.55e-07	4.42	2.96e-05	5.26	3.99e-07	4.95

Table 7.1: Accuracy of VI-EP4 scheme for advecting vortex problem with a flat topography and no Coriolis effects. N corresponds to the number of cells in each coordinate direction.

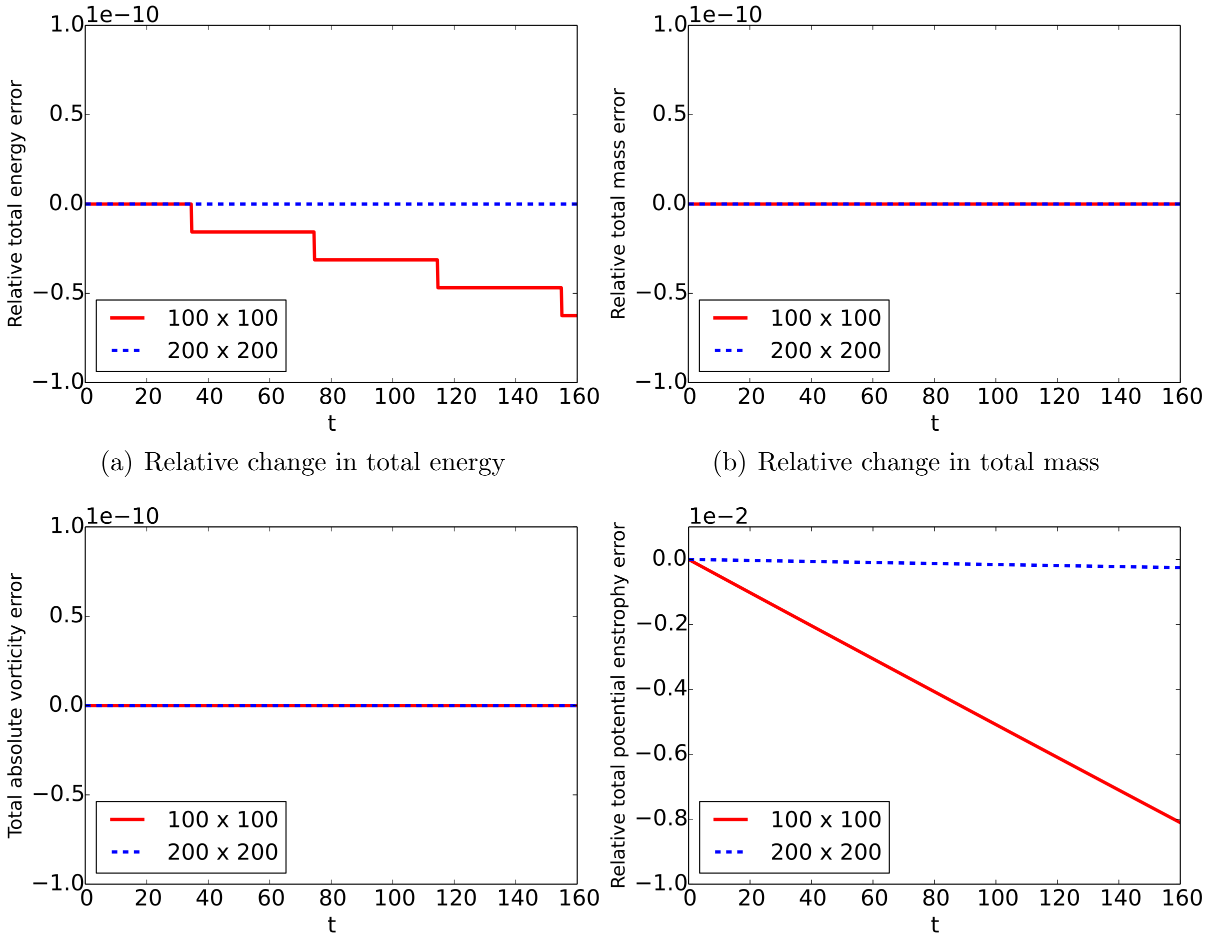
Preservation of invariant quantities

Although VI-EP4 has been designed to conserve the total energy (at the semi-discrete level), we would like to test its ability to preserve other invariants discussed in Section 7.1.1. In particular, we would like to observe the discrete evolution of total mass, total absolute vorticity and total potential enstrophy. We set $(x_c, y_c) = (0, 0)$, $h_s = 0$, $M = 1$, $\mathscr{F} = 0$ and choose a final time of $t_f = 160$ with CFL=0.5. For $M = 1$, the vortex completes two full cycles at the final time. We discretize the domain with 100 cells in each direction, which ensures that there about 16 cells spanning the diameter of the vortex. In Figure 7.1, we plot the changes in the integral quantities from their initial values. These differences are scaled by their values at $t = 0$, except for the total absolute vorticity which is almost zero ($\sum \vartheta_0 \approx -1.02e - 12$). The VI-EP4 scheme is able to preserve total mass and total absolute vorticity quite well till the final time, with their relative differences being kept at zero. The relative difference in total energy deviates from zero by an order of 10^{-11}, indicating a minor decay in energy. This in not surprising as energy is known to be dissipated or generated depending on the time-marching scheme used to integrate the semi-discrete scheme [116] (also see Chapter 8). Unfortunately, there is a significant decay in the total enstrophy. By doubling the number of cells in each direction i.e., $N = 200$, the preservation of total enstrophy is greatly improved, as is the preservation of total energy.

7.3.2 Effect of Coriolis force

Inclusion of the effects Coriolis forces can substantially change the solution structure for the shallow water equations. To see this, we set $M = 0$ and discretize the domain with $N = 200$ cells in each direction, with a final time $t_f = 40$. A constant time step $\Delta t = 0.2$ is chosen, which corresponds to an effective CFL ≈ 0.56. We first consider the case when there is no Coriolis forces active, i.e., $\mathscr{F} = 0$. In this case, the vortex continues to be centered at the origin with a steady clockwise spin, as shown in Figure 7.2. However, on choosing $\mathscr{F} = 1$, the vortex remains centered at the origin, but periodically changes its direction of rotation, as can be seen in Figure 7.3. The degree of change in vortex rotation depends on the strength of the Coriolis forces. For smaller values of $\mathscr{F}$, say $\mathscr{F} = 10^{-1}$, we have observed (not presented here) that the direction of rotation does not change but the strength of the spin oscillates.

The evolution of relative change in invariants for $\mathscr{F} = 1$ are depicted in Figure 7.4. Note that we can now plot the relative change in total absolute vorticity, since the initial value total absolute vorticity is no longer close to zero with the inclusion of a non-zero $\mathscr{F}$. The total mass and absolute vorticity are preserved by VI-EP4. The total energy shows a small decay, while the relative change in total enstrophy oscillates close to zero with an amplitude of order 10^{-9}. If we reduce the time step to $\Delta t = 0.05$, we can see from Figure 7.5 that the conservation of total energy is drastically improved. This clearly shows that the loss of energy can be attributed to the time integration scheme. However, their seems to be no noticeable improvement in the conservation of total enstrophy. On the other hand, if we double the number of cells in each direction and work with the original time step of $\Delta t = 0.1$, we observe that while there is no improvement in the conservation of total energy, the amplitude of oscillations in the relative change in total enstrophy reduces by a factor of 10.



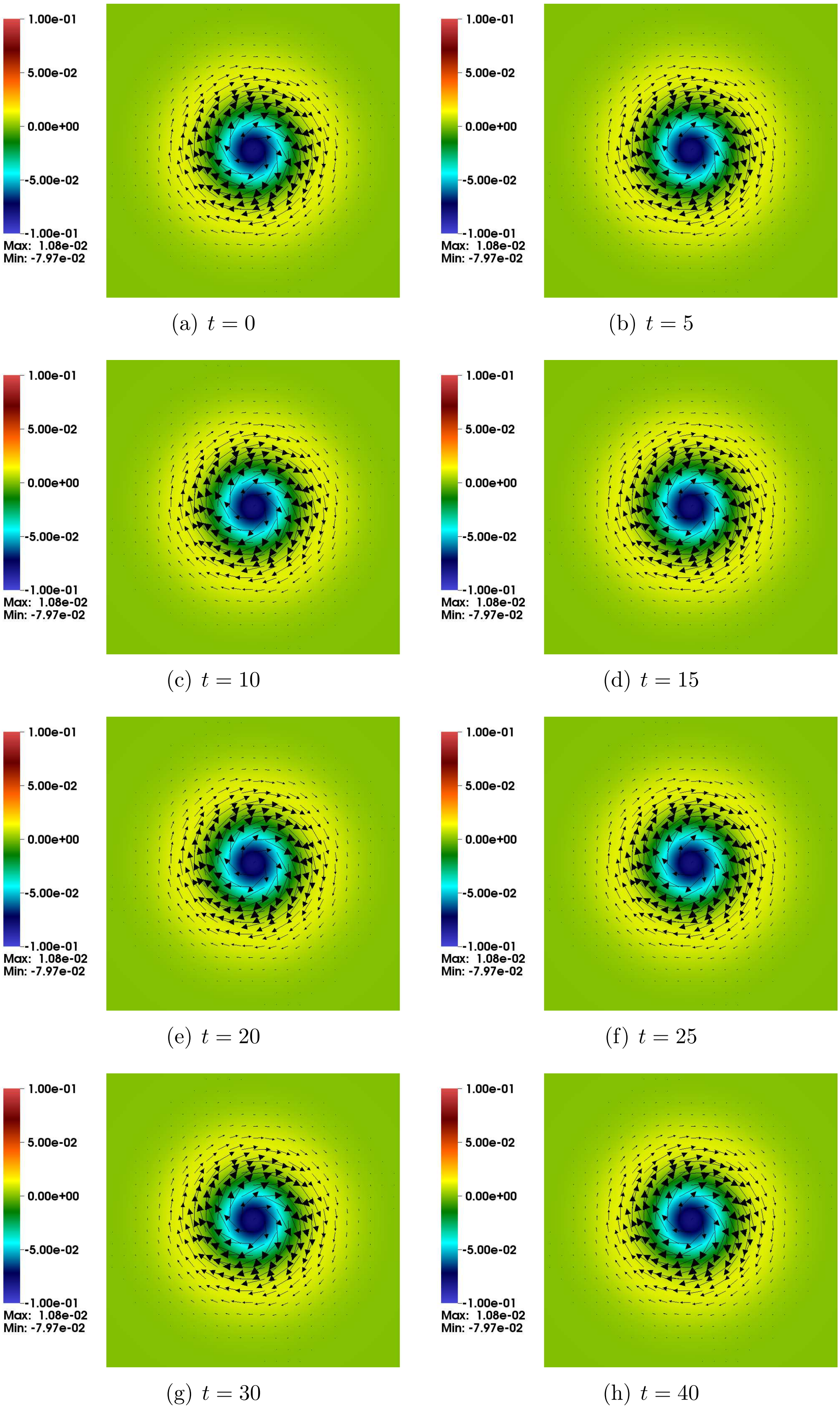
(a) Relative change in total energy

(b) Relative change in total mass

(c) Change in absolute vorticity

(d) Relative change in total enstrophy

Figure 7.1: Evolution of invariants for advecting vortex with $M = 1$ with VI-EP4.



(a) $t = 0$

(b) $t = 5$

(c) $t = 10$

(d) $t = 15$

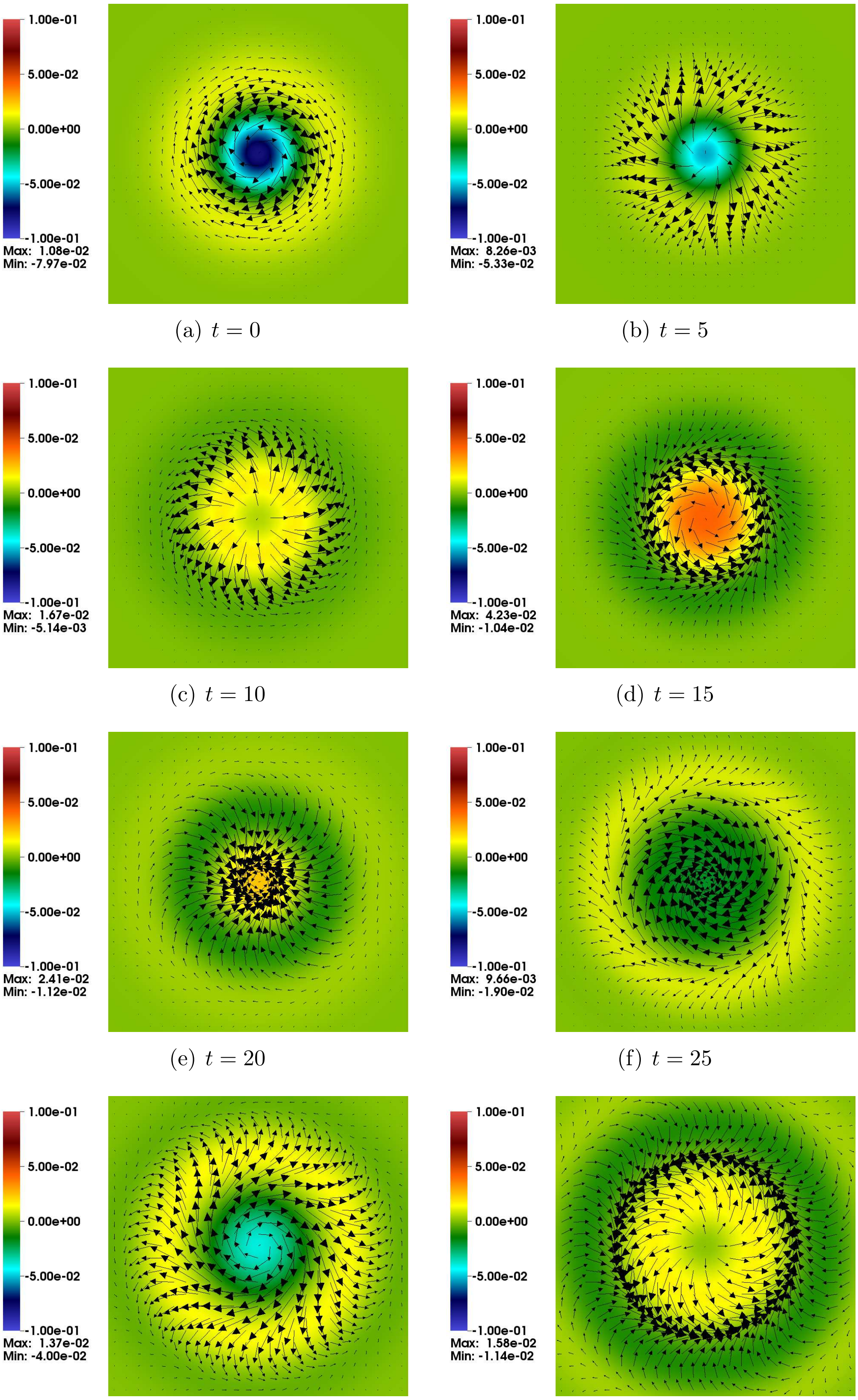
(e) $t = 20$

(f) $t = 25$

(g) $t = 30$

(h) $t = 40$

Figure 7.2: Stationary vortex solution with $\mathscr{F} = 0$. The color plots represent the magnitude of vorticity ω, while the vectors describe the velocity field.



(a) $t = 0$

(b) $t = 5$

(c) $t = 10$

(d) $t = 15$

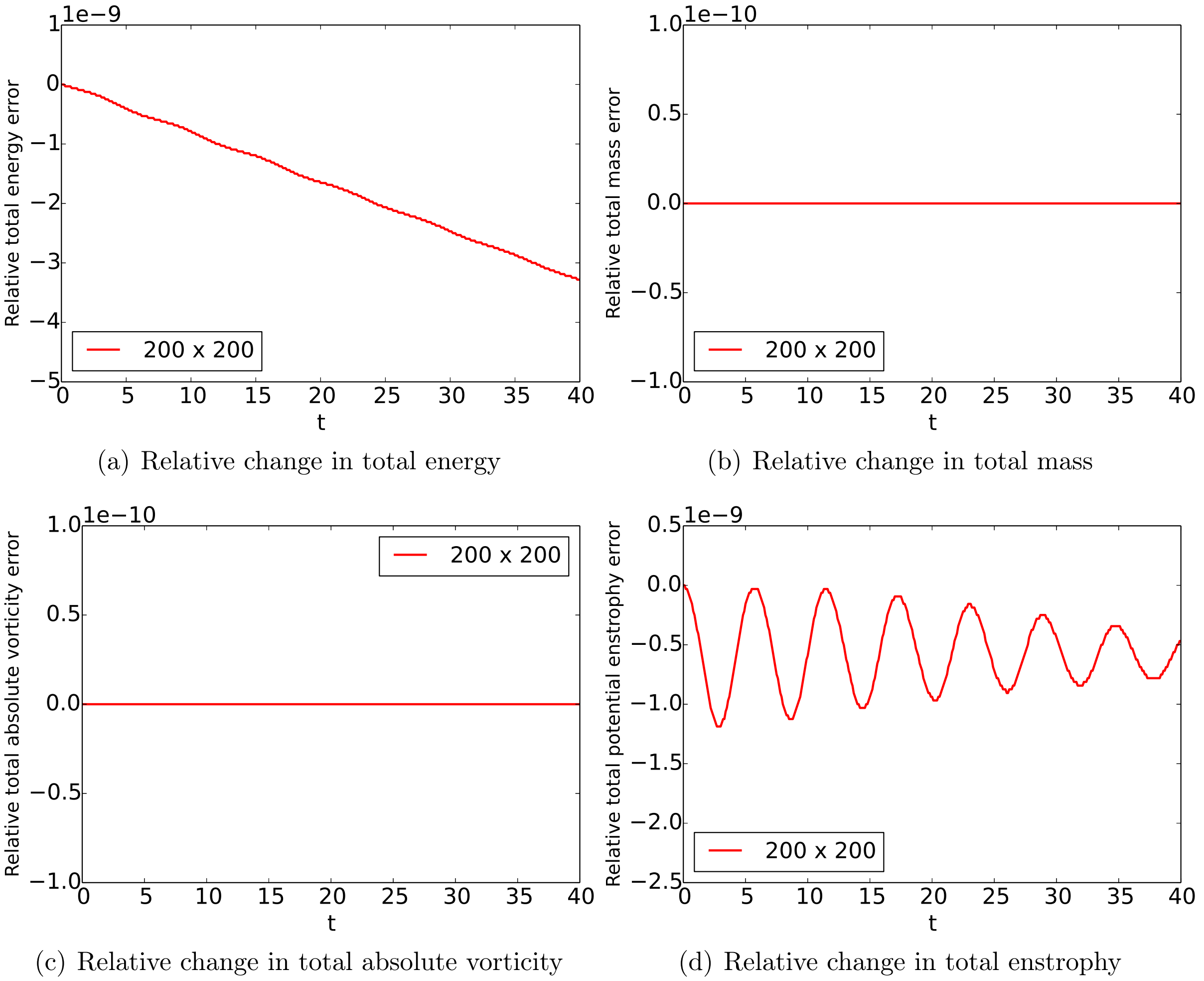
(e) $t = 20$

(f) $t = 25$

(g) $t = 30$

(h) $t = 40$

Figure 7.3: Stationary vortex solution with $\mathscr{F} = 1$. The color plots represent the magnitude of vorticity ω, while the vectors describe the velocity field.



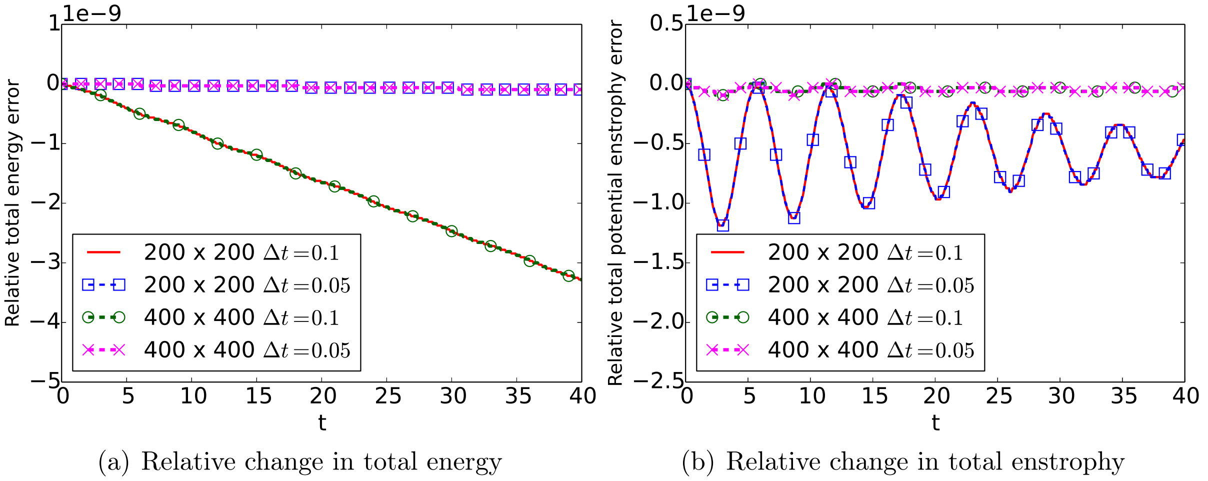
(a) Relative change in total energy

(b) Relative change in total mass

(c) Relative change in total absolute vorticity

(d) Relative change in total enstrophy

Figure 7.4: Evolution of relative change in invariants for vortex with $M = 0$ and $\mathscr{F} = 1$, with $\Delta t = 0.1$.



(a) Relative change in total energy

(b) Relative change in total enstrophy

Figure 7.5: Evolution of relative change in total energy and total enstrophy for vortex with $M = 0$ and $\mathscr{F} = 1$ on two meshes, with different values of Δt.

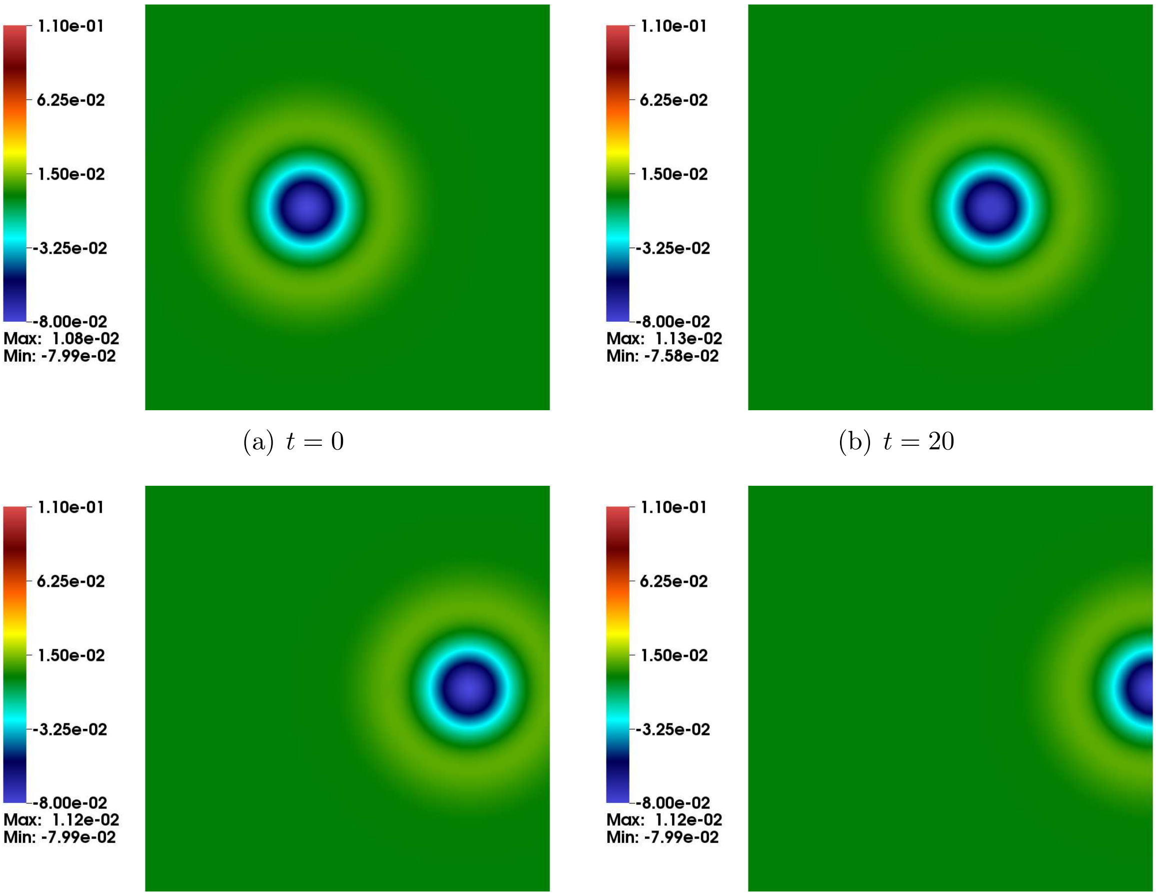
Non-flat bottom topography

The test problems considered above assumed a flat bottom topography. We now consider a hump bottom topography given by

$$h_s(x,y) = h_0 \exp\left(-r_0^2\right), \quad r_0 = \sqrt{(x-x_0)^2 + (y-y_0)^2}, \tag{7.21}$$

where $h_0 = 0.05$ represents the peak height of the bottom surface, with the hump centered at $(x_0, y_0) = (5, 0)$. The vortex parameters are chosen as $(x_c, y_c) = (-5, 0)$, $M = 0.5$ and $\mathscr{F} = 0$. The domain is discretized using 200 cells in each direction, with a final time of $t_f = 60$ and CFL=0.5.

The vortex is able to pass over the hump without losing its structure, as shown in Figure 7.6. The background field of the solution interacts with the hump and generates weak wave fronts scattering through the domain, which simply pass through the advecting vortex, as can been seen in Figure 7.7. As was observed in the previous test cases, the total energy and total potential enstrophy deviate slightly from the expected constant values (see Figure 7.8). However, the results improve significantly on refining the mesh.



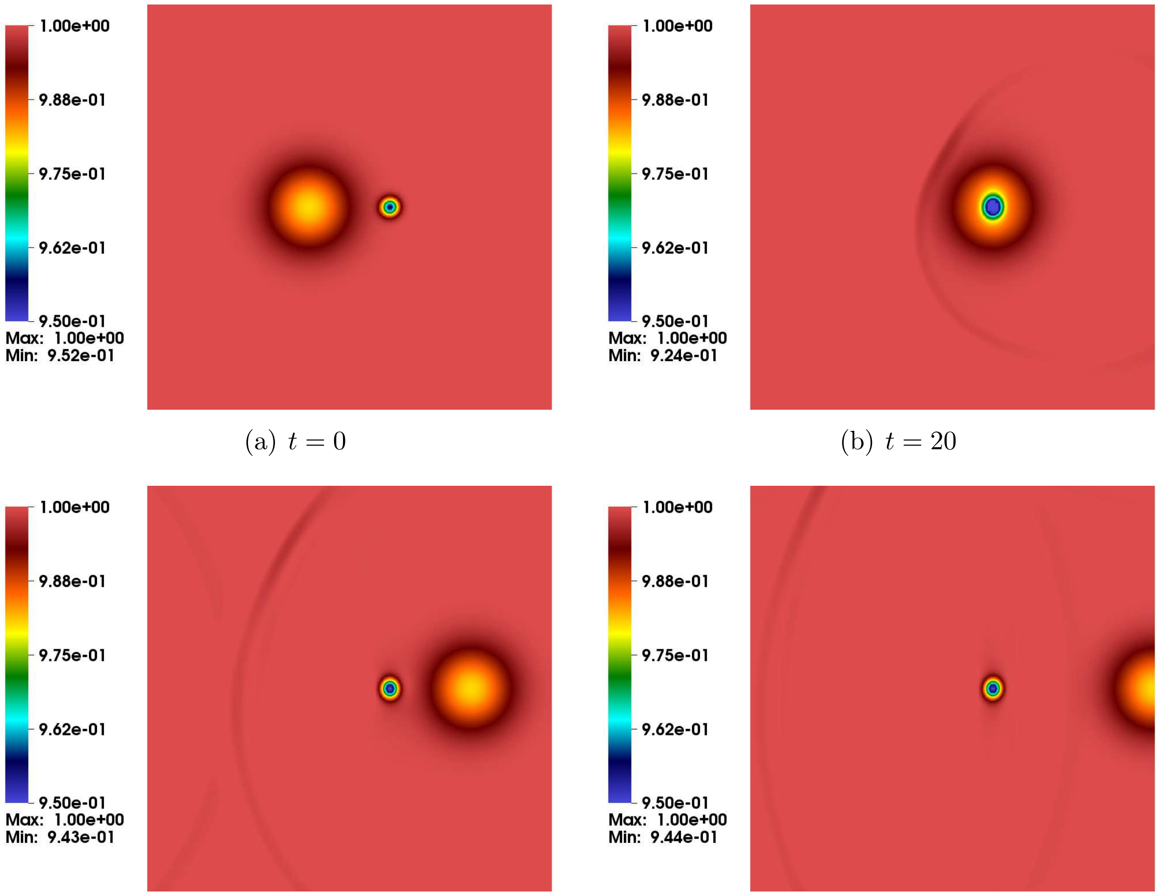
(a) $t = 0$

(b) $t = 20$

(c) $t = 40$

(d) $t = 60$

Figure 7.6: Absolute vorticity plots for the advecting vortex solution with non-flat surface. The plot is zoomed to $[-20, 20]^2$.



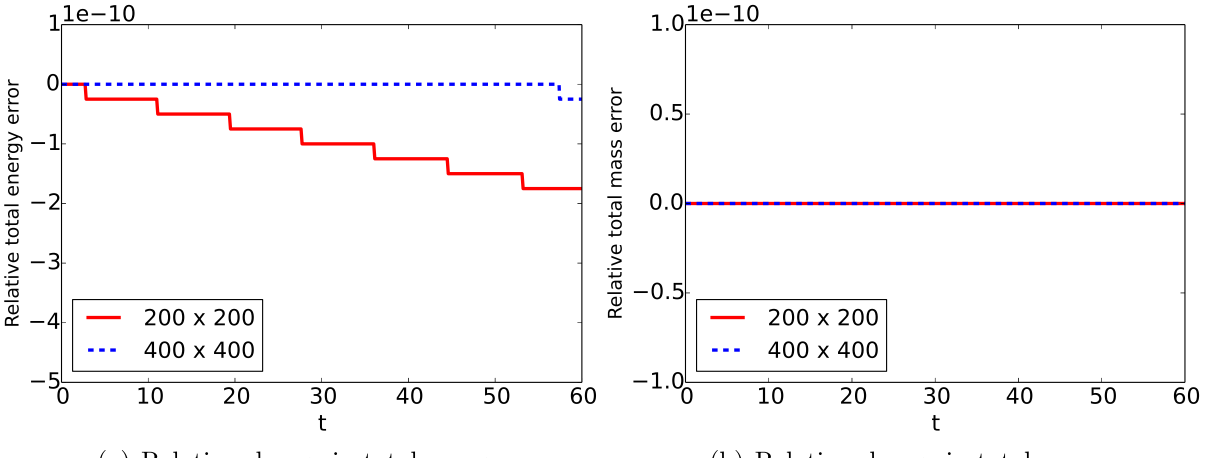
(a) $t = 0$

(b) $t = 20$

(c) $t = 40$

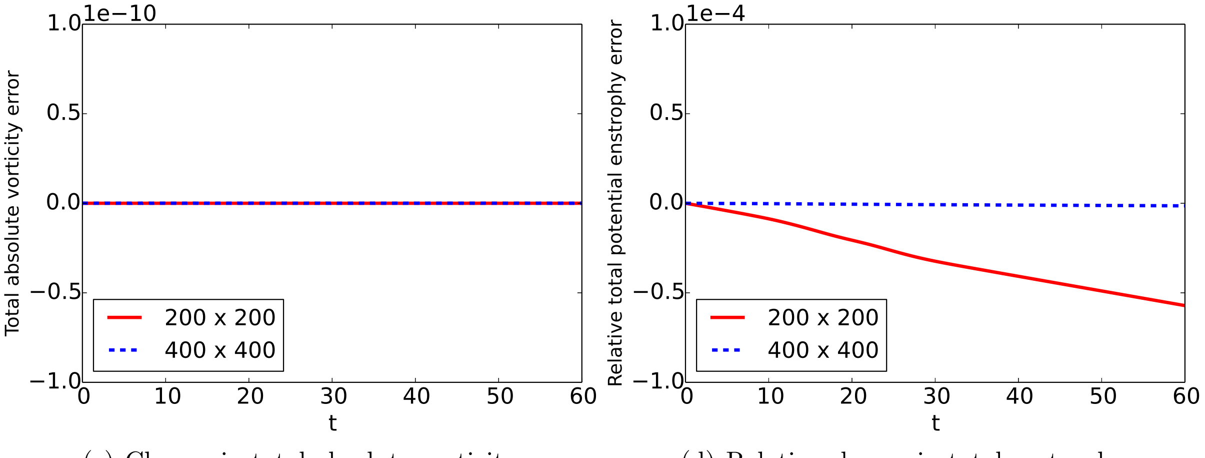
(d) $t = 60$

Figure 7.7: Height h plots for the advecting vortex solution with non-flat surface



(a) Relative change in total energy

(b) Relative change in total mass



(c) Change in total absolute vorticity

(d) Relative change in total enstrophy

Figure 7.8: Evolution of invariants for the advecting vortex solution with non-flat surface.

7.3.3 Perturbed geostrophic balance

We now test the robustness of VI-EP4 to approximate more complex solutions. We consider two test cases, in which we start with a solution of the rotating shallow water equations which is in geostrophic balance i.e., the solutions are steady in time. We work with solutions which are not stable. In other words, the solution profile eventually diverges from steady state when a small perturbation is introduced into the initial profile.

Roll-up of vorticity

We consider the domain $[-5,5]\times[-5,5]$ and choose

$$h=\widetilde{h}(y)=1+0.1\left[\tanh\left(\frac{1-y^2}{\epsilon}\right)+1\right], \tag{7.22}$$

with $\epsilon=0.5$. We assume a flat bottom topography ($h_s\equiv 0$) and choose a non-zero constant Coriolis term $\mathscr{F}=1$. The initial flow is assume to be purely zonal in the x-direction, with $u_2\equiv 0$. The the remaining variables describing geostrophic balance can be obtained as

$$u_1(y)=-\frac{g}{\mathscr{F}}\partial_y\widetilde{h}(y),\qquad \omega=\frac{g}{\mathscr{F}}\partial_{yy}\widetilde{h}(y), \tag{7.23}$$

where we set $g=1$.

The steady state solution is perturbed by introducing a small "mountain-like" perturbation to the height

$$h(x,y)=\widetilde{h}(y)+h_\delta\exp\left(-r^2\right),\quad r=\sqrt{x^2+y^2}$$

where h_δ determines the peak height of the perturbation. We choose $h_\delta=0.005$ and simulate the solution till a final time of $t_f=50$ with CFL=0.1.

We also compare the solutions obtained with the VI-EP4 scheme for the vector-invariant model, to those obtained by numerical schemes formulated for the more commonly used conservative form of the shallow water equations with rotation. In other words, we consider the model

$$\partial_t\mathbf{U}+\partial_x\mathbf{f}_1(\mathbf{U})+\partial_y\mathbf{f}_2(\mathbf{U})+\boldsymbol{S}=0, \tag{7.24}$$

where

$$\mathbf{U}=\begin{pmatrix}hu_1\\hu_2\\h\end{pmatrix},\qquad \mathbf{f}_1(\mathbf{U})=\begin{pmatrix}hu_1^2+\frac{1}{2}gh^2\\hu_1u_2\\hu_1\end{pmatrix},\qquad \mathbf{f}_2(\mathbf{U})=\begin{pmatrix}hu_1u_2\\hu_2^2+\frac{1}{2}gh^2\\hu_2\end{pmatrix}, \tag{7.25}$$

while the source term is $\boldsymbol{S}=(-hu_2\mathscr{F},\ hu_1\mathscr{F},\ 0)^\top$.

We work with two important schemes corresponding to the conservative model. The first is an entropy conservative scheme [125], with the energy chosen as the entropy function. The energy/entropy preserving flux is given by

$$\mathbf{F}^x_{i+\frac{1}{2},j}=\begin{pmatrix}\overline{h}\ \overline{u_1}^2+\frac{g}{2}\overline{h^2}\\\overline{h}\ \overline{u_1}\ \overline{u_2}\\\overline{h}\ \overline{u_1}\end{pmatrix}_{i+\frac{1}{2},j},\qquad \mathbf{F}^y_{i,j+\frac{1}{2}}=\begin{pmatrix}\overline{h}\ \overline{u_1}\ \overline{u_2}\\\overline{h}\ \overline{u_2}^2+\frac{g}{2}\overline{h^2}\\\overline{h}\ \overline{u_2}\end{pmatrix}_{i,j+\frac{1}{2}}. \tag{7.26}$$

Using (5.15), we get a fourth-order energy preserving scheme, which we term as *CM-EP4*. The second scheme considered, is a WENO5 finite difference scheme, with the following flux splitting

$$\begin{aligned} (\mathbf{f}_1)^{\mathscr{U}}(\mathbf{U}) &= \frac{1}{2}\left(\mathbf{f}_1(\mathbf{U}) + \alpha^x \mathbf{U}\right), \quad (\mathbf{f}_1)^{\mathscr{D}}(\mathbf{U}) = \frac{1}{2}\left(\mathbf{f}_1(\mathbf{U}) - \alpha^x \mathbf{U}\right), \\ (\mathbf{f}_2)^{\mathscr{U}}(\mathbf{U}) &= \frac{1}{2}\left(\mathbf{f}_2(\mathbf{U}) + \alpha^y \mathbf{U}\right), \quad (\mathbf{f}_2)^{\mathscr{D}}(\mathbf{U}) = \frac{1}{2}\left(\mathbf{f}_2(\mathbf{U}) - \alpha^y \mathbf{U}\right), \end{aligned} \tag{7.27}$$

where we choose $\alpha^x = \max_{i,j}\{|(u_1)_{i,j}| + \sqrt{gh_{i,j}}\}$, $\alpha^y = \max_{i,j}\{|(u_2)_{i,j}| + \sqrt{gh_{i,j}}\}$. This scheme is termed as *CM-WENO5*. Unlike CM-EP4, we cannot a priori prove that CM-WENO5 is energy/entropy preserving.

Since vorticity is not evaluated in the conservative model given by (7.24)-(7.25), it is approximated in each cell from the velocity field by

$$\omega_{i,j} = \partial_x^h (u_2)_{i,j} - \partial_y^h (u_1)_{i,j}.$$

To obtained a $2p$-th order accurate approximation of ω, the following partial derivative approximations of the velocity field are use

$$\partial_y^h (u_1)_{i,j} = \sum_{k=-p}^{p} (\alpha_k)(u_1)_{i,j+k}, \qquad \partial_x^h (u_2)_{i,j} = \sum_{k=-p}^{p} (\alpha_k)(u_2)_{i+k,j}. \tag{7.28}$$

For $p = 2$, the weights in (7.28) are given by

$$\alpha_{-2} = -\alpha_2 = \frac{1}{12}, \quad \alpha_{-1} = -\alpha_1 = -\frac{2}{3}, \quad \alpha_0 = 0. \tag{7.29}$$

The plots for the variables h and $\mathbf{u}$ shown in Figures 7.9 - 7.11 clearly show that the solutions with CM-WENO5 are the smoothest, while those with CM-EP4 are the polluted with small scale oscillations. The solutions with VI-EP4 are comparable to CM-WENO5. Using the 4-th order approximation of ω given by (7.28) and (7.29), leads to a very noisy profile for CM-EP4 (see Figure 7.12). This is expected since the approximated velocity field with CM-EP4 is not regular enough. The vorticity evaluated directly by the VI-EP4 scheme is smooth and comparable to the 4-th order accurate vorticity for CM-WENO5.

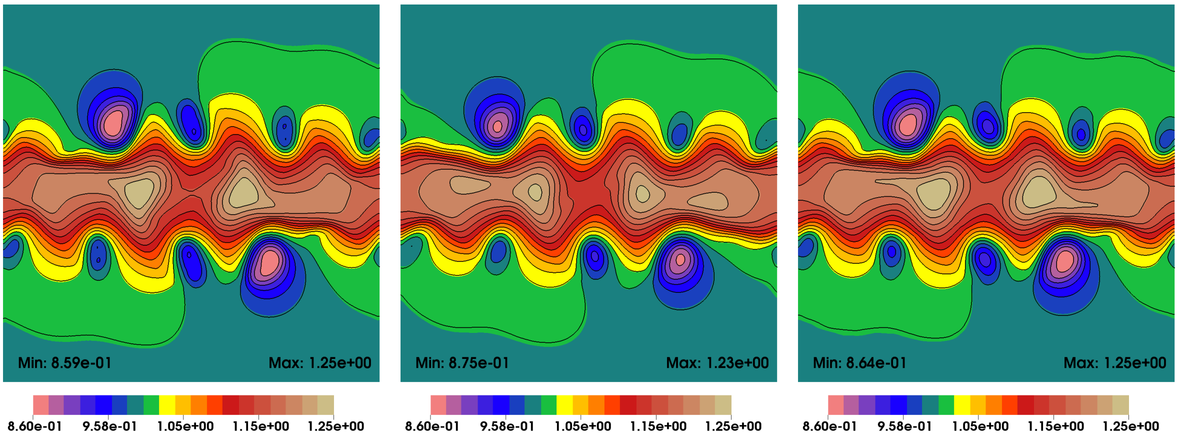
One can also approximate the vorticity from the velocity field obtained with the VI-EP4 scheme. However, Figure 7.13 clearly shows that such approximations can be noisy if the velocity field is not very regular. This demonstrates an advantage of solving for (absolute) vorticity as an independent variable, especially when the numerical scheme is not very dissipative.

Next, we compare the evolution of relative change in total energy and total enstrophy by the schemes. Figure 7.14 shows that CM-WENO5 is the most dissipative of the three schemes, while CM-EP4 and VI-EP4 are both able to preserve the total energy of the flow. The evolution of total enstrophy in Figure 7.15 once again indicates that CM-WENO5 is the most dissipative. At first glance, it seems that CM-EP4 performs the best at preserving total enstrophy. However, recall that enstrophy is evaluated using vorticity, which is poorly approximated by CM-EP4 at later times. Thus, the results shown in 7.15 for CM-EP4 are misleading.

Potential vorticity ϑ/h can also be used as an indicator to judge the performance of numerical schemes. Subtracting ϑ/h times the second equation of (7.8) from the third equation of (7.8) gives the following evolution equation for potential vorticity

$$\partial_t\left(\frac{\vartheta}{h}\right)+u_1\partial_x\left(\frac{\vartheta}{h}\right)+u_2\partial_y\left(\frac{\vartheta}{h}\right)=0. \tag{7.30}$$

Since (7.30) is simply an advection equation, the potential vorticity should satisfy the *maximum principle*. In other words, the minimum and maximum values of potential vorticity should not change in time. We plot the bounds of potential vorticity approximated by the schemes in Figure 7.16. Till time $t = 30$, both VI-EP4 and CM-EP4 are able to preserve the bounds, while CM-WENO5 shows a decrease in the upper bound and an increase in the lower bound. This is expected since CM-WENO5 is the most dissipative. After $t = 30$, the vorticity approximated with the solution from CM-EP4 is oscillatory and thus the bounds of of potential vorticity quickly deviate from its initial values. The VI-EP4 scheme is able to preserve the bounds quite well till the end of the simulation.



(a) CM-EP4

(b) CM-WENO5

(c) VI-EP4

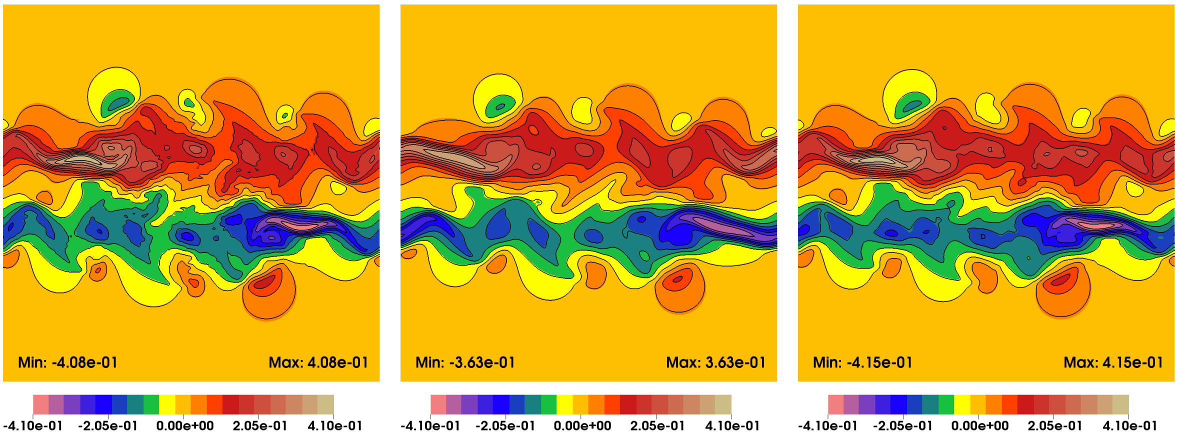
Figure 7.9: Contour and pseudo plots of height at t=50 for roll-up of vorticity test case.

Flow over an isolated mountain

This test case has been taken from [124], and was initially designed for simulation on the sphere (see test case 5 in [134]). The domain is taken as $[-\pi a, \pi a] \times [-\pi a, \pi a]$, where $a = 6.37 \times 10^6$m corresponds to the Earth's radius. Thus, the x- and y-extents of the domain are equal to the Earth's circumference. We start with a geostrophic balance obtained by considering a flat bottom topography and choosing

$$H(y)=\widetilde{h}-\frac{a}{g}\Omega\widetilde{u}\sin^2\left(\frac{y}{a}\right), \quad u_1(y)=\widetilde{u}\cos\left(\frac{y}{a}\right), \quad u_2\equiv 0, \quad \mathscr{F}(y)=2\Omega\sin\left(\frac{y}{a}\right),$$

where $\widetilde{h} = 5960$m, $\widetilde{u} = 20\text{ms}^{-1}$ and $\Omega = 7.292 \times 10^{-5}\text{s}^{-1}$ is the rotation rate of the Earth. Note that unlike the previous test cases, the Coriolis force is a function of space.



(a) CM-EP4 (b) CM-WENO5 (c) VI-EP4

Figure 7.10: Contour and pseudo plots of u_1 at t=50 for roll-up of vorticity test case.

The geostrophic balance is perturbed by suddenly introducing a mountain in the bottom topography

$$h_s(x,y) = \widetilde{h}_s\left(1 - \frac{r}{R}\right), \quad r = \min\left(R, \sqrt{(x-x_c)^2 + (y-y_c)^2}\right),$$

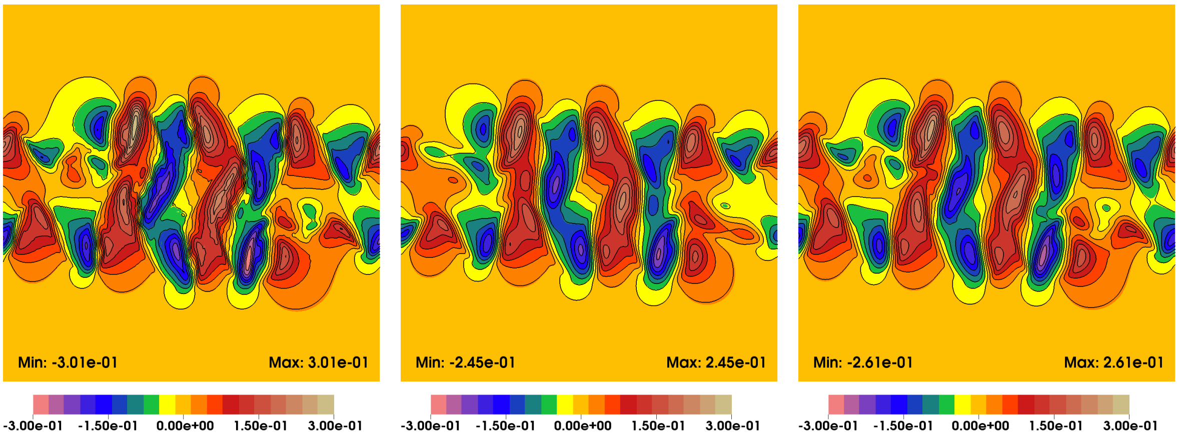
where $\widetilde{h}_s = 2000\text{m}$, $R = \pi a/9$ and the mountain is centered at $(x_c, y_c) = (-\pi a/2, \pi a/6)$. Note that H is still a function of only y, but $h \neq H$ with the introduction of the mountain. To compare with the results in [124], we consider a 400 × 400 mesh with two time steps i) $\Delta t = 6s$ with an effective CFL ≈ 0.016, and ii) $\Delta t = 30s$ with an effective CFL $\approx 0.08s$. Since the scheme proposed in [124] is second-order accurate, we also simulate results on a coarser mesh of size 200 × 200. The vorticity contours at the end of 7 days is shown in Figure 7.17, and are indistinguishable for various meshes and time-steps considered. Furthermore, the results are not polluted by small scale oscillations near the mountain, as is the case in [124].

