

**RUNGE-KUTTA DISCONTINUOUS GALERKIN METHOD BASED ON
FLUX VECTOR SPLITTING WITH CONSTRAINED
OPTIMIZATION-BASED TVB(D)-MINMOD LIMITER FOR SOLVING
HYPERBOLIC CONSERVATION LAWS**

ZHENGRONG XIE, XIAOFENG CAI, AND HAIBIAO ZHENG

Date: November 26, 2024.

2020 Mathematics Subject Classification. 65M60, 65M99, 35L65.

Key words and phrases. Flux Vector Splitting; RKDG; Hyperbolic Conservation Law; TVB(D)-minmod Limiter; Lagrange Multiplier Method; Local Characteristic Decomposition.

ABSTRACT. The flux vector splitting (FVS) method, employed for solving hyperbolic conservation systems, is frequently observed in finite difference and finite volume schemes, where it serves to reconstruct the numerical fluxes at the interfaces of adjacent cells. This method now has been incorporated into the discontinuous Galerkin (DG) framework for reconstructing the numerical fluxes required for the spatial semi-discrete formulation, setting it apart from the conventional DG approaches that typically utilize the Lax-Friedrichs flux scheme or classical Riemann solvers such as HLL and HLLC. The FVS method inherently does not introduce any error. Consequently, the control equations of hyperbolic conservation systems are initially reformulated into a flux-split form. Subsequently, a variational approach is applied to this flux-split form, from which a DG spatial semi-discrete scheme based on FVS is derived. When solving hyperbolic conservation laws, DG schemes are prone to numerical pseudo-oscillations, necessitating the incorporation of limiters for correction. Initially, the smoothness measurement function IS from the WENO limiter is integrated into the TVB(D)-minmod limiter, constructing an optimization problem based on the smoothness factor constraint, thereby realizing a TVB(D)-minmod limiter applicable to arbitrary high-order polynomial approximation. Subsequently, drawing on the “reconstructed polynomial and the original high-order scheme’s L^2 -error constraint” from the literature [1], combined with our smoothness factor constraint, a bi-objective optimization problem is formulated to enable the TVB(D)-minmod limiter to balance oscillation suppression and high precision. The aforementioned constrained optimization-type TVB(D)-minmod limiter has been extended to two-dimensional scenarios. When solving hyperbolic conservation systems, limiters are typically required to be used in conjunction with local characteristic decomposition. To transform polynomials from the physical space to the characteristic space, an interpolation-based characteristic transformation scheme has been proposed, and its equivalence with the original moment characteristic transformation has been demonstrated in one-dimensional scenarios. For variable-coefficient and nonlinear equations, the transformation matrix utilized in characteristic transformation must be determined through local freezing. To achieve this, we employ the arithmetic mean or Roe average of the integral means on the common interface of adjacent cells for local freezing, rather than the conventional approach of using the arithmetic mean or Roe average of the cell integral means of each adjacent cell for this purpose. Finally, the concept of “flux vector splitting based on Jacobian eigenvalue decomposition” has been applied to the conservative linear scalar transport equations and the nonlinear Burgers’ equation. This approach has led to the re-derivation of the classical Lax-Friedrichs flux scheme and the provision of a Steger-Warming flux scheme for scalar equations.

CONTENTS

1. Introduction	3
2. Overview of RKDG	9
3. Overview of Flux Vector Splitting for Hyperbolic Conservative System in One-dimension	13
3.1. Based on Jacobian eigenvalue Splitting	13

3.2. Based on Mach Number Splitting	16
4. Roe Average	19
5. DG Based on Flux Vector Splitting in One Dimension (1D-FVS-DG)	20
6. DG Based on Flux Vector Splitting in Two-dimension (2D-FVS-DG)	22
6.1. DG Based on Flux Vector Splitting in Two-dimension	23
6.2. Normal Roe Average Employed by DG in Two-dimension	25
7. A Novel TVB(D)-minmod Limiter for Numerical Pseudo-Oscillation Treatment	26
7.1. Commonly Used Discontinuity Indicators	26
7.2. Constrained Optimization-based TVB(D)-minmod Limiter Compatible with High-Order Polynomial Approximation	29
8. Reconstruction in Characteristic Field	39
8.1. Local Freezing Based on Cell Interface Integral Mean	43
8.2. Interpolation-based Characteristic Transformation	43
8.3. Equivalence of Interpolation-based Characteristic Transformation and Moment Characteristic Transformation	45
9. Numerical Results	46
9.1. Accuracy Tests for FVS-DG	46
9.2. Performance of the IS- L^2 -TVB(D)-minmod Limiter for Scalar Conservation Law	52
9.3. Solving the Riemann Problems for Hyperbolic Conservative Systems Using FVS-DG with IS- L^2 -TVB(D)-minmod Limiter	62
10. Conclusion	72
Acknowledgments	72
References	73
Appendix A. Vector, Matrix, and Tensor Operations in FVS-DG	75
Appendix B. Flux Vector Splitting Method for Shallow Water System	77
B.1. Flux Vector Splitting for Shallow Water Wave Equations in One-dimension	79
B.2. Flux Vector Splitting for Shallow Water Wave Equations in Two-dimension	80
Appendix C. Development of Numerical Flux Formats for Scalar Equations Based on the System's Jacobian-FVS Method	81
C.1. The Classical Lax-Friedrichs Flux Format from the Jacobian-FVS Perspective	81
C.2. Constructing Steger-Warming flux for the scalar equation based on Steger-Warming splitting for system	82