The relative change in total energy with the VI-EP4 scheme shown in Figure 7.18 is zero up to machine precision error, on both meshes. The relative change in total enstrophy deviates from zero, which improves with mesh refinement. However, there is no change in the results when the time step is reduced from 30s to 6s. Note that the numerical scheme in [124] is able to conserve total enstrophy more accurately, since the scheme has been designed to preserve enstrophy.

We also plot the bounds of the approximated potential vorticity in Figure 7.19. Note that the initial maximum (minimum) values on the two meshes differ by an order of 10^{-11}. However, the values on the courser 200 × 200 mesh are eventually indistinguishable from those on the finer mesh.

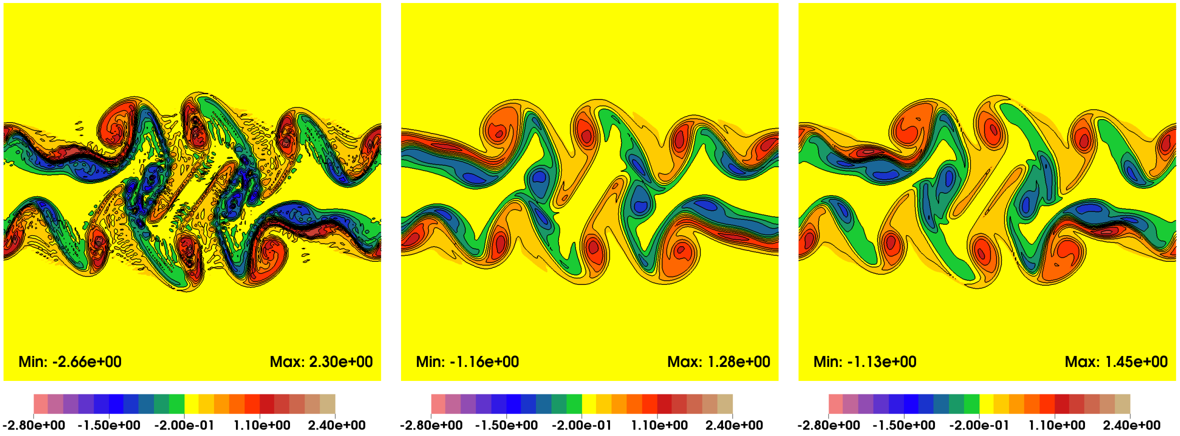
We make the following remarks based on the numerical results discussed in the chapter:

- The results for the VI-EP4 scheme show that it can be advantageous to solve for vorticity as an independent variable.
- The VI-EP4 scheme uses a central flux for evaluating h and $\mathbf{u}$, which may lead Gibbs-oscillations when the solution has steep gradients. This is clearly observed



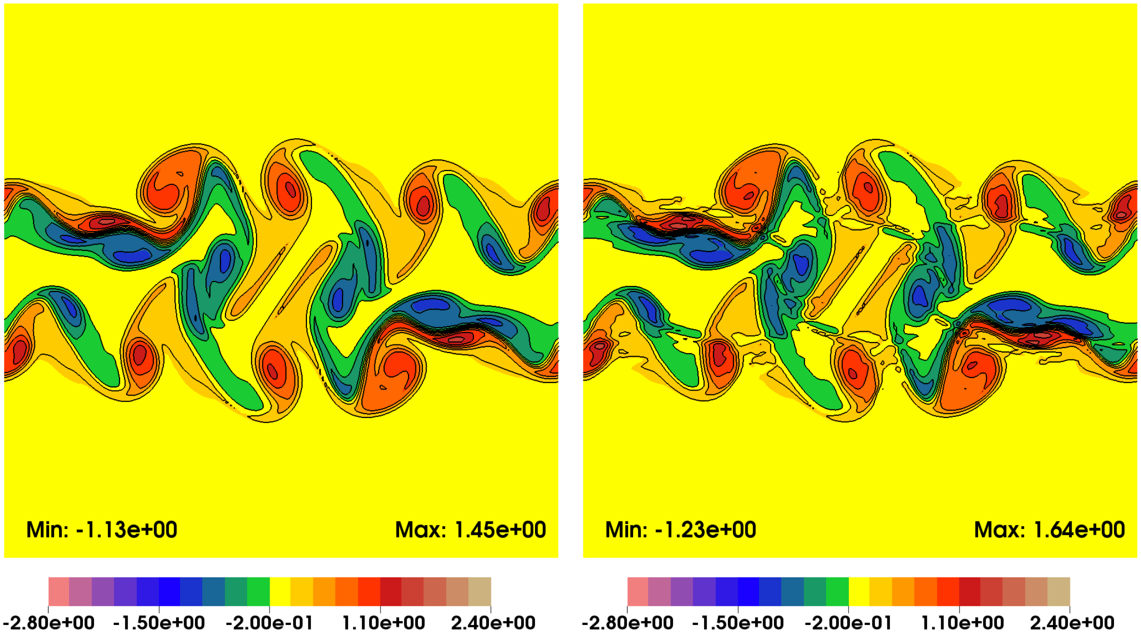
(a) CM-EP4 (b) CM-WENO5 (c) VI-EP4

Figure 7.11: Contour and pseudo plots of u_2 at t=50 for roll-up of vorticity test case.



(a) CM-EP4 (b) CM-WENO5 (c) VI-EP4

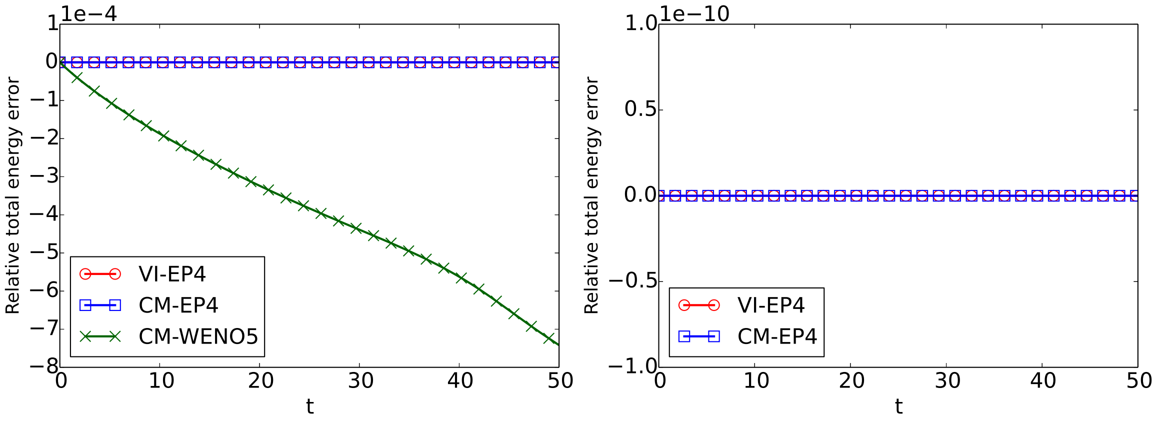
Figure 7.12: Contour and pseudo plots of ω at t=50 for roll-up of vorticity test case. The 4-th order finite difference approximation is used for CM-EP4 and CP-WENO5



(a) ω from scheme

(b) 4-th order ω FD approx.

Figure 7.13: Comparison of ω obtained from the scheme and the 4-th order finite difference approximations from the velocity field for VI-EP4 at t=50, for roll-up of vorticity test case.



(a) Relative change in total energy

(b) Zoomed profile

Figure 7.14: Evolution of relative change in total energy for roll-up of vorticity test case.

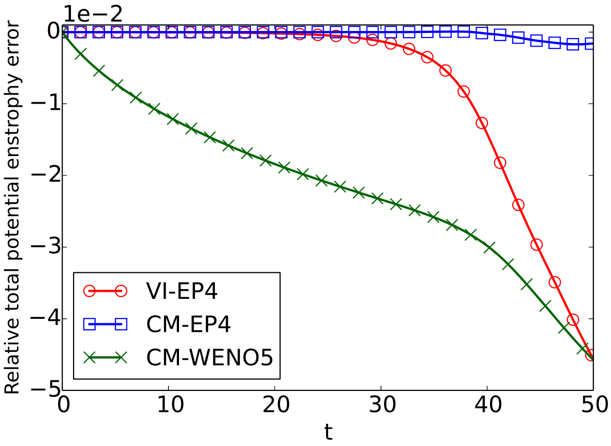
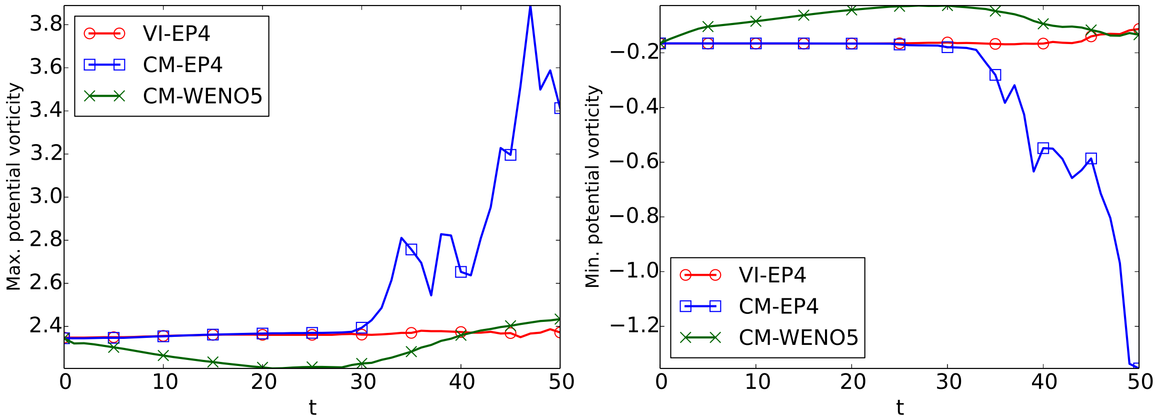


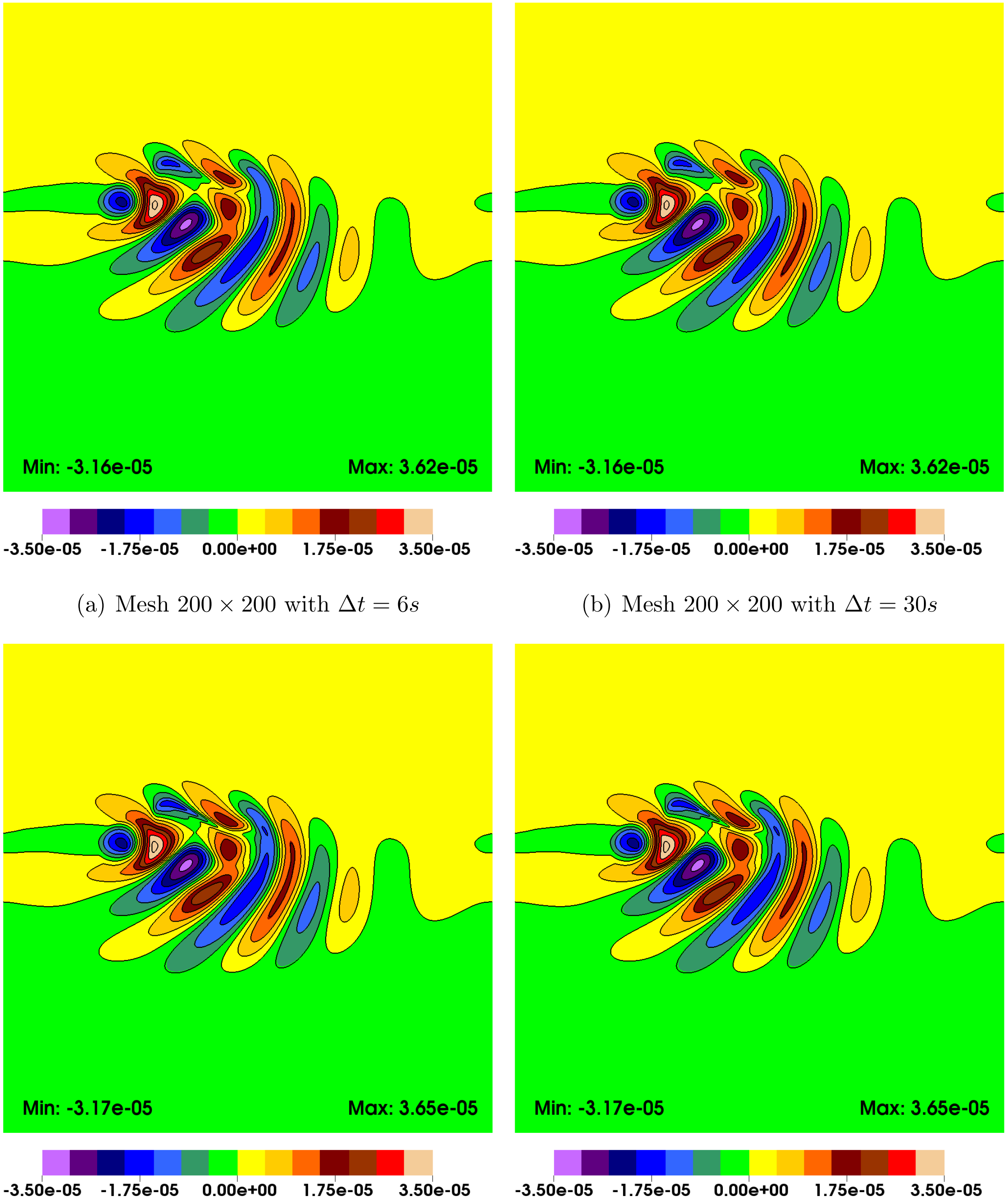
Figure 7.15: Evolution of relative change in total enstrophy for roll-up of vorticity test case. A 4-th order of the finite difference method has been used to approximate ω for CM-EP4 and CM-WENO5.



(a) Maximum value of potential vorticity

(b) Minimum value of potential vorticity

Figure 7.16: Evolution of bounds of potential vorticity for roll-up of vorticity test case. A 4-th order of the finite difference method has been used to approximate ω for CM-EP4 and CM-WENO5.



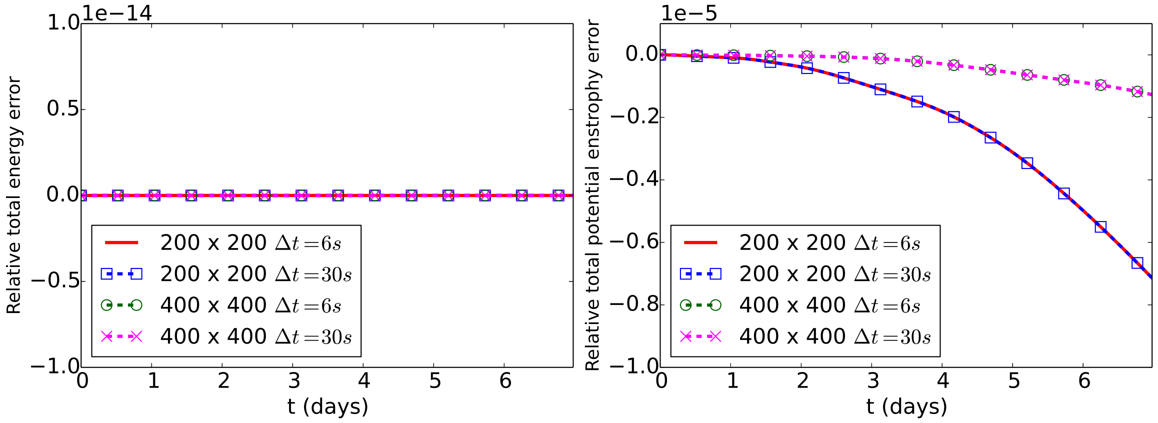
(a) Mesh 200 × 200 with $\Delta t = 6s$

(b) Mesh 200 × 200 with $\Delta t = 30s$

(c) Mesh 400 × 400 with $\Delta t = 6s$

(d) Mesh 400 × 400 with $\Delta t = 30s$

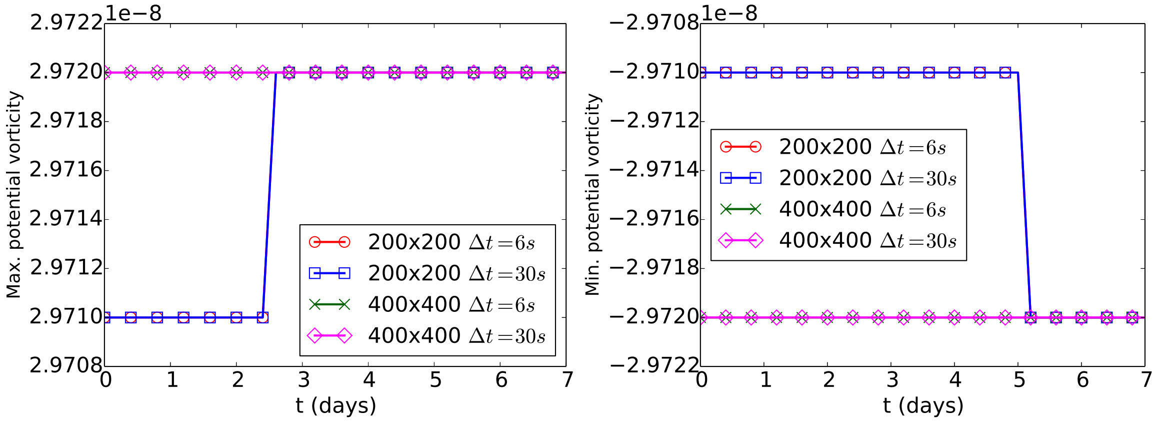
Figure 7.17: Comparison of ω obtained with the VI-EP4 scheme for flow over an isolated mountain.



(a) Relative change in total energy

(b) Relative change in total enstrophy

Figure 7.18: Evolution of relative change in total energy with VI-EP4 for flow over an isolated mountain.



(a) Maximum value of potential vorticity

(b) Minimum value of potential vorticity

Figure 7.19: Evolution of bounds of potential vorticity with VI-EP4 for flow past an isolated mountain.

with the CM-EP4 scheme, which also uses a central flux. However, using a finite difference WENO5 method with upwind flux splitting for (absolute) vorticity, seems to implicitly generate the required amount of dissipation to stabilize the solutions with VI-EP4.

- The dissipation in VI-EP4 is not as severe as compared to the CM-WENO5 scheme. Thus, VI-EP4 is able to preserve several important invariants associated the shallow water equations, for a reasonable amount of time.

8. A fully-discrete kinetic energy preserving and entropy stable scheme

In the previous chapters, we have discussed the construction of semi-discrete schemes which are entropy conservative. However, entropy need not be conserved if we are not careful about the temporal discretization. It has been shown in [116] that the implicit backward Euler time discretization can lead to decay of entropy, while the explicit forward Euler method can lead to production of entropy. A Crank-Nicolson type scheme on the other hand, can ensure that entropy is conserved. In addition to entropy conservation, we would like kinetic energy to be preserved by the fully discrete scheme for compressible flows. Subbareddy and Candler [112] have proposed an implicit fully discrete kinetic energy preserving finite difference scheme, by choosing appropriate time averaged states. However, this scheme cannot be shown to be conserve entropy.

In this chapter, we propose a fully-discrete second-order finite difference scheme for the Euler equations, which is both entropy conservative and kinetic energy preserving. When used in conjunction with the the viscous fluxes, it leads to the formulation of a kinetic energy preserving and entropy stable scheme for the Navier-Stokes equations.

8.1 Notations

We introduce some convenient notations required to construct the fully-discrete finite difference scheme. Consider the semi-discrete finite difference scheme (5.30) for the one-dimensional Navier-Stokes equations, with the numerical fluxes at the cell interfaces written as two-point fluxes in terms of the entropy variables $\mathbf{V}$

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}(\mathbf{V}_i, \mathbf{V}_{i+1}), \qquad \mathbf{G}_{i+\frac{1}{2}} = \mathbf{G}(\mathbf{V}_i, \mathbf{V}_{i+1}).$$

We rewrite (5.30) in terms of a *residual function* $\boldsymbol{\mathcal{R}}$

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbf{U}_i = \boldsymbol{\mathcal{R}}_i = \boldsymbol{\mathcal{R}}(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}), \tag{8.1}$$

where

$$\boldsymbol{\mathcal{R}}(\mathbf{a},\mathbf{b},\mathbf{c}) = \begin{pmatrix} \mathcal{R}^{\rho}(\mathbf{a},\mathbf{b},\mathbf{c}) \\ \mathcal{R}^{m}(\mathbf{a},\mathbf{b},\mathbf{c}) \\ \mathcal{R}^{e}(\mathbf{a},\mathbf{b},\mathbf{c}) \end{pmatrix} = \frac{1}{\Delta x}\left(-\mathbf{F}(\mathbf{b},\mathbf{c}) + \mathbf{F}(\mathbf{a},\mathbf{b}) + \mathbf{G}(\mathbf{b},\mathbf{c}) - \mathbf{G}(\mathbf{a},\mathbf{b})\right). \tag{8.2}$$

The discrete kinetic energy evolution equation discussed in Section 5.6.1 can be rewritten in terms of the the residual function as

$$\sum_i \Delta x \frac{\mathrm{d}\mathcal{K}_i}{\mathrm{d}t} = \sum_i \left[-\frac{1}{2} u_i^2 \mathcal{R}_i^\rho + u_i \mathcal{R}_i^m \right] \Delta x, \tag{8.3}$$

where $u := u(\mathbf{V})$. Define the quantity

$$\mathcal{S}_i = \mathcal{S}(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) = \frac{\Delta u_{i+\frac{1}{2}}}{\Delta x} \widetilde{p}_{i+\frac{1}{2}} - \frac{4}{3}\mu \sum_i \left(\frac{\Delta u_{i+\frac{1}{2}}}{\Delta x} \right)^2, \tag{8.4}$$

with $\Delta u_{i+\frac{1}{2}}, \widetilde{p}_{i+\frac{1}{2}}$ are obtained from $\mathbf{U}(\mathbf{V}_i), \mathbf{U}(\mathbf{V}_{i+1})$. Then, the following result relates $\mathcal{S}$ with the residual function $\boldsymbol{\mathcal{R}}$.

Lemma 8.1.1. *Assume that the residual function $\boldsymbol{\mathcal{R}}_i$ is evaluated with an inviscid numerical flux $\mathbf{F}_{i+\frac{1}{2}}$ satisfying the condition* (5.9)*, and with the viscous flux discretized according to* (5.32)*. Then the following relation holds*

$$\begin{aligned} \sum_i \mathcal{S}(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) \Delta x = & \sum_i \left[-\frac{1}{2} u(\mathbf{V}_i)^2 \mathcal{R}^\rho(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) \right] \Delta x \\ & + \sum_i \left[u(\mathbf{V}_i) \mathcal{R}^m(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) \right] \Delta x. \end{aligned} \tag{8.5}$$

Proof. Based on the derivations in Section 5.6.1, we have

$$\sum_i \Delta x \frac{\mathrm{d}\mathcal{K}_i}{\mathrm{d}t} = \sum_i \mathcal{S}(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) \Delta x,$$

since the inviscid and viscous numerical fluxes are assumed to satisfy the necessary conditions. Using the relation (8.3) gives the desired result (8.5). □

Finally, we consider the entropy relation corresponding to (8.1). Taking scalar product of (8.1) with $\mathbf{V}_i$ and summing over all cells leads to

$$\sum_i \frac{\mathrm{d}\eta_i}{\mathrm{d}t} \Delta x = \sum_i \left\langle \mathbf{V}_i, \boldsymbol{\mathcal{R}}(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) \right\rangle \Delta x. \tag{8.6}$$

We introduce the following quantity

$$\mathcal{V}_i = \mathcal{V}(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) = -\left[\frac{8\mu\overline{\beta}_{i+\frac{1}{2}}}{3} \left(\frac{\Delta u_{i+\frac{1}{2}}}{\Delta x} \right)^2 + \frac{\kappa}{R T_i T_{i+1}} \left(\frac{\Delta T_{i+\frac{1}{2}}}{\Delta x} \right)^2 \right], \tag{8.7}$$

where $\theta_i, \theta_{i+1}, \Delta u_{i+\frac{1}{2}}, \Delta\theta_{i+\frac{1}{2}}$ are obtained from $\mathbf{U}(\mathbf{V}_i), \mathbf{U}(\mathbf{V}_{i+1})$. Note that $\mathcal{V}_i \leqslant 0$. The quantity $\mathcal{V}$ can also be related to the $\boldsymbol{\mathcal{R}}$ in the following manner.

Lemma 8.1.2. *Assume that the residual function $\boldsymbol{\mathcal{R}}_i$ is evaluated with an entropy conservative numerical flux $\mathbf{F}_{i+\frac{1}{2}}$ and with the viscous flux discretized according to* (5.32)*. Then the following relation holds*

$$\sum_i \mathcal{V}(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) \Delta x = \sum_i \left\langle \mathbf{V}_i, \boldsymbol{\mathcal{R}}(\mathbf{V}_{i-1}, \mathbf{V}_i, \mathbf{V}_{i+1}) \right\rangle \Delta x. \tag{8.8}$$

Proof. As discussed in Section 5.6.2, if we choose an entropy conservative inviscid flux and discretize the viscous flux according to (5.32), then the inviscid flux contributions cancel one another when summed over cells leading to

$$\sum_i \frac{\mathrm{d}\eta_i}{\mathrm{d}t}\Delta x = \sum_i \mathcal{V}_i \Delta x \leqslant 0.$$

Using the relation (8.6), we have the desired result (8.8). □

8.2 Fully discrete finite volume scheme

Let $\widetilde{\mathbf{V}}(\mathbf{U}^*, \mathbf{U}^{**})$ be a centered average entropy variable that is yet to be specified, and define

$$\mathbf{V}_i^{n+\frac{1}{2}} = \widetilde{\mathbf{V}}(\mathbf{U}_i^n, \mathbf{U}_i^{n+1}),$$

which depends on the solution at time levels n and $n+1$. Consider the implicit finite difference scheme

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} + \frac{\mathbf{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x} = \frac{\mathbf{G}_{i+\frac{1}{2}}^{n+\frac{1}{2}} - \mathbf{G}_{i-\frac{1}{2}}^{n+\frac{1}{2}}}{\Delta x}, \tag{8.9}$$

where the fluxes are evaluated as

$$\mathbf{F}_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \mathbf{F}(\mathbf{V}_i^{n+\frac{1}{2}}, \mathbf{V}_{i+1}^{n+\frac{1}{2}}), \qquad \mathbf{G}_{i+\frac{1}{2}}^{n+\frac{1}{2}} = \mathbf{G}(\mathbf{V}_i^{n+\frac{1}{2}}, \mathbf{V}_{i+1}^{n+\frac{1}{2}}). \tag{8.10}$$

Since this is a Crank-Nicholson type scheme, it is second-order accurate. Note that the scheme can also be written as

$$\frac{\mathbf{U}_i^{n+1} - \mathbf{U}_i^n}{\Delta t} = \boldsymbol{\mathcal{R}}(\mathbf{V}_{i-1}^{n+\frac{1}{2}}, \mathbf{V}_i^{n+\frac{1}{2}}, \mathbf{V}_{i+1}^{n+\frac{1}{2}}),$$

where the residual function $\boldsymbol{\mathcal{R}}$ is given by (8.2).

8.2.1 Fully discrete kinetic energy preserving scheme

Assume that the velocity corresponding to the average value $\widetilde{\mathbf{V}}$ is given by

$$\widetilde{u}^{n+\frac{1}{2}} = u(\widetilde{\mathbf{V}}(\mathbf{U}^n, \mathbf{U}^{n+1})) = u(\mathbf{V}^{n+\frac{1}{2}}) = \frac{\sqrt{\rho^n}u^n + \sqrt{\rho^{n+1}}u^{n+1}}{\sqrt{\rho^n} + \sqrt{\rho^{n+1}}}. \tag{8.11}$$

This leads to the following result for the discrete kinetic energy $\mathcal{K}_i^n = \frac{1}{2}\rho_i^n \left(u_i^n\right)^2$.

Theorem 8.2.1 (Subbareddy and Candler [112]). *If the time-averaged velocity is defined by* (8.11)*, then the following relation is satisfied*

$$\frac{\mathcal{K}_i^{n+1} - \mathcal{K}_i^n}{\Delta t} = -\frac{1}{2}\left(\widetilde{u}_i^{n+\frac{1}{2}}\right)^2 \frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} + \widetilde{u}_i^{n+\frac{1}{2}} \frac{(\rho u)_i^{n+1} - (\rho u)_i^n}{\Delta t}, \tag{8.12}$$

which is consistent with continuous relation given by the first equation of (3.16).

The following theorem ensures that the scheme (8.9) is kinetic energy preserving, provided the fluxes are chosen appropriately.

Theorem 8.2.2. *Consider the fully-discrete numerical scheme* (8.9), *with the numerical fluxes chosen as follows:*

- *The inviscid momentum flux is approximated by*

$$F_{i+\frac{1}{2}}^{n+\frac{1}{2},m} = \widetilde{p}_{i+\frac{1}{2}}^{n+\frac{1}{2}} + \left(\frac{\widetilde{u}_i^{n+\frac{1}{2}} + \widetilde{u}_{i+1}^{n+\frac{1}{2}}}{2} \right) F_{i+\frac{1}{2}}^{n+\frac{1}{2},\rho}, \tag{8.13}$$

  where $\widetilde{u}_i^{n+\frac{1}{2}}$ *is given by* (8.11) *and* $\widetilde{p}_{i+\frac{1}{2}}^{n+\frac{1}{2}}$, $F_{i+\frac{1}{2}}^{n+\frac{1}{2},\rho}$ *are any consistent time-averaged approximations at the cell-interface* $x_{i+\frac{1}{2}}$.

- *The viscous flux is discretized using* (5.32), *with all the space-differences and space-averages evaluated using consistent time-averaged states.*

Then the total kinetic energy evolves at a discrete level according to

$$\sum_i \Delta x \frac{K_i^{n+1} - K_i^n}{\Delta t} = \sum_i \mathcal{S}(\mathbf{V}_{i-1}^{n+\frac{1}{2}}, \mathbf{V}_i^{n+\frac{1}{2}}, \mathbf{V}_{i+1}^{i+\frac{1}{2}})\Delta x, \tag{8.14}$$

with the quantity $\mathcal{S}$ *defined by* (8.4) *and evaluated at time-averaged states. In other words, the fully discrete scheme is kinetic energy preserving.*

Proof. Summing the relation (8.12) over all cells gives us the discrete evolution equation

$$\begin{aligned}
\sum_i \Delta x \frac{\mathcal{K}_i^{n+1} - \mathcal{K}_i^n}{\Delta t} &= \sum \left[-\frac{1}{2} \left(\widetilde{u}_i^{n+\frac{1}{2}} \right)^2 \frac{\rho_i^{n+1} - \rho_i^n}{\Delta t} + \widetilde{u}_i^{n+\frac{1}{2}} \frac{(\rho u)_i^{n+1} - (\rho u)_i^n}{\Delta t} \right] \Delta x \\
&= \sum_i \left[-\frac{1}{2} \left(u(\mathbf{V}_i^{n+\frac{1}{2}}) \right)^2 \mathcal{R}^\rho(\mathbf{V}_{i-1}^{n+\frac{1}{2}}, \mathbf{V}_i^{n+\frac{1}{2}}, \mathbf{V}_{i+1}^{n+\frac{1}{2}}) \right] \Delta x \\
&\quad + \sum_i \left[u(\mathbf{V}_i^{n+\frac{1}{2}}) \mathcal{R}^m(\mathbf{V}_{i-1}^{n+\frac{1}{2}}, \mathbf{V}_i^{n+\frac{1}{2}}, \mathbf{V}_{i+1}^{n+\frac{1}{2}}) \right] \Delta x.
\end{aligned}$$

The inviscid numerical flux is chosen to satisfy (8.13), which is exactly the condition (5.9) but defined for time-averaged quantities. Furthermore, the viscous flux is discretized using (5.32). Thus, an application of Lemma 8.1.1 gives the results (8.14). □

Remark 8.2.1. *The KEP and KEPEC fluxes satisfy the condition* (8.13) *and can thus be used to construct fully-discrete kinetic energy preserving schemes.*

8.2.2 Fully discrete entropy conservative scheme

Assume that the averaged entropy variable $\widetilde{\mathbf{V}}(\mathbf{U}^*, \mathbf{U}^{**})$ satisfies

$$\widetilde{\mathbf{V}}(\mathbf{U}^*, \mathbf{U}^{**}) \cdot (\mathbf{U}^{**} - \mathbf{U}^*) = \eta(\mathbf{U}^{**}) - \eta(\mathbf{U}^*), \tag{8.15}$$

Then the following theorem details the construction of an entropy stable scheme.

Theorem 8.2.3. *Consider the fully-discrete numerical scheme* (8.9) *and assume that* $\mathbf{V}_i^{n+\frac{1}{2}}$ *satisfies* (8.15)*. Furthermore, let the numerical fluxes be chosen as follows:*

- *The inviscid numerical flux is entropy conservative. A sufficient condition to obtain such a flux is given by* (5.3)*.*
- *The viscous flux is discretized using* (5.32)*, with all the space-differences and space-averages evaluated at consistent time-averaged states.*

Then the following entropy estimate holds

$$\sum_i \frac{\eta(\mathbf{U}_i^{n+1}) - \eta(\mathbf{U}_i^n)}{\Delta t}\Delta x = \sum_i \mathcal{V}(\mathbf{V}_{i-1}^{n+\frac{1}{2}}, \mathbf{V}_i^{n+\frac{1}{2}}, \mathbf{V}_{i+1}^{n+\frac{1}{2}})\Delta x. \tag{8.16}$$

with the quantity $\mathcal{V}$ *defined by* (8.7) *and evaluated using time-averaged states. Since the quantity* $\mathcal{V} \leqslant 0$, (8.16) *ensures*

$$\sum_i \eta(\mathbf{U}_i^{n+1})\Delta x \leqslant \sum_i \eta(\mathbf{U}_i^n)\Delta x.$$

In other words, the scheme is entropy stable.

Proof. Since $\mathbf{V}_i^{n+\frac{1}{2}}$ satisfies (8.15), taking the scalar product of (8.9) with $\mathbf{V}_i^{n+\frac{1}{2}}$ leads to the cell entropy estimate

$$\frac{\eta(\mathbf{U}_i^{n+1}) - \eta(\mathbf{U}_i^n)}{\Delta t} = \left\langle \mathbf{V}_i^{n+\frac{1}{2}}, \boldsymbol{\mathcal{R}}(\mathbf{V}_{i-1}^{n+\frac{1}{2}}, \mathbf{V}_i^{n+\frac{1}{2}}, \mathbf{V}_{i+1}^{n+\frac{1}{2}}) \right\rangle. \tag{8.17}$$

Furthermore, the inviscid flux is entropy conservative and the viscous flux is discretized using (5.32). Thus, summing (8.17) over all cells and an application of Lemma 8.1.2 gives us the required result (8.16). □

Remark 8.2.2. *The entropy conservative ROE-EC and KEPEC fluxes can be used to construct fully-discrete entropy stable schemes.*

8.2.3 Construction of $\widetilde{\mathbf{V}}$

We now demonstrate a method to construct an average value $\widetilde{\mathbf{V}}$ satisfying (8.15). Moreover, to ensure kinetic energy preservation, the velocity corresponding to $\widetilde{\mathbf{V}}$ should be given by (8.11). This motivates us to introduce the parameter vector

$$\mathbf{Z} = \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} \sqrt{\rho} \\ \sqrt{\rho}u \\ p \end{bmatrix}.$$

Subbareddy and Candler [112] also use such an approach to construct fully-discrete kinetic energy preserving scheme, however they only considered the mass and momentum balance equations. We define the following jump and arithmetic averaging operators with respect to time

$$\delta(\cdot)^{n+\frac{1}{2}} := (\cdot)^{n+1} - (\cdot)^n, \quad \sigma(\cdot)^{n+\frac{1}{2}} := \frac{(\cdot)^n + (\cdot)^{n+1}}{2}.$$

We additionally define the logarithmic average operator with respect to time

$$\ddot{(\cdot)}^{n+\frac{1}{2}} := \frac{(\cdot)^{n+1} - (\cdot)^n}{\ln(\cdot)^{n+1} - \ln(\cdot)^n}.$$

The superscript $n + \frac{1}{2}$ will be dropped to simplify the notation. We wish to find $\widetilde{\mathbf{V}}$ such that (8.15) is satisfied. In other words,

$$\widetilde{V}_1 \delta\rho + \widetilde{V}_2 \delta(\rho u) + \widetilde{V}_3 \delta E = -\frac{1}{\gamma - 1}\delta(\rho s). \tag{8.18}$$

In terms of the parameter vector $\mathbf{Z}$, we have the following exact linearizations

$$\begin{aligned}
\delta(\rho) &= 2\sigma(Z_1)\delta(Z_1), \\
\delta(\rho u) &= \sigma(Z_2)\delta(Z_1) + \sigma(Z_1)\delta(Z_2), \\
\delta(E) &= \frac{\delta(Z_3)}{\gamma - 1} + \sigma(Z_2)\delta(Z_2), \\
\delta(\rho s) &= \frac{\sigma(\rho)}{\ddot{(p)}}\delta(Z_3) + \left(2\sigma(s)\ \sigma(Z)_1 - \frac{2\gamma\sigma(\rho)\ \sigma(Z_1)}{\ddot{(\rho)}}\right)\delta(Z_1).
\end{aligned}$$

Equating the coefficients of $\delta(Z_2)$ and $\delta(Z_3)$ in (8.18) yields

$$\widetilde{V}_2 = -\widetilde{V}_3 \frac{\sigma(Z_2)}{\sigma(Z_1)}, \qquad \widetilde{V}_3 = -\frac{\sigma(\rho)}{\ddot{(p)}},$$

which in turn gives

$$\widetilde{\beta} = \frac{\sigma(\rho)}{2\ddot{(p)}}, \qquad \widetilde{u} = \frac{\sigma(Z_2)}{\sigma(Z_1)}. \tag{8.19}$$

Note that velocity $\widetilde{u}$ corresponding to $\widetilde{\mathbf{V}}$ satisfies (8.11). Finally equating the coefficients $\delta(Z_1)$ in (8.18) yields

$$\widetilde{V}_1 = \frac{1}{\gamma - 1}\left(\gamma \frac{\sigma(\rho)}{\ddot{(\rho)}} - \sigma(s)\right) - \widetilde{\beta}\widetilde{u}^2,$$

which gives

$$\widetilde{s} = \sigma(s) + \gamma\left(1 - \frac{\sigma(\rho)}{\ddot{(\rho)}}\right). \tag{8.20}$$

The above definitions of the averages $\widetilde{V}_1$, $\widetilde{V}_2$, $\widetilde{V}_3$ are consistent. The time-averaged entropy variable $\widetilde{\mathbf{V}}$ is most conveniently expressed as

$$\widetilde{\mathbf{V}} = \widetilde{\mathbf{V}}(\widetilde{s}, \widetilde{u}, \widetilde{\beta}),$$

with $\widetilde{s}, \widetilde{u}, \widetilde{\beta}$ given by (8.19) and (8.20).

8.3 Time integration

The discrete system of equations we need to solve is of the form

$$\frac{\mathbf{U}^{n+1} - \mathbf{U}^n}{\Delta t} - \boldsymbol{\mathcal{R}}(\mathbf{U}^n, \mathbf{U}^{n+1}) = 0,$$

where $\boldsymbol{\mathcal{R}}$ is a non-linear function, making the system implicit in $\mathbf{U}^{n+1}$. We consider two methods to solve for the update $\mathbf{U}^{n+1}$.

8.3.1 Explicit iteration

Assuming that $\mathbf{U}^n$ is given, we consider the fixed point problem in $\mathbf{U}$

$$\mathbf{U} = \mathbf{U}^n + \Delta t \boldsymbol{\mathcal{R}}(\mathbf{U}^n, \mathbf{U}) = J(\mathbf{U}).$$

This is solved using Picard's iteration

$$\mathbf{U}^{(s+1)} = J(\mathbf{U}^{(s)}), \tag{8.21}$$

with the initialisation $\mathbf{U}^{(0)} = \mathbf{U}^n$. The iterations are stopped when the norm of relative residual drops below some threshold

$$\frac{|(\mathbf{U}^{(s+1)} - \mathbf{U}^n)/(\Delta t) - \boldsymbol{\mathcal{R}}(\mathbf{U}^n, \mathbf{U}^{(s)})|}{|\boldsymbol{\mathcal{R}}(\mathbf{U}^n, \mathbf{U}^n)|} < \epsilon.$$

At the end of the iterations, we set $\mathbf{U}^{n+1} = \mathbf{U}^{(s+1)}$.

In practice, the explicit iteration method is found to converge rather slowly. This can be understood as follows. Assume for the sake of simplicity that $\boldsymbol{\mathcal{R}}$ is Lipschitz with a Lipschitz constant M

$$|\boldsymbol{\mathcal{R}}(\mathbf{U}) - \boldsymbol{\mathcal{R}}(\mathbf{V})| \leqslant M|\mathbf{U} - \mathbf{V}|.$$

Then we have

$$|J(\mathbf{U}) - J(\mathbf{V})| = \Delta t|\boldsymbol{\mathcal{R}}(\mathbf{U}) - \boldsymbol{\mathcal{R}}(\mathbf{V})| \leqslant \Delta t M|\mathbf{U} - \mathbf{V}|.$$

If $\Delta t M < 1$, we can use the contraction mapping principle to get a unique fixed point $\mathbf{U}^*$, to which the sequence defined by (8.21) converges. If the Lipschitz constant is large, it can put a severe restriction on the time-step Δt.

8.3.2 Newton-GMRES

This approach involves solving for the root of the system

$$\mathcal{L}(\mathbf{U}) = \frac{\mathbf{U} - \mathbf{U}^n}{\Delta t} - \boldsymbol{\mathcal{R}}(\mathbf{U}^n, \mathbf{U}) = 0.$$

We make use of an iterative Newton method

$$\mathcal{L}'(\mathbf{U}^{(s)})(\mathbf{U}^{(s+1)} - \mathbf{U}^{(s)}) + \mathcal{L}(\mathbf{U}^{(s)}) = 0, \qquad \mathbf{U}^{(0)} = \mathbf{U}^n.$$

The iterations are stopped when

$$\frac{|\mathcal{L}(\mathbf{U}^{(s+1)})|}{|\mathcal{L}(\mathbf{U}^n)|} < \epsilon,$$

and we set $\mathbf{U}^{n+1} = \mathbf{U}^{(s+1)}$. We need to evaluate two quantities in each Newton iteration, namely the Jacobian $\mathcal{L}'(\mathbf{U}^{(s)})$ and the step size $\boldsymbol{\delta} = \mathbf{U}^{(s+1)} - \mathbf{U}^{(s)}$. Instead of evaluating the Jacobian, we consider $\mathcal{L}'(\mathbf{U})\mathbf{W}$, which is the action of the Jacobian evaluated at $\mathbf{U}$

on the vector $\mathbf{W}$. We approximate $\mathcal{L}'(\mathbf{U})\mathbf{W}$ according to the following algorithm given in [67]

$$D_h\mathcal{L}(\mathbf{U},\mathbf{W}) = \begin{cases} 0, & \text{if } \mathbf{W}=0 \\ \frac{|\mathbf{W}|}{h|\mathbf{U}|}\left(\mathcal{L}\left(\mathbf{U}+h\frac{|\mathbf{U}|\mathbf{W}}{|\mathbf{W}|}\right)-\mathcal{L}(\mathbf{U})\right), & \text{if } \mathbf{W},\mathbf{U}\neq 0 \\ \frac{|\mathbf{W}|}{h}\left(\mathcal{L}\left(\frac{h\mathbf{W}}{|\mathbf{W}|}\right)-\mathcal{L}(\mathbf{U})\right), & \text{if } \mathbf{W}\neq 0, \mathbf{U}=0 \end{cases}$$

choosing $h = 10^{-7}$ to evaluate the above derivative. The step size $\boldsymbol{\delta}$ is approximated using the GMRES linear solver. The advantage of the GMRES method is that it makes use of the matrix vector products defined above. Moreover, if $\mathcal{L}' \in \mathbb{R}^{m\times m}$, then the method converges in atmost m steps. A detailed explaination of the GMRES method is given in Appendix D. The Newton-GMRES code used for simulating numerical results in this chapter has been taken from [67].

8.4 Numerical results

We demonstrate the performance of the fully discrete Crank-Nicolson type scheme (8.9), using the time-averaged vector $\widetilde{\mathbf{V}}$ described in Section 8.2.3, the KEPEC flux and the viscous discretization given by (5.32), which leads to a kinetic energy preserving and entropy stable scheme. We refer to this scheme as the *FDKEPES scheme*. The tolerance level required in the explicit iteration and Newton-GMRES algorithms is set to $\epsilon = 10^{-8}$. The time step Δt is determined by the CFL condition (4.5).

8.4.1 Smooth inviscid solution

We consider a smooth solution for the Euler equations, with the initial conditions given by

$$\rho_0 = 1 + 0.5\cos(\pi x), \qquad u_0 = 5.0, \qquad p_0 = 1.0,$$

on the domain $[-1,1]$ with periodic boundary conditions. The final time is chosen as $t_f = 2$, which corresponds to the completion of 10 periodic cycles of the wave. The Newton-GMRES scheme is used for time integration with CFL = 0.5. The errors and convergence rates shown in Table 8.1 indicate that the scheme is second-order accurate. Identical errors were obtained (not presented here) when the explicit iteration scheme was used for time integration.

N	L^1_h		L^∞_h	
	error	rate	error	rate
50	5.54e-02	-	5.35e-02	-
100	1.41e-02	1.97	1.22e-02	2.14
200	3.55e-03	2.00	2.86e-03	2.09
400	8.87e-04	2.00	7.08e-04	2.01

Table 8.1: Error with the FDKEPES scheme and Newton-GMRES for advecting density wave problem.

In order to judge which time integration method performs better, we discretize the mesh with $N = 200$ cells and vary the CFL number. For the explicit iteration method, we monitor the average number of Picard iteration steps needed to approximate the solution at the next time level. On the other hand, for the Newton-GMRES method, we have to look at the number of *inner* iteration steps needed for GMRES and the number of *outer* iterations steps used for the Newton solve. We infer from table 8.2 that the performance of both methods is quite similar, for small CFL numbers. However, as CFL number is increased beyond 1, the explicit method requires considerably more number of iterations to solve the non-linear time update problem, as compared to Newton-GMRES.

CFL	Explicit Iteration	Newton-GMRES		L_h^1	L_h^∞
	Picard-steps	GMRES-steps	Newton-steps		
0.5	7	5	5	3.5e-03	2.9e-03
1.0	12	6	8	4.3e-03	3.5e-03
1.5	15	6	10	5.6e-03	4.6e-03
2.0	1432	7	12	7.4e-03	6.1e-03

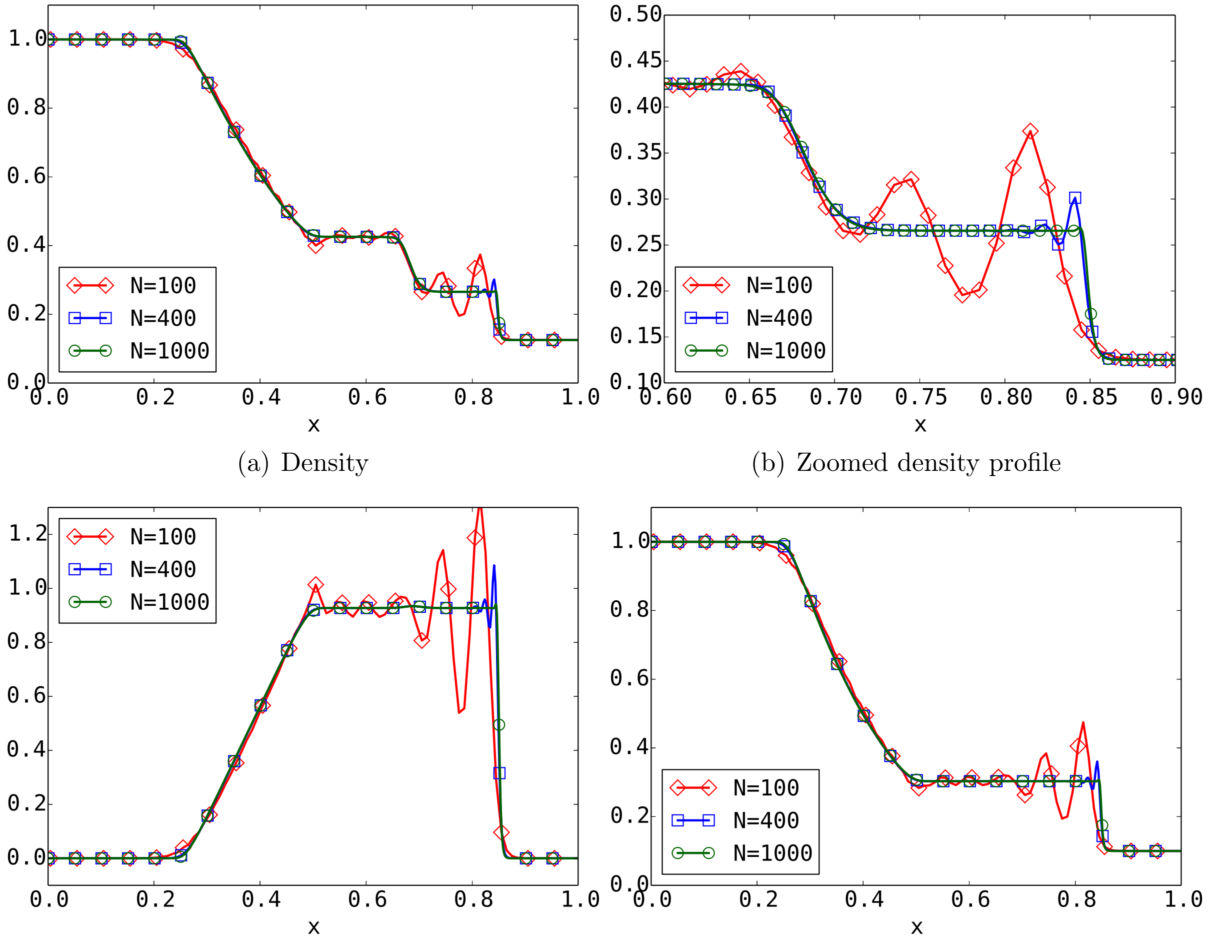
Table 8.2: Comparing the explicit iteration method and Newton-GMRES.

Remark 8.4.1. *The magnitude of errors increases as the CFL number is increased, thus one needs to be judicious about how large a CFL number is chosen for a particular problem.*

8.4.2 DNS of 1D Navier-Stokes: Sod test case

This is a shock tube problem for the Navier-Stokes equations, with the left state $(\rho_L, u_L, p_L) = (1.0, 0.0, 1.0)$ and the right state $(\rho_R, u_R, p_R) = (0.125, 0.0, 0.1)$. The domain is $[0, 1]$ with the initial discontinuity at $x = 0.5$. We use the Newton-GMRES scheme to integrate up to a final time of $t_f = 0.2$ and a CFL=1.5. We choose a constant viscosity coefficient $\mu = 2 \times 10^{-4}$ and discretize the mesh using $N = 100$ cells. The solutions shown in Figure 8.1 are highly oscillatory, indicating that physical viscosity (and the heat flux) are unable to suppress the oscillations in the solutions. The amplitude of oscillations are reduced with $N = 400$, and they finally subside with $N = 1000$. This behaviour can be explained by considering the *mesh Peclet number* $Pe = \rho u \Delta x / \mu$, which is essentially the local cell Reynolds number. It is well known that the if Pe is much larger than unity, oscillations can appear while performing DNS using central fluxes [59, 61]. The Pe corresponding to the different mesh sizes shown in Figure 8.2, validate this fact. A deeper analysis in this direction will be performed in Chapter 10, where we also look at the balance between the physical viscosity, heat flux and artificial viscosity introduced in the numerical flux.

We also compare the fully-discrete solution on a mesh with $N = 1000$ to the solution obtained by a semi-discrete scheme with the KEPEC flux and viscous discretization (5.32), integrated using the explicit SSP-RK3 scheme. The results in Figure 8.3 show that the solutions obtained by both methods are indistinguishable. Note that we cannot choose CFL $= 1.5$ with the semi-discrete scheme as the solution blows up. We choose a more conservative CFL=0.5. In other words, the fully-discrete scheme with Newton-GMRES permits the use of larger time steps.



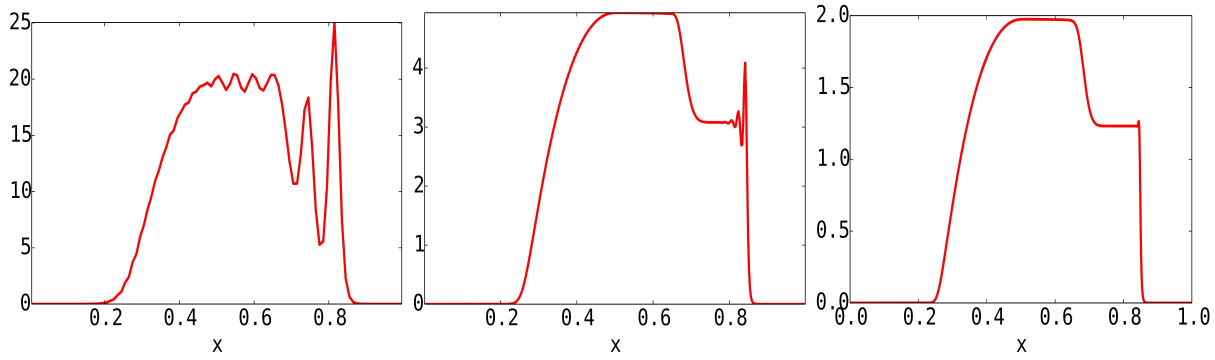
(a) Density

(b) Zoomed density profile

(c) Velocity

(d) Pressure

Figure 8.1: Sod test case with FDKEPES on different mesh sizes

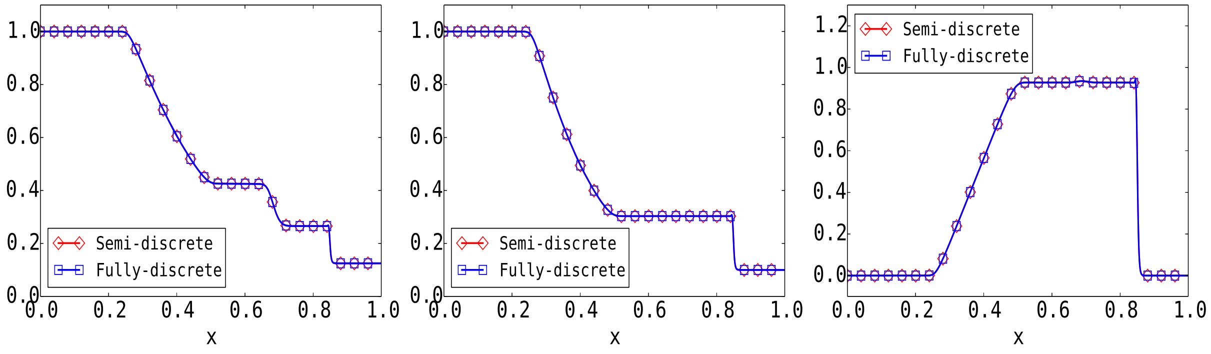


(a) N=100

(b) N=400

(c) N=1000

Figure 8.2: Mesh Peclet numbers with FDKEPES on different mesh sizes



(a) Density (b) Pressure (c) Velocity

Figure 8.3: Comparison of FDKEPES with a semi-discrete scheme using the KEPEC flux and SSP-RK3 on a mesh with N=1000.

9. Finite volume schemes for Euler equations

In the previous chapters, we described finite difference schemes on Cartesian grids, satisfying additional properties such as entropy stability and kinetic energy preservation. However, many applications of interest, particularly in engineering, involve domains with complex geometry which can be more easily discretized using *unstructured grids.* Finite volume methods can be easily applied on unstructured grids, which makes them useful for problems involving complex domains. In this chapter, we introduce the important ingredients of finite volume schemes on two-dimensional unstructured meshes for systems of conservation laws, with a focus on the compressible Euler equations. The work presented in this chapter has been published in [90].

9.1 Mesh

The domain $\Omega \subset \mathbb{R}^2$ is discretized by a collection of non-overlapping triangles, which forms the *primary mesh.* A triangle T is formed by vertices i, j, k, as shown in Figure 9.1(a). We use the notation $\mathbf{n}_i^T$ to describe the outward normal to the edge of T which is opposite to the vertex i. The normal has magnitude equal to the length of the corresponding edge. Furthermore, for each boundary edge e, we denote the triangle adjacent to it by T_e and the outward normal to the edge e as $\mathbf{n}_e$. These are depicted in Figure 9.1(b). Note that $\mathbf{n}_e \equiv \mathbf{n}_k^{T_e}$.

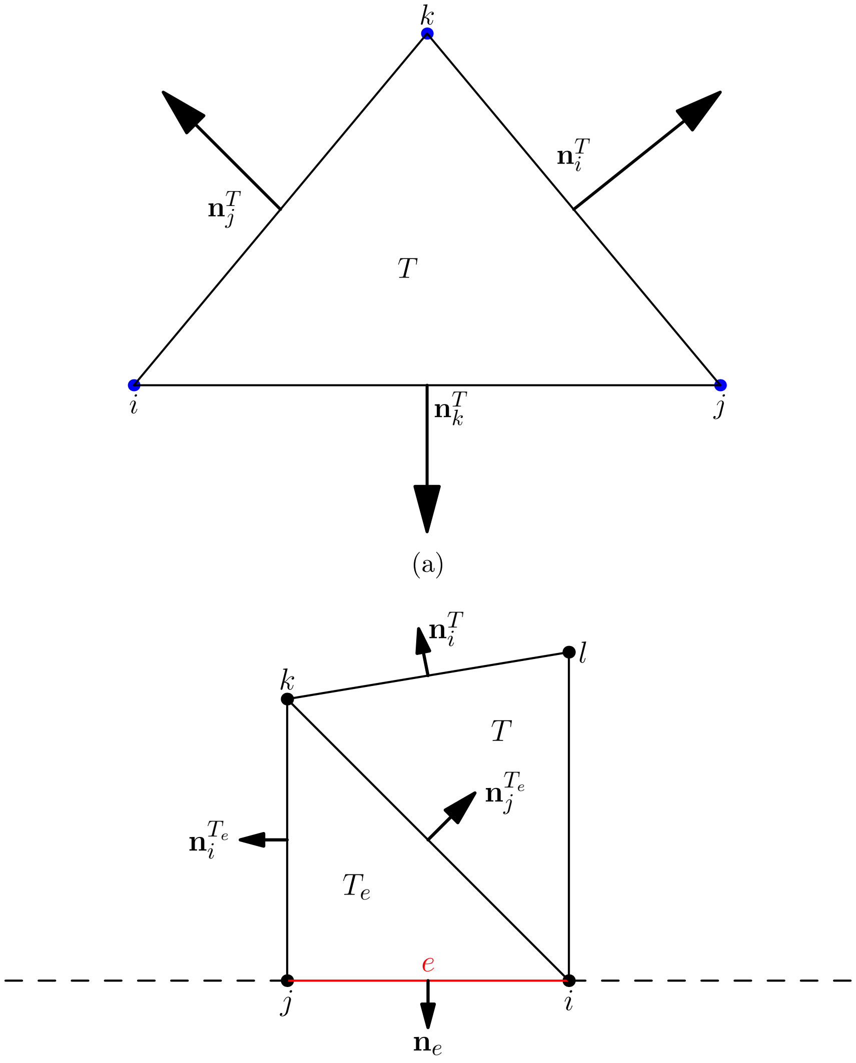
There are two types of finite volume formulations based on the type of cells used as control volumes, namely *cell-centered schemes* and *vertex-centered schemes.*

9.1.1 Cell-centered scheme

In cell-centered schemes, the primary triangles are chosen as control volumes and the solutions are stored at the centroids. Consider the primary grid shown in Figure 9.2. We index the various triangles using capital letters I, J, K,etc., with the centroid of triangle T_I denoted by G_I. For convenience, we use the notation $\mathbf{n}_{IJ}$ to describe the scaled outward normal with respect to the triangle T_I on the edge shared by T_I and T_J. Note that this is equivalent to $\mathbf{n}_i^{T_I}$ in terms of our earlier notation. A simple application of the divergence theorem gives

$$\sum_{J \in I} \mathbf{n}_{IJ} = \mathbf{0}, \tag{9.1}$$

where the notation $J \in I$ denotes the set of triangles T_J which share an edge with T_I.



(a)

(b)

Figure 9.1: Primary mesh triangles (a) T with its vertices and outward normals (b) boundary triangle T_e with outward normals.

9.1.2 Vertex-centered scheme

In this formulation the solution is stored at the vertices of the primary mesh, and a *dual cell* Ω_i is constructed around each vertex i. One way of constructing the dual cell, is by joining the centroids of each adjoining triangle to the mid-points of its edges. This leads to the *median dual cell* [132, 111, 2]. The *Voronoi dual cells* can also be generated in a similar manner, by joining the mid-point of the triangle edges to the circumcenters instead of the centroids [76, 1]. Examples of a primary mesh and corresponding dual meshes are depicted in Figure 9.3.

Consider the dual cell Ω_i around vertex i as shown in Figure 9.4. If j is a vertex

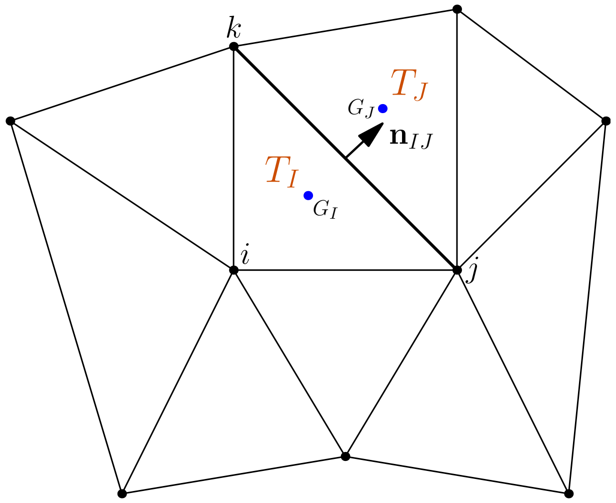


Figure 9.2: Primary mesh with triangles as control volumes.

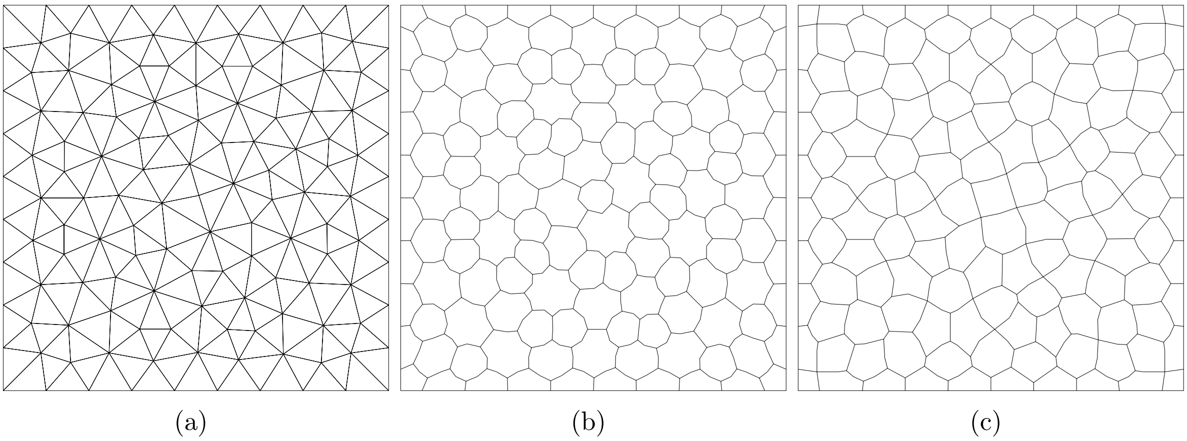


Figure 9.3: Mesh (a) primary; (b) median dual; (c) Voronoi dual.

connected to vertex i, then define

$$\mathbf{n}_{ij} = \int\limits_{\partial\Omega_i \cap \partial\Omega_j} \mathbf{n} ds = \mathbf{n}_{ij}^{(1)} + \mathbf{n}_{ij}^{(2)},$$

where $\mathbf{n}$ is the unit normal vector to the faces of dual cell Ω_i common with the dual cell Ω_j. The quantity $\mathbf{n}_{ij}$ has units of length. As before, the divergence theorem gives

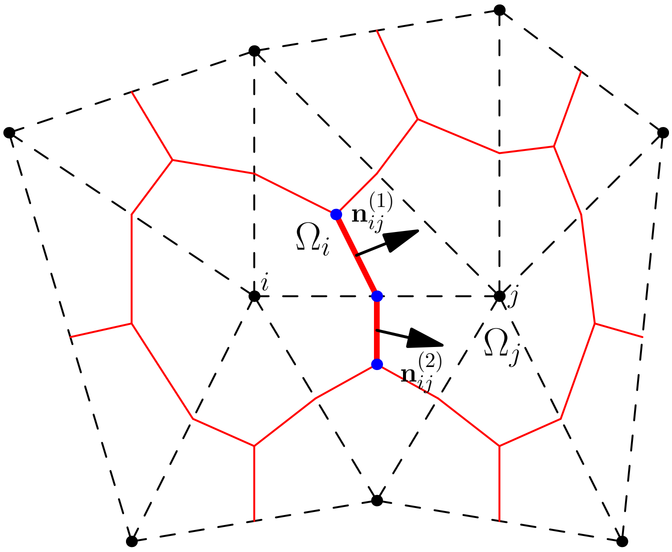


Figure 9.4: Dual cell interface and normal

$$\sum_{j\in i} \mathbf{n}_{ij} = \mathbf{0}, \tag{9.2}$$

where the notation $j \in i$ denotes the set of vertices j neighbouring the vertex i, i.e., which are connected to vertex i through a primary edge.

9.2 Semi-discrete finite volume scheme

Consider the integral formulation (2.1) with $d = 2$ over a control volume $\mathcal{C}$. We define the cell-averaged value of the conserved variable $\mathbf{U}$ over the volume $\mathcal{C}$ by

$$\widetilde{\mathbf{U}}(t) = \frac{1}{|\mathcal{C}|} \int\limits_{\mathcal{C}} \mathbf{U}(\mathbf{x}, t)\, d\mathbf{x},$$

where $|\mathcal{C}|$ is the cell volume. Let $\mathcal{N}$ be the set of control volumes sharing an edge with $\mathcal{C}$, i.e., $\mathcal{N} = \{$ all $\mathcal{C}'$ such that $\mathcal{C}' \neq \mathcal{C}$ and $\partial\mathcal{C} \cap \partial\mathcal{C}' \neq \emptyset\}$. Thus, we have

$$|\mathcal{C}| \frac{\mathrm{d}\widetilde{\mathbf{U}}}{\mathrm{d}t} = -\int\limits_{\partial\mathcal{C}} \mathbf{F}(\mathbf{U}, \mathbf{n}) \mathrm{d}\, S = -\sum_{\mathcal{C}' \in \mathcal{N}} \int\limits_{\partial\mathcal{C}\cap\partial\mathcal{C}'} \mathbf{F}(\mathbf{U}, \mathbf{n}) \mathrm{d}S, \tag{9.3}$$

where the notation $\mathbf{F}(\mathbf{U}, \mathbf{n})$ is defined in (3.13). We now look at the cell-centered and vertex-centered semi-discrete formulation independently.

9.2.1 Cell-centered semi-discrete scheme

As discussed in Section 9.1.1, the control volumes for a cell-centered scheme are the primary triangles. For each triangle T_I, we use the notation $\mathbf{U}_I$ to represent the cell average value. Thus (9.3) can be written as

$$|T_I|\frac{\mathrm{d}\mathbf{U}_I}{\mathrm{d}t} = -\sum_{J\in I}\int\limits_{\partial T_I\cap\partial T_J}\mathbf{F}(\mathbf{U},\mathbf{n})\mathrm{d}S. \tag{9.4}$$

The integrals are approximated using the mid-point quadrature rule, which is second-order accurate. The semi-discrete scheme is written as

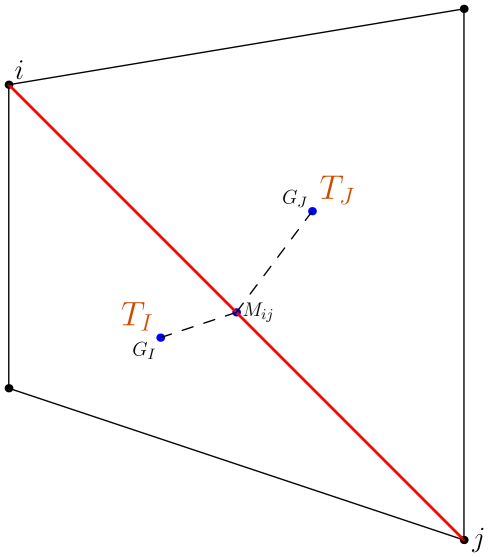


Figure 9.5: Evaluation of flux for cell-centered scheme.

$$|T_I|\frac{\mathrm{d}\mathbf{U}_I}{\mathrm{d}t} = -\sum_{J\in I}\mathbf{F}_{IJ}, \tag{9.5}$$

where $\mathbf{F}_{IJ} := \mathbf{F}(\mathbf{U}_I,\mathbf{U}_J,\mathbf{n}_{IJ})$ is the flux approximation at the edge mid-point M_{ij} (see figure 9.5). For a higher order approximation, one could use an N-point Gauss-Legendre quadrature. The numerical flux should have the following crucial properties

1. Consistency:
$$\mathbf{F}(\mathbf{U},\mathbf{U},\mathbf{n}) = \mathbf{F}(\mathbf{U},\mathbf{n}).$$

2. Conservation:
$$\mathbf{F}(\mathbf{U}_1,\mathbf{U}_2,\mathbf{n}) = -\mathbf{F}(\mathbf{U}_2,\mathbf{U}_1,-\mathbf{n}) \qquad \forall \quad \mathbf{U}_1,\mathbf{U}_2,\mathbf{n}.$$

We are interested in constructing central fluxes, to which suitable dissipation is added at a later stage to ensure stability. Thus, we wish to analyse the approximation of such central schemes. We consider the simplest central flux of the form

$$\mathbf{F}(\mathbf{U}_I,\mathbf{U}_J,\mathbf{n}_{IJ}) = \frac{1}{2}\Big[\mathbf{F}(\mathbf{U}_I,\mathbf{n}_{IJ}) + \mathbf{F}(\mathbf{U}_J,\mathbf{n}_{IJ})\Big]. \tag{9.6}$$

Assuming sufficient smoothness, the Taylor series expansion of the exact solution about a point $\mathbf{x}_0$ gives

$$\mathbf{U}(\mathbf{x},t) = \sum_{k=0}^{r-1} \frac{1}{k!} \sum_{|\boldsymbol{\alpha}|=k} (\mathbf{x}-\mathbf{x}_0)^{\boldsymbol{\alpha}} \partial^{\boldsymbol{\alpha}} \mathbf{U}(\mathbf{x}_0,t)\mathrm{d}\mathbf{x} + \mathcal{O}(|\mathbf{x}-\mathbf{x}_0|^r),$$

for some $r \geqslant 1$. Thus, the cell average value in T_I can be expanded around the centroid G_I as

$$\mathbf{U}_I(t) = \mathbf{U}(\mathbf{x}_{G_I},t) + \frac{1}{|T_I|}\int_{T_I} \nabla_{\mathbf{x}}\mathbf{U}(\mathbf{x}_{G_I},t)\cdot(\mathbf{x}-\mathbf{x}_{G_I})\mathrm{d}\mathbf{x} + \mathcal{O}(h^2),$$

where h denotes an appropriate length scale of the triangulation, for instance the length of the longest edge. The centroid satisfies the relation

$$\frac{1}{|T_I|}\int_{T_I} (\mathbf{x}-\mathbf{x}_{G_I})\mathrm{d}\mathbf{x} = 0,$$

which leads to the approximation

$$\mathbf{U}_I(t) = \mathbf{U}(\mathbf{x}_{G_I},t) + \mathcal{O}(h^2). \tag{9.7}$$

Using (9.7) in the central flux (9.6) leads to

$$\mathbf{F}(\mathbf{U}_I,\mathbf{U}_J,\mathbf{n}_{IJ}) = \frac{1}{2}\Big[\mathbf{F}\big(\mathbf{U}(\mathbf{x}_{G_I},t),\mathbf{n}_{IJ}\big) + \mathbf{F}\big(\mathbf{U}(\mathbf{x}_{G_J},t),\mathbf{n}_{IJ}\big)\Big] + \mathcal{O}(h^2). \tag{9.8}$$

Furthermore,

$$\begin{aligned}\mathbf{F}\big(\mathbf{U}(\mathbf{x}_{G_I},t),\mathbf{n}_{IJ}\big) &= \mathbf{F}\big(\mathbf{U}(\mathbf{x}_{M_{ij}},t),\mathbf{n}_{IJ}\big) + \nabla_{\mathbf{x}}\mathbf{F}\big(\mathbf{U}(\mathbf{x}_{M_{ij}},t),\mathbf{n}_{IJ}\big)\cdot(\mathbf{x}_{G_I}-\mathbf{x}_{M_{ij}}) + \mathcal{O}(h^2),\\ \mathbf{F}\big(\mathbf{U}(\mathbf{x}_{G_J},t),\mathbf{n}_{IJ}\big) &= \mathbf{F}\big(\mathbf{U}(\mathbf{x}_{M_{ij}},t),\mathbf{n}_{IJ}\big) + \nabla_{\mathbf{x}}\mathbf{F}\big(\mathbf{U}(\mathbf{x}_{M_{ij}},t),\mathbf{n}_{IJ}\big)\cdot(\mathbf{x}_{G_J}-\mathbf{x}_{M_{ij}}) + \mathcal{O}(h^2).\end{aligned} \tag{9.9}$$

Looking at Figure 9.5, we note that G_i, M_{ij}, G_j need not be collinear and $|\mathbf{x}_{G_I}-\mathbf{x}_{M_{ij}}| \neq |\mathbf{x}_{M_{ij}}-\mathbf{x}_{G_J}|$ in general. Thus, the $\mathcal{O}(h)$ contribution in (9.9) will not cancel out when both expression are added, leading to

$$\mathbf{F}(\mathbf{U}_I,\mathbf{U}_J,\mathbf{n}_{IJ}) = \mathbf{F}(\mathbf{U}(\mathbf{x}_{M_{ij}},t),\mathbf{n}_{IJ}) + \mathcal{O}(h).$$

9.2.2 Vertex-centered semi-discrete scheme

In this case, the control volumes are taken to be dual cells Ω_i around each vertex i, with the notation $\mathbf{U}_i = \widetilde{\mathbf{U}}$. Thus (9.3) can be written as

$$|\Omega_i|\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = -\sum_{j\in i}\int\limits_{\partial\Omega_i\cap\partial\Omega_j} \mathbf{F}(\mathbf{U},\mathbf{n})\mathrm{d}S. \tag{9.10}$$

The integrals are approximated using a rule similar to the mid-point quadrature

$$\sum_{j\in i}\int\limits_{\partial\Omega_i\cap\partial\Omega_j} \mathbf{F}(\mathbf{U},\mathbf{n})\mathrm{d}S \approx \sum_{j\in i}\mathbf{F}\big(\mathbf{U}(M_{ij},t),\mathbf{n}_{ij}\big), \tag{9.11}$$

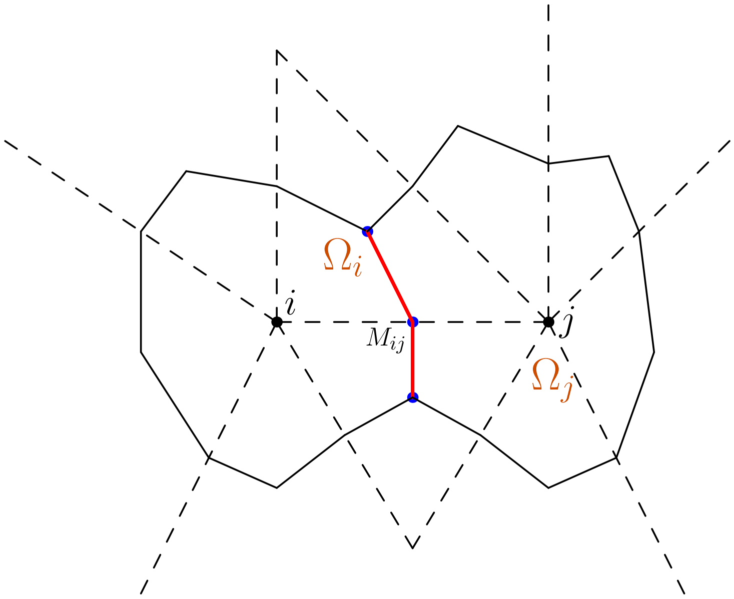


Figure 9.6: Evaluation of flux for vertex-centered scheme.

which is also second-order accurate [65]. Here M_{ij} is the mid-point of the primary edge joining vertices i and j, as shown in Figure 9.6. The flux evaluation at the mid-point M_{ij} is approximated by the central flux

$$\mathbf{F}_{ij} := \mathbf{F}(\mathbf{U}_i, \mathbf{U}_j, \mathbf{n}_{ij}) = \frac{1}{2}\Big[\mathbf{F}(\mathbf{U}_i, \mathbf{n}_{ij}) + \mathbf{F}(\mathbf{U}_j \mathbf{n}_{ij})\Big]. \tag{9.12}$$

Assuming sufficient smoothness, we expand the cell average in Ω_i about the vertex i

$$\mathbf{U}_i(t) = \mathbf{U}(\mathbf{x}_i, t) + \frac{1}{|\Omega_i|}\int_{\Omega_i} \nabla_{\mathbf{x}}\mathbf{U}(\mathbf{x}_i, t)\cdot(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x} + \mathcal{O}(h^2).$$

Unfortunately, the vertex i need not be the barycenter of the dual cell Ω_i, thus

$$\frac{1}{|\Omega_i|}\int_{\Omega_i}(\mathbf{x}-\mathbf{x}_i)\mathrm{d}\mathbf{x} \neq 0,$$

in general. Thus, unlike the cell-centered approach, we only have a first-order approximation

$$\mathbf{U}_i(t) = \mathbf{U}(\mathbf{x}_i, t) + \mathcal{O}(h),$$

which in turn leads to the following first-order approximation when used with the central flux

$$\mathbf{F}(\mathbf{U}_i, \mathbf{U}_j, \mathbf{n}_{ij}) = \frac{1}{2}(\mathbf{F}(\mathbf{U}(\mathbf{x}_i, t), \mathbf{n}_{ij}) + \mathbf{F}(\mathbf{U}(\mathbf{x}_j, t), \mathbf{n}_{ij})) + \mathcal{O}(h). \tag{9.13}$$

Additionally

$$\begin{aligned}\mathbf{F}\big(\mathbf{U}(\mathbf{x}_i, t), \mathbf{n}_{ij}\big) &= \mathbf{F}\big(\mathbf{U}(\mathbf{x}_{M_{ij}}, t), \mathbf{n}_{ij}\big) + \nabla_{\mathbf{x}}\mathbf{F}\big(\mathbf{U}(M_{ij}, t), \mathbf{n}_{ij}\big)\cdot(\mathbf{x}_i - \mathbf{x}_{M_{ij}}) + \mathcal{O}(h^2),\\ \mathbf{F}\big(\mathbf{U}(\mathbf{x}_j, t), \mathbf{n}_{ij}\big) &= \mathbf{F}\big(\mathbf{U}(\mathbf{x}_{M_{ij}}, t), \mathbf{n}_{ij}\big) + \nabla_{\mathbf{x}}\mathbf{F}\big(\mathbf{U}(M_{ij}, t), \mathbf{n}_{ij}\big)\cdot(\mathbf{x}_j - \mathbf{x}_{M_{ij}}) + \mathcal{O}(h^2).\end{aligned} \tag{9.14}$$

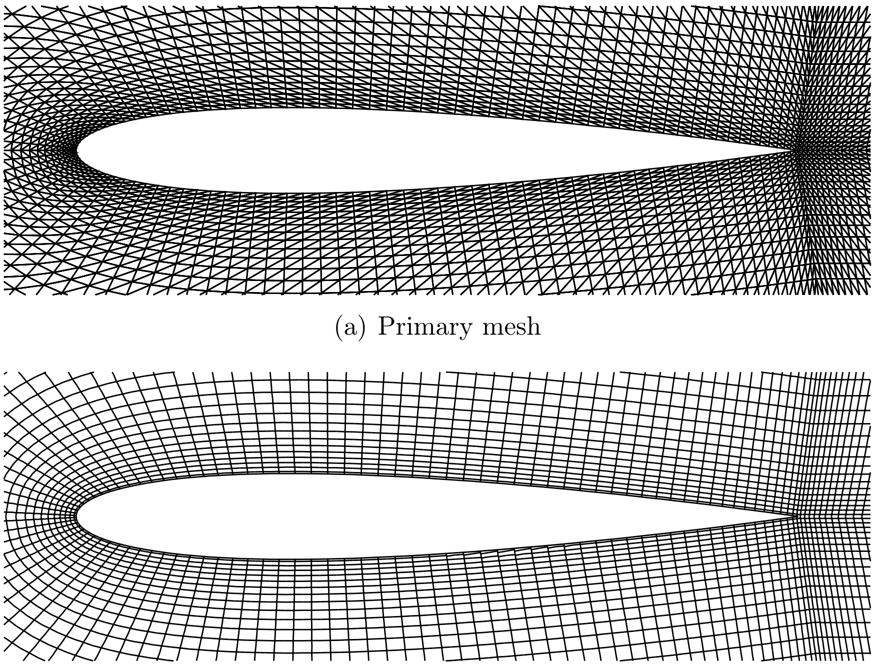
Since the vertices i, j and the face mid-point M_{ij} are collinear with $|\mathbf{x}_i - \mathbf{x}_{M_{ij}}| = |\mathbf{x}_{M_{ij}} - \mathbf{x}_j|$, the $\mathcal{O}(h)$ contributions in (9.14) will cancel out when the two expansions are added together. However, we are still left with the $\mathcal{O}(h)$ term in (9.13) which appeared as a result of the vertices not corresponding to the barycenters of the dual cells. Thus, we only have

$$\mathbf{F}(\mathbf{U}_i, \mathbf{U}_j, \mathbf{n}_{ij}) = \mathbf{F}(\mathbf{U}(\mathbf{x}_{M_{ij}}, t), \mathbf{n}_{ij}) + \mathcal{O}(h).$$

We conclude from the above discussion that for an arbitrary unstructured mesh, both cell-centered and vertex-centered approaches lead to first-order schemes when central fluxes are used. However, the first-order error terms may cancel out if the mesh is better behaved, leading to a second-order global spatial error. The vertex-centered approach has a big advantage when the viscous components of Navier-Stokes are included in the formulation (see Chapter 10). Firstly, the stencil to obtain the viscous fluxes is more compact. Secondly, the (voronoi) dual cells can be used to generate flat rectangular cells, which are crucial in approximating solutions in boundary layers. An example of a flat celled mesh for the NACA-0012 airfoil is shown in Figure 9.7. For the rest of this chapter, we use the vertex-centered scheme

$$|\Omega_i| \frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = -\sum_{j \in i} \mathbf{F}_{ij}, \tag{9.15}$$

with $\mathbf{F}_{ij}$ being a consistent and conservative numerical flux.



(a) Primary mesh

(b) Voronoi dual mesh

Figure 9.7: Mesh for a NACA-0012 airfoil. The dual cells of the Voronoi mesh near the boundary of the airfoil are flat and rectangular, making it suitable for approximating boundary layers in viscous flows.

9.3 Entropy conservative and entropy stable schemes

Having laid the groundwork for finite volume schemes on unstructured meshes, we now aim to construct entropy stable schemes to approximate (2.2). In other words, we want the numerical scheme to satisfy a discrete version of the entropy condition (2.9). This has already been discussed in Chapter 5 for finite difference schemes on Cartesian meshes. An analogous formulation can be made for finite volume schemes. Following the approach of Tadmor [115] and the recent paper [74], the first step is the design of an entropy conservative finite volume scheme.

9.3.1 Entropy conservative scheme

Definition 9.3.1. *The numerical scheme* (9.15) *is said to be entropy conservative if it satisfies the discrete entropy relation*

$$\frac{\mathrm{d}\eta(\mathbf{U}_i)}{\mathrm{d}t} + \frac{1}{|\Omega_i|}\sum_{j\in i} q^*_{ij} = 0, \tag{9.16}$$

where q^*_{ij} is a consistent numerical entropy flux.

We introduce the notations

$$\Delta(\cdot)_{ij} = (\cdot)_j - (\cdot)_i, \qquad \overline{(\cdot)}_{ij} = \frac{(\cdot)_i + (\cdot)_j}{2},$$

and the entropy potential

$$\Psi(\mathbf{U},\mathbf{n}) := \langle \mathbf{V}(\mathbf{U}), \mathbf{F}(\mathbf{U},\mathbf{n})\rangle - q(\mathbf{U},\mathbf{n}), \tag{9.17}$$

where $q(\mathbf{U},\mathbf{n})$ is defined in (3.13). The following theorem gives a sufficient condition to construct an entropy conservative numerical flux, which is a variant of the result proved for cell-centered schemes in [74].

Theorem 9.3.1. *The numerical scheme* (9.15) *with the flux* $\mathbf{F}^*$ *is entropy conservative if*

$$\left\langle \Delta\mathbf{V}_{ij}, \mathbf{F}^*_{ij}\right\rangle = \Psi(\mathbf{U}_j,\mathbf{n}_{ij}) - \Psi(\mathbf{U}_i,\mathbf{n}_{ij}). \tag{9.18}$$

Specifically, it satisfies (9.16) *with numerical entropy flux given by*

$$q^*_{ij} = q^*(\mathbf{U}_i,\mathbf{U}_j,\mathbf{n}_{ij}) = \left\langle \overline{\mathbf{V}}_{ij}, \mathbf{F}^*_{ij}\right\rangle - \frac{1}{2}\left(\Psi(\mathbf{U}_j,\mathbf{n}_{ij}) + \Psi(\mathbf{U}_i,\mathbf{n}_{ij})\right).$$

Proof. Taking the inner-product of (9.15) with the entropy variables $\mathbf{V}_i$, we get

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{U}_i) &= -\frac{1}{|\Omega_i|}\sum_{j\in i}\left\langle \mathbf{V}_i, \mathbf{F}^*_{ij}\right\rangle \\
&= -\frac{1}{|\Omega_i|}\sum_{j\in i}\left\langle\left(\overline{\mathbf{V}}_{ij} - \frac{1}{2}\Delta\mathbf{V}_{ij}\right), \mathbf{F}^*_{ij}\right\rangle \\
&= -\frac{1}{|\Omega_i|}\sum_{j\in i}\left(\left\langle\overline{\mathbf{V}}_{ij}, \mathbf{F}^*_{ij}\right\rangle - \frac{1}{2}\left(\Psi(\mathbf{U}_j, \mathbf{n}_{ij}) - \Psi(\mathbf{U}_i, \mathbf{n}_{ij})\right)\right) \\
&= -\frac{1}{|\Omega_i|}\sum_{j\in i} q^*_{ij} - \frac{1}{|\Omega_i|}\sum_{j\in i}\Psi(\mathbf{U}_i, \mathbf{n}_{ij}) \\
&= -\frac{1}{|\Omega_i|}\sum_{j\in i} q^*_{ij},
\end{aligned}$$

since we can show that $\sum_{j\in i}\Psi(\mathbf{U}_i, \mathbf{n}_{ij}) = 0$ using (9.2). □

Remark 9.3.1. *The KEPEC and ROE-EC fluxes introduced in Chapter 5 for finite difference schemes, also serve as important examples of entropy conservative fluxes for finite volume schemes, since they satisfy* (9.18)*. Their expressions in the context of finite volume schemes are given in Appendix C.*

Remark 9.3.2. *The condition for kinetic energy preservation can be extended to finite volume schemes on unstructured grids [59] as*

$$\mathbf{F}^m = p\mathbf{n} + \overline{\mathbf{u}}F^\rho, \tag{9.19}$$

where p and F^ρ are any consistent approximations for pressure and the mass flux respectively. Since both KEP and KEPEC fluxes (see Appendix C) satisfy (9.19)*, they are kinetic energy preserving.*

9.3.2 First order entropy stable scheme

While entropy is conserved for smooth solutions, it is dissipated near discontinuities in accordance to the entropy condition (2.9). Hence, we introduce additional dissipation terms to construct entropy stable schemes.

Definition 9.3.2. *The numerical scheme* (9.15) *is said to be* entropy stable *if it satisfies the discrete entropy relation*

$$\frac{d\eta(\mathbf{U}_i)}{dt} + \frac{1}{|\Omega_i|}\sum_{j\in i} q_{ij} \leqslant 0, \tag{9.20}$$

where q_{ij} is a consistent numerical entropy flux.

Entropy stable schemes can be characterized by satisfaction of an *E-flux* condition, which is a generalized extension of Osher's E-flux condition [83] to system of conservation laws. This has been proved for the cell-centered setup in [74], which we adapt for vertex-centered schemes.

Theorem 9.3.2. *The semi-discrete numerical scheme* (9.15) *with numerical flux* $\mathbf{F}_{ij}$ *satisfying the E-flux condition*

$$\langle \Delta \mathbf{V}_{ij}, \mathbf{F}_{ij} \rangle \leqslant \Psi(\mathbf{U}_j, \mathbf{n}_{ij}) - \Psi(\mathbf{U}_i, \mathbf{n}_{ij}), \tag{9.21}$$

is entropy stable; specifically, it satisfies the discrete entropy inequality (9.20) *with numerical entropy flux given by*

$$q_{ij} = \left\langle \overline{\mathbf{V}}_{ij}, \mathbf{F}_{ij} \right\rangle - \frac{1}{2} \left(\Psi(\mathbf{U}_j, \mathbf{n}_{ij}) + \Psi(\mathbf{U}_i, \mathbf{n}_{ij}) \right).$$

Proof. Taking the inner-product of (9.15) with $\mathbf{V}_i$ and following the algebraic manipulations similar to those in Theorem 9.3.1, we get

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{U}_i) &= -\frac{1}{|\Omega_i|} \sum_{j \in i} \langle \mathbf{V}_i, \mathbf{F}_{ij} \rangle \\
&= -\frac{1}{|\Omega_i|} \sum_{j \in i} \left[\left\langle \overline{\mathbf{V}}_{ij}, \mathbf{F}_{ij} \right\rangle - \frac{1}{2} \langle \Delta \mathbf{V}_{ij}, \mathbf{F}_{ij} \rangle \right] \\
&\leqslant -\frac{1}{|\Omega_i|} \sum_{j \in i} \left[\left\langle \overline{\mathbf{V}}_{ij}, \mathbf{F}_{ij} \right\rangle - \frac{1}{2} \left(\Psi(\mathbf{U}_j, \mathbf{n}_{ij}) - \Psi(\mathbf{U}_i, \mathbf{n}_{ij}) \right) \right] \\
&= -\frac{1}{|\Omega_i|} \sum_{j \in i} q_{ij}.
\end{aligned}$$

□

Several such entropy stable fluxes satisfying the E-flux have been proposed in [6]. Alternately, we can consider the approach proposed by Tadmor [116, 35, 74], where entropy variable based numerical dissipation is augmented to the entropy conservative numerical flux $\mathbf{F}^*_{ij}$ in the form

$$\mathbf{F}_{ij} = \mathbf{F}^*_{ij} - \frac{1}{2} \mathbf{D}_{ij} \Delta \mathbf{V}_{ij}, \tag{9.22}$$

for a symmetric and positive semi-definite matrix $\mathbf{D}_{ij}$. The diffusion matrix must also satisfy $\mathbf{D}_{ij} = \mathbf{D}_{ji}$ to ensure that the numerical flux is conservative. Clearly (9.22) satisfies the E-flux condition, provide $\mathbf{F}^*_{ij}$ satisfies (9.18). As discussed in Section 5.5.1, we choose the matrix $\mathbf{D}_{ij}$ of the form

$$\mathbf{D}_{ij} = \mathbf{R}_{ij} \mathbf{\Lambda}_{ij} \mathbf{R}^\top_{ij}, \tag{9.23}$$

where $\mathbf{R} = \mathbf{R}(\mathbf{U}, \mathbf{n})$ is the matrix of scaled eigenvectors of the flux Jacobian $\partial_{\mathbf{U}} \mathbf{F}(\mathbf{U}, \mathbf{n})$ and $\mathbf{\Lambda} = \mathbf{\Lambda}(\mathbf{U}, \mathbf{n})$ is the non-negative diagonal matrix depending on the eigenvalues of the flux Jacobian. In particular, we choose the Roe-type dissipation operator. The expression for the dissipation operator in the context of finite volume schemes is given in Appendix C.

9.3.3 High-order diffusion operators

Since $\Delta \mathbf{V}_{ij} = \mathcal{O}(h)$ for smooth solutions, the numerical flux (9.22) leads to a first-order accurate scheme. A higher-order scheme can be obtained by suitably reconstructing the solution to the cell interfaces. Consider the cell interface between two control volumes

Ω_i and Ω_j. Corresponding to this particular cell interface, let $\mathbf{V}_{ij}$ and $\mathbf{V}_{ji}$ be the reconstructed values of $\mathbf{V}$ from cell Ω_i and Ω_j respectively, and define the jump at the interface by

$$[\![\mathbf{V}]\!]_{ij} := \mathbf{V}_{ji} - \mathbf{V}_{ij}. \tag{9.24}$$

Using the higher order jump (9.24) instead of $\Delta\mathbf{V}_{ij}$ in the numerical flux (9.22), leads to a high-order flux. The following lemma (proved for Cartesian meshes in Section 5.5.2) gives a sufficient condition for the reconstruction, ensuring that the entropy stability of the scheme is retained.

Lemma 9.3.1. *For each pair of vertices (i, j) which are connected to one another by a primary edge, let $\mathbf{R}_{ij}$ be non-singular, let $\mathbf{\Lambda}_{ij}$ be any non-negative diagonal matrix, and define the numerical diffusion matrix*

$$\mathbf{D}_{ij} = \mathbf{R}_{ij}\mathbf{\Lambda}_{ij}\mathbf{R}_{ij}^\top.$$

Let $\mathbf{V}_{ij}$ and $\mathbf{V}_{ji}$ be the reconstructed values of the entropy variables at the interface between Ω_i and Ω_j. Assume that the reconstruction ensures the existence of a diagonal matrix $\mathbf{B}_{ij} \geqslant 0$ such that

$$[\![\mathbf{V}]\!]_{ij} = \left(\mathbf{R}_{ij}^\top\right)^{-1}\mathbf{B}_{ij}\mathbf{R}_{ij}^\top\Delta\mathbf{V}_{ij}. \tag{9.25}$$

Then the scheme with the numerical flux

$$\mathbf{F}_{ij} = \mathbf{F}_{ij}^* - \frac{1}{2}\mathbf{D}_{ij}[\![\mathbf{V}]\!]_{ij}, \tag{9.26}$$

is entropy stable with numerical entropy flux

$$q_{ij} := q_{ij}^* - \frac{1}{2}\overline{\mathbf{V}}_{ij}^\top\mathbf{D}_{ij}[\![\mathbf{V}]\!]_{ij}.$$

Proof. As in the proof of Lemma 9.3.2, consider (9.15) with the flux defined by (9.26), and take the inner-product with the entropy variables $\mathbf{V}_i$ to get

$$\begin{aligned}
\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{U}_i) &= -\frac{1}{|C_i|}\sum_{j\in i}\langle\mathbf{V}_i, \mathbf{F}_{ij}\rangle \\
&= -\frac{1}{|C_i|}\sum_{j\in i} q_{ij}^* + \frac{1}{2|C_i|}\sum_{j\in\mathcal{N}_i}\left\langle\overline{\mathbf{V}}_{ij} - \frac{1}{2}\Delta\mathbf{V}_{ij}, \mathbf{D}_{ij}[\![\mathbf{V}]\!]_{ij}\right\rangle \\
&= -\frac{1}{|C_i|}\sum_{j\in i}\left[q_{ij}^* - \frac{1}{2}\left\langle\overline{\mathbf{V}}_{ij}, \mathbf{D}_{ij}[\![\mathbf{V}]\!]_{ij}\right\rangle\right] \\
&\quad - \frac{1}{4|C_i|}\sum_{j\in i}\left\langle\Delta\mathbf{V}_{ij}, \mathbf{R}_{ij}\mathbf{\Lambda}_{ij}\mathbf{B}_{ij}\mathbf{R}_{ij}^\top\Delta\mathbf{V}_{ij}\right\rangle.
\end{aligned}$$

Since $\mathbf{R}_{ij}\mathbf{\Lambda}_{ij}\mathbf{B}_{ij}\mathbf{R}_{ij}^\top$ is symmetric and positive semi-definite, we get

$$\frac{\mathrm{d}}{\mathrm{d}t}\eta(\mathbf{U}_i) + \frac{1}{|C_i|}\sum_{j\in i} q_{ij} \leqslant 0.$$

□

Remark 9.3.3. *The quantities $\mathbf{R}_{ij}$, $\mathbf{\Lambda}_{ij}$ are evaluated at some average value corresponding to $\mathbf{V}_i$, $\mathbf{V}_j$. Note that $\mathbf{F}^*_{ij} = \mathbf{F}^*(\mathbf{V}_i, \mathbf{V}_j, \mathbf{n}_{ij})$, i.e., it is evaluated using the solution at the vertices and only the dissipation flux makes use of the reconstructed values.*

Remark 9.3.4. *On Cartesian grids, the interpolation formula* (5.14) *can be used to construct high-order entropy conservative fluxes. However, no such formula is known at present in the context of entropy conservative finite volume fluxes on unstructured meshes. The entropy conservative fluxes on unstructured grids are only (formally) second-order accurate. Thus, we restrict ourselves to the linear reconstruction of entropy variables at the interface, to obtain second-order dissipation operators.*

9.3.4 Reconstruction procedure and the sign-property

In order to use Lemma 9.3.1, we describe a reconstruction procedure that satisfies (9.25). As done in Section 5.5.3, we define by $\mathbf{Z} = \mathbf{R}_{ij}^\top \mathbf{V}$ the scaled entropy variables for the interface between the neighbouring cells Ω_i and Ω_j. Let $\mathbf{Z}_{ij}$, $\mathbf{Z}_{ji}$ be the reconstructed values of $\mathbf{Z}$ at the interface from cell Ω_i and Ω_j respectively. We further define

$$\mathbf{V}_{ij} = (\mathbf{R}_{ij}^\top)^{-1}\mathbf{Z}_{ij}, \qquad \mathbf{V}_{ji} = (\mathbf{R}_{ij}^\top)^{-1}\mathbf{Z}_{ji} \qquad \Longrightarrow \qquad [\![\mathbf{V}]\!]_{ij} = (\mathbf{R}_{ij}^\top)^{-1}[\![\mathbf{Z}]\!]_{ij}.$$

Thus, the dissipation terms in the flux given by (9.26) can be written as $\mathbf{D}_{ij}[\![\mathbf{V}]\!]_{ij} = \mathbf{R}_{ij}\mathbf{\Lambda}_{ij}[\![\mathbf{Z}]\!]_{ij}$, with the condition (9.25) reducing to the component-wise sign-property $\mathbf{Z}$:

$$\operatorname{sign}\Big([\![\mathbf{Z}]\!]_{ij}\Big) = \operatorname{sign}\Big(\Delta \mathbf{Z}_{ij}\Big). \tag{9.27}$$

We describe a slope-limited linear reconstruction procedure of scaled entropy variables appearing in the dissipation terms, which satisfies the sign-property. For neighbouring control volumes Ω_i and Ω_j, the scaled entropy variables with respect to the interface between vertices i and j are given by

$$\mathbf{Z}_i = \mathbf{R}_{ij}^\top \mathbf{V}_i, \qquad \mathbf{Z}_j = \mathbf{R}_{ij}^\top \mathbf{V}_j. \tag{9.28}$$

In order to perform the reconstruction, we need more information along the line joining vertices i and j, so that we can get some information about the smoothness of the function. Let us extend the line by an equal length on either side, to obtain the additional vertices $i-1$ and $j+1$ (Figure 9.8(a)). Assuming the values $\mathbf{Z}_{i-1}$, $\mathbf{Z}_{j+1}$ are known, we define following differences

- The forward differences

$$\Delta^f_{ij} = \Delta \mathbf{Z}_{ij}, \qquad \Delta^f_{ji} = \mathbf{Z}_{j+1} - \mathbf{Z}_j. \tag{9.29}$$

- The backward differences

$$\Delta^b_{ij} = \mathbf{Z}_i - \mathbf{Z}_{i-1} \qquad \Delta^b_{ji} = \Delta \mathbf{Z}_{ij}. \tag{9.30}$$

The reconstructed values of $\mathbf{Z}$ at the interface are given by

$$\mathbf{Z}_{ij} = \mathbf{Z}_i + \frac{1}{2}\mathcal{M}\left(\Delta^f_{ij}, \Delta^b_{ij}\right), \qquad \mathbf{Z}_{ji} = \mathbf{Z}_j - \frac{1}{2}\mathcal{M}\left(\Delta^f_{ji}, \Delta^b_{ji}\right). \tag{9.31}$$

where we have used the minmod slope limiter function given by (4.9). There are several methods available in the literature to obtain the additional information $\mathbf{Z}_{i-1}$ and $\mathbf{Z}_{j+1}$, which need not correspond to actual points in the mesh.

- The values at the vertices $i-1$ and $j+1$ can be evaluated through continuation and interpolation from neighbouring vertices [11], as shown in Figure 9.8(a).
- The differences Δ^b_{ij} and Δ^f_{ji} can be estimated if we know the gradients of $\mathbf{Z}$ at the vertices [133].
- For the edge joining the vertices i and j, one considers the *upstream* and *downstream* triangles T_{ij} and T_{ji} through which the extended edge would pass (see Figure 9.8(b)). The gradients evaluated on these triangles can be used instead of gradients at the vertices [11, 99, 4].

In our reconstruction procedure, we use vertex gradients to evaluate the differences as follows

$$\Delta^f_{ji} = \mathbf{Z}_{j+1} - \mathbf{Z}_j = 2\nabla^h \mathbf{Z}_j \cdot (\mathbf{x}_j - \mathbf{x}_i) - \Delta \mathbf{Z}_{ij}, \tag{9.32}$$

$$\Delta^b_{ij} = \mathbf{Z}_i - \mathbf{Z}_{i-1} = 2\nabla^h \mathbf{Z}_i \cdot (\mathbf{x}_j - \mathbf{x}_i) - \Delta \mathbf{Z}_{ij}. \tag{9.33}$$

The approximation of the gradients at the vertices, is described in Section 9.3.5.

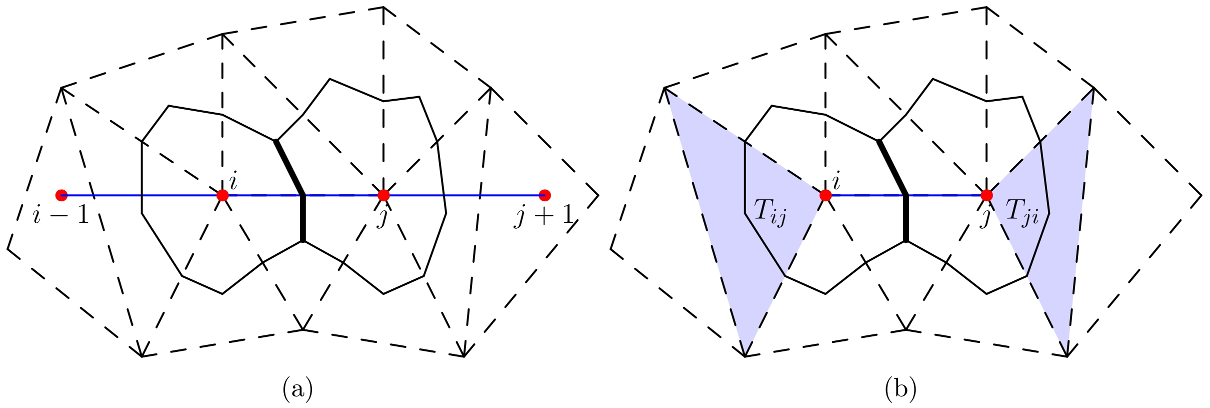


Figure 9.8: Stencil for linear reconstruction (a) extension and interpolation, (b) extension into upstream and downstream triangles.

Lemma 9.3.2. *The reconstruction of the scaled entropy variables described by* (9.31), (9.29) *and* (9.30) *satisfies the sign property* (9.27).

Proof. For any component Z of $\mathbf{Z}$, the reconstruction scheme gives

$$Z_{ji} - Z_{ij} = (Z_j - Z_i) - \frac{1}{2}\left[\mathcal{M}\left(\Delta^f_{ji}, \Delta^b_{ji}\right) + \mathcal{M}\left(\Delta^f_{ij}, \Delta^b_{ij}\right)\right].$$

If $Z_j - Z_i \geqslant 0$, then

$$\mathcal{M}\left(\Delta^f_{ij}, \Delta^b_{ij}\right) \leqslant \Delta^f_{ij}, \qquad \mathcal{M}\left(\Delta^f_{ji}, \Delta^b_{ji}\right) \leqslant \Delta^b_{ji} = \Delta^f_{ij}.$$

Thus,

$$Z_{ji} - Z_{ij} \geqslant (Z_j - Z_i) - \frac{1}{2}\left[2\Delta^f_{ij}\right] = 0.$$

Similarly, if $Z_j - Z_i \leqslant 0$, then

$$\mathcal{M}\left(\Delta^f_{ij}, \Delta^b_{ij}\right) \geqslant \Delta^f_{ij}, \qquad \mathcal{M}\left(\Delta^f_{ji}, \Delta^b_{ji}\right) \geqslant \Delta^b_{ji} = \Delta^f_{ij},$$

giving us

$$Z_{ji} - Z_{ij} \leqslant (Z_j - Z_i) - \frac{1}{2}\left[2\Delta^f_{ij}\right] = 0.$$

Hence, the reconstruction satisfies the sign property. □

Remark 9.3.5. *The above reconstruction with the minmod limiter is one possible option which has the sign-property. One could instead use the second-order ENO scheme (ENO-2), which also satisfies the sign-property. Note that the ENO-2 scheme reduces to the minabs limiter. Numerical tests have yielded almost indistinguishable results with both minmod and ENO-2 reconstruction. Thus, we adhere to presenting results with the minmod limiter.*

Remark 9.3.6. *Either of the reconstruction methods described above would lead to the sign-property, when used with the minmod or minabs limiter. What is crucial to obtain the sign-property is that we evaluate $\Delta^f_{ij} = \Delta^b_{ji} = \Delta\mathbf{Z}_{ij}$.*

9.3.5 Computation of gradients

The second-order limited reconstruction described above requires the evaluation of vertex gradients of scaled entropy variables. We evaluate these gradient as

$$\nabla^h \mathbf{Z}_i = \mathbf{R}^\top_{ij} \nabla^h \mathbf{V}_i, \tag{9.34}$$

where $\nabla^h \mathbf{V}_i$ must be numerically approximated.

Consider the vertex i and the set of neighbouring primary triangular cells denoted by $T \in i$. Using the Green's theorem combined with the trapezoidal rule for integration [26, 1], the gradient of a scalar valued function ϕ on each triangle T is approximated by

$$\begin{aligned}\nabla^h \phi^T &= \frac{1}{|T|}\left[\frac{(\phi_i + \phi_j)}{2}\mathbf{n}^T_k + \frac{(\phi_j + \phi_k)}{2}\mathbf{n}^T_i + \frac{(\phi_k + \phi_i)}{2}\mathbf{n}^T_j\right] \\ &= -\frac{1}{2|T|}\left[\phi_i \mathbf{n}^T_i + \phi_j \mathbf{n}^T_j + \phi_k \mathbf{n}^T_k\right].\end{aligned} \tag{9.35}$$

This approximation is exact for affine functions, and thus second-order accurate. Similarly, the gradient approximation of a vector valued function $\boldsymbol{\phi} \in \mathbb{R}^m$, is given by

$$\begin{aligned}\nabla^h \boldsymbol{\phi}^T &= \frac{1}{|T|}\left[\frac{(\boldsymbol{\phi}_i + \boldsymbol{\phi}_j)}{2} \otimes \mathbf{n}^T_k + \frac{(\boldsymbol{\phi}_j + \boldsymbol{\phi}_k)}{2} \otimes \mathbf{n}^T_i + \frac{(\boldsymbol{\phi}_k + \boldsymbol{\phi}_i)}{2} \otimes \mathbf{n}^T_j\right] \\ &= -\frac{1}{2|T|}\left[\boldsymbol{\phi}_i \otimes \mathbf{n}^T_i + \boldsymbol{\phi}_j \otimes \mathbf{n}^T_j + \boldsymbol{\phi}_k \otimes \mathbf{n}^T_k\right].\end{aligned} \tag{9.36}$$

Finally, the gradient at vertex i is approximated for the scalar ϕ or the vector $\boldsymbol{\phi}$ by

$$\nabla^h \phi_i = \frac{\sum\limits_{T\in i} |T| \nabla^h \phi^T}{\sum\limits_{T\in i} |T|}, \qquad \nabla^h \boldsymbol{\phi}_i = \frac{\sum\limits_{T\in i} |T| \nabla^h \boldsymbol{\phi}^T}{\sum\limits_{T\in i} |T|}, \tag{9.37}$$

respectively. Note that the expression in (9.37) are also exact for affine functions and hence second-order accurate. The gradient of the entropy variables at the vertices are approximated using (9.36) and (9.37).

Remark 9.3.7. *In actual implementation of the scheme, we never compute $\mathbf{V}_{ij}$, $\mathbf{V}_{ji}$, which would be expensive since it requires the inversion of the matrix $\mathbf{R}_{ij}$. The numerical flux can be directly computed as*

$$\mathbf{F}_{ij} = \mathbf{F}^*_{ij} - \frac{1}{2}\mathbf{R}_{ij}\mathbf{\Lambda}_{ij}(\mathbf{Z}_{ji} - \mathbf{Z}_{ij}), \tag{9.38}$$

thus avoiding some costly operations.

Remark 9.3.8. *One could also approximate $\nabla^h \mathbf{Z}_i$ and $\nabla^h \mathbf{Z}^T$ directly from the scaled entropy variables at each vertex. Since the scaling depends on the particular dual mesh interface at which the reconstruction is being performed, this would require the computation of several vertex and triangular gradients for each vertex and triangle. In order to avoid this additional computational cost and storage requirement, we simply scale the gradients evaluated for the original entropy variables, as given by* (9.34).

9.4 Numerical results

We now present the numerical results of the scheme discussed above on several standard two dimensional test cases. The numerical flux nomenclatures KEPEC, KEPES, KEPES-TeCNO and Roe have already been introduced in the beginning of Section 5.8. In addition we introduce the *KEPES2 flux*, which is identical to the KEPES-TeCNO flux, except the scaled entropy variables are reconstructed using an unlimited second order-reconstruction

$$\mathbf{Z}_{ij} = \mathbf{Z}_i + \frac{1}{2}\nabla^h \mathbf{Z}_i \cdot (\mathbf{x}_j - \mathbf{x}_i), \qquad \mathbf{Z}_{ji} = Z_j - \frac{1}{2}\nabla^h \mathbf{Z}_j \cdot (\mathbf{x}_j - \mathbf{x}_i).$$

Note that KEPES2 is not necessarily entropy stable, as the unlimited reconstruction need not satisfy the sign-property.

The semi-discrete scheme is integrated in time using the explicit SSP-RK3. The local time steps are evaluated as

$$\Delta t_i = \frac{\mathrm{CFL} \cdot |\Omega_i|}{\lambda_i}, \qquad \lambda_i = \sum_{j \in i} \left[|\mathbf{u}_i \cdot \mathbf{n}_{ij}| + a_i |\mathbf{n}_{ij}| \right].$$

In all test cases we consider the ideal gas with $\gamma = 1.4$, except when indicated otherwise.

9.4.1 Smooth density wave

In order to test the order of accuracy of the proposed schemes, we consider the simple problem of a smooth advecting wave on a domain $[0,1] \times [0,1]$ with periodic boundary conditions. The initial conditions are given by

$$\rho = 10 + \sin(2\pi x)\sin(2\pi y), \quad p = 5, \quad u_1 = u_2 = 1,$$

which corresponds to the ρ profile moving with a constant velocity $(1, 1)$ and constant pressure. The solution completes one full cycle at time $t_f = 1$. For the simulations, we choose CFL=0.4. To compute the order of accuracy, we consider a series of nested triangular primary meshes. The first mesh, denoted by $L0$, consists of 40 uniformly spaced vertices each in the horizontal and vertical directions. The base mesh is refined by splitting each primary triangle into four similar triangles to obtain mesh Ln, where n refers to the number of mesh refinements. With each refinement, the number of vertices are doubled in both the horizontal and vertical directions. Thus, mesh Ln contains $2^n \times 40$ vertices in each direction. The corresponding Voronoi dual meshes resemble Cartesian grids (for instance, see the mesh shown in Figure 9.10 for the shock-tube problem).

The rate of convergence for the schemes in the discrete L^1_h and L^∞_h norms are shown in Table 9.1. The central entropy conservative KEPEC scheme is second-order accurate due to the cancellation of first-order terms. The KEPES is only first-order accurate due to the $\mathcal{O}(h)$ jump term in the dissipation operator. The convergence rate improves to a certain degree with a minmod reconstruction of the jump in the KEPES-TeCNO scheme. The minmod limiter leads to clipping of smooth extrema [84, 85]. This becomes even more evident by observing the rate of convergence in L^∞_h norm. In the KEPES2 flux, an unlimited second-order reconstruction is used for the jump in the dissipation term. This helps in the recovery of full second-order accuracy, with the errors for KEPEC and KEPES2 being almost identical.

On unstructured meshes (with Voronoi dual grids), a purely central scheme such as KEPEC, is unable to completely filter out the small scale noise caused by dispersive errors, even with mesh refinement (see Figure 9.9). This leads to problems in convergence, giving an incorrect order of convergence. Artificial dissipation needs to be added for simulations on unstructured meshes, especially in the absence of any other form of physical diffusion. The analysis in Section 9.2.2 suggests that the central KEPEC flux is first-order accurate on general unstructured meshes. Thus, the KEPES and KEPES2 schemes (both of which use KEPEC as the central flux) give first-order convergence with mesh refinement, as indicated in Table 9.2, although KEPES2 gives much smaller errors.

9.4.2 Modified shock tube problem

This corresponds to the Sod type test case described in Section 5.8.2. We consider a rectangular domain $[0, 1] \times [0, 0.4]$ and discretize it with 100 vertices in the direction of the flow, and 80 vertices along the flow cross-section. The primary and Voronoi dual meshes used for the simulations are shown in Figure 9.10. The left state is given by $(\rho, u_1, u_2, p)_L = (1.0, 0.75, 0.0, 1.0)$ and the right state is given by $(\rho, u_1, u_2, p)_R = (0.125, 0.0, 0.0, 0.1)$, with the initial discontinuity along $x = 0.3$.

The Roe scheme gives an entropy violating jump in the expansion region where the flow becomes sonic, as shown in Figure 9.11. The entropy stable KEPES and KEPES-TeCNO schemes, are able to overcome this issue to a large extent, The comparison in Figure 9.12 shows that the solutions with KEPES-TeCNO are much better resolved as compared to KEPES. Convergence is demonstrated in Figure 9.13, where the solutions are evaluated using KEPES-TeCNO on three levels of uniform grid refinements, with the number of vertices along the streamwise direction being $N = 100, 200$ and 400 respectively.

The KEPES and KEPES-TeCNO also give rise to a small jump near the sonic point,

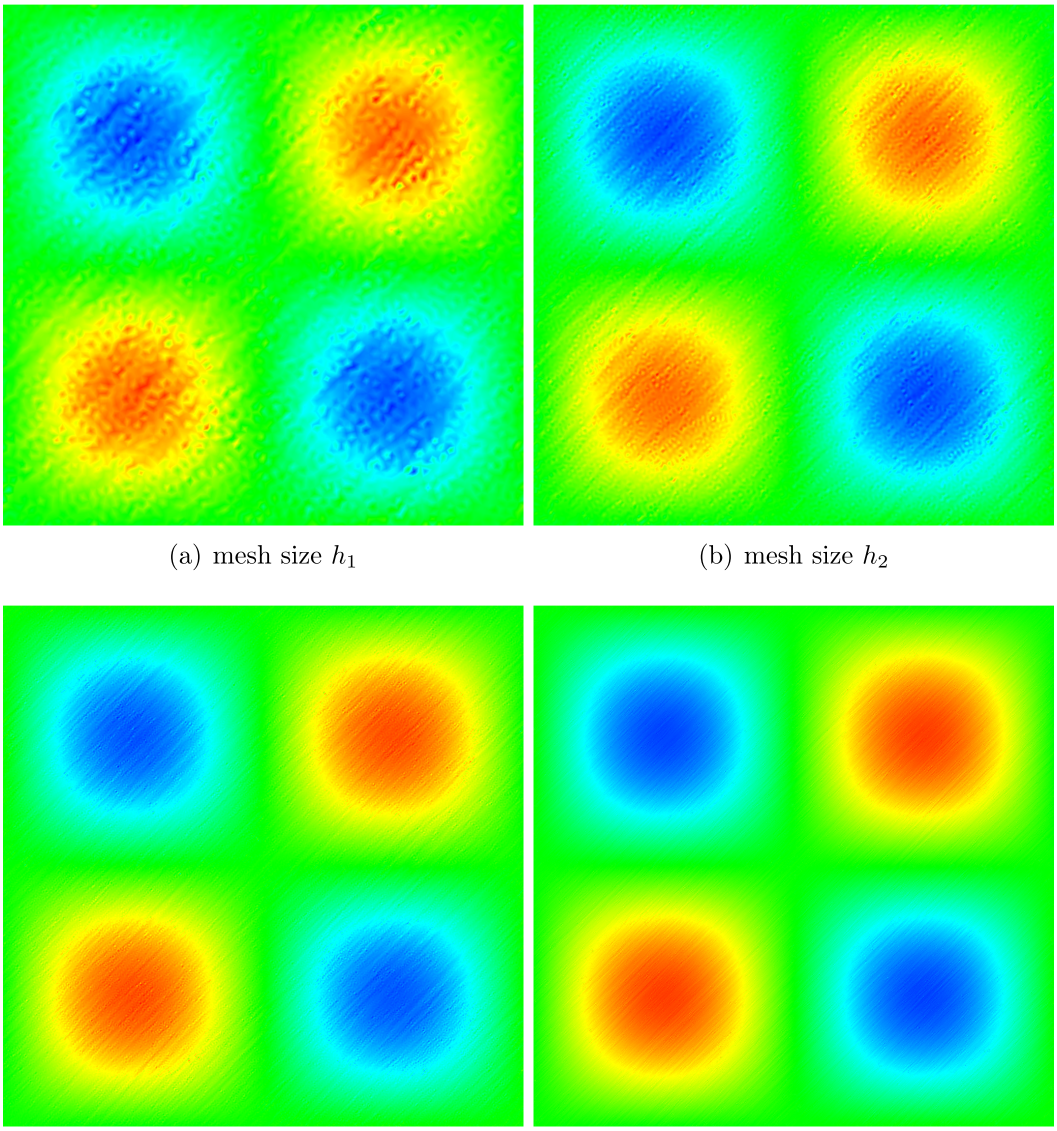
	KEPEC				KEPES			
N	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
40	1.72e-02	-	1.14e-01	-	2.57e-01	-	2.54e-00	-
80	4.21e-03	2.03	2.75e-02	2.05	1.59e-01	0.69	1.57e-00	0.69
160	1.04e-03	2.02	6.87e-03	2.00	8.91e-02	0.84	8.79e-01	0.84
320	2.58e-04	2.01	1.69e-03	2.02	4.72e-02	0.92	4.65e-01	0.92
640	6.44e-05	2.00	4.26e-04	1.99	2.43e-02	0.96	2.39e-01	0.96

	KEPES-TeCNO				KEPES2			
N	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
40	4.35e-02	-	7.32e-01	-	1.77e-02	-	1.38e-01	-
80	1.19e-02	1.86	3.18e-01	1.20	4.73e-03	2.06	3.00e-02	2.19
160	3.55e-03	1.75	1.32e-01	1.26	1.16e-03	2.02	7.01e-03	2.10
320	1.01e-03	1.81	5.43e-02	1.29	2.87e-04	2.01	1.69e-03	2.05
640	2.80e-04	1.85	2.20e-02	1.30	6.44e-05	2.00	4.26e-04	1.99

Table 9.1: Order of convergence for the smooth density wave problem.

	KEPES				KEPES2			
h	L_h^1		L_h^∞		L_h^1		L_h^∞	
	error	rate	error	rate	error	rate	error	rate
h_0	2.32e-01	-	2.17e-00	-	2.16e-02	-	2.35e-01	-
h_1	1.45e-01	0.68	1.31e-00	0.72	6.13e-03	1.81	9.98e-02	1.23
h_2	8.17e-02	0.83	7.33e-01	0.84	2.22e-03	1.47	5.27e-02	0.92
h_3	4.35e-02	0.91	3.89e-01	0.91	1.01e-03	1.14	2.69e-02	0.96
h_4	2.24e-02	0.95	2.00e-01	0.96	4.92e-04	1.03	1.36e-02	0.98

Table 9.2: Order of convergence for the smooth density wave problem on unstructured mesh, with base mesh size $h_0 = 2.5 \times 10^{-2}$ and refined mesh size $h_i = h_0/2^i$.



(a) mesh size h_1

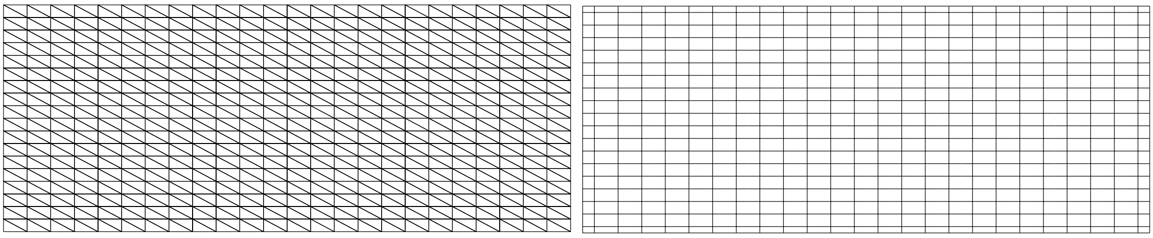
(b) mesh size h_2

(c) mesh size h_3

(d) mesh size h_4

Figure 9.9: Smooth density wave problem with KEPEC flux on unstructured mesh, with base mesh size $h_0 = 2.5 \times 10^{-2}$ and refined mesh size $h_i = h_0/2^i$.

which reduces with mesh refinement unlike the jump observed with the Roe scheme (see Figure 5.6). This jump could be attributed to the absence of the right amount of dissipation. Using the entropy consistent modification (5.24) can fix this issue, as shown in Figures 9.14 and 9.15. Focusing on the region near the sonic point in Figures 9.14(b) and 9.15(b), we observe that for $\alpha_{EC} = {}^{1}\!/_{6}$ the jump reduces significantly.



(a) Primal grid

(b) Voronoi dual grid

Figure 9.10: Grid used for shock tube problem.

9.4.3 Supersonic flow over wedge

This test case involves a weak oblique shock, which occurs when a supersonic flow is *turned into itself* due to the presence of a wedge. The wedge is inclined at an angle of 10 degrees to the horizontal. The farfield Mach number is 2, with slip boundary conditions on the wedge. The mesh (see Figure 9.16) has 18848 vertices and we use median dual cells as control volumes. As can be seen in Figure 9.17, the shock profile is quite dissipated with KEPES. But, the minmod reconstruction in KEPES-TeCNO scheme leads to a much sharper shock profile, that is comparable to the one computed by the Roe scheme with MUSCL type reconstruction and van Albada limiter (see Section 4.1.4).

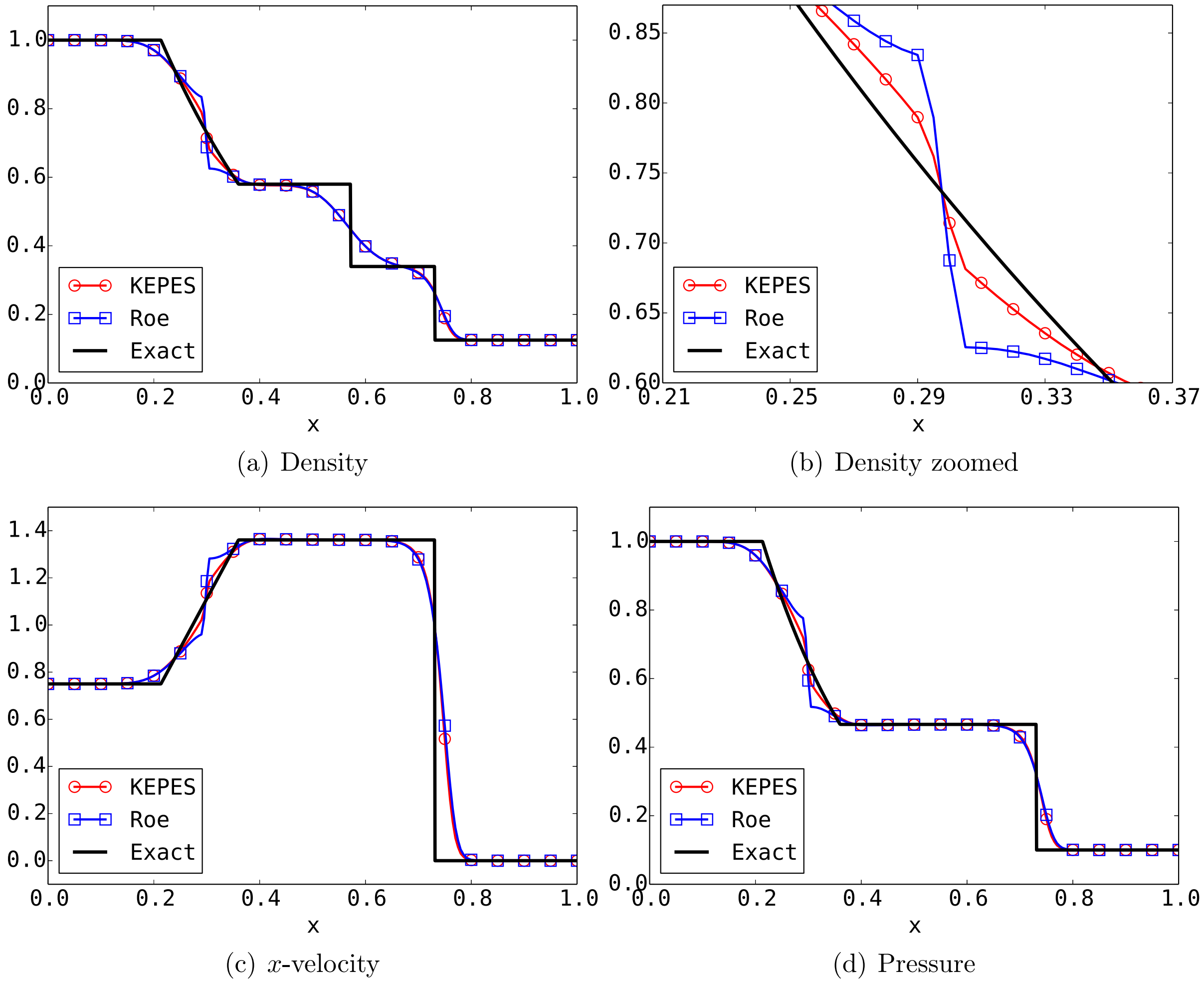
9.4.4 Transonic flow past NACA-0012 airfoil

This is an example of a symmetric NACA-0012 airfoil placed in a freestream Mach number of 0.85, with an angle of attack of 2 degrees. A zoomed view of the primary mesh and the corresponding median dual mesh used for this test case, is shown in Figure 9.18. The mesh contains 180 points on the airfoil surface, and 20 points on the farfield boundary which is a circle, with a total of 6402 vertices. The flow develops shocks both on the upper and lower airfoil surfaces. The Mach contour plots in Figure 9.19 show that KEPES-TeCNO gives much better shock resolution than KEPES, and comparable to the high-resolution Roe-MUSCL scheme.

The pressure coefficient for compressible flows is given by

$$\mathrm{C}_p = \frac{2}{\gamma \mathrm{M}_\infty^2}\left(\frac{p}{p_\infty} - 1\right),$$

where p is the pressure at the vertices, while p_∞ and M_∞ are the farfield pressure and Mach numbers respectively. We consider the vertex values of C_p on the surface of the airfoil, as shown in Figure 9.20. The x-axis represents the normalized wingspan, while



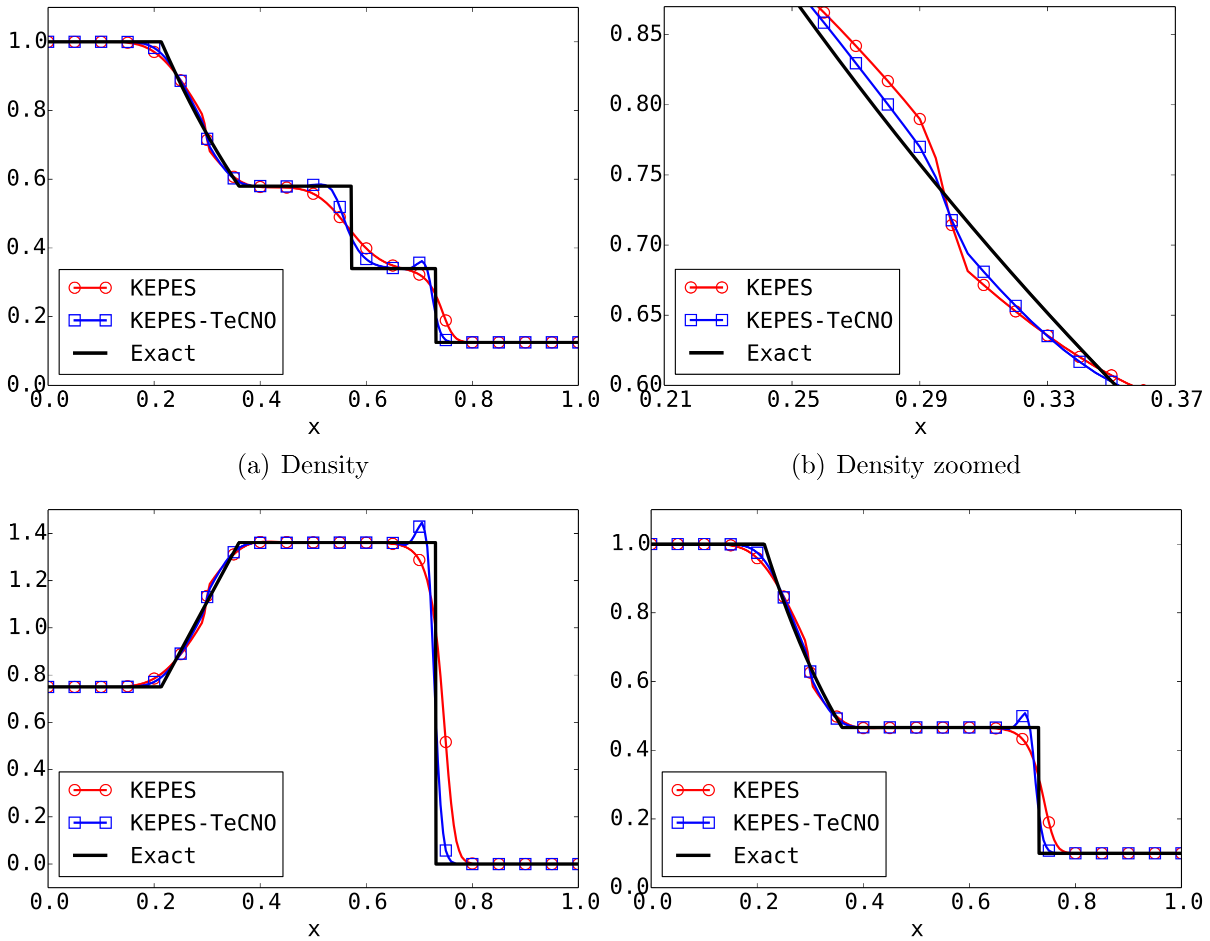
(a) Density

(b) Density zoomed

(c) x-velocity

(d) Pressure

Figure 9.11: Modified shock tube problem using first order schemes.



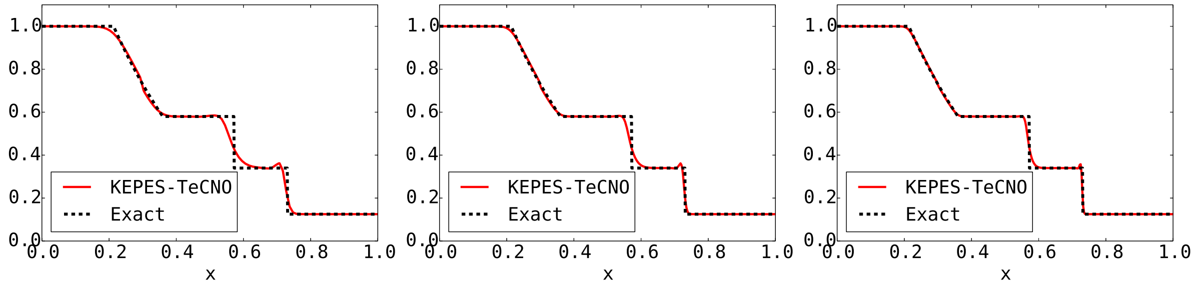
(a) Density

(b) Density zoomed

(c) x-velocity

(d) Pressure

Figure 9.12: Comparison of KEPES and KEPES-TeCNO schemes.

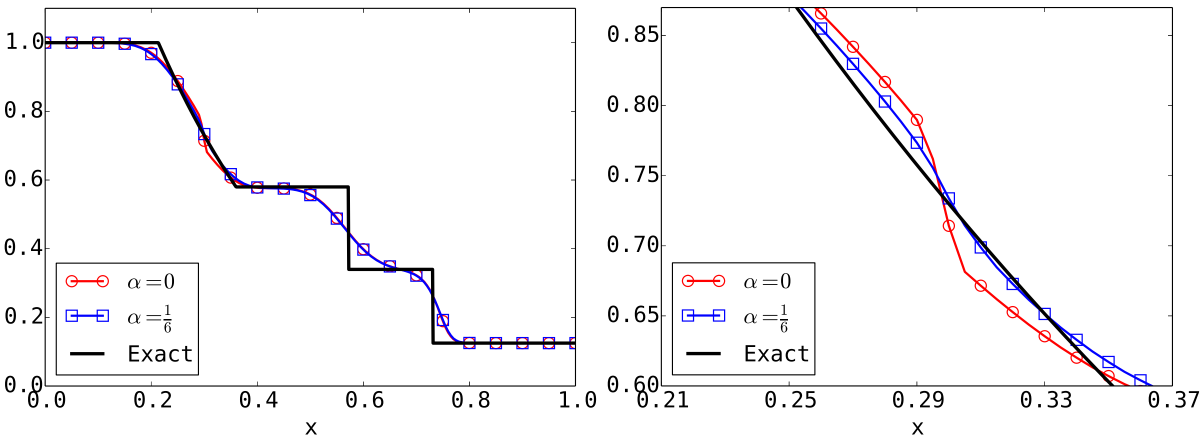


(a) $N = 100$

(b) $N = 200$

(c) $N = 400$

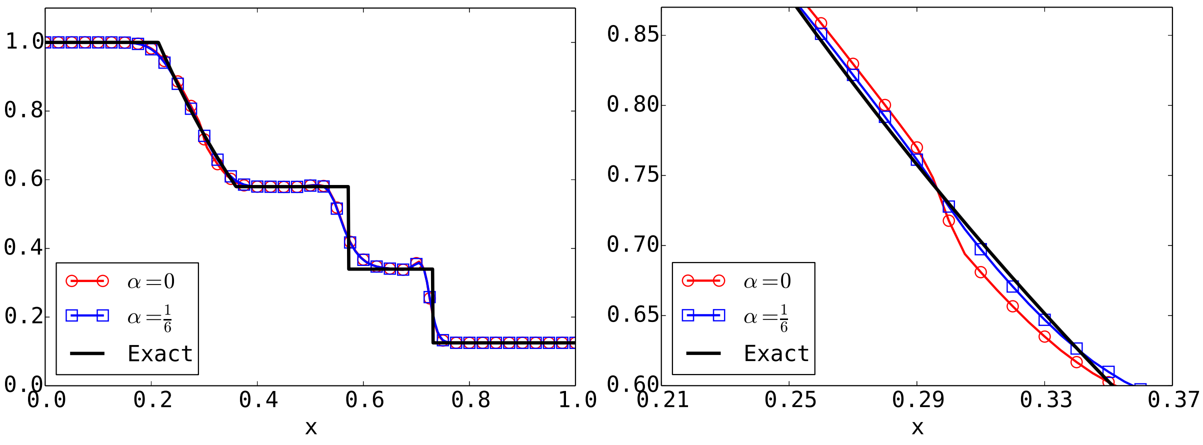
Figure 9.13: Density plot; grid refinement study with KEPES-TeCNO.



(a) Density

(b) Density zoomed

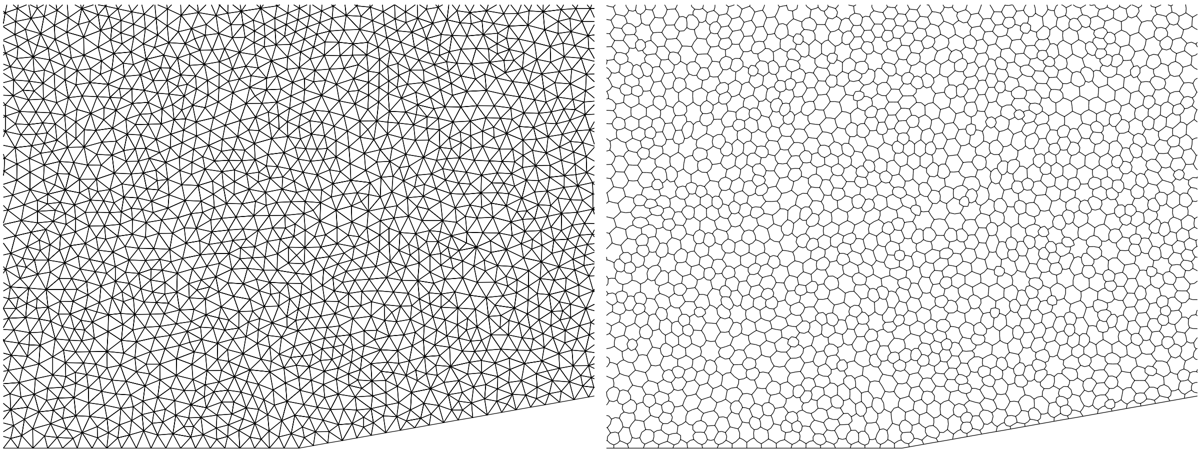
Figure 9.14: Density plot for N=100 and the KEPES scheme, with the entropy consistent modification.



(a) Density

(b) Density zoomed

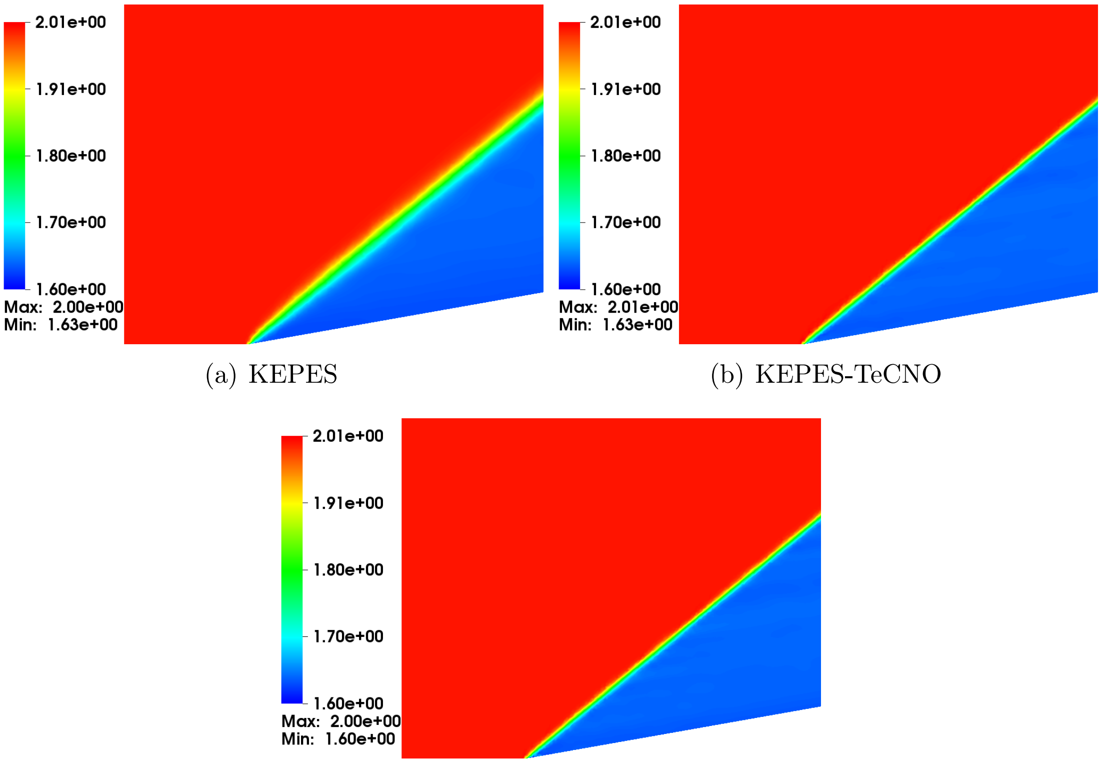
Figure 9.15: Density plot for N=100 and the KEPES-TeCNO scheme, with the entropy consistent modification.



(a) Primary

(b) Median dual

Figure 9.16: Mesh for flow over wedge.

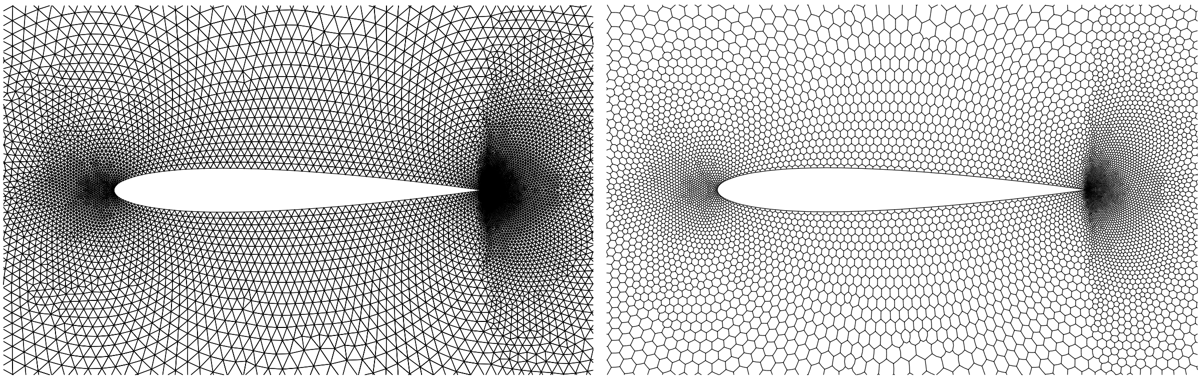


(a) KEPES

(b) KEPES-TeCNO

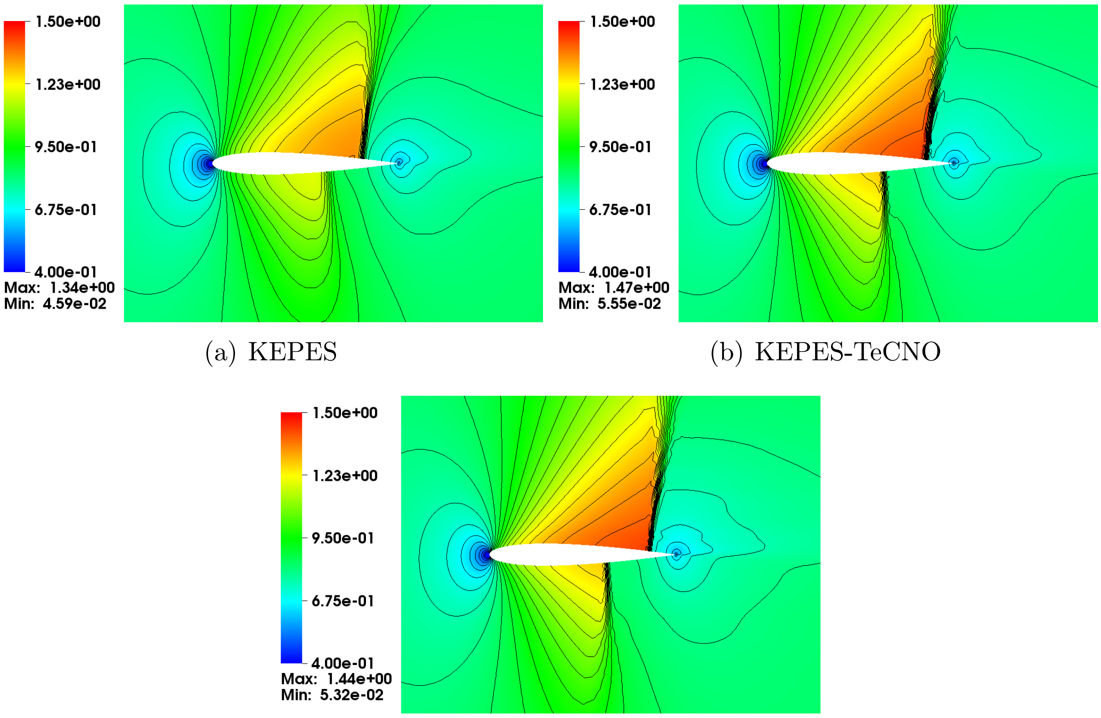
(c) Roe (MUSCL)

Figure 9.17: Mach number plots for a supersonic flow past a wedge.



(a) Primary (b) Median dual

Figure 9.18: Mesh for flow past NACA-0012 airfoil.

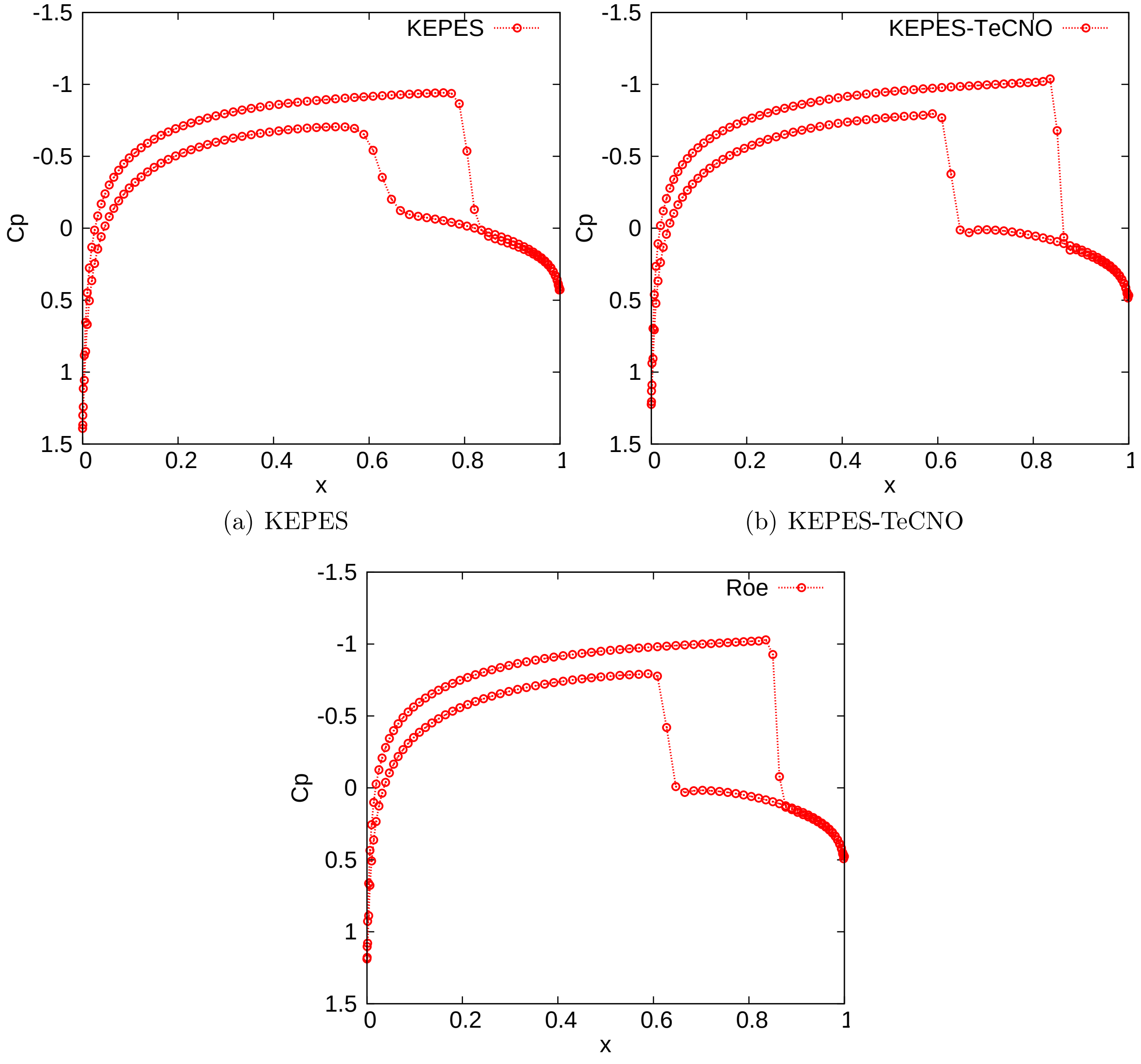


(a) KEPES (b) KEPES-TeCNO

(c) Roe (MUSCL)

Figure 9.19: Mach number, 30 equally spaced contours between 0.04 and 1.5.

the y-axis represents the inverted pressure coefficient. Thus, the upper surface of the wing, which has a much lower pressure distribution as compared to the lower surface, appears at the top of the plot. There is a sudden change in pressure across the shock that develops on both surfaces, and is clearly visible in the C_p plots. The area enclosed by the graph in the plots represents the lift experienced by the airfoil. Again, the high resolution KEPES-TeCNO was indistinguishable in accuracy compared to the standard high resolution Roe-MUSCL scheme.



(a) KEPES

(b) KEPES-TeCNO

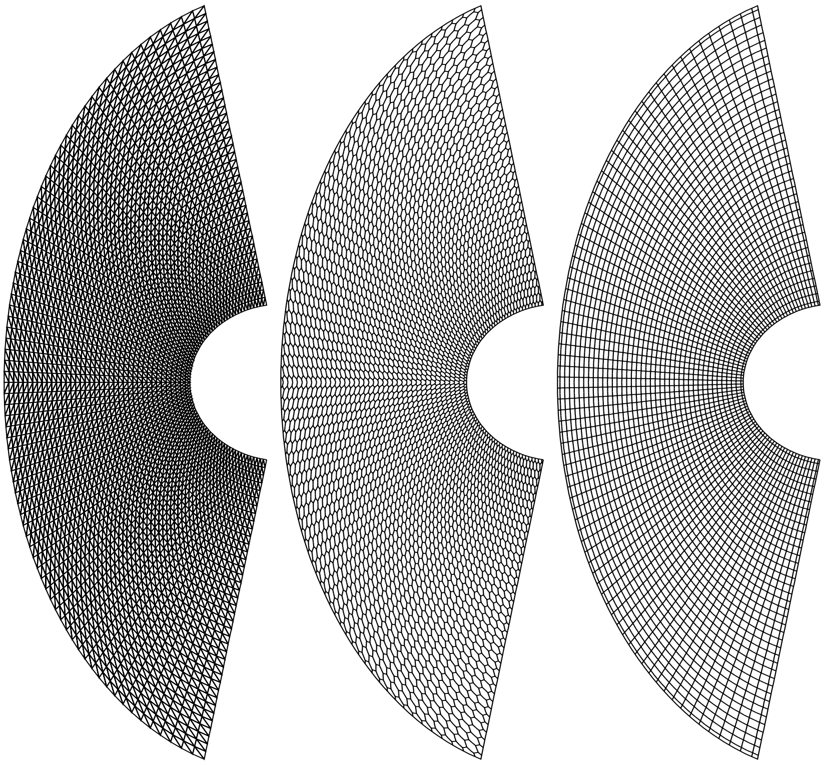
(c) Roe (MUSCL)

Figure 9.20: Pressure coefficient plots of the surface of the airfoil with $p_\infty = 0.9886$, $M_\infty = 0.85$.

9.4.5 Supersonic flow past a cylinder

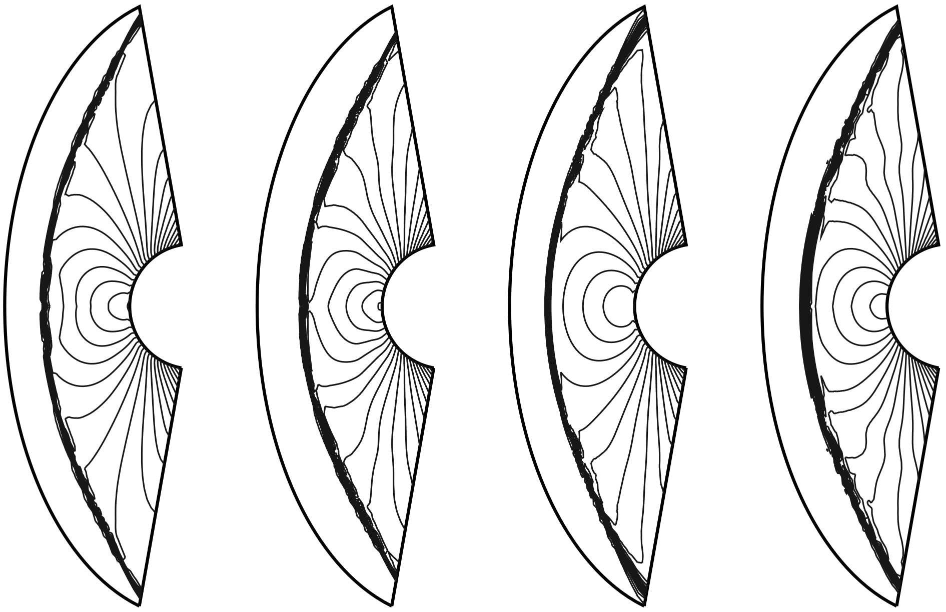
Most shock-capturing numerical schemes, except for a few highly dissipative schemes like the Rusanov scheme, can lead to numerical instabilities, particularly when approximating strong shocks. One of the most common anomalies is the *carbuncle phenomenon* [93, 94], which is produced when computing a supersonic flow past a blunt body such as a circular cylinder. Instead of having a smooth bow shock profile upstream of the cylinder, a protuberance appears ahead of the bow shock along the stagnation line. This effect seems to be more pronounced the more closely the grid is aligned to the bow shock.

Simulations were performed for the inviscid supersonic flow over a semi-cylinder. The primary triangular grid and the corresponding median and Voronoi dual meshes are shown in Figure 9.21. The Voronoi cells lead to nearly structured type grids, and can thus give rise to the carbuncle problem, since the shock will be aligned with the cell faces to a greater extent compared to the median dual cells. At free-stream Mach number $M_\infty = 2$, KEPES and KEPES-TeCNO give carbuncle free solutions on both median dual and Voronoi dual meshes, as can be seen in Figure 9.22. The bow shock is well resolved in each case. Similar results were observed when the schemes were used to simulate a hypersonic flow with $M_\infty = 20$, as shown in Figure 9.23.



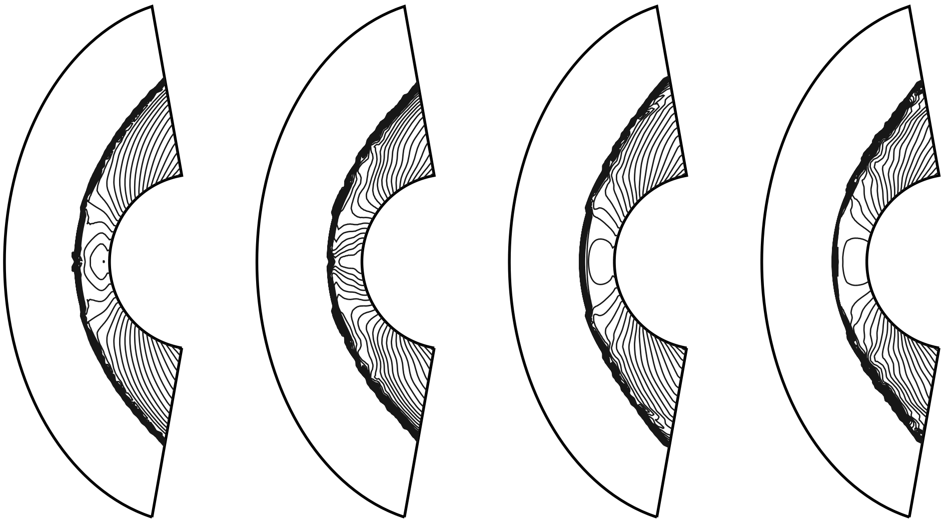
(a) Primal grid (b) Median dual grid (c) Voronoi dual grid

Figure 9.21: Grid used for supersonic cylinder problem.



(a) KEPES (b) KEPES-TeCNO (c) KEPES (d) KEPES-TeCNO

Figure 9.22: Density contours for supersonic cylinder, $M_\infty = 2$. (a)-(b) median dual grid; (c)-(d) Voronoi dual grid.

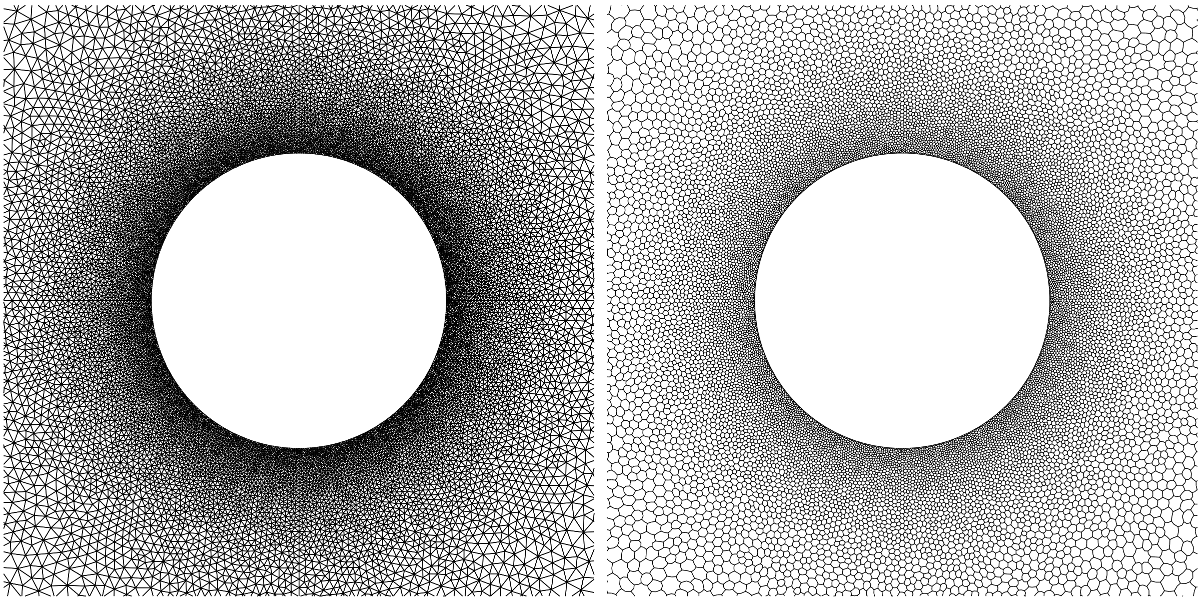


(a) KEPES (b) KEPES-TeCNO (c) KEPES (d) KEPES-TeCNO

Figure 9.23: Density contours for supersonic cylinder, $M_\infty = 20$. (a)-(b) median dual grid; (c)-(d) Voronoi dual grid.

9.4.6 Subsonic flow past a cylinder

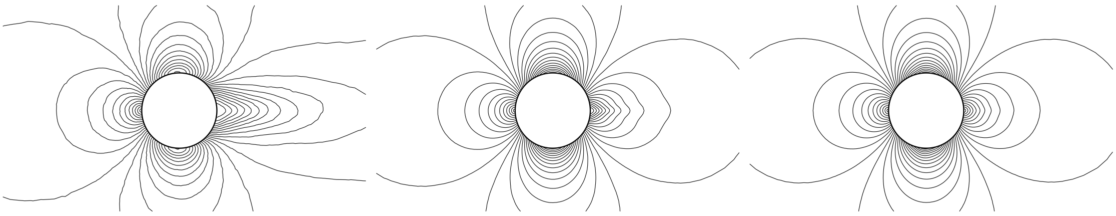
We consider an inviscid flow past a full cylinder at a low Mach number of 0.3. The mesh used for this problem is shown in Figure 9.24. The steady state solution has both top-bottom and left-right symmetry. The first-order KEPES solution loses its symmetry due to the excessive dissipation, as shown in Figure 9.25. The KEPES-TeCNO does a much better job at preserving the symmetry property, comparable to the approximate solution given by the KEPES2 scheme which uses an unlimited second-order reconstruction in the dissipation term.



(a) Primary

(b) Median dual

Figure 9.24: Mesh for a subsonic flow past a cylinder.



(a) KEPES

(b) KEPES-TeCNO

(c) KEPES2

Figure 9.25: Mach number, 30 equally spaced contours between 0.001 and 0.7.

The flow under consideration is nearly isentropic, i.e., the physical entropy of the flow around the cylinder should be nearly constant. Note that the physical entropy is given by $\widetilde{s} = \widetilde{s_0} + s$ (see Section 3.1.2). Since $\widetilde{s_0}$ is an arbitrary constant, the quantity s is constant for an isentropic flow. To demonstrate the ability of the schemes to preserve this constancy, the bounds in s obtained with each scheme, and their percentage deviation from the free-stream value s_∞, are mentioned in Table 9.3. We notice that KEPES gives the largest positive deviation, KEPES2 gives almost negligible positive deviation, while the limited KEPES-TeCNO scheme lies somewhere in between. Both the entropy stable schemes show no negative deviations, while the KEPES2 scheme gives almost negligible negative deviation. Although the KEPES2 performs the best in this scenario, we cannot theoretically prove any stability estimates with it. Moreover, the unlimited KEPES2 would perform rather poorly in the presence of shocks.

Scheme	Minimum	Maximum	Percent deviation from s_∞	
KEPES	2.07147	2.08695	+0.747 %	-0.000 %
KEPES-TeCNO	2.07147	2.07208	+0.029 %	-0.000 %
KEPES2	2.07139	2.07153	+0.003 %	-0.004 %

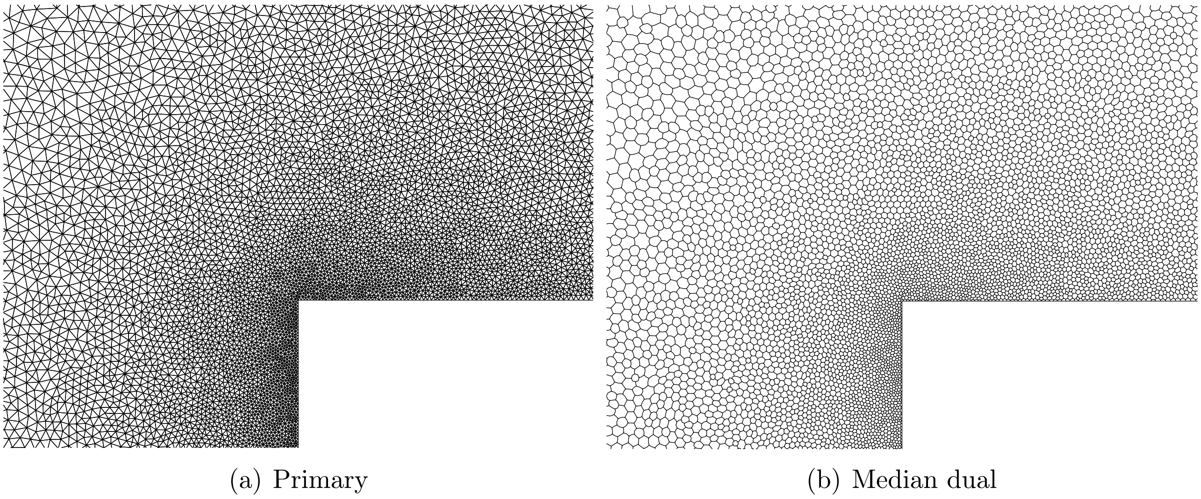
Table 9.3: Physical entropy bounds, with freestream $s_\infty = 2.07147$.

9.4.7 Step in wind tunnel

This test case is described in [135] involves an inviscid supersonic flow past a step in a wind tunnel, which is impulsively started with an initial Mach number of $M = 3$. The wind tunnel is one unit length wide and three unit lengths long. The step is 0.2 unit length high and is located 0.6 unit length from the left-hand end of the tunnel. At the left boundary, one imposes an inflow boundary condition. The exit boundary condition on the right has no effect on the flow, because the exit velocity is always supersonic. Slip boundary conditions are applied along the top and bottom walls of the tunnel . The simulation is run till $t_f = 4$ with a CFL=0.6.

The flow develops several shocks which undergo further reflections. A shock triple point intersection leads to the formation of a slip line. The corner of the step is the center of a rarefaction fan and hence is a singular point of the flow. The grid is adapted to be finer near the corner, where the spacing is of size ≈ 0.002 while the maximum spacing is of size ≈ 0.01. The total number of grid points is 70970. A close-up of the mesh close to the corner is shown in Figure 9.26. The density contours at time $t = 4$ are shown in Figure 9.27 using the KEPES-TeCNO scheme, which is able to resolve the main features of the flow very accurately.

Remark 9.4.1. *The numerical tests show that the proposed entropy stable schemes are able to preserve positivity of density and pressure, without any additional treatment on unstructured grids.*


(a) Primary

(b) Median dual

Figure 9.26: Mesh near the corner of of the forward step.

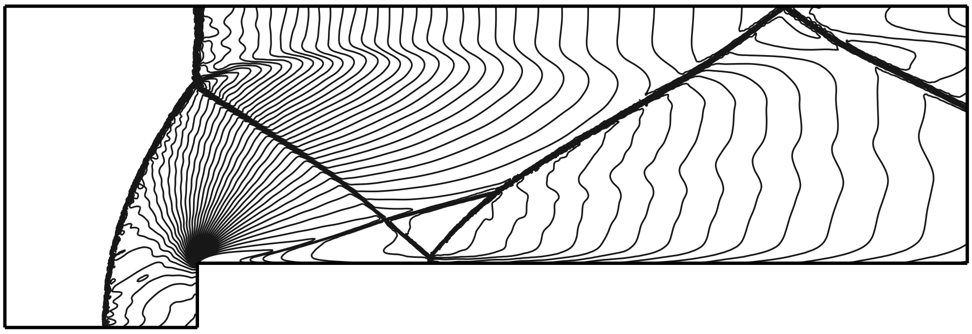


Figure 9.27: Density, 50 contour lines between 0.5 and 7.1 using KEPES-TeCNO at $t = 4$.

10. Finite volume scheme for Navier-Stokes equations

In Chapter 9 we introduced a vertex-centered finite volume scheme for the Euler equations, which is provably entropy stable. In this chapter, we extend the finite volume scheme to incorporate the viscous terms of Navier-Stokes equations. Based on the specific choice for the entropy-entropy flux pair (3.8), the viscous terms are approximated in terms of the entropy variables to exploit the symmetric formulation of the viscous fluxes. The final scheme is shown to be entropy stable.

10.1 Mesh notations and discretization

Most of the mesh notations have already been introduced in Chapter 9. We adopt the vertex-centered approach for evaluating the inviscid fluxes. The viscous fluxes, on the other hand, are approximated by evaluating the contributions on each primary triangle neighbouring a given vertex. We additionally define the following notations needed to describe the finite-volume scheme

$$\begin{aligned}
i \in T &= \{ \text{ all vertices } i \text{ belonging to triangle } T \ \} \ , \\
T \in i &= \{ \text{ all triangles } T \text{ having vertex } i \ \} \ , \\
\Gamma &= \{ \text{ all boundary edges of the primary mesh } \} \ , \\
\Gamma_i &= \{ \text{ all boundary edges of the primary mesh having vertex } i \ \} \ , \\
\partial\Omega_i^T &= \partial\Omega_i \cap \operatorname{int}(T).
\end{aligned}$$

If we consider the intersection of a dual cell Ω_i with a triangle T (see Figure 10.1), we have

$$\mathbf{n}_i^T = 2(\mathbf{n}_{ij}^{(1)} + \mathbf{n}_{ik}^{(2)}).$$

10.1.1 Approximation of gradient and divergence operators

In Section 9.3.5, we approximated the gradient of scalar and vector valued functions on triangles by (9.35) and (9.36) respectively. These approximations are used for an *interior triangle* T_I, i.e., a triangle with none of its faces in Γ. A triangle T_e adjoining a boundary face $e \in \Gamma$, with e joining vertices i, j as shown in Figure 9.1(b), will be termed as a

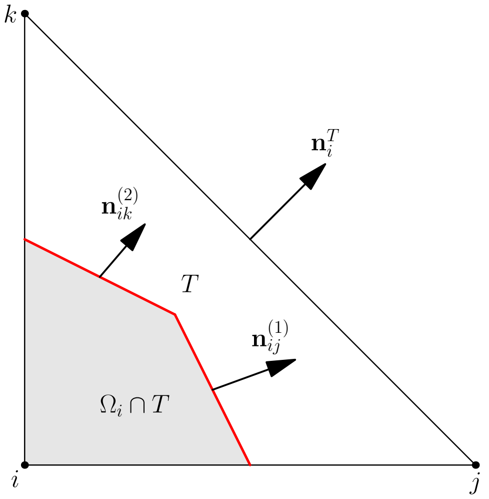


Figure 10.1: Relation between outward normals of Ω_i and T.

boundary triangle. The gradient approximation of a scalar function ϕ on T_e is chosen as

$$\nabla^h \phi^{T_e} = \frac{1}{|T_e|}\left[\frac{(\phi_i^b + \phi_j^b)}{2}\mathbf{n}_e + \frac{(\phi_j + \phi_k)}{2}\mathbf{n}_i^{T_e} + \frac{(\phi_k + \phi_i)}{2}\mathbf{n}_j^{T_e}\right], \tag{10.1}$$

where ϕ_i^b, ϕ_j^b are obtained from prescribed boundary conditions. Similarly, the gradient of a vector valued function $\boldsymbol{\phi} \in \mathbb{R}^m$ on T_e is approximated by

$$\nabla^h \boldsymbol{\phi}^{T_e} = \frac{1}{|T_e|}\left[\frac{(\boldsymbol{\phi}_i^b + \boldsymbol{\phi}_j^b)}{2} \otimes \mathbf{n}_e + \frac{(\boldsymbol{\phi}_j + \boldsymbol{\phi}_k)}{2} \otimes \mathbf{n}_i^{T_e} + \frac{(\boldsymbol{\phi}_k + \boldsymbol{\phi}_i)}{2} \otimes \mathbf{n}_j^{T_e}\right]. \tag{10.2}$$

The gradient at the vertex i is approximated using (9.37). Note that the evaluation of (9.36), (10.2) and (9.37) require the values of $\boldsymbol{\phi}$ to be defined at the vertices.

We next approximate the first-order spatial partial derivatives of a vector $\boldsymbol{\vartheta} \in \mathbb{R}^m$, at the vertices. Integrating $\partial_x \boldsymbol{\vartheta}$ over the dual cell Ω_i gives us

$$\int\limits_{\Omega_i} \partial_x \boldsymbol{\vartheta}\ \mathrm{d}\mathbf{x} = \int\limits_{\partial\Omega_i \setminus \Gamma} \boldsymbol{\vartheta} n_1 \mathrm{d}S + \int\limits_{\partial\Omega_i \cap \Gamma} \boldsymbol{\vartheta} n_1 \mathrm{d}S = \sum_{T \in i} \int\limits_{\partial\Omega_i^T} \boldsymbol{\vartheta} n_1 \mathrm{d}S + \sum_{e \in \Gamma_i} \int\limits_{\partial\Omega_i \cap e} \boldsymbol{\vartheta} n_1 \mathrm{d}S,$$

where $\mathbf{n} = (n_1, n_2)$ is the unit normal vector to the faces of dual cell Ω_i. We assume that $\boldsymbol{\vartheta}$ is defined on each triangle T by the constant vector $\boldsymbol{\vartheta}(T)$, with the same value used for $\boldsymbol{\vartheta}$ on $\partial\Omega_i^T$. Furthermore, we assume that the value of $\boldsymbol{\vartheta}$ on the boundary edge e is given by the constant vector value in the adjoining boundary triangle T_e. Thus, we get the following approximation for $\partial_x \boldsymbol{\vartheta}$ at the vertex i

$$\begin{aligned}\partial_x^h \boldsymbol{\vartheta}_i &= \frac{1}{|\Omega_i|}\left[\sum_{T \in i} \boldsymbol{\vartheta}(T) \int\limits_{\partial\Omega_i^T} n_1 \mathrm{d}S + \sum_{e \in \Gamma_i} \boldsymbol{\vartheta}(T_e) \int\limits_{\partial\Omega_i \cap e} n_1 \mathrm{d}S\right] \\ &= \frac{1}{|\Omega_i|}\left[\frac{1}{2}\sum_{T \in i} \boldsymbol{\vartheta}(T) n_{i,1}^T + \frac{1}{2}\sum_{e \in \Gamma_i} \boldsymbol{\vartheta}(T_e) n_{e,1}\right],\end{aligned} \tag{10.3}$$

where $\mathbf{n}_i^T = (n_{i,1}^T, n_{i,2}^T)$ and $\mathbf{n}_e = (n_{e,1}, n_{e,2})$. Similarly, we can get the following expression for $\partial_y^h \boldsymbol{\vartheta}_i$

$$\partial_y^h \boldsymbol{\vartheta}_i = \frac{1}{|\Omega_i|}\left[\frac{1}{2}\sum_{T\in i}\boldsymbol{\vartheta}(T)n_{i,2}^T + \frac{1}{2}\sum_{e\in\Gamma_i}\boldsymbol{\vartheta}(T_e)n_{e,2}\right]. \tag{10.4}$$

Note that in the approximation (9.37), the function values are known at the vertices, which is first used to find the gradient on triangles using (9.36), and then used to find the gradient at the vertex. On the other hand, (10.3) and (10.4) assume that the function values on triangles are directly available. With the approximations (10.3), we can prove the following summation-by-parts property.

Theorem 10.1.1. *Let the vectors $\boldsymbol{\phi}, \boldsymbol{\vartheta} \in \mathbb{R}^m$ be defined at the mesh vertices and on triangles respectively. Then the following SBP property holds*

$$\sum_i \left\langle \boldsymbol{\phi}_i, \partial_x^h \boldsymbol{\vartheta}_i\right\rangle |\Omega_i| = -\sum_T \left\langle \partial_x^h \boldsymbol{\phi}^T, \boldsymbol{\vartheta}(T)\right\rangle |T| + \sum_{e\in\Gamma}\left\langle \frac{\boldsymbol{\phi}_i^b + \boldsymbol{\phi}_j^b}{2}, \boldsymbol{\vartheta}(T_e)\right\rangle n_{e,1}, \tag{10.5}$$

where $\partial_x^h \boldsymbol{\vartheta}_i$ is evaluated at the vertices using (10.3)*, while $\partial_x^h \boldsymbol{\phi}^T$ is evaluated on triangles using* (9.36) *and* (10.2)*. Note that Ω_i is the dual cell centered at the vertex i.*

Proof. Using the expression (10.3), we have

$$\sum_i \left\langle \boldsymbol{\phi}_i, \partial_x^h \boldsymbol{\vartheta}_i\right\rangle |\Omega_i| = \frac{1}{2}\sum_i\sum_{T\in i}\left\langle \boldsymbol{\phi}_i, \boldsymbol{\vartheta}(T)\right\rangle n_{i,1}^T + \frac{1}{2}\sum_i\sum_{e\in\Gamma_i}\left\langle \boldsymbol{\phi}_i, \boldsymbol{\vartheta}(T_e)\right\rangle n_{e,1}.$$

Interchanging the order of summation and considering the summations over interior and boundary triangles separately, we get

$$\begin{aligned}\sum_i \left\langle \boldsymbol{\phi}_i, \partial_x^h \boldsymbol{\vartheta}_i\right\rangle |\Omega_i| = & \underbrace{\frac{1}{2}\sum_{T_I}\sum_{i\in T_I}\left\langle \boldsymbol{\phi}_i, \boldsymbol{\vartheta}(T_I)\right\rangle n_{i,1}^{T_I}}_{I} + \underbrace{\frac{1}{2}\sum_{T_e}\sum_{i\in T_e}\left\langle \boldsymbol{\phi}_i, \boldsymbol{\vartheta}(T_e)\right\rangle n_{i,1}^{T_e}}_{II} \\ & \underbrace{+\sum_{e\in\Gamma}\left\langle \frac{\boldsymbol{\phi}_i + \boldsymbol{\phi}_j}{2}, \boldsymbol{\vartheta}(T_e)\right\rangle n_{e,1}}_{III}.\end{aligned}$$

Using (9.36), we get

$$I = \sum_{T_I}\left\langle \frac{1}{2}\sum_{i\in T_I}\boldsymbol{\phi}_i n_{i,1}^{T_I}, \boldsymbol{\vartheta}(T_I)\right\rangle = -\sum_{T_I}\left\langle \partial_x^h \boldsymbol{\phi}^{T_I}, \boldsymbol{\vartheta}(T_I)\right\rangle |T_I|. \tag{10.6}$$

Consider the contribution of a fixed boundary triangle T_e (refer to Figure 9.1(b)) to the summation II. Using (10.2) and the fact that $\mathbf{n}_i^{T_e} + \mathbf{n}_j^{T_e} + \mathbf{n}_e = 0$, we have

$$\begin{aligned} II\Big|_{T_e} &= \frac{1}{2}\left\langle \left(\boldsymbol{\phi}_i n_{i,1}^{T_e} + \boldsymbol{\phi}_j n_{j,1}^{T_e} + \boldsymbol{\phi}_k n_{e,1}\right), \boldsymbol{\vartheta}(T_e)\right\rangle \\ &= -\left\langle \left(\frac{\boldsymbol{\phi}_i + \boldsymbol{\phi}_j}{2}n_{e,1} + \frac{\boldsymbol{\phi}_j + \boldsymbol{\phi}_k}{2}n_{i,1}^{T_e} + \frac{\boldsymbol{\phi}_k + \boldsymbol{\phi}_i}{2}n_{j,1}^{T_e}\right), \boldsymbol{\vartheta}(T_e)\right\rangle \\ &= -\left\langle \partial_x^h \boldsymbol{\phi}^{T_e}, \boldsymbol{\vartheta}(T_e)\right\rangle |T_e| + \left\langle \left(\frac{\boldsymbol{\phi}_i^b + \boldsymbol{\phi}_j^b}{2} - \frac{\boldsymbol{\phi}_i + \boldsymbol{\phi}_j}{2}\right), \boldsymbol{\vartheta}(T_e)\right\rangle n_{e,1}. \end{aligned}$$

Similarly, the contribution of T_e to the summation III is given by

$$III\Big|_{T_e} = \left\langle \frac{\boldsymbol{\phi}_i^b + \boldsymbol{\phi}_j^b}{2}, \boldsymbol{\vartheta}(T_e) \right\rangle n_{e,1}.$$

Thus, we get

$$II + III = -\sum_{T_e} \left\langle \partial_x^h \boldsymbol{\phi}^{T_e}, \boldsymbol{\vartheta}(T_e) \right\rangle |T_e| + \sum_{e\in\Gamma} \left\langle \frac{\boldsymbol{\phi}_i^b + \boldsymbol{\phi}_j^b}{2}, \boldsymbol{\vartheta}(T_e) \right\rangle n_{e,1}. \tag{10.7}$$

Adding (10.6) and (10.7) gives the result (10.5). □

A formula similar (10.5) can be shown to hold for the approximation (10.4). This leads to the following corollary.

Corollary 10.1.1. *Let the vector $\boldsymbol{\phi} \in \mathbb{R}^m$ be defined at mesh vertices. Consider the function $\boldsymbol{\vartheta} = (\boldsymbol{\vartheta}^x, \boldsymbol{\vartheta}^y)$, where the component vectors $\boldsymbol{\vartheta}^x, \boldsymbol{\vartheta}^y \in \mathbb{R}^m$ are defined on triangles. Then the following SBP property holds*

$$\sum_i \left\langle \boldsymbol{\phi}_i, \left(\partial_x^h \boldsymbol{\vartheta}_i^x + \partial_y^h \boldsymbol{\vartheta}_i^y\right) \right\rangle |\Omega_i| = -\sum_T \left\langle \widetilde{\nabla}^h \boldsymbol{\phi}^T, \widetilde{\boldsymbol{\vartheta}}(T) \right\rangle |T| + \sum_{e\in\Gamma} \left\langle \frac{\boldsymbol{\phi}_i^b + \boldsymbol{\phi}_j^b}{2}, \boldsymbol{\vartheta}(T_e)\cdot \mathbf{n}_e \right\rangle, \tag{10.8}$$

where $\partial_x^h \boldsymbol{\vartheta}_i, \partial_y^h \boldsymbol{\vartheta}_i$ are evaluated at the vertices using (10.3) *and* (10.4) *respectively, while*

$$\widetilde{\boldsymbol{\vartheta}}(T) = \begin{pmatrix} \boldsymbol{\vartheta}^x(T) \\ \boldsymbol{\vartheta}^y(T) \end{pmatrix} \in \mathbb{R}^{2m}, \quad \widetilde{\nabla}^h \boldsymbol{\phi}^T = \begin{pmatrix} \partial_x^h \boldsymbol{\phi}^T \\ \partial_y^h \boldsymbol{\phi}^T \end{pmatrix} \in \mathbb{R}^{2m},$$

with the components of $\widetilde{\nabla}^h \boldsymbol{\phi}^T$ evaluated on triangles using (9.36) *and* (10.2) *(compare with the notation $\widetilde{\nabla}$ defined in Section 3.4). Note that Ω_i is the dual cell centered at the vertex i.*

10.1.2 Finite volume scheme

The finite volume scheme is obtained by integrating the Navier-Stokes equation (3.20) over the control volume Ω_i, and approximating the flux integrals over the control volume boundary.

$$\begin{aligned}
|\Omega_i|\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} &= -\sum_{j\in i} \int\limits_{\partial\Omega_i\cap\partial\Omega_j} \mathbf{F}(\mathbf{U},\mathbf{n})\mathrm{d}S + \sum_{T\in i} \int\limits_{\partial\Omega_i^T} \mathbf{G}(\mathbf{V},\nabla\mathbf{V},\mathbf{n})\mathrm{d}S \\
&\quad - \int\limits_{\partial\Omega_i\cap\Gamma} \mathbf{F}(\mathbf{U},\mathbf{n})\mathrm{d}S + \int\limits_{\partial\Omega_i\cap\Gamma} \mathbf{G}(\mathbf{V},\nabla\mathbf{V},\mathbf{n})\mathrm{d}S \\
&\approx -\sum_{j\in i} \mathbf{F}_{ij} + \sum_{T\in i} \left(\mathbf{G}^T \cdot \int\limits_{\partial\Omega_i^T} \mathbf{n}\mathrm{d}S \right) - \sum_{e\in\Gamma_i} \mathbf{F}_{ie} + \sum_{e\in\Gamma_i} \left(\mathbf{G}^e \cdot \int\limits_{\partial\Omega_i\cap e} \mathbf{n}\mathrm{d}S \right) \\
&= -\sum_{j\in i} \mathbf{F}_{ij} + \sum_{T\in i} \mathbf{G}^T \cdot \frac{\mathbf{n}_i^T}{2} - \sum_{e\in\Gamma_i} \mathbf{F}_{ie} + \sum_{e\in\Gamma_i} \mathbf{G}^e \cdot \frac{\mathbf{n}_e}{2}.
\end{aligned}$$

The semi-discrete finite volume scheme is taken to be of the form

$$|\Omega_i|\frac{\mathrm{d}\mathbf{U}_i}{\mathrm{d}t} = -\sum_{j\in i}\mathbf{F}_{ij} + \sum_{T\in i}\mathbf{G}^T\cdot\frac{\mathbf{n}_i^T}{2} - \sum_{e\in\Gamma_i}\mathbf{F}_{ie} + \sum_{e\in\Gamma_i}\mathbf{G}^e\cdot\frac{\mathbf{n}_e}{2}, \tag{10.9}$$

where $\mathbf{U}_i$ is the cell averaged value in cell Ω_i, $\mathbf{F}_{ij}$ is a consistent and conservative inviscid numerical flux across the interior faces of the dual volumes, and $\mathbf{F}_{ie}$ is a consistent inviscid flux across the boundary face $\partial\Omega_i \cap e$. We take the following consistent approximation $\mathbf{G}^T = (\mathbf{G}_1^T, \mathbf{G}_2^T)$ for the viscous fluxes in term of the entropy variables in each triangle T

$$\mathbf{G}_\alpha^T = \mathbf{K}_{\alpha 1}^T\partial_x^h\mathbf{V}^T + \mathbf{K}_{\alpha 2}^T\partial_y^h\mathbf{V}^T, \qquad \alpha = 1, 2, \tag{10.10}$$

where matrices of $\mathbf{K}_{1,1}, \mathbf{K}_{1,2}, \mathbf{K}_{2,1}, \mathbf{K}_{2,2}$ are precisely the matrices described in Section 3.2.1, evaluated at some appropriate averaged state in triangle T (discussed in theroem 10.2.1). Furthermore, $\nabla^h\mathbf{V}^T = (\partial_x^h\mathbf{V}^T, \partial_y^h\mathbf{V}^T)$ is the gradient approximation of the entropy variables on the triangle T evaluated using (9.36) and (10.2). We define the viscous flux across the boundary edge e as $\mathbf{G}^e = (\mathbf{G}_1^e, \mathbf{G}_2^e)$ such that

$$\mathbf{G}^e\cdot\mathbf{n}_e = \mathbf{G}^{T_e}\cdot\mathbf{n}_e + \mathbf{C}^e, \tag{10.11}$$

where $\mathbf{C}^e$ is a correction term to be defined later (see Theorem 10.2.2), needed to construct consistent boundary viscous numerical fluxes. Note that if we ignore the correction term, then the viscous flux approximation in (10.9) is essentially given by (10.3) and (10.4).

10.2 Discrete entropy estimates

We now present two results that describe the sufficient conditions on the scheme (10.9), needed to obtain discrete global entropy estimates analogous to those described in Section 3.4. In Theorem 10.2.1, we discuss how the various numerical flux terms in (10.9) need to be prescribed to obtain an entropy estimate consistent with (3.22). However, we do not make any assumptions on type of boundary conditions imposed. This leaves the correction term $\mathbf{C}^e$ introduced in (10.11) unspecified. In Theorem 10.2.2, we assume homogeneous boundary conditions and specify how $\mathbf{C}^e$ must be specified, to obtain consistent boundary fluxes and discrete entropy estimates consistent with (3.27) or (3.26).

Theorem 10.2.1. *Consider the numerical scheme* (10.9) *for the Navier-Stokes system* (3.10)*, with the various flux terms chosen as follows:*

1. *The numerical inviscid flux* $\mathbf{F}_{ij}$ *is chosen to satisfy the E-flux condition* (9.21)*.*

2. *The boundary inviscid flux is approximated by*

$$\begin{aligned}\mathbf{F}_{ie} &= \begin{pmatrix} F_{ie}^\rho \\ \frac{1}{2}(p_i^b\mathbf{n}_e) + \mathbf{u}_i F_{ie}^\rho \\ F_{ie}^E \end{pmatrix}, \quad F_{ie}^\rho = \left(\rho\frac{u_{\mathbf{n}_e}^b}{2}\right)_i, \\ F_{ie}^E &= \left[\left(\rho\frac{|\mathbf{u}|^2}{2} + \frac{\gamma p}{\gamma-1}\right)\frac{u_{\mathbf{n}_e}^b}{2}\right]_i + \left[(p^b - p)\frac{u_{\mathbf{n}_e}}{2}\right]_i,\end{aligned} \tag{10.12}$$

where $u_\mathbf{n} = \mathbf{u}\cdot\mathbf{n}$ *and the superscript 'b' implies that the quantity in question is evaluated from the prescribed boundary conditions (when available).*

3. The viscous flux on interior triangles is approximated by (10.10) *and* (9.36)*, with*

$$\mathbf{K}^T = \mathbf{K}\left(\mathbf{V}^T\right), \qquad \mathbf{V}^T = \frac{1}{3}\sum_{i\in T}\mathbf{V}_i. \tag{10.13}$$

4. The boundary viscous flux on the boundary edge e (refer to Figure 9.1) is approximated by (10.11) *and* (10.2)*, with*

$$\mathbf{K}^{T_e} = \mathbf{K}\left(\mathbf{V}^{T_e}\right), \qquad \mathbf{V}^{T_e} = \frac{(\mathbf{V}_i^b + \mathbf{V}_j^b)^\top}{2}. \tag{10.14}$$

Then the scheme satisfies the following discrete entropy estimate for the Navier-Stokes system

$$\begin{aligned}\frac{d}{dt}\sum_i \eta_i|\Omega_i| \leqslant & -\sum_{e\in\Gamma}\left[-\left(\frac{\rho s}{\gamma-1}\frac{u^b_{\mathbf{n}_e}}{2}\right)_i - \left(\frac{\rho s}{\gamma-1}\frac{u^b_{\mathbf{n}_e}}{2}\right)_j\right] \\ & \sum_{e\in\Gamma}\left[\left\langle\frac{(\mathbf{V}_i^b+\mathbf{V}_j^b)}{2}, \mathbf{G}^{T_e}\cdot\mathbf{n}_e\right\rangle + \left\langle\frac{(\mathbf{V}_i+\mathbf{V}_j)}{2}, \mathbf{C}^e\right\rangle\right].\end{aligned} \tag{10.15}$$

Note that (10.15) *is consistent with* (3.22) *if* $\mathbf{C}^e = 0$*. In fact, this will be true for all cases except wall boundary conditions with a non-zero heat flux (see Theorem 10.2.2).*

Proof. Taking the scalar product of (10.9) with $\mathbf{V}_i$ and summing over all vertices i results in

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t}\sum_i \eta_i|\Omega_i| =& -\sum_i\sum_{j\in i}\langle\mathbf{V}_i,\mathbf{F}_{ij}\rangle - \sum_i\sum_{e\in\Gamma_i}\langle\mathbf{V}_i,\mathbf{F}_{ie}\rangle + \frac{1}{2}\sum_i\sum_{T\in i}\left\langle\mathbf{V}_i,\mathbf{G}^T\cdot\mathbf{n}_i^T\right\rangle \\ &+ \frac{1}{2}\sum_i\sum_{e\in\Gamma_i}\langle\mathbf{V}_i,\mathbf{G}^e\cdot\mathbf{n}_e\rangle \\ =& \underbrace{-\sum_i\sum_{j\in i}\langle\mathbf{V}_i,\mathbf{F}_{ij}\rangle}_{I} \underbrace{-\sum_{e\in\Gamma}\left(\langle\mathbf{V}_i,\mathbf{F}_{ie}\rangle + \langle\mathbf{V}_j,\mathbf{F}_{je}\rangle\right)}_{II} + \underbrace{\frac{1}{2}\sum_T\sum_{i\in T}\left\langle\mathbf{V}_i,\mathbf{G}^T\cdot\mathbf{n}_i^T\right\rangle}_{III} \\ &+ \underbrace{\sum_{e\in\Gamma}\left\langle\frac{(\mathbf{V}_i+\mathbf{V}_j)}{2},\mathbf{G}^{T_e}\cdot\mathbf{n}_e\right\rangle}_{IV(a)} + \underbrace{\sum_{e\in\Gamma}\left\langle\frac{(\mathbf{V}_i+\mathbf{V}_j)}{2},\mathbf{C}^e\right\rangle}_{IV(b)}.\end{aligned}$$

Since $\mathbf{F}_{ij}$ satisfies (9.21), we obtain

$$\begin{aligned}I &= -\sum_i\sum_{j\in i}\left\langle\left(\overline{\mathbf{V}}_{ij} - \frac{1}{2}\Delta\mathbf{V}_{ij}\right),\mathbf{F}_{ij}\right\rangle \\ &\leqslant -\sum_i\sum_{j\in i}\left\langle\overline{\mathbf{V}}_{ij},\mathbf{F}_{ij}\right\rangle + \frac{1}{2}\sum_i\sum_{j\in i}\left(\Psi(\mathbf{U}_j,\mathbf{n}_{ij}) - \Psi(\mathbf{U}_i,\mathbf{n}_{ij})\right) \\ &= -\sum_i\sum_{j\in i}\left\langle\overline{\mathbf{V}}_{ij},\mathbf{F}_{ij}\right\rangle + \frac{1}{2}\sum_i\sum_{j\in i}\left(\Psi(\mathbf{U}_j,\mathbf{n}_{ij}) + \Psi(\mathbf{U}_i,\mathbf{n}_{ij})\right) - \sum_i\sum_{j\in i}\Psi(\mathbf{U}_i,\mathbf{n}_{ij}).\end{aligned} \tag{10.16}$$

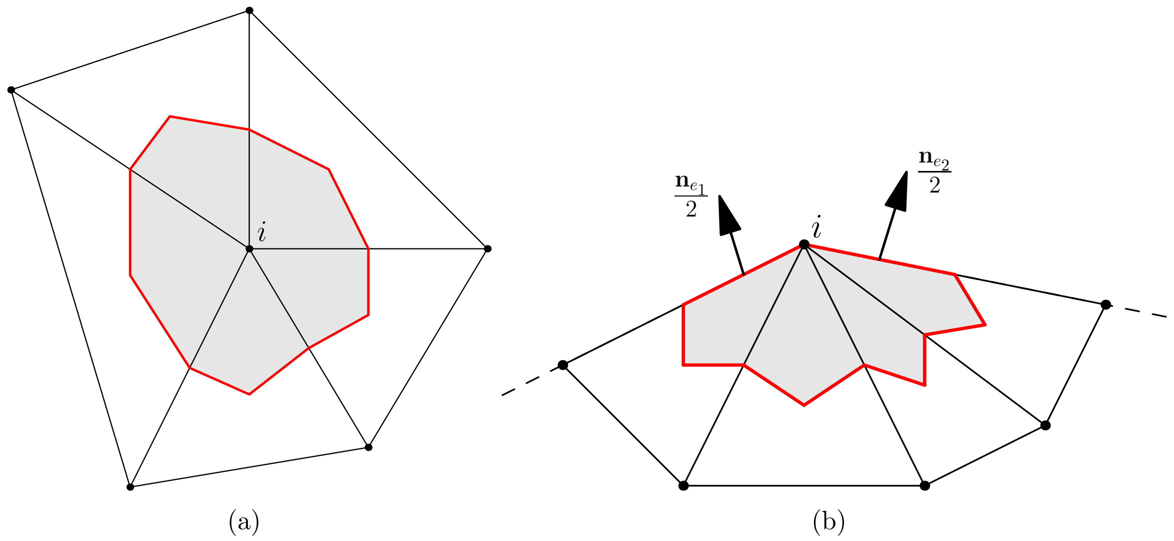


Figure 10.2: Dual cell for (a) an interior vertex; (b) boundary vertex with boundary edges e_1 and e_2.

Noting that $\overline{\mathbf{V}}_{ij} = \overline{\mathbf{V}}_{ji}$ and $\mathbf{F}_{ij} = -\mathbf{F}_{ji}$, the first term in (10.16) sums to zero. Similarly, the second term in (10.16) drops out since $\Psi(\mathbf{U}_i, \mathbf{n}_{ij}) = -\Psi(\mathbf{U}_i, \mathbf{n}_{ji})$. Thus, let us focus on the remaining third term in (10.16). If i is an interior vertex as shown in Figure 10.2(a), then $\sum\limits_{j\in i} \Psi(\mathbf{U}_i, \mathbf{n}_{ij}) = 0$ as $\sum\limits_{j\in i} \mathbf{n}_{ij} = 0$. However, if i is a boundary vertex as shown in Figure 10.2(b), then

$$\begin{aligned}
\sum_{j\in i} \Psi(\mathbf{U}_i, \mathbf{n}_{ij}) &= \sum_{\alpha=1,2} \left((\langle \mathbf{V}_i, \mathbf{f}_\alpha(\mathbf{U}_i)\rangle - q_\alpha(\mathbf{U}_i)) \sum_{j\in i} n_{ij,\alpha} \right) \\
&= - \sum_{\alpha=1,2} \left((\langle \mathbf{V}_i, \mathbf{f}_\alpha(\mathbf{U}_i)\rangle - q_\alpha(\mathbf{U}_i)) \sum_{e\in\Gamma_i} \frac{1}{2} n_{e,\alpha} \right) \\
&= - \sum_{e\in\Gamma_i} \Psi\left(\mathbf{U}_i, \frac{\mathbf{n}_e}{2}\right).
\end{aligned}$$

Thus,

$$I \leqslant \sum_{e\in\Gamma} \left(\Psi\left(\mathbf{U}_i, \frac{\mathbf{n}_e}{2}\right) + \Psi\left(\mathbf{U}_j, \frac{\mathbf{n}_e}{2}\right)\right),$$

which in turn implies

$$I + II \leqslant \sum_{e\in\Gamma} \left[\left(\Psi\left(\mathbf{U}_i, \frac{\mathbf{n}_e}{2}\right) - \langle \mathbf{V}_i, \mathbf{F}_{ie}\rangle \right) + \left(\Psi\left(\mathbf{U}_j, \frac{\mathbf{n}_e}{2}\right) - \langle \mathbf{V}_j, \mathbf{F}_{je}\rangle \right) \right]. \tag{10.17}$$

The Navier-Stokes equations with the entropy framework given by (3.8), has the entropy potential function $\Psi(\mathbf{U}, \mathbf{n}) = \rho u_{\mathbf{n}}$. Choosing $\mathbf{F}_{ie}$ as (10.12), we get

$$\langle \mathbf{V}_i, \mathbf{F}_{ie}\rangle - \Psi\left(\mathbf{U}_i, \frac{\mathbf{n}_e}{2}\right) = - \left(\frac{\rho s}{\gamma - 1} \frac{u^b_{\mathbf{n}_e}}{2} \right)_i. \tag{10.18}$$

For the viscous terms III and $IV(a)$, we use Corollary 10.1.1 with $\boldsymbol{\phi}_i \equiv \mathbf{V}_i$, $\boldsymbol{\vartheta}^x(T) \equiv \mathbf{G}_1^T$ and $\boldsymbol{\vartheta}^y(T) \equiv \mathbf{G}_2^T$ to get

$$III + IV(a) = -\sum_T |T| \left\langle \widetilde{\nabla}^h \mathbf{V}^{T_I}, \mathbf{K}^T \widetilde{\nabla}^h \mathbf{V}^T \right\rangle + \left\langle \frac{(\mathbf{V}_i^b + \mathbf{V}_j^b)}{2}, \mathbf{G}^{T_e} \cdot \mathbf{n}_e \right\rangle . \tag{10.19}$$

Finally, using (10.17), (10.18) and (10.19), we get

$$\begin{aligned}\frac{\mathrm{d}}{\mathrm{d}t} \sum_i \eta_i |\Omega_i| \ &\leqslant - \sum_{e \in \Gamma} \left[- \left(\frac{\rho s}{\gamma - 1} \frac{u^b_{\mathbf{n}_e}}{2} \right)_i - \left(\frac{\rho s}{\gamma - 1} \frac{u^b_{\mathbf{n}_e}}{2} \right)_j \right] - \sum_T |T| \left\langle \mathbf{K}^T \widetilde{\nabla}^h \mathbf{V}^T, \widetilde{\nabla}^h \mathbf{V}^T \right\rangle \\ &\sum_{e \in \Gamma} \left[\left\langle \frac{(\mathbf{V}_i^b + \mathbf{V}_j^b)}{2}, \mathbf{G}^{T_e} \!\cdot \mathbf{n}_e \right\rangle + \left\langle \frac{(\mathbf{V}_i + \mathbf{V}_j)}{2}, \mathbf{C}^e \right\rangle \right] .\end{aligned}$$

Since $\mathbf{K}^T \geqslant 0$, we get (10.15). □

Remark 10.2.1. *1. Ignoring the viscous fluxes and considering the scheme*

$$|\Omega_i| \frac{d\mathbf{U}_i}{dt} = - \sum_{j \in i} \mathbf{F}_{ij} + \sum_{e \in \Gamma_i} \mathbf{F}_{ie}, \tag{10.20}$$

with the inviscid fluxes approximated as above, we obtain the following entropy estimate for the Euler system

$$\frac{d}{dt} \sum_i \eta_i |\Omega_i| \ \leqslant - \sum_{e \in \Gamma} \left[- \left(\frac{\rho s}{\gamma - 1} \frac{u^b_{\mathbf{n}_e}}{2} \right)_i - \left(\frac{\rho s}{\gamma - 1} \frac{u^b_{\mathbf{n}_e}}{2} \right)_j \right], \tag{10.21}$$

which is consistent with its analytical counterpart

$$\frac{d}{dt} \int_{\Omega} \eta d\mathbf{x} \leqslant - \int_{\partial \Omega} q(\mathbf{U}, \mathbf{n}) dS. \tag{10.22}$$

2. Assuming positivity of pressure and density at the vertices, the pressure and density obtained on the triangles from the averaged entropy variables $\mathbf{V}^T$ and $\mathbf{V}^{T_e}$ are also positive. This is discussed in Appendix E.

3. Due to the presence of physical viscosity, the estimates (10.15) *would also hold true if $\mathbf{F}_{ij}$ is chosen to satisfy* (9.18) *instead of* (9.21).

Note that we have discretized the symmetric formulation of the viscous fluxes in terms of the entropy variables, using (9.36), (10.2), (10.3) and (10.4). This allowed us to obtain consistent entropy estimates for the Navier-Stokes equations. Alternately, we can write the viscous fluxes directly in terms of the stress tensor $\boldsymbol{\tau}$, and use the same discrete operators to approximate the derivatives of the velocity vector. This has approach has been used in [17] to construct a semi-discrete scheme for the Navier-Stokes equation, which is kinetic energy preserving.

Next, we consider homogeneous boundary conditions and specify how $\mathbf{C}^e$ must be chosen to obtain consistent discrete entropy stability estimates. Note that the correction term only appears in the boundary viscous terms.

Theorem 10.2.2. *Consider the numerical scheme* (10.9) *for Navier-Stokes equation, satisfying the conditions described in Theorem 10.2.1. Let the correction term be chosen as*

$$\mathbf{C}^e = \begin{cases} -\left(0 \quad 0 \quad 0 \quad \left(h^b + \frac{\kappa}{R(V^{T_e,(4)})^2}\left(\nabla^h \mathbf{V}^{T_e,(4)} \cdot \mathbf{n}_e\right)\right)\right)^\top, & \textit{if } \mathbf{Q}\cdot \mathbf{n}\Big|_{\partial\Omega} = h^b \textit{ given} \\ \mathbf{0}, & \textit{otherwise} \end{cases}, \tag{10.23}$$

for Navier-Stokes with no-slip boundary conditions. Then, for an adiabatic wall ($h^b = 0$), we obtain the consistent entropy stability estimate

$$\frac{d}{dt}\sum_i \eta_i|\Omega_i| \leqslant 0 \tag{10.24}$$

and boundary flux expressions $\mathbf{F}_{ie} = (0,\ p_i\mathbf{n}_e/2,\ 0)^\top$, $\mathbf{G}^e\cdot \mathbf{n}_e = \left(0,\ \boldsymbol{\tau}^h\cdot \mathbf{n}_e,\ 0\right)^\top$, *which are consistent with* (3.24)*. On the other hand, for an isothermal wall with* $\theta\Big|_\Omega = \theta^b$*, we obtain the estimate*

$$\frac{d}{dt}\sum_i \eta_i|\Omega_i| \leqslant \sum_{e\in\Gamma}\left[\frac{\kappa}{R\overline{V^{e,(4)}}}\left(\nabla^h \mathbf{V}^{T_e,(4)}\cdot \mathbf{n}_e\right)\right], \quad \overline{V^{e,(4)}} = -\frac{1}{R}\left(\frac{\theta_i^b + \theta_j^b}{\theta_i^b\theta_j^b}\right),$$

which is consistent with (3.26)*. Note that* $V^{(4)}$ *denotes the last component of the vector of entropy variables, since we are in two-dimensions.*

Proof. If we consider no-slip boundary conditions for the Navier-Stokes system, then for each boundary edge e we have $u^b = 0$ and according to (10.14)

$$\mathbf{V}^{T_e} = \begin{pmatrix} V^{T_e,(1)} \\ \mathbf{0} \\ V^{T_e,(4)} \end{pmatrix}, \qquad \mathbf{G}^{T_e}\cdot \mathbf{n}_e = \begin{pmatrix} 0 \\ \boldsymbol{\tau}^{\boldsymbol{h}}\cdot \mathbf{n}_e \\ \frac{\kappa}{R(V^{T_e,(4)})^2}\left(\nabla^h \mathbf{V}^{T_e,(4)}\cdot \mathbf{n}_e\right) \end{pmatrix},$$

$$\left\langle \frac{(\mathbf{V}_i^b + \mathbf{V}_j^b)}{2}, \mathbf{G}^{T_e}\cdot \mathbf{n}_e \right\rangle = \frac{\kappa}{RV^{T_e,(4)}}\left(\nabla^h \mathbf{V}^{T_e,(4)}\cdot \mathbf{n}_e\right),$$

$$\boldsymbol{\tau}^{\boldsymbol{h}} = \frac{1}{V^{T_e,(4)}}\begin{pmatrix} -\frac{4}{3}\mu\partial_x^h \mathbf{V}^{T_e,(2)} + \frac{2}{3}\mu\partial_y^h \mathbf{V}^{T_e,(3)} & -\mu\partial_x^h \mathbf{V}^{T_e,(3)} - \mu\partial_y^h \mathbf{V}^{T_e,(2)} \\ -\mu\partial_x^h \mathbf{V}^{T_e,(3)} - \mu\partial_y^h \mathbf{V}^{T_e,(2)} & \frac{2}{3}\mu\partial_x^h \mathbf{V}^{T_e,(2)} - \frac{4}{3}\mu\partial_y^h \mathbf{V}^{T_e,(3)} \end{pmatrix}.$$

Thus, (10.15) reduces to

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_i \eta_i|\Omega_i| \leqslant \sum_{e\in\Gamma}\left[\frac{\kappa}{RV^{T_e,(4)}}\left(\nabla^h \mathbf{V}^{T_e,(4)}\cdot \mathbf{n}_e\right) + \left\langle \frac{(\mathbf{V}_i + \mathbf{V}_j)}{2}, \mathbf{C}^e \right\rangle\right].$$

Additionally, assume the heat flux at the boundary is prescribed as h^b, and note that for this particular type of boundary condition

$$V^{T_e,(4)} = \frac{(V_i^{b,(4)} + V_j^{b,(4)})}{2} = \frac{(V_i^{(4)} + V_j^{(4)})}{2},$$

since the temperature, density and pressure are not prescribed at the boundary. Choosing $\mathbf{C}^e$ according to (10.23) leads to the viscous boundary flux $\mathbf{G}^e\cdot\mathbf{n}_e = \big(0,\ \boldsymbol{\tau}^h\cdot\mathbf{n}_e,\ -h^b\big)^\top$ and the discrete entropy estimate

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_i \eta_i|\Omega_i| \leqslant \sum_{e\in\Gamma}\left[\frac{V^{T_e,(4)}}{2R}h^b\right].$$

Setting $h^b = 0$ gives us the entropy stability estimate (10.24), and the corresponding boundary flux expressions for adiabatic no-slip wall conditions.

On the other hand, if we have isothermal no-slip wall conditions, $\mathbf{C}^e = \mathbf{0}$ according to (10.23). In this case, (10.15) reduces to

$$\frac{\mathrm{d}}{\mathrm{d}t}\sum_i \eta_i|\Omega_i| \leqslant \sum_{e\in\Gamma}\left[\frac{\kappa}{RV^{T_e,(4)}}\left(\nabla^h\mathbf{V}^{T_e,(4)}\cdot\mathbf{n}_e\right)\right], \quad V^{T_e,(4)} = \frac{(V_i^{b,(4)}+V_j^{b,(4)})}{2} = \overline{V^{e,(4)}}.$$

□

10.3 Code implementation

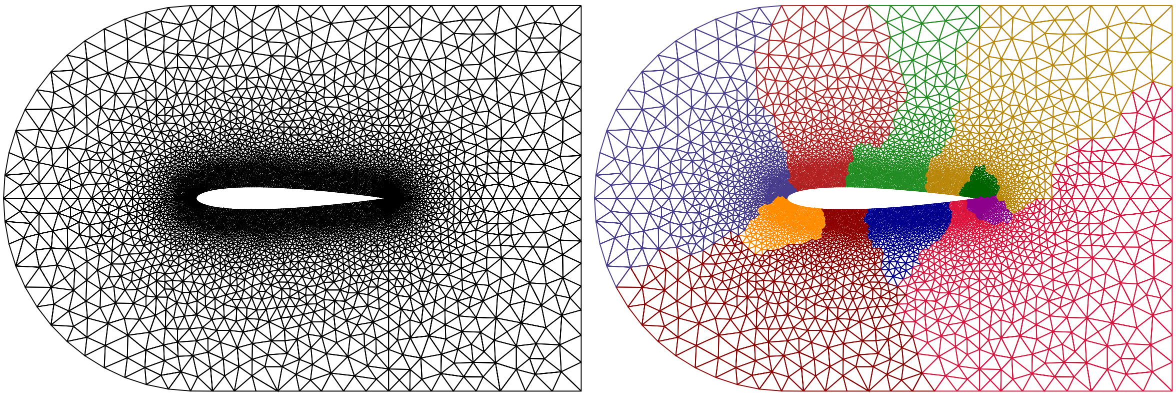
We briefly describe the important aspects of the implementation of the numerical solver. The code has been termed as *TEnSUM*, which stands for **T**wo-dimensional **En**tropy stable **S**olver on **U**nstructured **M**eshes. The code has been implemented in C++, and parallelized using MPI standards.

10.3.1 Mesh generation and partitioning

The unstructured meshes are generated using Gmsh [41], which is an open-source finite element grid generator with a built-in CAD engine, and equipped with several important post-processing tools. An important tool implemented inside Gmsh is METIS [63], which is a package used for graph partitioning. We use the information provided by METIS to partition the mesh based on the *cell-graph*. As an example, consider the un-partitioned mesh around a NACA-0012 airfoil, as shown in Figure 10.3(a). Using METIS, the mesh is partitioned into 10 sub-meshes, as shown in Figure 10.3(b), with each partition depicted by a different colour. The partitioning algorithm must ensure i) each sub-mesh has approximately the same number of cells, which is also termed as *load balancing*, and ii) the amount of communication between processors is minimum, which depends on the length of the boundary between sub-meshes and the number of neighbouring sub-meshes. Both these objectives are achieved by using a *multilevel k-way partitioning algorithm* [64].

10.3.2 Parallelization

Once the mesh is partitioned, say into N partitions, MPI is used to initialize N processors. Each processor solves the problem by running TEnSUM on one of the N sub-meshes. Each primary triangle of the original mesh is owned by only one processor. However, vertices on the inter-partition boundaries are shared by two or more processors. At each (sub) time-step of the Runge-Kutta algorithm, data at the shared vertices may need to be



(a) Un-partitioned mesh

(b) Partitioned mesh

Figure 10.3: Partitioning the mesh around a NACA-0012 airfoil into 10 sub-meshes using METIS.

communicated between neighbouring processors. As an example, consider the situation shown in Figure 10.4, where the triangles T_1, T_5 in light-grey belong to partition 1, while the triangles T_2, T_3, T_4 in dark-grey belong to partition 2. Furthermore, the vertex j_1 belongs to partition 1, the vertices j_3, j_4 belong to partition 2 and the vertices i, j_2, j_5 are shared by the two partitions. We need to evaluate the right hand side of the scheme (10.9) at the vertex i. Keeping with the terminology introduced is Chapter 8, we denote the right hand side as the residual function $\boldsymbol{\mathcal{R}}$. Since the inter-partition boundary never intersects with the domain boundary, there are no boundary terms to be evaluated in $\boldsymbol{\mathcal{R}}_i$.

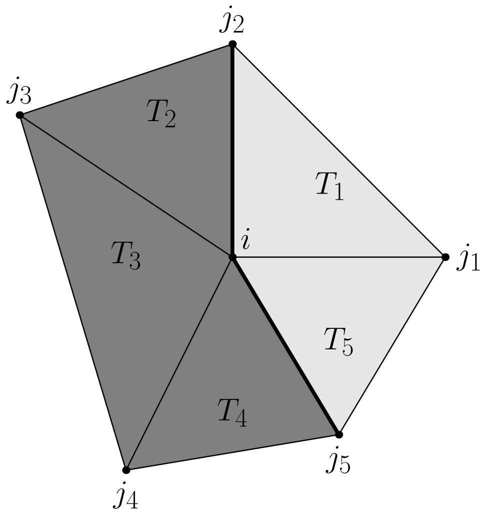


Figure 10.4: Partitioning about vertex i. The triangles in light-grey belong to partition 1 while the ones in dark-grey belong to partition 2.

In partition 1, the residual function is evaluated as

$$\boldsymbol{\mathcal{R}}_i = -w_{ij_1}\mathbf{F}_{ij_1} - w_{ij_2}\mathbf{F}_{ij_2} - w_{ij_5}\mathbf{F}_{ij_5} + \frac{1}{2}\left(\mathbf{G}_1^T \cdot \mathbf{n}_i^{T_1} + \mathbf{G}_5^T \cdot \mathbf{n}_i^{T_5}\right), \tag{10.25}$$

while in partition 2,

$$\boldsymbol{\mathcal{R}}_i = -w_{ij_2}\mathbf{F}_{ij_2} - w_{ij_3}\mathbf{F}_{ij_3} - w_{ij_4}\mathbf{F}_{ij_4} - w_{ij_5}\mathbf{F}_{ij_5} + \frac{1}{2}\left(\mathbf{G}_2^T \cdot \mathbf{n}_i^{T_2} + \mathbf{G}_3^T \cdot \mathbf{n}_i^{T_3} + \mathbf{G}_4^T \cdot \mathbf{n}_i^{T_4}\right). \tag{10.26}$$

We have multiplied the inviscid fluxes with weights w_{ij}, to accommodate for the fact that the inviscid fluxes on inter-partition edges are evaluated in both partitions. Thus, $w_{ij} = 0.5$ if both vertices i, j are shared between the two partitions, otherwise $w_{ij} = 1$. Once (10.25) and (10.26) have been evaluated in the respective partitions, the values are communicated between the processors handling partitions 1 and 2 using *asynchronous* MPI send and receive commands, following which they are added to the existing values on each partition. Note that vertex gradients of (entropy) variables are also required if a higher-order inviscid flux is used (see Chapter 9). In this case, the vertex gradients are evaluated locally on each partition using (9.37) and then the gradients at shared vertices are also communicated between processors.

10.4 Numerical results

We now present numerical results with the scheme discussed above, on several standard test cases. The inviscid flux is chosen to be one of the fluxes described in the beginning of Section 9.4. The semi-discrete scheme is integrated in time using the explicit SSP-RK3, and the local time step depends on the convective and viscous contributions [12]:

$$\Delta t_i = \frac{\mathrm{CFL}\cdot|\Omega_i|}{\lambda_i}, \qquad \lambda_i = \sum_{j\in i}\left[|\mathbf{u}_i\cdot \mathbf{n}_{ij}| + a_i|\mathbf{n}_{ij}| + \frac{\mu|\mathbf{n}_{ij}|}{|\mathbf{x}_j - \mathbf{x}_i|\rho_i}\right].$$

In all test cases we consider the ideal gas equation of state with $\gamma = 1.4$. The Prandtl number and the ideal gas constant are chosen as $Pr = 1.0$, $R = 1$ respectively, except when indicated otherwise.

10.4.1 Shock tube problem

This test case corresponds to the shock tube problem of the Sod type discussed in Section 8.4.2 for the Navier-Stokes equations. The aim of this test case is to study the balancing effects of physical viscosity, the heat flux and artificial numerical viscosity, from the point of view of obtaining non-oscillatory solution profiles. The initial left state is given by $(\rho, u_1, u_2, p)_L = (1.0, 0.0, 0.0, 1.0)$ and the right state is given by $(\rho, u_1, u_2, p)_R = (0.125, 0.0, 0.0, 0.1)$, with the initial discontinuity placed at $x = 0.5$. The domain is taken to be $[0, 1] \times [0, 0.4]$. The primary and the Voronoi dual meshes used for the simulations are similar to the ones shown in Figure 9.10. We discretize the domain with 100 vertices in the direction of the flow and 80 vertices along the flow cross-section. This will be the *coarse mesh*, which corresponds to a mesh size of $h = 10^{-2}$. We also consider a finer mesh obtained by taking 10 times the original number of vertices in each direction, leading to a mesh size of $h = 10^{-3}$. Inflow farfield conditions are imposed on the left wall and outflow conditions on the right. The top and bottom walls have periodic boundary conditions. The final time is set as $t_f = 0.2$ with CFL = 0.5. All simulations are performed with the KEPES-TeCNO flux.

We consider two different viscous regimes determined by the value of μ. In each regime, we switch off the heat flux, physical viscosity or artificial numerical viscosity by setting $\mu = 0$, $\kappa = 0$ or $D_{ij} = 0$ respectively. Note that if the heat flux is active in the absence of physical viscosity, κ is first evaluated from the given non-zero value of μ using (3.12), following which μ is set to zero. We use the notations "ND", "VISC" and "HEAT" to indicate whether the numerical viscosity, physical viscosity and heat flux are respectively active during a particular simulation. The solution with ND+VISC+HEAT will be used as a reference solution, as it gives oscillation-free solutions in all the scenarios considered below.

Regime 1 ($\mu = 2.0 \times 10^{-4}$)

The solutions on the coarse mesh for all configurations with ND active, are almost indistinguishable with non-oscillatory solution profiles, as can be seen in Figure 10.5(a). However, the solutions evaluated with ND switched off are highly oscillatory, as shown in Figures 10.5(b)-(c), indicating that the physical viscosity and heat flux are not enough to stabilize the oscillations in low viscosity regimes. This is not surprising as the mesh Peclet number for the given set-up is quite large, as shown in Figure 10.5(d). The oscillations dampen down on the finer mesh with $h = 10^{-3}$, as shown in Figure 10.6. In fact, the solution with VISC + HEAT is non-oscillatory and indistinguishable with the solution obtained if ND is active as well. Note that the solutions seem to converge to a different profile in the absence of the heat flux.

Regime 2 ($\mu = 2.0 \times 10^{-3}$)

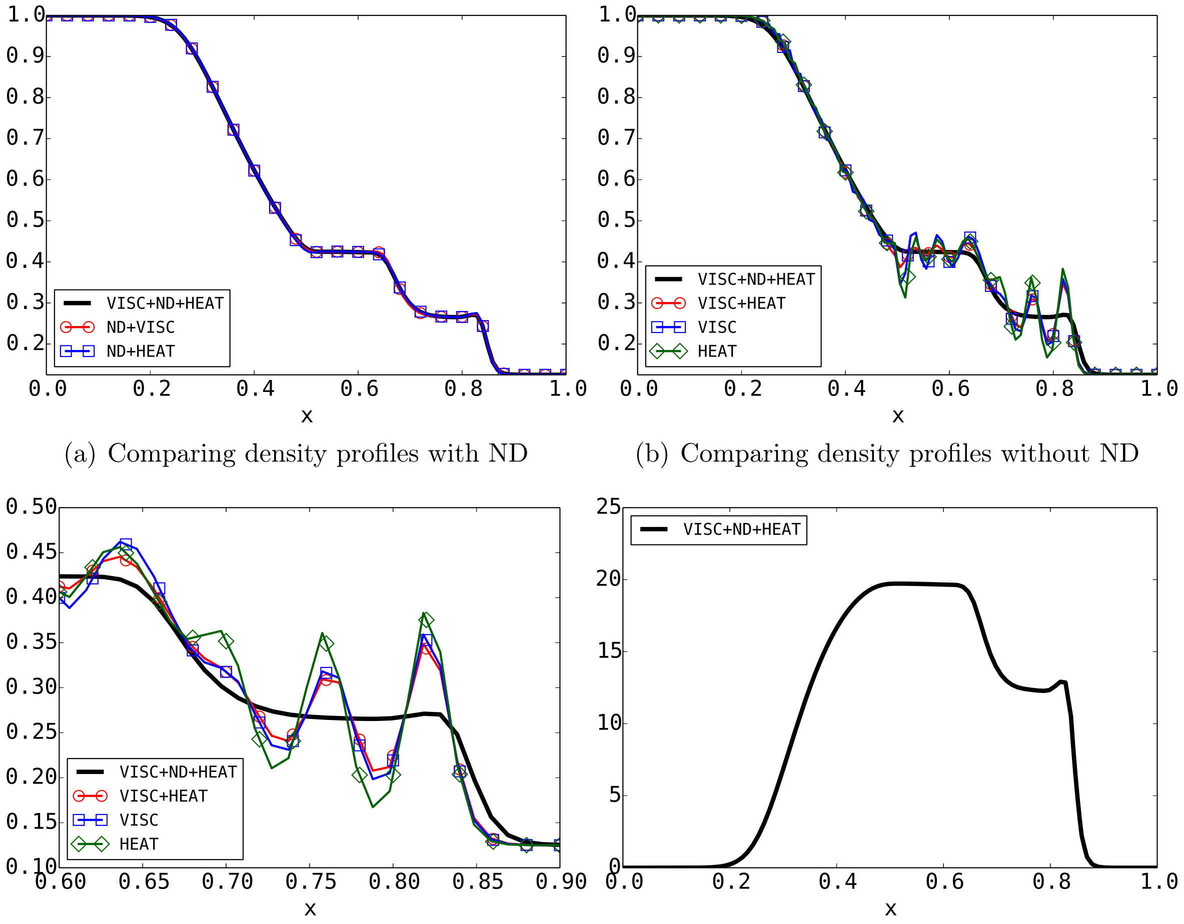
In this regime, the viscosity is increased by a factor of 10. We observe from Figure 10.7 that the solutions are once again oscillatory in the absence of numerical diffusion. However, the profile is non-oscillatory with only physical viscosity and heat flux active. This is expected, as the mesh Peclet number is within a reasonable range to preform DNS of the Navier-Stokes equation. Moreover, the oscillations with purely physical viscosity or heat flux active, are much milder, as compared to those observed in the previous regime. The solutions on a finer mesh (see Figure 10.8) have the same characteristics as those observed in the previous regime.

Based on the above results, we make the following remarks:

1. When Pe is much larger than unity, as in the case in Figure 10.5(b), the solution is highly oscillatory in the absence of ND.

2. When Pe below 2.0, the scheme with only VISC+HEAT active seems to give *almost* non-oscillatory solutions. The term almost is used due to the existence of very minor oscillations, as is shown in Figure 10.9. Also note that the initial condition is discontinuous, which furthers the need of introducing artificial numerical dissipation to eliminate oscillations due to Gibbs phenomenon. Thus, we prefer to use the with scheme ND+VISC+HEAT for problems with discontinuous initial data or having coarse meshes.

3. Physical viscosity seems to play an important role in suppressing oscillations in the region corresponding to a *shock* for the inviscid (Euler) solution. Similarly,



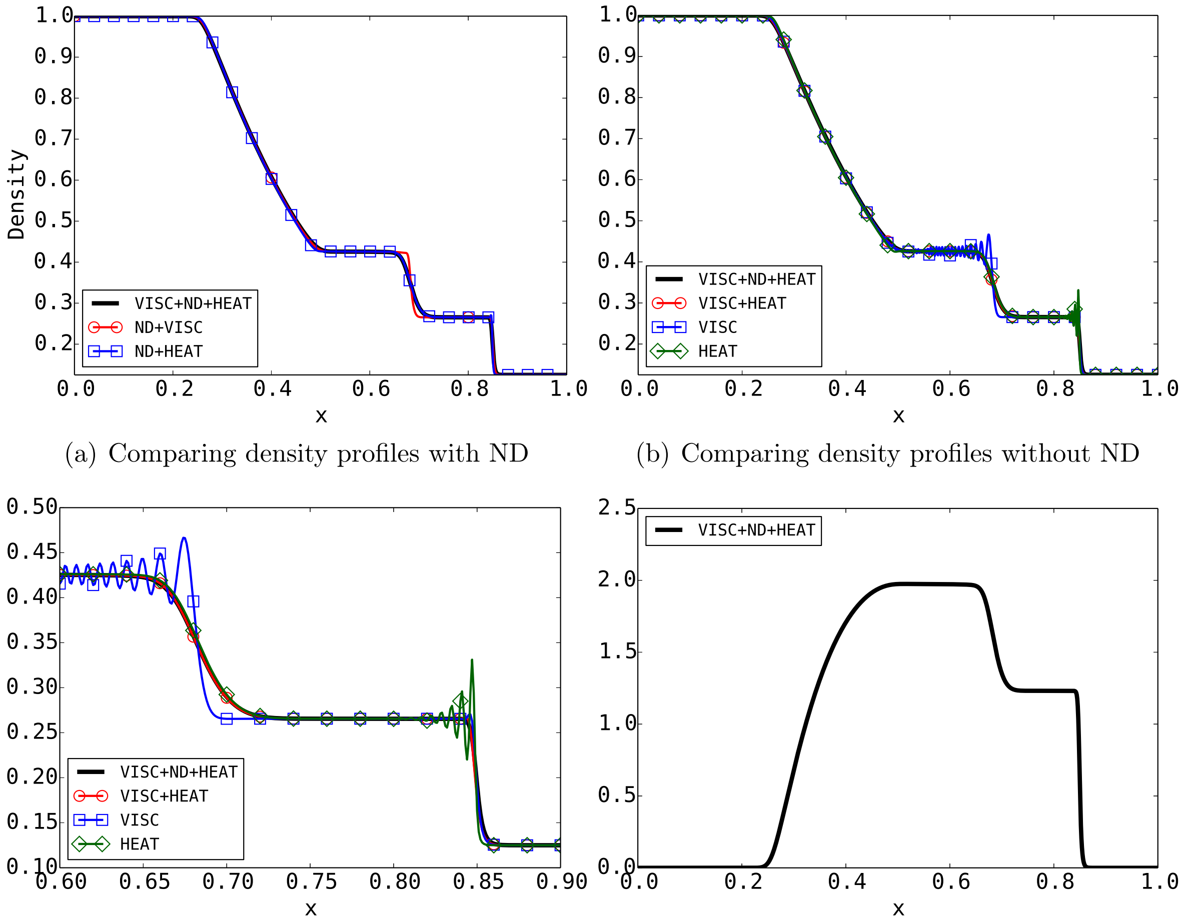
(a) Comparing density profiles with ND

(b) Comparing density profiles without ND

(c) Zoomed density profiles without ND

(d) Mesh Peclet Number

Figure 10.5: Shock tube for $\mu = 2.0 \times 10^{-4}$ on a mesh with $h = 10^{-2}$, solved with KEPES-TeCNO.



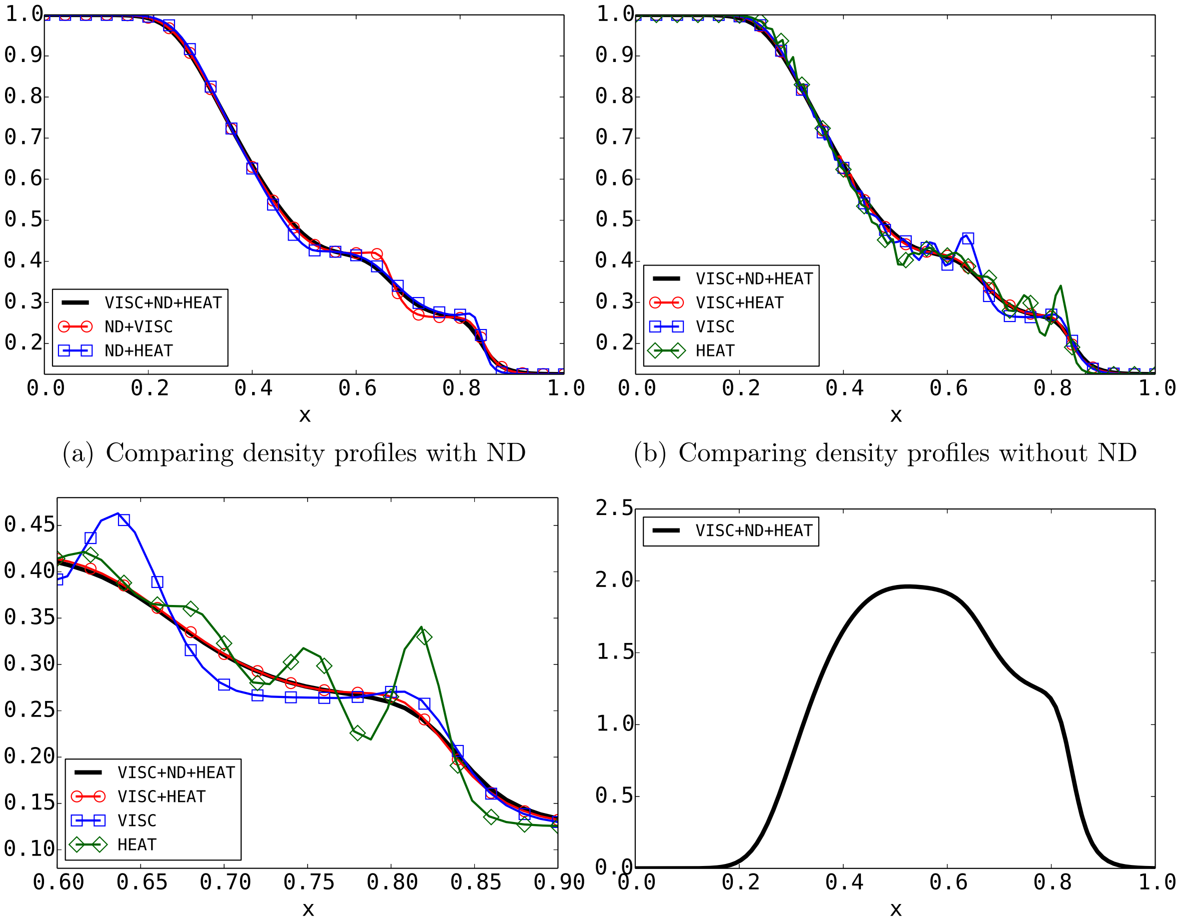
(a) Comparing density profiles with ND

(b) Comparing density profiles without ND

(c) Zoomed density profiles without ND

(d) Mesh Peclet Number

Figure 10.6: Shock tube for $\mu = 2.0 \times 10^{-4}$ on a mesh with $h = 10^{-3}$, solved with KEPES-TeCNO.



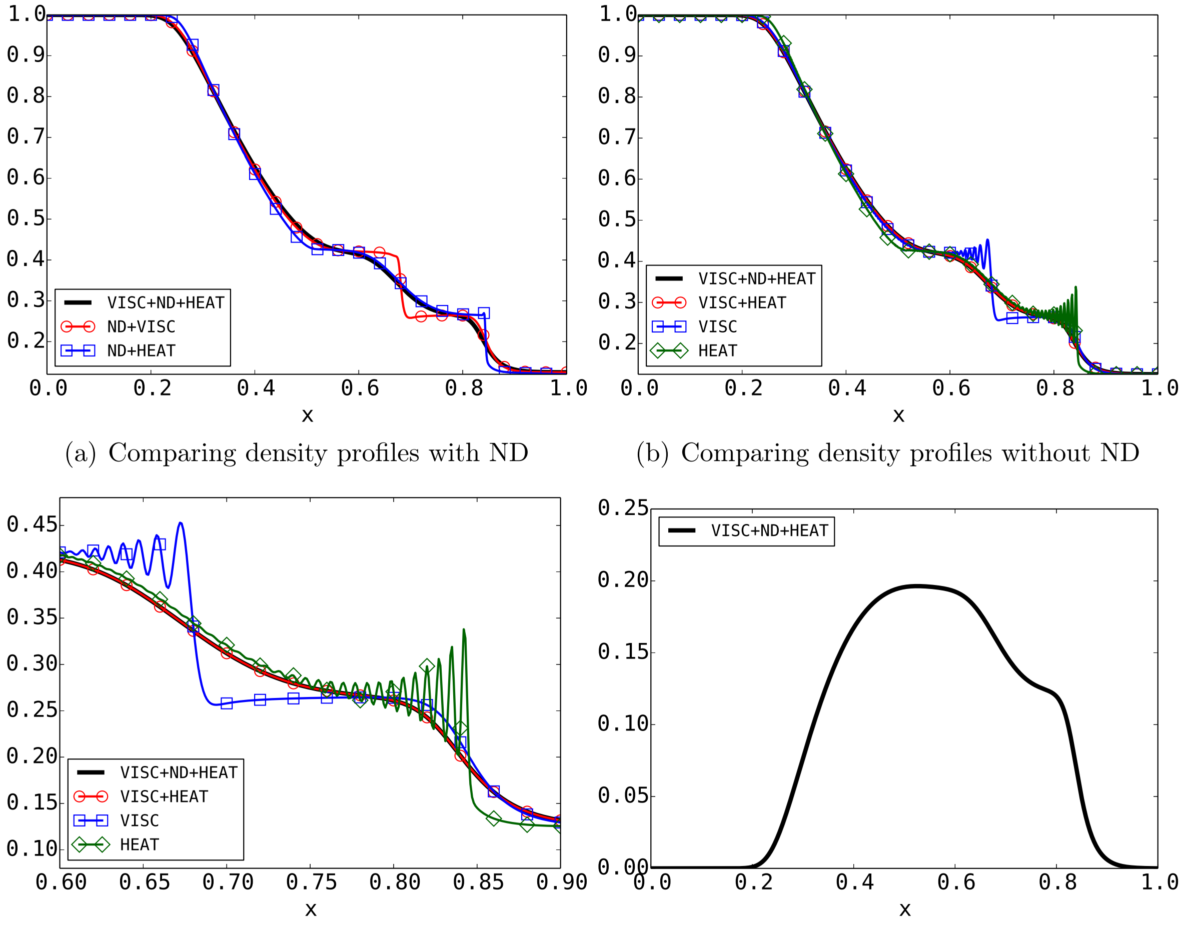
(a) Comparing density profiles with ND

(b) Comparing density profiles without ND

(c) Zoomed density profiles without ND

(d) Mesh Peclet Number

Figure 10.7: Shock tube for $\mu = 2.0 \times 10^{-4}$ on a mesh with $h = 10^{-2}$, solved with KEPES-TeCNO.



(a) Comparing density profiles with ND

(b) Comparing density profiles without ND

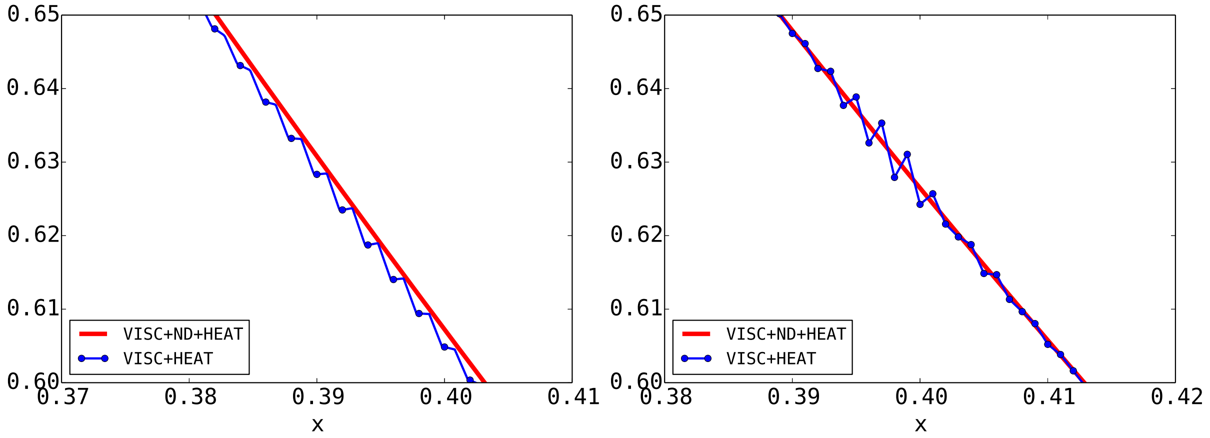
(c) Zoomed density profiles without ND

(d) Mesh Peclet Number

Figure 10.8: Shock tube for $\mu = 2.0 \times 10^{-4}$ on a mesh with $h = 10^{-3}$, solved with KEPES-TeCNO.

the heat flux seems to be key in obtaining non-oscillatory solutions in the region corresponding to a *contact discontinuity* for the inviscid solution.

4. It is evident from Figure 10.8 that, in the absence of heat flux or in the absence of physical viscosity, the solutions converge to very different profiles, as compared to the case when both physical viscosity and heat flux are active.



(a) $\mu = 2.0 \times 10^{-4}$

(b) $\mu = 2.0 \times 10^{-3}$

Figure 10.9: Zoomed density profile for the shock tube, with mesh size $h = 10^{-3}$, solved with KEPES-TeCNO.

10.4.2 Vortex advection

This test involves the advection of a smooth vortex in a viscous flow. The initial conditions of the flow are the same as those prescribed in Section 5.8.4 for the isentropic vortex. We choose $M = 2\sqrt{2}$, $\alpha = 45°$ and a constant coefficient of viscosity $\mu = 10^{-5}$. The vortex is advected till time $t = 50$ with a CFL=0.4, during which the vortex completes 10 cycles along the domain diagonal. We assume periodic boundary conditions.

The initial and and final density profiles with the KEPES-TeCNO scheme are shown in Figure 10.10. The profile at $t_f = 50$ seems a little diffused as compared to the initial profile, which can be attributed to physical viscosity. For an entropy stable scheme, the total (mathematical) entropy should decrease with time, in accordance with the results proved in Section 10.2. This behaviour is clearly observed in Figure 10.11 with the KEPES-TeCNO scheme.

10.4.3 Laminar flat-plate boundary layer

This problem corresponds to a viscous flow over a flat plate, which leads to the development of a boundary layer near the plate surface. The computational domain is taken as $[0, 1.5] \times [0, 0.25]$, with the primary and median dual grids shown in Figure 10.12. There is an initial inlet portion of the domain of length 0.5 units on which slip boundary condition is imposed, followed by the no-slip boundary corresponding to the flat plate of length

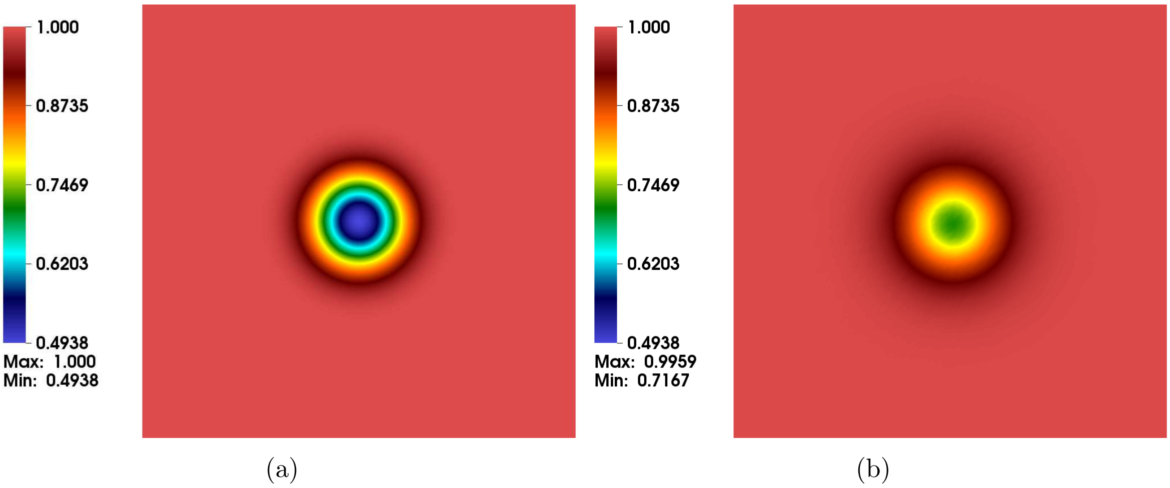


Figure 10.10: Density profile for the advecting vortex solved with KEPES-TeCNO (a) t=0, (b) t=50.

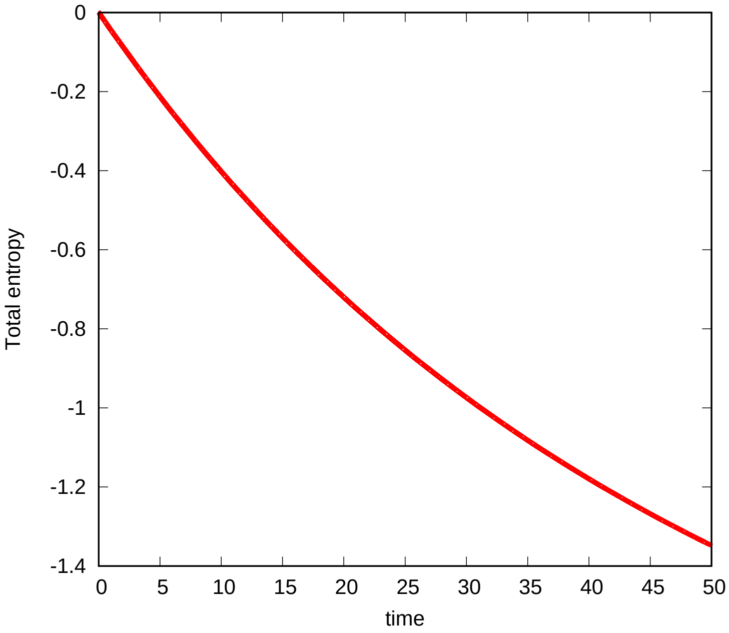


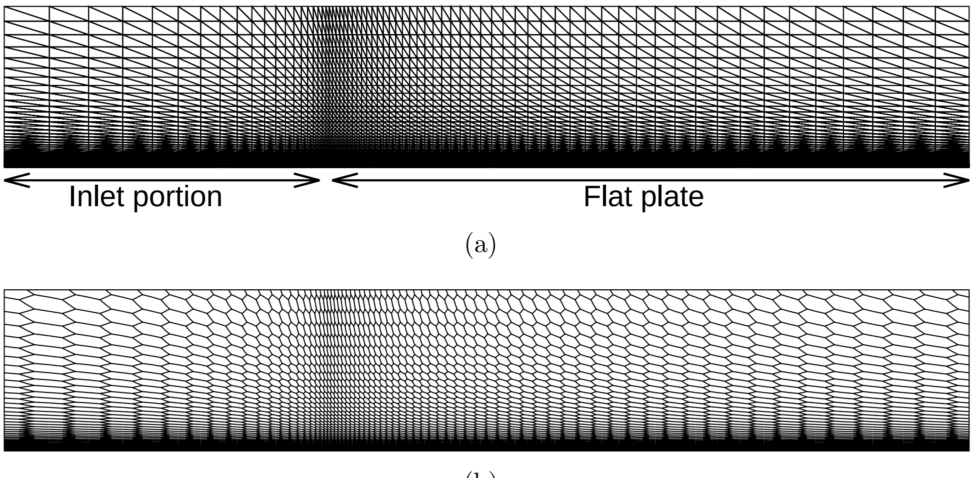
Figure 10.11: Evolution of total entropy for advecting vortex solved with KEPES-TeCNO.

1 unit. Adiabatic conditions are used on the flat plate boundary, with the Reynolds number corresponding to the plate length being 10^5. At the top and outlet, the free-stream pressure is specified, while at the inlet the free-stream values are used together with the numerical flux function to compute the flux. The free-stream values used for the simulations are

$$p_\infty = 8610, \theta_\infty = 300, \mathbf{u}_\infty = (34.7189, 0)^\top,$$

with $Pr = 0.72$ and $R = 287$. The flow is initialized using the free-stream values, which has a Mach number of 0.1.

Figure 10.13 shows the comparison of the numerically approximated velocities with the Blasius semi-analytical solution in the standard non-dimensional units. These results are taken on the vertical line through the point on the plate, at a distance $x = 0.8$ from the plate tip. The solutions obtained using the KEPES-TeCNO scheme and Roe-MUSCL scheme are almost identical. The vertical velocity component is much weaker than the stream-wise component, and thus probably more sensitive to numerical dissipation and compressibility effects. The KEPES-TeCNO scheme is able to capture this profile quite accurately, indistinguishable from the standard Roe solver.



(a)

(b)

Figure 10.12: Computational domain for flat-plat problem: (a) Primal mesh, (b) Median dual mesh.

10.4.4 Lid-driven cavity

This is a standard problem used for validation of numerical methods for incompressible flows [66, 14, 42]. The problem describes a fluid at Reynolds number 10^3, in a unit square domain with no-slip conditions on three sides, while the top boundary of the domain moves with a velocity $\mathbf{u} = (1, 0)^\top$. All walls are isothermal and the flow is initialized with

$$p_0 = \theta_0 = 71.429, \mathbf{u}_0 = (1, 0)^\top,$$

with $Pr = 0.7$ and θ_0 corresponding to the wall temperature. The laminar solution of the problem is steady, with a Mach number of 0.1 corresponding to the moving lid. The

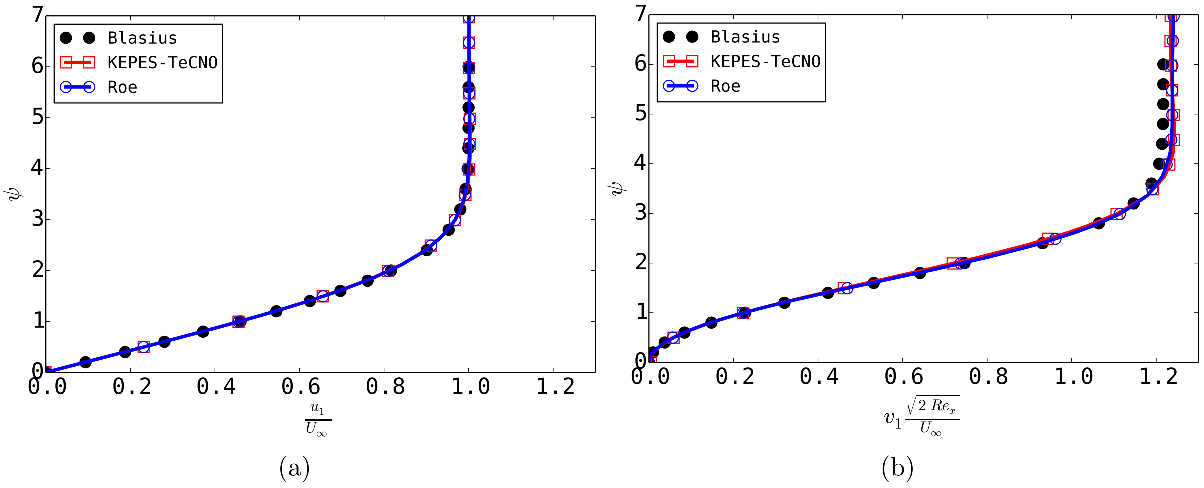


Figure 10.13: Laminar flat plate boundary layer: (a) Stream-wise velocity, (b) Vertical velocity. Re_x is the Reynolds number corresponding to the plate length at $x = 0.8$, while $\psi = y\sqrt{(0.5Re_x)}/x$ is the non-dimensionalised vertical distance from the plate at $x = 0.8$.

numerical data of Ghia et al. [42] is used as a reference solution. We plot the horizontal velocity profile along the vertical line through the midpoint of the domain, and vertical velocity profile along the horizontal line through the midpoint of the domain, as shown in Figure 10.14. The KEPES-TeCNO scheme leads to solutions that match the reference profiles quite accurately, and are also comparable to the solution from the Roe-MUSCL solver.

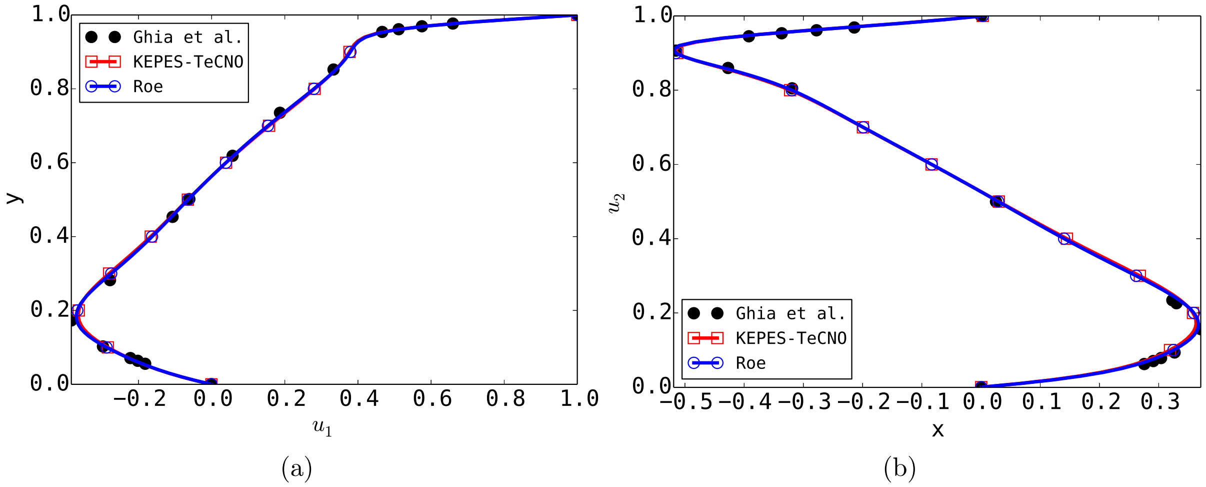
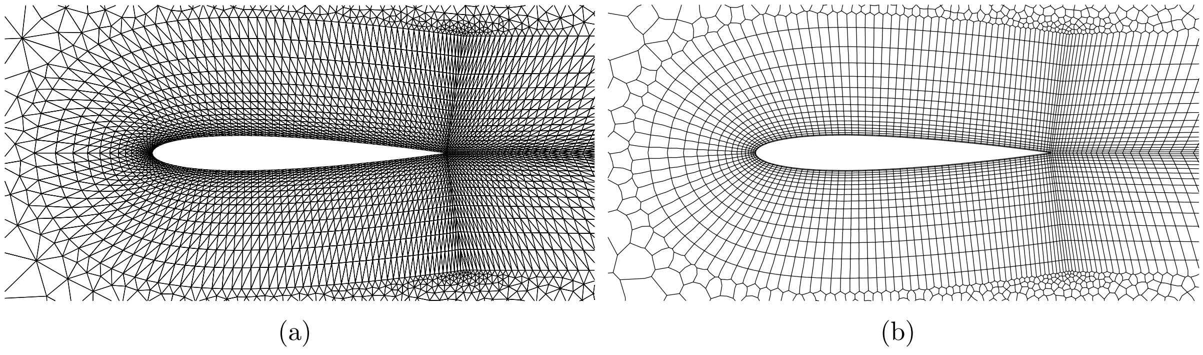


Figure 10.14: Lid driven cavity: (a) Streamwise velocity, (b) Vertical velocity

10.4.5 Transonic flow past NACA-0012 airfoil

This test case involves a steady flow past a symmetric NACA-0012 airfoil. A zoomed view of the primary and (Voronoi) dual meshes of the computational domain are shown in Figure 10.15, which will correspond to base mesh $L0$. The primary mesh has 4948 vertices, with 60 vertices lying on each of the top and bottom airfoil surfaces. The mesh is refined by splitting each primary triangle into four similar triangles to obtain the mesh Ln, where n refers to the number of mesh refinements. Adiabatic no-slip wall conditions are imposed on the surface of the air-foil. The dual mesh cells are flat and clustered near the surface of the airfoil, to capture the boundary layer and reduce cross-diffusion.



(a) (b)

Figure 10.15: Mesh $L0$ for NACA airfoil: (a) Primal mesh, (b) Voronoi dual mesh.

We also consider the following quantities for comparison:

- Surface pressure coefficient C_p along the airfoil surface

$$C_p = \frac{p - p_\infty}{\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2 l_\infty}.$$

- Surface skin friction coefficient C_f along the airfoil surface

$$C_f = \frac{(\tau \mathbf{n})\cdot \mathbf{n}^\perp}{\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2 l_\infty}.$$

- Pressure induced lift and drag force coefficients:

$$c_{dp} = \frac{\int\limits_S p(\mathbf{n}\cdot \Psi_d)\mathrm{d}s}{\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2 l_\infty}, \qquad c_{lp} = \frac{\int\limits_S p(\mathbf{n}\cdot \Psi_l)\mathrm{d}s}{\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2 l_\infty}.$$

- Lift and drag force coefficients due to viscous forces:

$$c_{df} = \frac{\int\limits_S (\tau \mathbf{n})\cdot \Psi_d \mathrm{d}s}{\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2 l_\infty}, \qquad c_{lf} = \frac{\int\limits_S (\tau \mathbf{n})\cdot \Psi_l \mathrm{d}s}{\frac{1}{2}\rho_\infty |\mathbf{u}_\infty|^2 l_\infty}.$$

In the above expressions, ρ_∞ is the free-stream density, $\mathbf{u}_\infty$ is the free-stream velocity magnitude and l_∞ corresponds to the characteristic length scale which is chosen to be the span of the airfoil i.e., $l_\infty = 1$. Furthermore, $\mathbf{n}$ is the outward normal at domain boundary, $\mathbf{n}^\perp$ is the the tangent at the domain boundary, $\Psi_d = (\cos\alpha, \sin\alpha)^\top, \Psi_l =$

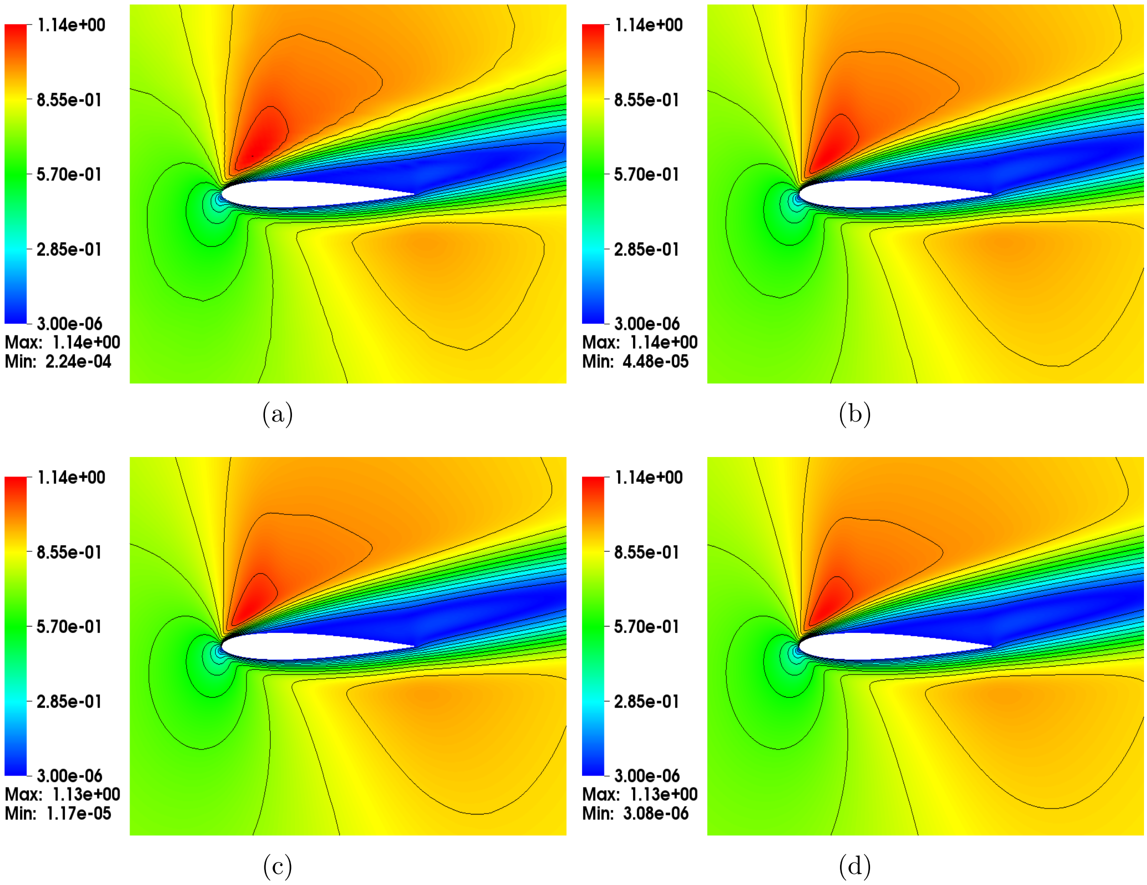


Figure 10.16: Mach number plots for flow past a NACA0012 airfoil, at Re=500, M=0.8 and a.o.a 10° with KEPES-TeCNO scheme (a) L0, (b) L1, (c) L2, (d) L3

$(-\sin\alpha, \cos\alpha)^\top$, with α being the angle of attack (a.o.a.). For our simulations, we choose $\rho_\infty = 1$, $p_\infty = 2.85$ and $Pr = 0.72$. We consider two flow configurations, and the value of $\mathbf{u}_\infty$ depends on the configuration under consideration. The flow is initialized using the free-stream values.

The first one corresponds to a flow with Reynolds number 500, free-stream Mach number of 0.8 and 10 degree angle of attack. The free-stream velocity is set at $\mathbf{u}_\infty = (1.574, 0.277)^\top$. The Mach number profiles with the high-resolution KEPES-TeCNO scheme on meshes $L0 - L3$ are shown in Figure 10.16. Apart from small changes in the maximum and minimum values of the Mach number, no significant qualitative differences can be observed with mesh refinement.

A mesh refinement study (see Figure 10.17) indicates the convergence of C_p and C_f to the reference data obtained from [114]. The C_p plots for meshes $L2$ and $L3$ are almost indistinguishable, while there is a significant improvement in the peak values of C_f on going from mesh $L2$ to $L3$. This re-affirms the well known fact that the accurate evaluation of C_f is a much harder task as compared to the evaluation of C_p. The lift and drag coefficients computed on the finest mesh (see Table 10.1), are comparable to the reference values taken from [114].

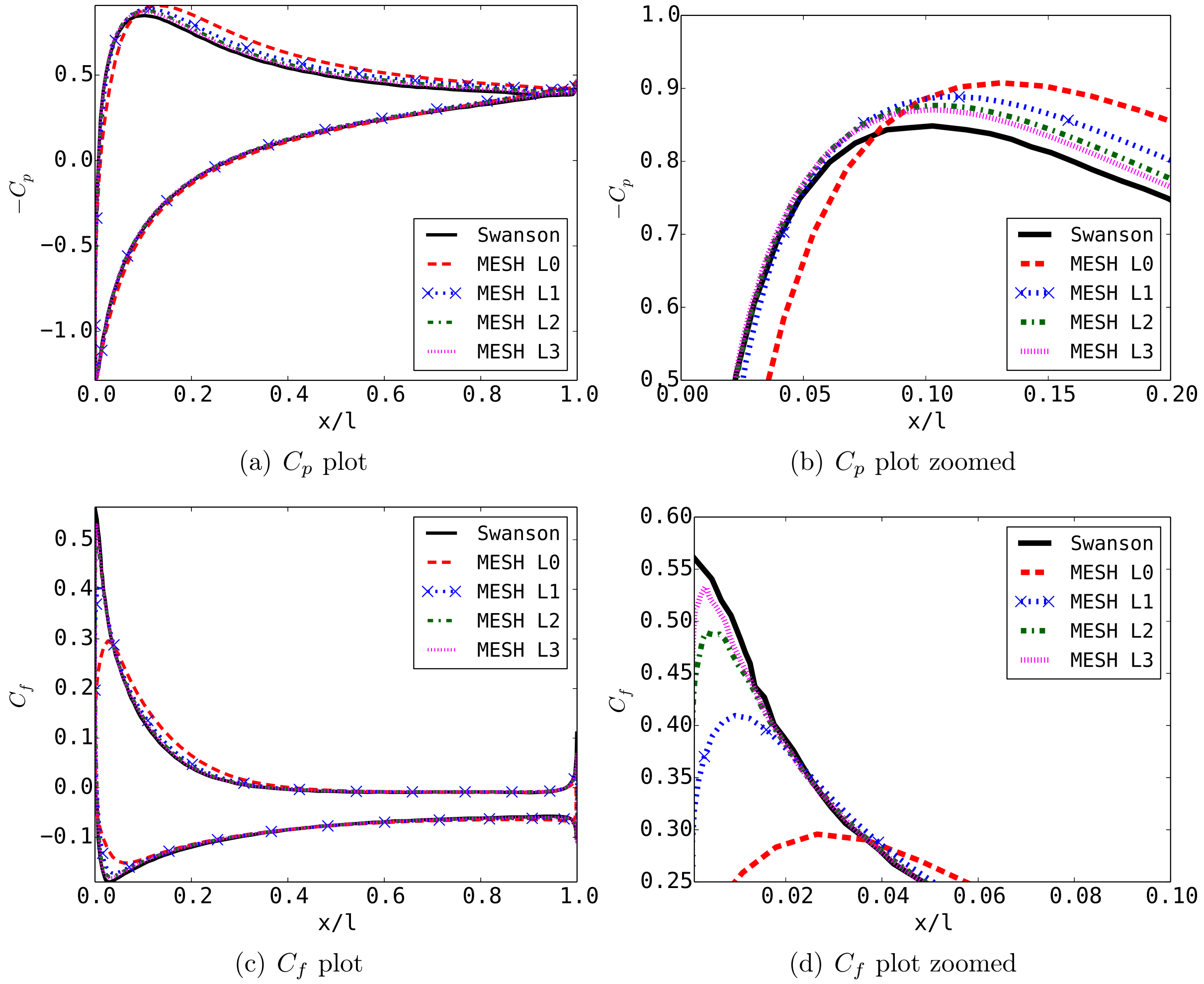
Mesh	c_{dp}	c_{df}	c_{lp}	c_{lf}	$c_{lp} + c_{lf}$
L0	1.637e-01	1.247e-01	4.946e-01	-4.468e-03	4.901e-01
L1	1.538e-01	1.237e-01	4.636e-01	-3.443e-03	4.601e-01
L2	1.495e-01	1.235e-01	4.520e-01	-3.669e-03	4.483e-01
L3	1.476e-01	1.235e-01	4.470e-01	-3.945e-03	4.430e-01
Reference:	1.475e-01	1.275e-01	-	-	4.363e-01

Table 10.1: NACA-0012 at Re=500, M=0.8 and a.o.a 10°, lift and drag coefficients, with KEPES-TeCNO scheme. Reference values taken from [114].

The second flow configuration considered corresponds to a laminar flow at a Reynolds number of 5000, Mach number of 0.5 and zero degree angle of attack. The free-stream velocity is set at $\mathbf{u}_\infty = (0.9987, 0)^\top$. Once again, the Mach number profiles shown in Figure 10.18 are qualitatively very similar to each other on the various mesh levels. Mesh refinement leads to a substantial improvement in the C_p and C_f plots as shown in Figure 10.19. The lift and drag coefficients on various mesh levels are given in Table 10.2. Since the a.o.a is zero degrees, the lift coefficients should be almost zero. Furthermore, the lift coefficients shown in 10.2, converge to the reference values taken from [129].

Mesh	c_{dp}	c_{df}	c_{lp}	c_{lf}
L0	3.134e-02	3.089e-02	-4.913e-06	6.341e-06
L1	2.513e-02	3.121e-02	8.074e-05	5.891e-06
L2	2.308e-02	3.188e-02	-8.021e-06	2.539e-06
L3	2.254e-02	3.221e-02	4.775e-05	1.476e-06
Reference:	2.249e-02	3.291e-02	-	-

Table 10.2: NACA-0012 at Re=5000, M=0.5 and a.o.a 0°, lift and drag coefficients, with the KEPES-TeCNO scheme. Reference values taken from [129].



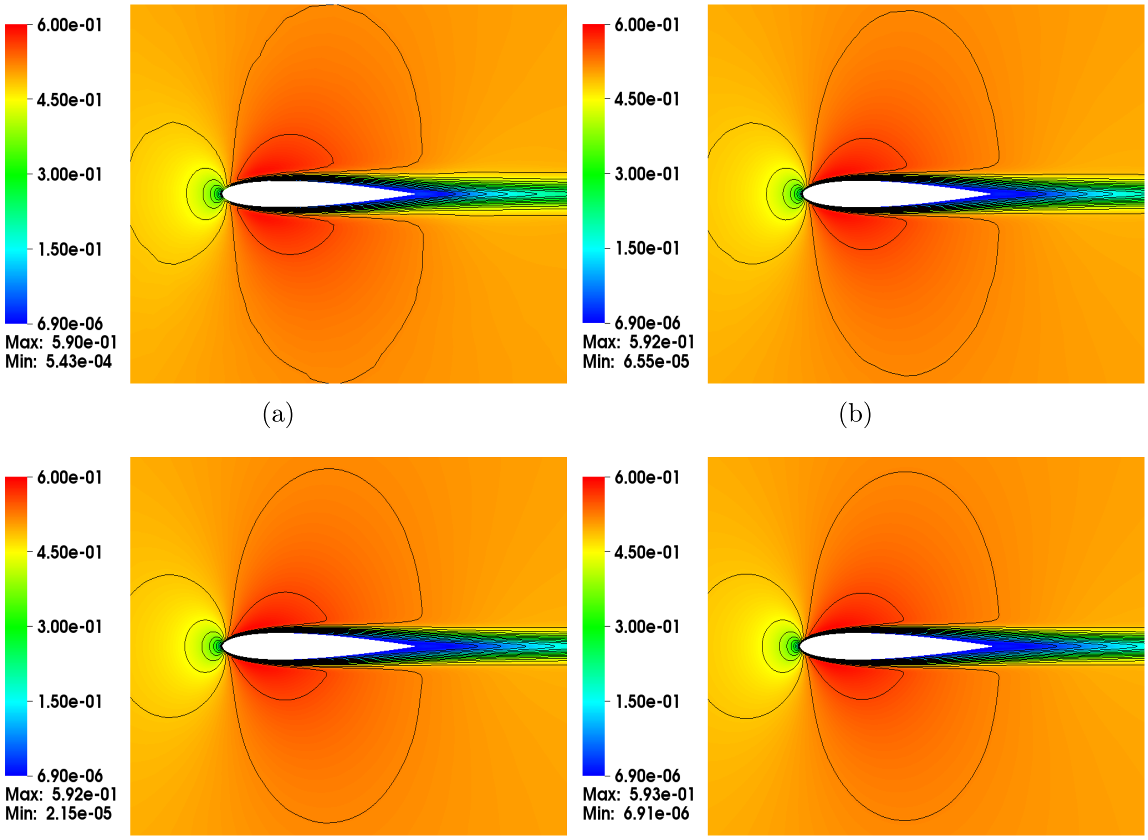
(a) C_p plot

(b) C_p plot zoomed

(c) C_f plot

(d) C_f plot zoomed

Figure 10.17: Transonic flow past a NACA0012 airfoil, at Re=500, M=0.8 and a.o.a $10°$, with KEPES-TeCNO scheme.



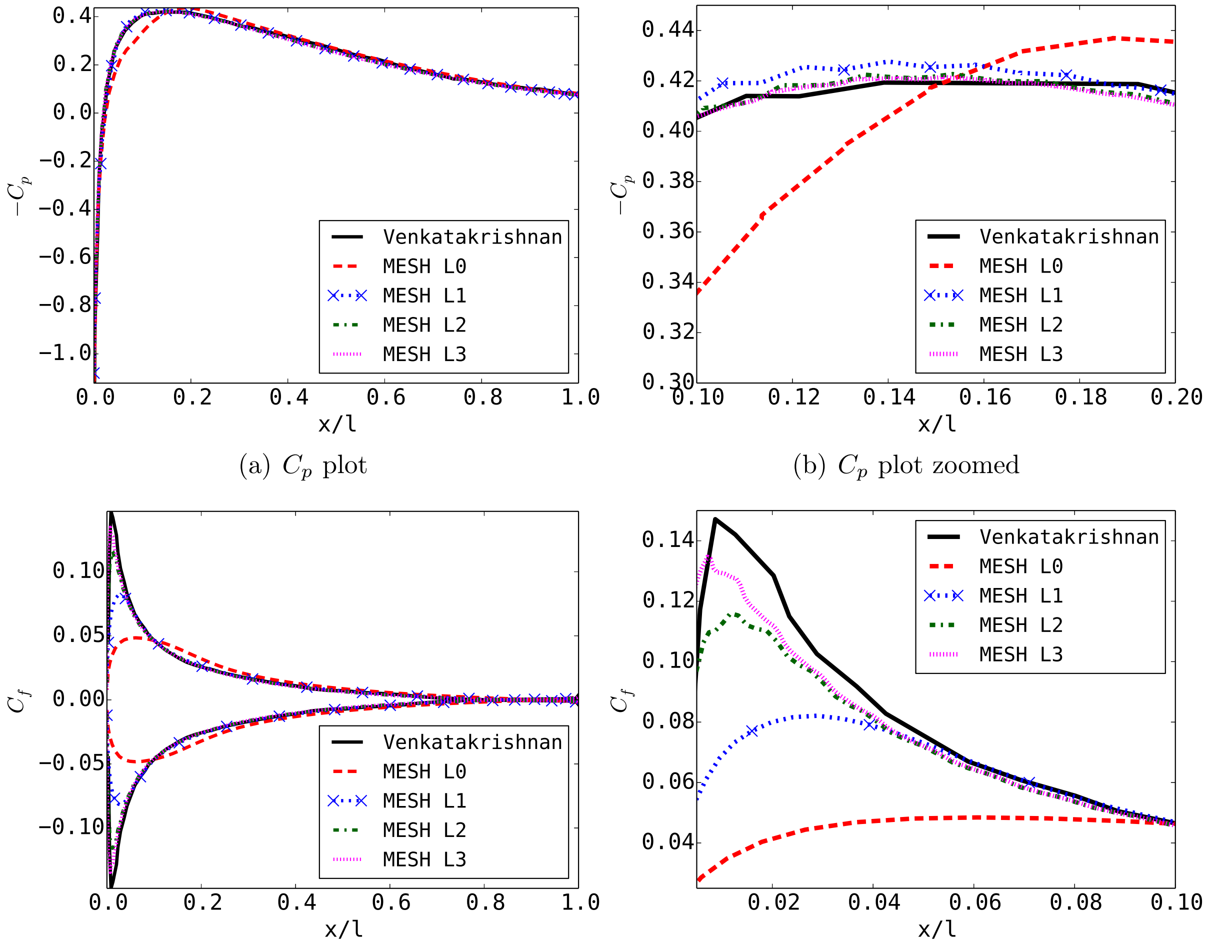
(a)

(b)

(c)

(d)

Figure 10.18: Mach number plots for flow past a NACA0012 airfoil, at Re=5000, M=0.5 and a.o.a 0° **, with KEPES-TeCNO scheme (a) L0, (b) L1, (c) L2, (d) L3**



(a) C_p plot

(b) C_p plot zoomed

(c) C_f plot

(d) C_f plot zoomed

Figure 10.19: Transonic flow past a NACA0012 airfoil, at Re=5000, M=0.5 and a.o.a $0°$, with KEPES-TeCNO scheme.

10.4.6 Flow past a cylinder

This test case involves a laminar unsteady flow past a cylinder inside a channel [102]. The geometry of the domain is shown in Figure 10.20. The cylinder is offset somewhat from the center of the channel to destabilize the otherwise steady symmetric flow. On the left, the inflow boundary condition is imposed with $p = \theta = 160.7143$, $u_1 = 4u_m y(H - y)/H^2$ and $u_2 = 0$, where $u_m = 1.5$ is the maximum velocity and the Mach number corresponding to u_m is 0.1. Isothermal no-slip boundary conditions are imposed on the cylinder surface, and the top and bottom boundaries. The Reynolds number of the flow corresponding to the cylinder is 100.

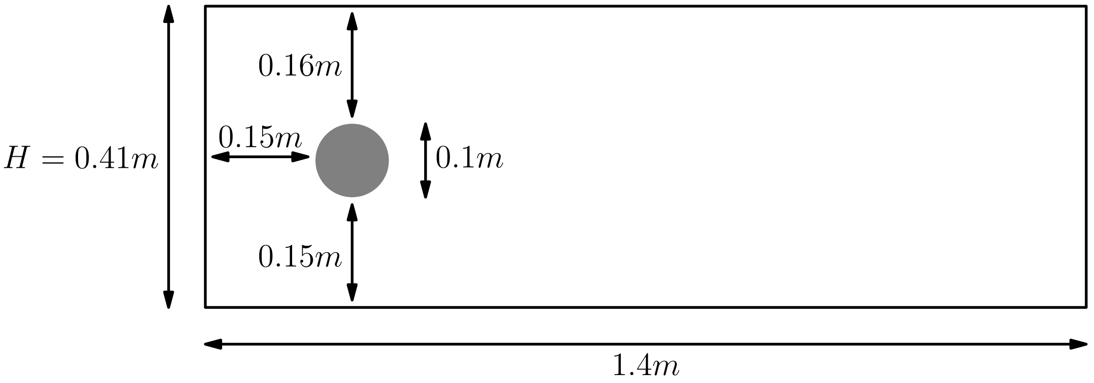


Figure 10.20: Geometry of domain for flow past a cylinder.

After some time, a Von Karman vortex street appears with a periodic shedding of eddies from alternate sides of the cylinder. This is typical for slow flows past a slender body. Simulations are performed with CFL=0.6 using the KEPEC scheme, which is the KEPES-TeCNO scheme without any artificial dissipation. Note that this would still lead to entropy stability as discussed in remark 10.2.1. Since we are performing a DNS for the test problem, we choose a fine mesh. The total number of grid points for the simulation is 427640, which ensures that the mesh Peclet number is close to unity throughout the mesh. The vorticity profile is shown in Figure 10.21, which clearly depicts the periodic vortex shedding. The periodic behavior can also be observed in Figure 10.22, where the evolution of the coefficient of total lift $c_l = c_{lp} + c_{lf}$ and the coefficient of total drag $c_d = c_{dp} + c_{df}$ on the surface of cylinder are shown. Another quantity of interest from the point of view of the oscillations, is the Strouhal number given by $St = \mathcal{F}D/\bar{u}$, where $\mathcal{F}$ is the frequency of oscillation for the lift, $D = 0.1$ is the diameter of the cylinder and $\bar{u} = 1$ is the mean inflow velocity. The time period of oscillation as observed from the evolution of c_l is $\tau = 0.333$. The peak c_l, c_d values and the Strouhal number for this simulation are presented in Table 10.3, which are close to the range of results given in [102].

	$\max c_l$	$\max c_d$	St
Simulation values	0.96068	3.2189	0.3003
Reference range	[0.99,1.01]	[3.22,3.24]	[0.284,0.300]

Table 10.3: Comparison of quantities of interest for flow past a cylinder.

The boundary conditions are imposed weakly in the numerical scheme. However, the scheme is able to ensure the numerical solution are consistent with the conditions near the boundaries. In Figure 10.23, we can clearly see the velocity field near the top-left and top-right cylinder boundary matches the no-slip boundary conditions quite well.

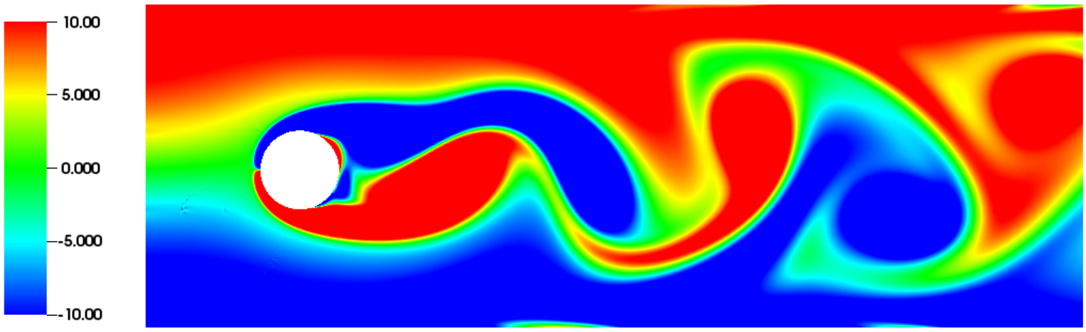
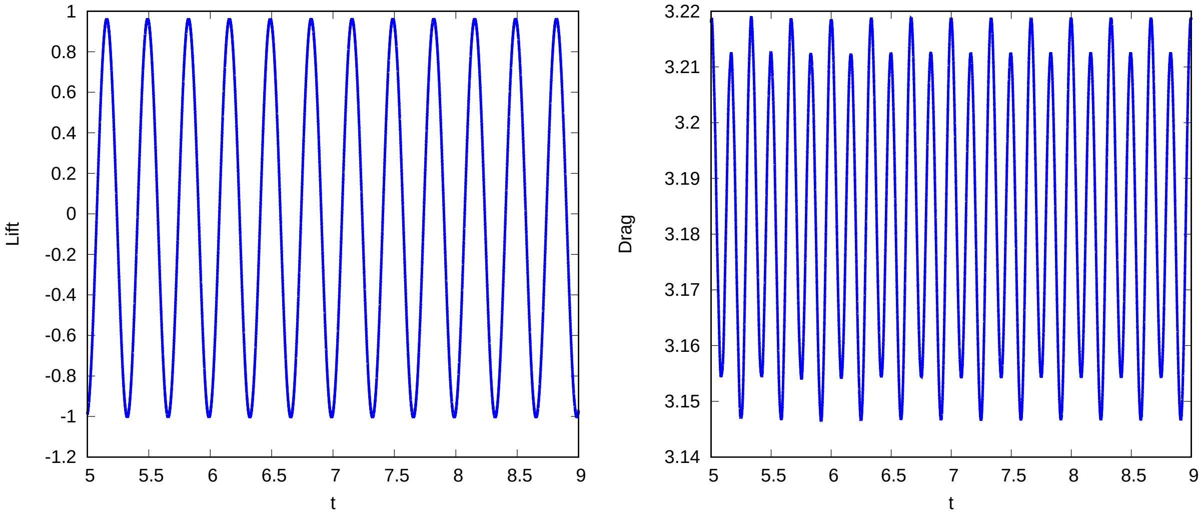


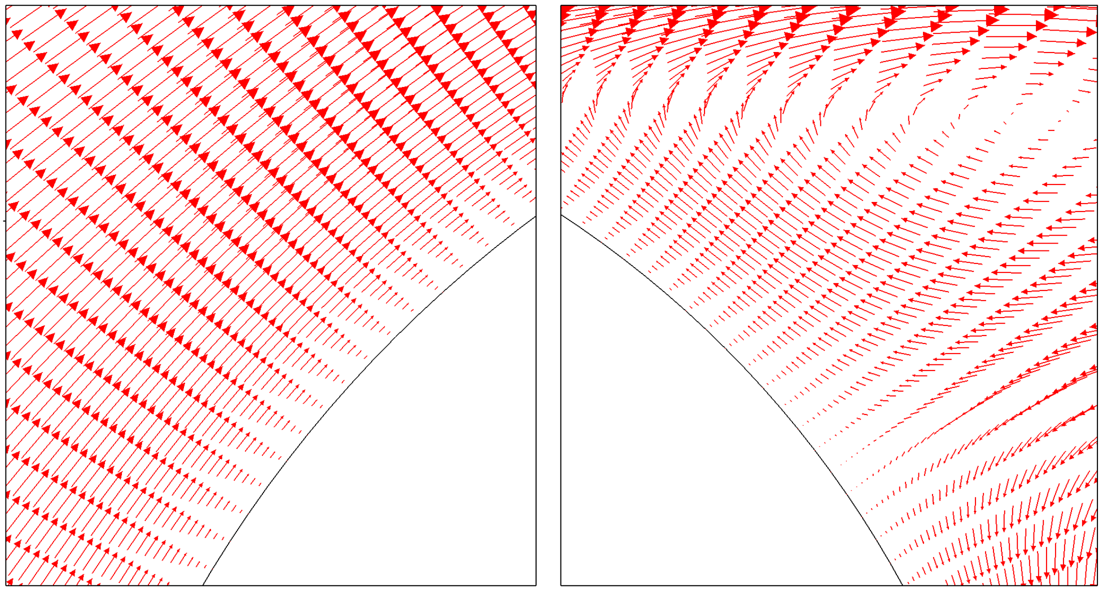
Figure 10.21: Von Karman vortex street for flow past a cylinder with KEPEC scheme; vorticity plot.



(a) c_l

(b) c_d

Figure 10.22: Evolution of c_l and c_d on the surface of the cylinder, with the KEPEC scheme.



(a) Top-left surface

(b) Top-right surface

Figure 10.23: Velocity field near cylinder surface.

11. Conclusions

We have investigated the performance of high-order entropy conservative finite difference schemes for the Euler equations on Cartesian grids. Comparing three central fluxes, namely, KEP, ROE-EC and KEPEC, we observe that the results with KEPEC are the most accurate for the DNS of the Euler equations with smooth solutions. Thus, we conjecture that a numerical flux that is both entropy conservative and kinetic energy preserving, is more accurate for long-time simulations of smooth solutions, as compared to a flux that has only one of these two properties.

Entropy stable schemes, i.e., schemes satisfying a discrete form of the entropy inequality, are constructed by combining high-order entropy conservative fluxes and entropy variable based numerical dissipation terms. The high-order dissipation terms are obtained by reconstructing the scaled entropy variables such that a sign property is satisfied at each cell-interface. Only a handful of reconstruction techniques are known to satisfy this property. We proposed a new WENO-type reconstruction, called SP-WENO, which satisfies the sign-property and possesses additional symmetry and stability properties. Unlike ENO interpolation techniques (which are also sign-preserving), SP-WENO does not show a deterioration in the order of accuracy when used in the TeCNO framework. While SP-WENO is shown to perform well for scalar conservation laws, it leads to undesirably large oscillations close to discontinuities when tested with the Euler equations. The oscillations can be attributed to the absence of numerical dissipation in the proximity of a shock or contact discontinuity. A modification to the reconstruction is suggested to ensure the dissipation does not vanish in key areas, while maintaining high-order of accuracy in smooth regions. The new method termed as SP-WENOc preserves most of the crucial properties of the original SP-WENO method, and gives better control of overshoots near discontinuities.

For the one-dimensional Navier-Stokes equations, it is possible to discretize the viscous flux to ensure that the semi-discrete scheme is kinetic energy preserving and entropy stable [16], when used in conjunction with the KEPEC flux. This discretization cannot be extended to higher dimensions due to the existence of cross derivative terms in the viscous flux. We proposed a multi-dimensional SBP-type discretization for the viscous terms, which leads to a kinetic energy preserving scheme if the viscous fluxes are written in terms of the stress tensor, while an entropy stable scheme can be obtained if the symmetric formulation of the viscous fluxes (in terms of the entropy variables) is used. However, it does not seem to be possible to discretize the viscous fluxes to satisfy both properties simultaneously for the higher-dimensional problem. DNS of the three-dimensional Taylor-Green vortex was performed with both types of discretizations, to test its ability to simulate turbulent flows when the mesh is under-resolved. Both methods were able to simulate the problem in a stable manner, while predicting the evolution of integral quantities such as the total kinetic energy and potential enstrophy quite accurately. On the other hand, the simulation blows up when a simple central average flux is used to approximate the inviscid flux.

We considered the vector-invariant formulation of the shallow water equations, which is a commonly used model in the meteorological community. The model is coupled with the scalar conservation law for absolute vorticity since i) obtaining vorticity indirectly from the velocity vector may not give a good approximation, ii) an accurate approximation of vorticity is needed, as several preserved invariants such as the total potential enstrophy are evaluated in terms of vorticity. We also included rotational effects due to Coriolis terms and allowed a non-flat bottom topography. A fourth-order energy preserving scheme is proposed, which is also capable of preserving the potential enstrophy reasonably well for smooth solutions.

While integrating a semi-discrete entropy conservative finite difference scheme in time, the temporal discretization can destroy the conservation property. However, a Crank-Nicolson type scheme for time-integration can ensure that entropy is conserved. On the other hand, Subbareddy and Candler [112] have proposed an implicit fully discrete scheme which is kinetic energy preserving, but cannot be shown to conserve entropy. We proposed a fully-discrete second-order accurate finite difference scheme, which is both entropy conservative and kinetic energy preserving when used in conjunction with the KEPEC flux. The scheme was used to perform DNS of the one-dimensional shock tube problem with the Navier-Stokes equations, and gave oscillation-free solutions when the mesh Peclet number was small enough to resolve all the necessary scales.

Many industrial problems involve complex domains, which are more easily discretized by unstructured grids. Thus, we designed a high-resolution vertex-centered finite volume scheme for conservation laws, which is provably entropy stable. The underlying computational domain was discretized using triangles and a dual cell is constructed around each vertex on which the conservation law is satisfied. The proposed scheme was constructed by combining entropy conservative fluxes, and numerical dissipation operators based on piecewise linear reconstruction of scaled entropy variables using the minmod limiter (which satisfies the sign property). In particular, the scheme with the KEPEC flux was used, and was termed as the KEPES-TeCNO scheme. To the best of our knowledge, the proposed KEPES-TeCNO scheme for the Euler equations is one of the first high-resolution finite volume scheme that is provably entropy stable on unstructured grids. The scheme is robust in approximating complex flow features such as strong (supersonic) shocks, shock reflections, slip lines and near incompressible flows. The robustness of the scheme is demonstrated through a large number of benchmark numerical experiments, that illustrate that the KEPES-TeCNO is at least as accurate as a standard high-resolution Roe-MUSCL method. The numerical tests show that the scheme is able to preserve positivity of density and pressure without any additional treatment on unstructured grids.

The KEPES-TeCNO scheme has been extended for the initial-boundary-value problem of the compressible Navier-Stokes equations. The viscous fluxes are evaluated on triangles in terms of the entropy variables, to preserve the symmetric structure of the continuous system. The boundary conditions are weakly imposed by constructing suitable inviscid boundary fluxes, based on the numerical value at the boundary node and the given boundary data. Additionally, the gradient of entropy variables evaluated in boundary cells are corrected using the boundary data, which in turn ensures the proper evaluation of viscous fluxes. The above ingredients together lead to the derivation of discrete non-linear entropy estimates for the Navier-Stokes equations with homogeneous boundary conditions.

11.1 Future scope

There are several extensions and possibly some unanswered questions associated with the work presented in this thesis. We list out a few of these crucial points below:

- The conjecture made about the accuracy of the KEPEC scheme for long-time simulations, is based on the results of the advecting isentropic vortex for the Euler equations. While these results seem promising, the claim needs to be substantiated with further experimentation, and possibly a theoretical proof.

- The SP-WENO gives oscillatory solutions for the Euler equations, which can be attributed to the vanishing of the reconstructed jump in key areas. Although the proposed correction of SP-WENO is able to control the overshoots near the discontinuities for the test cases considered in this thesis, one cannot guarantee that the new SP-WENOc will give oscillation-free solutions for other test problems. The fact that we were able to successfully modify SP-WENO while preserving most of its crucial properties, indicates that it is possible to find other types of modifications, thus giving more room to explore and experiment. It would also be fruitful to find a version of SP-WENO that has smoother weights, as is the case with the original WENO scheme [62].

- While we can construct arbitrarily high-order entropy conservative/stable finite difference schemes for the Euler equations, the SBP-type discretizations of the viscous fluxes considered in this thesis are only second-order accurate. Performing a DNS of the Navier-Stokes equations in tandem with a high-order entropy conservative flux would certainly give smaller errors, but the overall scheme would still only be second-order accurate. High-order viscous discretizations satisfying kinetic energy preservation and/or entropy stability need to be investigated.

- The energy preserving VI-EP4 scheme for the shallow water equation has been designed to preserve total energy, while the total mass and total absolute vorticity are preserved due to the conservative flux discretization. However, numerical results indicate that VI-EP4 is also capable of preserving the total enstrophy quite accurately, even though the scheme has not been designed to ensure this. Thus, the VI-EP4 scheme is well suited for DNS simulations of the viscous shallow water equations. Furthermore, it is worth looking at the possible extension of the VI-EP4 scheme to more generalized grids. This will be investigated in future work.

- A big drawback with entropy stable finite volume schemes for conservation laws on unstructured grids, is that the entropy conservative fluxes are in general only first-order accurate. Unlike finite difference schemes on Cartesian grids, there is no interpolation formula available to get high-order accurate entropy conservative finite volume fluxes. Thus, it would be fruitful to investigate methods to construct a genuinely second-order accurate flux which is entropy conservative.

- Boundary fluxes have been constructed for the initial boundary value problem for the Navier-Stokes equations, which lead to entropy stability estimates for homogeneous boundary conditions. It might be possible to construct boundary fluxes for a more general boundary condition while satisfying suitable entropy estimates.

- Although the analysis for the finite volume schemes have been presented in two-dimensions on triangular grids, it can easily be extended to the three-dimensional Navier-Stokes equations on tetrahedral grids.

A. Viscous flux symmetrization

Consider the entropy variables $\mathbf{V}$ given in (3.9). The first order spatial derivatives can be evaluated in terms in the entropy variables as

$$\partial_{x_j} u_1 = -\frac{\partial_{x_j}(V^{(2)})}{V^{(5)}} + \frac{V^{(2)}\partial_{x_j}(V^{(5)})}{(V^{(5)})^2}, \quad \partial_{x_j} u_2 = -\frac{\partial_{x_j}(V^{(3)})}{V^{(5)}} + \frac{V^{(3)}\partial_{x_j}(V^{(5)})}{(V^{(5)})^2},$$
$$\partial_{x_j} u_3 = -\frac{\partial_{x_j}(V^{(4)})}{V^{(5)}} + \frac{V^{(4)}\partial_{x_j}(V^{(5)})}{(V^{(5)})^2}, \quad \partial_{x_j} T = \frac{\partial_{x_j}(V^{(5)})}{R(V^{(5)})^2},$$

for $j = 1, 2, 3$. We define the following notations

$$\chi = \frac{4}{3}\mu, \quad \xi \ = -\frac{2}{3}\mu, \quad e_1 = \frac{V^{(2)}}{V^{(5)}}, \quad e_2 = \frac{V^{(3)}}{V^{(5)}}, \quad e_3 = \frac{V^{(4)}}{V^{(5)}}, \quad e_4 = \frac{1}{V^{(5)}}.$$

Based on a slightly rescaled version of the expressions reported in [57], the viscous flux components can be written as

$$\mathbf{g}_i = \sum_{j=1}^{3} \mathbf{K}_{ij} \partial_{x_j} \mathbf{V}, \quad i = 1, 2, 3.$$

where

$$\mathbf{K}_{11} = \mathbf{K}_{11}^{\top} = e_4 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -\chi & 0 & 0 & \chi e_1 \\ 0 & 0 & -\mu & 0 & \mu e_2 \\ 0 & 0 & 0 & -\mu & \mu e_3 \\ 0 & \chi e_1 & \mu e_2 & \mu e_3 & -\chi e_1^2 - \mu(e_2^2 + e_3^2) + \frac{\kappa}{R} e_4 \end{pmatrix},$$

$$\mathbf{K}_{22} = \mathbf{K}_{22}^{\top} = e_4 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -\mu & 0 & 0 & \mu e_1 \\ 0 & 0 & -\chi & 0 & \chi e_2 \\ 0 & 0 & 0 & -\mu & \mu e_3 \\ 0 & \mu e_1 & \chi e_2 & \mu e_3 & -\chi e_2^2 - \mu(e_1^2 + e_3^2) + \frac{\kappa}{R} e_4 \end{pmatrix},$$

$$\mathbf{K}_{33} = \mathbf{K}_{33}^{\top} = e_4 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & -\mu & 0 & 0 & \mu e_1 \\ 0 & 0 & -\mu & 0 & \mu e_2 \\ 0 & 0 & 0 & -\chi & \chi e_3 \\ 0 & \mu e_1 & \mu e_2 & \chi e_3 & -\chi e_3^2 - \mu(e_1^2 + e_2^2) + \frac{\kappa}{R} e_4 \end{pmatrix},$$

$$\mathbf{K}_{12} = \mathbf{K}_{21}^{\top} = e_4 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & -\xi & 0 & \xi e_2 \\ 0 & -\mu & 0 & 0 & \mu e_1 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & \mu e_2 & \xi e_1 & 0 & -(\mu+\xi)e_1 e_2 \end{pmatrix},$$

$$\mathbf{K}_{13} = \mathbf{K}_{31}^{\top} = e_4 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\xi & \xi e_3 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & -\mu & 0 & 0 & \mu e_1 \\ 0 & \mu e_3 & 0 & \xi e_1 & -(\mu+\xi)e_1 e_3 \end{pmatrix},$$

$$\mathbf{K}_{23} = \mathbf{K}_{32}^{\top} = e_4 \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\xi & \xi e_3 \\ 0 & 0 & -\mu & 0 & \mu e_2 \\ 0 & 0 & \mu e_3 & \xi e_2 & -(\mu+\xi)e_2 e_3 \end{pmatrix}.$$

Since $\mathbf{K}_{ij} = \mathbf{K}_{ji}^{\top}$ for $i, j = 1, 2, 3$, the matrix

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}_{11} & \mathbf{K}_{12} & \mathbf{K}_{13} \\ \mathbf{K}_{21} & \mathbf{K}_{22} & \mathbf{K}_{23} \\ \mathbf{K}_{31} & \mathbf{K}_{32} & \mathbf{K}_{33} \end{pmatrix} \in \mathbb{R}^{15\times 15},$$

is symmetric. Furthermore, $\mathbf{K}$ can be shown to be positive semi-definite [57, 31].

B. Logarithmic average

Let a be a positive quantity of interest with two states a_L and a_R. The logarithmic mean of a_L and a_R is defined as

$$\widehat{a} = \frac{a_L - a_R}{\ln(a_L) - \ln(a_R)}.$$

However, this is not numerically well-posed when a_L and a_R are nearly equal. The following stable algorithm to evaluate the logarithmic average has been given in [58].

Let $\zeta = a_L/a_R$, so that

$$\widehat{a} = \frac{a_L + a_R}{\ln(\zeta)} \left(\frac{\zeta - 1}{\zeta + 1} \right),$$

where we use the series expansion of $\ln \zeta$

$$\ln \zeta = 2 \left(\frac{1-\zeta}{1+\zeta} + \frac{1}{3} \frac{(1-\zeta)^3}{(1+\zeta)^3} + \frac{1}{5} \frac{(1-\zeta)^5}{(1+\zeta)^5} + \frac{1}{7} \frac{(1-\zeta)^7}{(1+\zeta)^7} + O(\zeta^9) \right),$$

to obtain a numerically well-formed logarithmic mean as follows

```
eps  = 1.0e-2
z    = a_L/a_R
f    = (z-1)/(z+1)
u    = f^2
if (u < eps)
   F = 1 + u/3 +  u^2/5 + u^3/7
else
   F = log(z)/2/f
logavg_a =  (a_L + a_R)/2/F
```

C. Numerical flux expression in three-dimensions

Let $\mathbf{n} = (n_1, n_2, n_3) \in \mathbb{R}^3$. We describe below the three-dimensional expressions of several key numerical fluxes for the Euler equations.

C.1 Central fluxes

The central fluxes are of the form $\mathbf{F}(\mathbf{U}_l, \mathbf{U}_r, \mathbf{n})$, where $\mathbf{U}_l$, $\mathbf{U}_r$ are the left and right states at the cell face at which the flux is being evaluated, while $\mathbf{n}$ corresponds to the normal to the face. The fluxes are evaluated at an averaged state depending on $\mathbf{U}_l$ and $\mathbf{U}_r$.

C.1.1 KEPEC

The kinetic energy and entropy conservative flux proposed in [16] is given by

$$\mathbf{F}(\mathbf{U}_l, \mathbf{U}_r, \mathbf{n}) = \begin{pmatrix} F^\rho \\ F^{m_1} \\ F^{m_2} \\ F^{m_3} \\ F^e \end{pmatrix} = \begin{pmatrix} \widehat{\rho}\overline{u}_n \\ \widetilde{p}n_1 + \overline{u_1}F^\rho \\ \widetilde{p}n_2 + \overline{u_2}F^\rho \\ \widetilde{p}n_3 + \overline{u_3}F^\rho \\ F^e \end{pmatrix}, \ F^e = \left[\frac{1}{2(\gamma-1)\widehat{\beta}} - \frac{1}{2}\overline{|\mathbf{u}|^2}\right] F^\rho + \overline{\mathbf{u}} \cdot \mathbf{F}^m,$$

where

$$\mathbf{F}^m = \left(F^{m_1},\ F^{m_2},\ F^{m_3}\right)^\top, \qquad \overline{u}_n = \mathbf{u} \cdot \mathbf{n}, \qquad \widetilde{p} = \frac{\overline{\rho}}{2\overline{\beta}},$$

and $\widehat{\rho}, \widehat{\beta}$ are the logarithmic averages of the respective quantities.

C.1.2 ROE-EC

Roe [58] defined the parameter vector

$$\mathbf{Z}^\top = \begin{pmatrix} Z_1 & Z_2 & Z_3 & Z_4 & Z_5 \end{pmatrix} = \sqrt{\frac{\rho}{p}} \begin{pmatrix} 1 & u_1 & u_2 & u_3 & p \end{pmatrix},$$

and proposed the following entropy conservative flux in terms of $\mathbf{Z}$

$$\mathbf{F}(\mathbf{U}_l, \mathbf{U}_r, \mathbf{n}) = \begin{pmatrix} F^\rho \\ F^{m_1} \\ F^{m_2} \\ F^{m_3} \\ F^e \end{pmatrix} = \begin{pmatrix} \overline{Z_n}\widehat{Z_5} \\ \frac{\overline{Z_5}}{\overline{Z_1}} n_1 + \frac{\overline{Z_2}}{\overline{Z_1}} F^\rho \\ \frac{\overline{Z_5}}{\overline{Z_1}} n_2 + \frac{\overline{Z_3}}{\overline{Z_1}} F^\rho \\ \frac{\overline{Z_5}}{\overline{Z_1}} n_3 + \frac{\overline{Z_4}}{\overline{Z_1}} F^\rho \\ F^e \end{pmatrix}, \; F^e = \frac{1}{2\overline{Z_1}} \left[\frac{(\gamma+1)}{(\gamma-1)} \frac{F^\rho}{\widehat{Z_1}} + \sum_{k=1}^{3} \overline{Z}_{k+1} F^{m_k} \right],$$

where

$$\overline{Z_n} = \overline{Z_2} n_1 + \overline{Z_3} n_2 + \overline{Z_4} n_3.$$

The ROE-EC scheme is not kinetic energy preserving.

C.1.3 KEP

Jameson [59] proposed the following simple central flux

$$\mathbf{F}(\mathbf{U}_l, \mathbf{U}_r, \mathbf{n}) = \begin{pmatrix} F^\rho \\ F^{m_1} \\ F^{m_2} \\ F^{m_3} \\ F^e \end{pmatrix} = \begin{pmatrix} \overline{\rho}\,\overline{u}_n \\ \overline{p} n_1 + \overline{u_1} F^\rho \\ \overline{p} n_2 + \overline{u_2} F^\rho \\ \overline{p} n_3 + \overline{u_3} F^\rho \\ \overline{\rho H}\,\overline{u}_n \end{pmatrix},$$

which clearly satisfies the condition (5.9) and is thus kinetic energy preserving. However, the flux is not entropy conservative.

C.2 Dissipation matrix used in entropy stable schemes

Let $\widetilde{\mathbf{n}}$ be the unit normal corresponding to $\mathbf{n}$. Then the Roe-type dissipation matrix is given by

$$\mathbf{D}(\mathbf{U}_l, \mathbf{U}_r, \mathbf{n}) = \mathbf{R}\mathbf{\Lambda}\mathbf{R}^\top$$

where

$$\mathbf{R} = \begin{pmatrix} 1 & 1 & 0 & 0 & 1 \\ u_1 - a\tilde{n}_1 & u_1 & \tilde{n}_2 & -\tilde{n}_3 & u_1 + a\tilde{n}_1 \\ u_2 - a\tilde{n}_2 & u_2 & -\tilde{n}_1 & 0 & u_2 + a\tilde{n}_2 \\ u_3 - a\tilde{n}_3 & u_3 & 0 & \tilde{n}_1 & u_3 + a\tilde{n}_3 \\ H - au_{\tilde{n}} & \frac{1}{2}|\mathbf{u}|^2 & u_1\tilde{n}_2 - u_2\tilde{n}_1 & u_3\tilde{n}_1 - u_1\tilde{n}_3 & H + au_{\tilde{n}} \end{pmatrix} \mathbf{S}^{\frac{1}{2}},$$

$$\mathbf{S} = \operatorname{diag}\left(\tfrac{\rho}{2\gamma}, \; \tfrac{(\gamma-1)\rho}{\gamma}, \; p, \; p, \; \tfrac{\rho}{2\gamma} \right),$$

$$\mathbf{\Lambda} = \mathbf{\Lambda}^{Roe} = |\mathbf{n}| \operatorname{diag}\left(|u_{\tilde{n}} - a|, \; |u_{\tilde{n}}|, \; |u_{\tilde{n}}|, \; |u_{\tilde{n}}|, \; |u_{\tilde{n}} + a| \right).$$

In the above expressions, $\mathbf{S}$ is the scaling matrix for the eigenvectors and $u_{\tilde{n}} = \mathbf{u} \cdot \tilde{\mathbf{n}}$. The following average states are used to evaluate the above matrices:

$$\mathbf{u} = \overline{\mathbf{u}}, \qquad \rho = \widehat{\rho}, \qquad p = \frac{\overline{\rho}}{2\overline{\beta}}, \qquad a = \sqrt{\frac{\gamma}{2\widehat{\beta}}}, \qquad H = \frac{1}{\gamma - 1} a^2,$$

which ensures that the KEPEC along with the Roe-type dissipation is able to resolve stationary contact discontinuities exactly [16].

One can also construct the Rusanov-type dissipation matrix by choosing

$$\mathbf{\Lambda} = \mathbf{\Lambda}^{Rus} = |\mathbf{n}|(|u_{\widetilde{n}}| + a)\mathbf{I},$$

where $\mathbf{I}$ is the identity matrix.

Remark C.2.1. *A finite difference flux for the Euler equations formed using one of the central fluxes described in Section C.1, and augmented with dissipation given in Section C.2, has the expression*

$$\begin{aligned}
\mathbf{F}^x_{i+\frac{1}{2},j,k} &= \mathbf{F}(\mathbf{U}_{i,j,k}, \mathbf{U}_{i+1,j,k}, \mathbf{e}_1) - \frac{1}{2}\mathbf{D}(\mathbf{U}_{i,j,k}, \mathbf{U}_{i+1,j,k}, \mathbf{e}_1)\Delta\mathbf{V}_{i+\frac{1}{2},j,k},\\
\mathbf{F}^x_{i-\frac{1}{2},j,k} &= \mathbf{F}(\mathbf{U}_{i-1,j,k}, \mathbf{U}_{i,j,k}, \mathbf{e}_1) - \frac{1}{2}\mathbf{D}(\mathbf{U}_{i-1,j,k}, \mathbf{U}_{i,j,k}, \mathbf{e}_1)\Delta\mathbf{V}_{i-\frac{1}{2},j,k},
\end{aligned}$$

where $\mathbf{e}_1 = (1,0,0)^\top$ *is the unit vector along the positive x-direction. Note that we used* $\mathbf{n} = \mathbf{e}_1$ *in* $\mathbf{F}^x_{i-\frac{1}{2},j,k}$ *instead of* $\mathbf{n} = -\mathbf{e}_1$ *since the direction of the face normals is already accounted for in the formulation of the finite difference scheme. The remaining fluxes are obtained in a similar manner as follows:*

$$\begin{aligned}
\mathbf{F}^y_{i,j+\frac{1}{2},k} &= \mathbf{F}(\mathbf{U}_{i,j,k}, \mathbf{U}_{i,j+1,k}, \mathbf{e}_2) - \frac{1}{2}\mathbf{D}(\mathbf{U}_{i,j,k}, \mathbf{U}_{i,j+1,k}, \mathbf{e}_2)\Delta\mathbf{V}_{i,j+\frac{1}{2},k},\\
\mathbf{F}^y_{i,j-\frac{1}{2},k} &= \mathbf{F}(\mathbf{U}_{i,j-1,k}, \mathbf{U}_{i,j,k}, \mathbf{e}_2) - \frac{1}{2}\mathbf{D}(\mathbf{U}_{i,j-1,k}, \mathbf{U}_{i,j,k}, \mathbf{e}_2)\Delta\mathbf{V}_{i,j-\frac{1}{2},k},\\
\mathbf{F}^z_{i,j,k+\frac{1}{2}} &= \mathbf{F}(\mathbf{U}_{i,j,k}, \mathbf{U}_{i,j,k+1}, \mathbf{e}_3) - \frac{1}{2}\mathbf{D}(\mathbf{U}_{i,j,k}, \mathbf{U}_{i,j,k+1}, \mathbf{e}_3)\Delta\mathbf{V}_{i,j,k+\frac{1}{2}},\\
\mathbf{F}^z_{i,j,k-\frac{1}{2}} &= \mathbf{F}(\mathbf{U}_{i,j,k-1}, \mathbf{U}_{i,j,k}, \mathbf{e}_3) - \frac{1}{2}\mathbf{D}(\mathbf{U}_{i,j,k-1}, \mathbf{U}_{i,j,k}, \mathbf{e}_3)\Delta\mathbf{V}_{i,j,k-\frac{1}{2}},
\end{aligned}$$

where $\mathbf{e}_2 = (0,1,0)^\top$*,* $\mathbf{e}_3 = (0,0,1)^\top$*. To obtain high-order fluxes, the central flux is used in tandem with the interpolation formula* (5.14) *in a dimension by dimension manner, while high-order scaled entropy variable jumps replace the first order jumps.*

Remark C.2.2. *The expression for the central flux and dissipation matrices are also used in the formulation of finite volume schemes discussed in Chapter 9. For finite volume schemes the vector* $\mathbf{n}$ *corresponds the the outward normal to the control volume face, and has a magnitude equal to the length of the face.*

D. GMRES method

The following GMRES algorithm has been taken from [67]. Consider the following system of linear equations

$$\mathbf{Ax} = \mathbf{b}.$$

We assume that the matrix $\mathbf{A} \in \mathbb{R}^{m\times m}$ is invertible and $|\mathbf{b}| = 1$. The n-th Krylov subspace for this problem is given as

$$K_n = K_n(\mathbf{A}, \mathbf{b}) = \operatorname{span}\{\mathbf{b}, \mathbf{Ab}, \mathbf{A}^2\mathbf{b}, \ldots, \mathbf{A}^{n-1}\mathbf{b}\}.$$

GMRES approximates the exact solution of $\mathbf{Ax} = \mathbf{b}$, by the vector $\mathbf{x}_n \in K_n$ that minimizes the norm of the residual

$$|\mathbf{r}_n| = |\mathbf{Ax}_n - \mathbf{b}|.$$

We find an orthonormal basis $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_n$ of K_n using the Arnoldi iteration

1. Define $\mathbf{r}_0 = \mathbf{b}$ and $\mathbf{v}_1 = \mathbf{r}_0/|\mathbf{r}_0|$, with the initial guess $\mathbf{x}_0 = 0$.
2. For $i = 1, 2, ..., n-1$

$$\mathbf{v}_{i+1} = \frac{\mathbf{Av}_i - \sum_{j=1}^{i} \langle \mathbf{Av}_i, \mathbf{v}_j\rangle \mathbf{v}_j}{|\mathbf{Av}_i - \sum_{j=1}^{i} \langle \mathbf{Av}_i, \mathbf{v}_j\rangle \mathbf{v}_j|}.$$

Thus, the vector $\mathbf{x}_n$ can be written as $\mathbf{x}_n = \mathbf{W}_n\mathbf{y}_n$ for some $\mathbf{y}_n \in \mathbb{R}^n$, where $\mathbf{W}_n \in \mathbb{R}^{m\times n}$ is the matrix with the basis vectors of K_n as the columns. If denominator in the second step of Algorithm arnoldi is zero for some index i, then the solution to $\mathbf{Ax} = \mathbf{b}$ is in K_i (see Lemma 3.4.1 in [67]).

The Arnoldi process also produces an $(n+1)\times n$ upper Hessenberg matrix $\mathbf{H}_n$ such that

$$\mathbf{AW}_n = \mathbf{W}_{n+1}\mathbf{H}_n.$$

Setting $\beta = |\mathbf{r}_0|$ and taking $\mathbf{e}_1 = (1, 0, 0, ..., 0)^\top \in \mathbb{R}^{n+1}$, we get

$$|\mathbf{r}_n| = |\mathbf{b} - \mathbf{Ax}_n|_2 = |\mathbf{W}_{n+1}(\beta\mathbf{e}_1 - \mathbf{H}_n\mathbf{y}_n)| = |\beta\mathbf{e}_1 - \mathbf{H}_n\mathbf{y}_n|$$

as $\mathbf{W}_{n+1}$ is orthogonal. Hence, $\mathbf{x}_n$ can be determined by minimizing the Euclidean norm of the residual $\mathbf{r}_n$, which is a linear least squares problem of size n.

The above process is repeated till the relative norm of the residue is below a threshold

$$\frac{|\mathbf{b} - \mathbf{Ax}_n|}{|\mathbf{b}|} < \epsilon.$$

E. Cell averages of entropy variables and positivity

For the two-dimensional Navier-Stokes equations, we wish to express the primitive variables $\mathbf{P} = (\rho, u_1, u_2, p)^\top$ in terms of the entropy variables $\mathbf{V} = (V^{(1)}, V^{(2)}, V^{(3)}, V^{(4)})^\top$. Clearly

$$u_1 = -\frac{V^{(2)}}{V^{(4)}}, \qquad u_2 = -\frac{V^{(3)}}{V^{(4)}}, \qquad \theta = \frac{p}{\rho R} = -\frac{1}{RV^{(4)}}, \qquad \beta = -\frac{V^{(4)}}{2}$$

Furthermore,

$$\begin{aligned} \ln\left(\frac{p}{\rho^\gamma}\right) = s &= -\left(\beta|\mathbf{u}|^2 + V^{(1)}\right)(\gamma - 1) + \gamma \\ &= -\left(-\frac{(V^{(2)})^2 + (V^{(3)})^2}{2V^{(4)}} + V^{(1)}\right)(\gamma - 1) + \gamma, \end{aligned}$$

or

$$\frac{\rho^\gamma}{p} = \exp\left[\left(-\frac{(V^{(2)})^2 + (V^{(3)})^2}{2V^{(4)}} + V^{(1)}\right)(\gamma - 1) - \gamma\right] = h(\mathbf{V}).$$

The above relations give us

$$\rho = \left(\frac{h(\mathbf{V})}{-V^{(4)}}\right)^{\frac{1}{\gamma-1}}, \qquad p = (h(\mathbf{V}))^{\frac{1}{\gamma-1}}\left(\frac{-1}{V^{(4)}}\right)^{\frac{\gamma}{\gamma-1}}.$$

Note that $V^{(4)} < 0$ ensures that ρ and p are well-defined and positive, since $h(\mathbf{V}) > 0$ and $\gamma > 1$. Thus, any average state $\overline{\mathbf{V}}$ of the entropy variables for which $\overline{V}^{(4)} < 0$ will ensure the positivity of the corresponding density and pressure. In particular, this holds true for the average states $\mathbf{V}^T$ and $\mathbf{V}^{T_e}$ defined in (10.13) and (10.14) respectively.

Bibliography

[1] W. K. Anderson. A Grid Generation and Flow Solution Method for the Euler Equations on Unstructured Grids. *Journal of Computational Physics*, 110(1):23 – 38, 1994.

[2] F. Angrand and F. C. Lafon. Flux Formulation using a Fully 2D Approximate Roe Riemann Solver. In Andrea Donato and Francesco Oliveri, editors, *Nonlinear Hyperbolic Problems: Theoretical, Applied, and Computational Aspects*, volume 43 of *Notes on Numerical Fluid Mechanics (NNFM)*, pages 15–22. Vieweg+Teubner Verlag, 1993.

[3] Akio Arakawa and Vivian R. Lamb. A potential enstrophy and energy conserving scheme for the shallow water equations. *Monthly Weather Review*, 109(1):18–36, 1981.

[4] P. Arminjon, A. Dervieux, L. Fezoui, H. Steve, and B. Stoufflet. Non-Oscillatory Schemes for Multidimensional Euler Calculations with Unstructured Grids. In J. Ballmann and R. Jeltsch, editors, *Nonlinear Hyperbolic Equations - Theory, Computation Methods, and Applications*, volume 24 of *Notes on Numerical Fluid Mechanics*, pages 1–10. Vieweg+Teubner Verlag, 1989.

[5] K. Atkinson. *An Introduction to Numerical Analysis, 2nd Edition.* John Wiley and Sons, 1989.

[6] T. Barth. On the role of involutions in the discontinuous galerkin discretization of maxwell and magnetohydrodynamic systems. In D. N. Arnold, P. B. Bochev, R. B. Lehoucq, R. A. Nicolaides, and M. Shashkov, editors, *Compatible Spatial Discretizations*, volume 142 of *The IMA Volumes in Mathematics and its Applications*, pages 69–88. Springer New York, 2006.

[7] T. Barth and D. Jespersen. *The design and application of upwind schemes on unstructured meshes.* American Institute of Aeronautics and Astronautics, 1989.

[8] T. J. Barth. Numerical methods for gasdynamic systems on unstructured meshes. In *An introduction to recent developments in theory and numerics for conservation laws (Freiburg/Littenweiler, 1997)*, volume 5 of *Lect. Notes Comput. Sci. Eng.*, pages 195–285. Springer, Berlin, 1999.

[9] Michele Benzi, Maxim A. Olshanskii, Leo G. Rebholz, and Zhen Wang. Assessment of a vorticity based solver for the Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 247-248:216 – 225, 2012.

[10] S. Bianchini and A. Bressan. Vanishing viscosity solutions of nonlinear hyperbolic systems. *Ann. of Math. (2)*, 161(1):223–342, 2005.

[11] V. Billey, J. Périaux, P. Perrier, and B. Stoufflet. 2-D and 3-D Euler computations with finite element methods in aerodynamics. In *Nonlinear hyperbolic problems (St. Etienne, 1986)*, volume 1270 of *Lecture Notes in Math.*, pages 64–81. Springer, Berlin, 1987.

[12] J. Blazek. *Computational Fluid Dynamics: Principles and Applications*. Elsevier Science, Oxford, second edition edition, 2005.

[13] A. Bressan, G. Crasta, and B. Piccoli. Well-posedness of the Cauchy problem for $n \times n$ systems of conservation laws. *Mem. Amer. Math. Soc.*, 146(694):viii+134, 2000.

[14] O. R. Burggraf. Analytical and numerical studies of the structure of steady separated flows. *Journal of Fluid Mechanics*, 24:113–151, 1966.

[15] Mark H. Carpenter, Travis C. Fisher, Eric J. Nielsen, and Steven H. Frankel. Entropy Stable Spectral Collocation Schemes for the Navier–Stokes Equations: Discontinuous Interfaces. *SIAM J. Sci. Comput.*, 36(5):B835–B867, 2014.

[16] P. Chandrashekar. Kinetic energy preserving and entropy stable finite volume schemes for compressible Euler and Navier-Stokes equations. *Commun. Comput. Phys.*, 14(5):1252–1286, 2013.

[17] Praveen Chandrashekar. Finite volume discretization of heat equation and compressible Navier–Stokes equations with weak Dirichlet boundary condition on triangular grids. *International Journal of Advances in Engineering Sciences and Applied Mathematics*, 8(3):174–193, 2016.

[18] X. Cheng and Y. Nie. A third-order entropy stable scheme for hyperbolic conservation laws. *Journal of Hyperbolic Differential Equations*, 13(01):129–145, 2016.

[19] E. Chiodaroli, C. De Lellis, and O. Kreml. Global ill-posedness of the isentropic system of gas dynamics. *Communications on Pure and Applied Mathematics*, 68(7):1157–1190, 2015.

[20] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework. *Math. Comp.*, 52(186):411–435, 1989.

[21] P. Colella. A direct Eulerian MUSCL scheme for gas dynamics. *SIAM J. Sci. Statist. Comput.*, 6(1):104–117, 1985.

[22] Georges-Henri Cottet, Bertrand Michaux, Sepand Ossia, and Geoffroy VanderLinden. A comparison of spectral and vortex methods in three-dimensional incompressible flows. *Journal of Computational Physics*, 175(2):702 – 712, 2002.

[23] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM J. Res. Dev.*, 11(2):215–234, March 1967.

[24] Michael G. Crandall and Andrew Majda. Monotone difference approximations for scalar conservation laws. *Math. Comp.*, 34(149):1–21, 1980.

[25] C. M. Dafermos. *Hyperbolic conservation laws in continuum physics*, volume 325 of *Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences].* Springer-Verlag, Berlin, third edition, 2010.

[26] H. Deconinck, P. L. Roe, and R. Struijs. A multidimensional generalization of Roe's flux difference splitter for the Euler equations. *Comput. & Fluids*, 22(2-3):215–222, 1993.

[27] J.-A. Désidéri and A. Dervieux. Compressible flow solvers using unstructured grids. In *Computational fluid dynamics, Vol. 1, 2*, volume 88 of *von Karman Inst. Fluid Dynam. Lecture Ser.*, page 115. von Karman Inst. Fluid Dynamics, Rhode-St-Genèse, 1988.

[28] R. J. DiPerna. Measure-valued solutions to conservation laws. *Archive for Rational Mechanics and Analysis*, 88(3):223–270, 1985.

[29] François Dubois and Philippe LeFloch. Boundary conditions for nonlinear hyperbolic systems of conservation laws. In *Nonlinear hyperbolic equations – theory, computation methods, and applications (Aachen, 1988)*, volume 24 of *Notes Numer. Fluid Mech.*, pages 96–104. Vieweg, Braunschweig, 1989.

[30] L. J. Durlofsky, B. Engquist, and S. Osher. Triangle based adaptive stencils for the solution of hyperbolic conservation laws. *Journal of Computational Physics*, 98(1):64 – 73, 1992.

[31] P. Dutt. Stable Boundary Conditions and Difference Schemes for Navier-Stokes Equations. *SIAM Journal on Numerical Analysis*, 25(2):245–267, 1988.

[32] C. Eldred and D. Randall. Total energy and potential enstrophy conserving schemes for the shallow water equations using Hamiltonian methods: Derivation and Properties (Part 1). *ArXiv e-prints*, September 2016.

[33] Lawrence C. Evans. *Partial differential equations*, volume 19 of *Graduate Studies in Mathematics.* American Mathematical Society, Providence, RI, second edition, 2010.

[34] U. S. Fjordholm, R. Käppeli, S. Mishra, and E. Tadmor. Construction of approximate entropy measure-valued solutions for hyperbolic systems of conservation laws. *Foundations of Computational Mathematics*, pages 1–65, 2015.

[35] U. S. Fjordholm, S. Mishra, and E. Tadmor. Arbitrarily High-order Accurate Entropy Stable Essentially Nonoscillatory Schemes for Systems of Conservation Laws. *SIAM Journal on Numerical Analysis*, 50(2):544–573, 2012.

[36] U. S. Fjordholm, S. Mishra, and E. Tadmor. ENO Reconstruction and ENO Interpolation Are Stable. *Foundations of Computational Mathematics*, 13:139–159, 2013.

[37] Ulrik S. Fjordholm, Siddhartha Mishra, and Eitan Tadmor. Well-balanced and energy stable schemes for the shallow water equations with discontinuous topography. *Journal of Computational Physics*, 230(14):5587 – 5609, 2011.

[38] Ulrik S. Fjordholm and Deep Ray. A sign preserving WENO reconstruction method. *Journal of Scientific Computing*, 68(1):42–63, 2016.

[39] Uriel Frisch. *Turbulence*. Cambridge University Press, Cambridge, 1995. The legacy of A. N. Kolmogorov.

[40] Thomas B. Gatski. Review of incompressible fluid flow computations using the vorticity-velocity formulation. *Applied Numerical Mathematics*, 7(3):227 – 239, 1991.

[41] Christophe Geuzaine and Jean-Francois Remacle. Gmsh: A 3-D finite element mesh generator with built-in pre- and post-processing facilities. *International Journal for Numerical Methods in Engineering*, 79(11):1309–1331, 2009.

[42] U. Ghia, K. N. Ghia, and C. T. Shin. High-Re solutions for incompressible flow using the Navier-Stokes equations and a multigrid method . *Journal of Computational Physics*, 48(3):387 – 411, 1982.

[43] J. Glimm. Solutions in the large for nonlinear hyperbolic systems of equations. *Comm. Pure Appl. Math.*, 18:697–715, 1965.

[44] E. Godlewski and P.A. Raviart. *Hyperbolic systems of conservation laws*. Mathématiques & applications. Ellipses, 1991.

[45] Edwige Godlewski and Pierre-Arnaud Raviart. *Numerical approximation of hyperbolic systems of conservation laws*, volume 118 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 1996.

[46] S. K. Godunov. A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Mat. Sb. (N.S.)*, 47 (89):271–306, 1959.

[47] S. K. Godunov. An interesting class of quasilinear systems. *Dokl. Akad. Nauk. SSSR*, 139:521—523, 1961.

[48] S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Rev.*, 43(1):89–112 (electronic), 2001.

[49] G. Guevremont, W. G. Habashi, and M. M. Hafez. Finite element solution of the Navier–Stokes equations by a velocity–vorticity method. *International Journal for Numerical Methods in Fluids*, 11(5):661–675, 1990.

[50] B Gustafsson and A Sundström. Incompletely Parabolic Problems in Fluid Dynamics. *SIAM J. Appl. Math.*, 35(2):343–357, 1978.

[51] A. Harten. High resolution schemes for hyperbolic conservation laws. *J. Comput. Phys.*, 49(3):357 – 393, 1983.

[52] A. Harten. On the symmetric form of systems of conservation laws with entropy. *J. Comput. Phys.*, 49(1):151–164, 1983.

[53] A. Harten and J. M. Hyman. Self adjusting grid methods for one-dimensional hyperbolic conservation laws. *J. Comput. Phys.*, 50(2):235 – 269, 1983.

[54] Ami Harten, Bjorn Engquist, Stanley Osher, and Sukumar R Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71(2):231 – 303, 1987.

[55] J S Hesthaven and D Gottlieb. A Stable Penalty Method for the Compressible Navier-Stokes Equations: I. Open Boundary Conditions. *SIAM J. Sci. Comput.*, 17(3):579–612, 1996.

[56] A. Hiltebrand and S. Mishra. Entropy stable shock capturing space–time discontinuous Galerkin schemes for systems of conservation laws. *Numerische Mathematik*, 126(1):103–151, 2014.

[57] T. J. R. Hughes, L. P. Franca, and M. Mallet. A new finite element formulation for computational fluid dynamics. I. Symmetric forms of the compressible Euler and Navier-Stokes equations and the second law of thermodynamics. *Comput. Methods Appl. Mech. Engrg.*, 54(2):223–234, 1986.

[58] F. Ismail and P. L. Roe. Affordable, entropy-consistent Euler flux functions II: Entropy production at shocks. *Journal of Computational Physics*, 228(15):5410 – 5436, 2009.

[59] A. Jameson. Formulation of kinetic energy preserving conservative schemes for gas dynamics and direct numerical simulation of one-dimensional viscous compressible flow in a shock tube using entropy and kinetic energy preserving schemes. *J. Sci. Comput.*, 34(2):188–208, 2008.

[60] A. Jameson and D. Mavriplis. Finite volume solution of the two-dimensional Euler equations on a regular triangular mesh. *AIAA Journal*, 24:611–618, April 1986.

[61] Antony Jameson. The construction of discretely conservative finite volume schemes that also globally conserve energy or entropy. *Journal of Scientific Computing*, 34(2):152–187, 2008.

[62] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126(1):202 – 228, 1996.

[63] George Karypis and Vipin Kumar. METIS – Unstructured Graph Partitioning and Sparse Matrix Ordering System, Version 2.0. Technical report, 1995.

[64] George Karypis and Vipin Kumar. Multilevelk-way partitioning scheme for irregular graphs. *J. Parallel Distrib. Comput.*, 48(1):96–129, January 1998.

[65] A. Katz and V. Sankaran. An efficient correction method to obtain a formally third-order accurate flow solver for node-centered unstructured grids. *J. Sci. Comput.*, 51(2):375–393, 2012.

[66] M. Kawaguti. Numerical Solution of the Navier-Stokes Equations for the Flow in a Two-Dimensional Cavity. *Journal of the Physical Society of Japan*, 16:2307, November 1961.

[67] C. Kelley. *Iterative Methods for Linear and Nonlinear Equations.* Society for Industrial and Applied Mathematics, 1995.

[68] D. Kröner, S. Noelle, and M. Rokyta. Convergence of higher order upwind finite volume schemes on unstructured grids for scalar conservation laws in several space dimensions. *Numerische Mathematik*, 71(4):527–560, 1995.

[69] D. Kröner and M. Rokyta. Convergence of Upwind Finite Volume Schemes for Scalar Conservation Laws in Two Dimensions. *SIAM Journal on Numerical Analysis*, 31(2):324–343, 1994.

[70] S. N. Kružkov. First order quasilinear equations in several independent variables. *Math. USSR Sb.*, 10:217–243, 1970.

[71] P. G. Lefloch, J. M. Mercier, and C. Rohde. Fully discrete, entropy conservative schemes of arbitrary order. *SIAM J. Numer. Anal.*, 40(5):1968–1992 (electronic), 2002.

[72] R. J. LeVeque. *Finite Volume Methods for Hyperbolic Problems.* Cambridge Texts in Applied Mathematics. Cambridge University Press, 2002.

[73] X.-D. Liu, S. Osher, and T. Chan. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115(1):200 – 212, 1994.

[74] A. Madrane, U. S. Fjordholm, S. Mishra, and E. Tadmor. Entropy conservative and entropy stable finite volume schemes for multi-dimensional conservation laws on unstructured meshes. In *Proceedings of European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS-2012)*, 2012.

[75] Andrew J. Majda and Andrea L. Bertozzi. *Vorticity and incompressible flow*, volume 27 of *Cambridge Texts in Applied Mathematics.* Cambridge University Press, Cambridge, 2002.

[76] D. J. Mavriplis. Adaptive Mesh Generation for Viscous Flows Using Delaunay Triangulation. *J. Comput. Phys.*, 90(2):271–291, September 1990.

[77] M. S. Mock. Systems of conservation laws of mixed type. *J. Differential Equations*, 37(1):70–88, 1980.

[78] Yohei Morinishi. Skew-symmetric form of convective terms and fully conservative finite difference schemes for variable density low-Mach number flows. *Journal of Computational Physics*, 229(2):276 – 300, 2010.

[79] François Murat. Compacité par compensation. *Annali della Scuola Normale Superiore di Pisa - Classe di Scienze*, 5(3):489–507, 1978.

[80] J Nordström. The use of characteristic boundary conditions for the Navier-Stokes equations. *Comput. Fluids*, 24(5):609–623, 1995.

[81] Jan Nordström and Magnus Svärd. Well-posed boundary conditions for the Navier-Stokes equations. *SIAM J. Numer. Anal.*, 43(3):1231–1255 (electronic), 2005.

[82] Maxim Olshanskii and Leo G. Rebholz. Note on helicity balance of the Galerkin method for the 3D Navier-Stokes equations . *Computer Methods in Applied Mechanics and Engineering*, 199(17-20):1032 – 1035, 2010.

[83] S. Osher. Riemann solvers, the entropy condition, and difference approximations. *SIAM J. Numer. Anal.*, 21(2):217–235, 1984.

[84] S. Osher and S. Chakravarthy. High resolution schemes and the entropy condition. *SIAM J. Numer. Anal.*, 21(5):955–984, 1984.

[85] S. Osher and E. Tadmor. On the convergence of difference approximations to scalar conservation laws. *Math. Comp.*, 50(181):19–51, 1988.

[86] A. Palha and M. Gerritsma. A mass, energy, enstrophy and vorticity conserving (MEEVC) mimetic spectral element discretization for the 2D incompressible Navier-Stokes equations. *Journal of Computational Physics*, 328:200–220, January 2017.

[87] M. Parsani, M. H. Carpenter, and E. J. Nielsen. Entropy stable wall boundary conditions for the three-dimensional compressible Navier-Stokes equations . *Journal of Computational Physics*, 292:88 – 113, 2015.

[88] B. Perthame and Y. Qiu. A Variant of Van Leer's Method for Multidimensional Systems of Conservation Laws. *J. Comput. Phys.*, 112(2):370–381, June 1994.

[89] Sergio Pirozzoli. Numerical methods for high-speed flows. In *Annual review of fluid mechanics. Volume 43, 2011*, volume 43 of *Annu. Rev. Fluid Mech.*, pages 163–194. Annual Reviews, Palo Alto, CA, 2011.

[90] 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*, 19:1111–1140, 5 2016.

[91] T. D. Ringler, J. Thuburn, J. B. Klemp, and W. C. Skamarock. A unified approach to energy conservation and potential vorticity dynamics for arbitrarily-structured C-grids. *J. Comput. Phys.*, 229(9):3065–3090, May 2010.

[92] Todd D. Ringler and David A. Randall. A potential enstrophy and energy conserving numerical scheme for solution of the shallow-water equations on a geodesic grid. *Monthly Weather Review*, 130(5):1397–1410, 2002.

[93] J.-Ch. Robinet, J. Gressier, G. Casalis, and J.-M. Moschetta. Shock wave instability and the carbuncle phenomenon: same intrinsic origin? *J. Fluid Mech.*, 417:237–263, 2000.

[94] P. Roe, H. Nishikawa, F. Ismail, and L. Scalabrin. On Carbuncles and Other Excrescences. In *17th AIAA Computational Fluid Dynamics Conference*, AIAA Paper 2005-4872, Toronto, 2005.

[95] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *J. Comput. Phys.*, 43(2):357–372, 1981.

[96] P. L. Roe. Sonic Flux Formulae. *SIAM J. Sci. Stat. Comput.*, 13(2):611–630, March 1992.

[97] P. L. Roe and J. Pike. Efficient Construction and Utilisation of Approximate Riemann Solutions. In *Proc. Of the Sixth Int'L. Symposium on Computing Methods in Applied Sciences and Engineering, VI*, pages 499–518, Amsterdam, The Netherlands, The Netherlands, 1985. North-Holland Publishing Co.

[98] A.M. Rogerson and E. Meiburg. A numerical study of the convergence properties of ENO schemes. *Journal of Scientific Computing*, 5(2):151–167, 1990.

[99] P. Rostand and B. Stoufflet. TVD schemes to compute compressible viscous flows on unstructured meshes. In *Nonlinear hyperbolic equations—theory, computation methods, and applications (Aachen, 1988)*, volume 24 of *Notes Numer. Fluid Mech.*, pages 510–520. Vieweg, Braunschweig, 1989.

[100] Rick Salmon. Poisson-bracket approach to the construction of energy- and potential-enstrophy-conserving algorithms for the shallow-water equations. *Journal of the Atmospheric Sciences*, 61(16):2016–2036, 2004.

[101] Rick Salmon. A general method for conserving energy and potential enstrophy in shallow-water models. *Journal of the Atmospheric Sciences*, 64(2):515–531, 2007.

[102] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. Benchmark computations of laminar flow around a cylinder. In ErnstHeinrich Hirschel, editor, *Flow Simulation with High-Performance Computers II*, volume 48 of *Notes on Numerical Fluid Mechanics (NNFM)*, pages 547–566. Vieweg+Teubner Verlag, 1996.

[103] F. Shakib, T. J. R. Hughes, and Z. Johan. A new finite element formulation for computational fluid dynamics: X. The compressible Euler and Navier-Stokes equations. *Computer Methods in Applied Mechanics and Engineering*, 89(1):141 – 219, 1991.

[104] Theodore G. Shepherd. Symmetries, conservation laws, and Hamiltonian structure in geophysical fluid dynamics. volume 32 of *Advances in Geophysics*, pages 287 – 338. Elsevier, 1990.

[105] Mohammad Shoeybi, Magnus Svärd, Frank E. Ham, and Parviz Moin. An adaptive implicit-explicit scheme for the DNS and LES of compressible flows on unstructured grids . *Journal of Computational Physics*, 229(17):5944 – 5965, 2010.

[106] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In Alfio Quarteroni, editor, *Advanced*

Numerical Approximation of Nonlinear Hyperbolic Equations, volume 1697 of *Lecture Notes in Mathematics*, pages 325–432. Springer Berlin Heidelberg, 1998.

[107] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83(1):32 – 78, 1989.

[108] Chi-Wang Shu. Numerical experiments on the accuracy of ENO and modified ENO schemes. *Journal of Scientific Computing*, 5(2):127–149, 1990.

[109] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77(2):439 – 471, 1988.

[110] G. A. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Computational Phys.*, 27(1):1–31, 1978.

[111] B. Stoufflet. Implicit finite element methods for the Euler equations. In *Numerical methods for the Euler equations of fluid dynamics (Rocquencourt, 1983)*, pages 409–434. SIAM, Philadelphia, PA, 1985.

[112] Pramod K. Subbareddy and Graham V. Candler. A fully discrete, kinetic energy consistent finite-volume scheme for compressible flows. *J. Comput. Phys.*, 228(5):1347–1364, March 2009.

[113] Magnus Svärd and Siddhartha Mishra. Entropy stable schemes for initial-boundary-value conservation laws. *Z. Angew. Math. Phys.*, 63(6):985–1003, 2012.

[114] R.C. Swanson and S. Langer. *Comparison of Naca 0012 Laminar Flow Solutions: Structured and Unstructured Grid Methods Nasa/Tm2016219003*. NASA technical memorandum. 2017.

[115] E. Tadmor. Numerical Viscosity and the Entropy Condition for Conservative Difference Schemes. *Mathematics of Computation*, 43(168):pp. 369–381, 1984.

[116] E. Tadmor. Entropy stability theory for difference approximations of nonlinear conservation laws and related time-dependent problems. *Acta Numer.*, 12:451–512, 2003.

[117] Eitan Tadmor. The numerical viscosity of entropy stable schemes for systems of conservation laws. I. *Math. Comp.*, 49(179):91–103, 1987.

[118] Eitan Tadmor and Weigang Zhong. *Energy-Preserving and Stable Approximations for the Two-Dimensional Shallow Water Equations*, pages 67–94. Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.

[119] K. Takano and M. G. Wurtele. A fourth order energy and potential enstrophy conserving difference scheme. Technical report, June 1982.

[120] L. Tartar. Compensated compactness and applications to partial differential equations. In *Nonlinear analysis and mechanics: Heriot-Watt Symposium, Vol. IV*, volume 39 of *Res. Notes in Math.*, pages 136–212. Pitman, Boston, Mass.-London, 1979.

[121] J. Thuburn and C. J. Cotter. A framework for mimetic discretization of the rotating shallow-water equations on arbitrary polygonal grids. *SIAM Journal on Scientific Computing*, 34(3):B203–B225, 2012.

[122] J. Thuburn, T.D. Ringler, W.C. Skamarock, and J.B. Klemp. Numerical representation of geostrophic modes on arbitrarily structured C-grids. *Journal of Computational Physics*, 228(22):8321 – 8335, 2009.

[123] E.F. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics: A Practical Introduction*. Springer Berlin Heidelberg, 2009.

[124] Michael D. Toy and Ramachandran D. Nair. A potential enstrophy and energy conserving scheme for the shallow water equations extended to generalized curvilinear coordinates. *Monthly Weather Review (to appear)*, 2016.

[125] S. Mishra U. S. Fjordholm and E. Tadmor. Energy preserving and energy stable schemes for the shallow water equations. In A. Pinkus F. Cucker and M. Todd, editors, *Foundations of Computational Mathematics: Proc. FoCM held in Hong Kong 2008*, London Math. Soc. Lecture Notes, Ser. 363, pages 93–139. 2009.

[126] G. D. van Albada, B. van Leer, and W. W. Roberts Jr. A Comparative Study of Computational Methods in Cosmic Gas Dynamics. In M. Y. Hussaini, B. van Leer, and J. Van Rosendale, editors, *Upwind and High-Resolution Schemes*, pages 95–103. Springer Berlin Heidelberg, 1997.

[127] B. van Leer. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *Journal of Computational Physics*, 32(1):101 – 136, 1979.

[128] Wim M. van Rees, Anthony Leonard, D.I. Pullin, and Petros Koumoutsakos. A comparison of vortex and pseudo-spectral methods for the simulation of periodic vortical flows at high Reynolds numbers. *Journal of Computational Physics*, 230(8):2794 – 2805, 2011.

[129] V. Venkatakrishnan. Viscous computations using a direct solver. *Computers & Fluids*, 18(2):191 – 204, 1990.

[130] V. Venkatakrishnan. *On the accuracy of limiters and convergence to steady state solutions*. American Institute of Aeronautics and Astronautics, 1993.

[131] V. Venkatakrishnan. Convergence to Steady State Solutions of the Euler Equations on Unstructured Grids with Limiters . *Journal of Computational Physics*, 118(1):120 – 130, 1995.

[132] G. Vijayasundaram. Transonic flow simulations using an upstream centered scheme of Godunov in finite elements. *J. Comput. Phys.*, 63(2):416–433, 1986.

[133] D. L. Whitaker, B. Grossman, and R. Löhner. Two-Dimensional Euler Computations on a Triangular Mesh Using an Upwind, Finite-Volume Scheme. In *27th Aerospace Sciences Meeting*, 1989.

[134] David L. Williamson, John B. Drake, James J. Hack, Rüdiger Jakob, and Paul N. Swarztrauber. A standard test set for numerical approximations to the shallow water equations in spherical geometry. *Journal of Computational Physics*, 102(1):211 – 224, 1992.

[135] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *J. Comput. Phys.*, 54(1):115–173, 1984.

[136] H.C. Yee, N.D. Sandham, and M.J. Djomehri. Low-dissipative high-order shock-capturing methods using characteristic-based filters. *Journal of Computational Physics*, 150(1):199 – 238, 1999.