1. INTRODUCTION

The Hyperbolic Conservation Law (HCL) is a homogeneous hyperbolic system of quasi-linear divergence form equations, which can be uniformly represented as follows:

$$\partial_t U + \nabla \cdot \mathcal{F}(U) = 0 \quad \text{in } \Omega \subset \mathbb{R}^d, \quad (1.1)$$

here, d denotes the spatial dimension, and U can represent either a scalar or a vector. In the one-dimensional case,

$$\nabla \cdot \mathcal{F}(U) = \partial_x(\mathcal{F}(U));$$

For higher-dimensional scenarios,

$$\begin{aligned} \mathcal{F}(U) &= (F_1(U), F_2(U), \dots, F_d(U)), \\ \nabla \cdot \mathcal{F}(U) &= \sum_{i=1}^{i=d} \partial_{x_i} F_i(U). \end{aligned}$$

Hyperbolic conservation laws are utilized to describe physical phenomena where the increment of extensive quantities within a system is balanced by the net flux passing through the system's boundaries [3]. Common examples of hyperbolic conservation laws include scalar transport equations, Euler equations, and source-free shallow water wave equations.

Numerical solutions to equation (1.1) commonly employ schemes such as the finite difference method (FD), the finite volume method (FV), and the discontinuous Galerkin method (DG). These methods share a common characteristic in that they are all "local" approximation schemes: first, the FD method, which is derived from a Taylor expansion around the grid framework points, inherently belongs to a local approximation technique; second, the FV method can be regarded as a special case of the DG method when using P0 polynomial approximation, and in the DG method, the basis functions are entirely discontinuous, i.e., they are piecewise defined according to the grid cells, allowing for different orders or types of basis functions on different grid cells. Accordingly, the DG method independently and in parallel carries out variation on each grid cell, ultimately yielding a series of weak equivalent integral equations that are as numerous as the grid cells. To distinguish from the global variational process of traditional finite element methods, we refer to this as local variation.

The FD scheme, when reconstructing the flux value $\mathcal{F}_{i+1/2}$ at the midpoint of the grid, faces the issue of choosing which side of the cell interface to use for interpolation based on the grid framework points. In both FV and DG schemes, each cell generates boundary integral terms during the variational process. However, due to the local nature of the variational form, the flux function \mathcal{F} is multivalued at the common interface of adjacent cells. Numerical flux techniques are employed in local approximation schemes to ensure the uniqueness of the flux at the common interfaces of neighboring cells. That is, numerical flux techniques map a multivalued function to a single-valued function, thereby restoring the mathematical structure and physical connection of the field functions within different cells. Let the mesh be partitioned as \mathcal{T}_h . Consider any cell $\Omega_k \in \mathcal{T}_h$, with its boundary denoted by $\partial\Omega_k$. Let Γ_k denote any part of the boundary $\partial\Omega_k$, such that $\Gamma_k \subset \partial\Omega_k$. Denote $C(\Gamma_k)$ as the set of all cells sharing the boundary segment Γ_k , and suppose there are q cells sharing Γ_k , i.e., $|C(\Gamma_k)| = q$. The numerical solution U_h is a multivalued function on the cell interface Γ_k , with q states $(\partial U_1, \partial U_2, \dots, \partial U_c, \dots, \partial U_q)$, where $\partial U_c = U_h^{(c)} \Big|_{\Gamma_k}$, $c \in C(\Gamma_k)$. The numerical

flux $\widehat{\mathcal{F}}$ is a single-valued function, such that

$$\begin{cases} U_h|_{\Gamma_k} = (\partial U_1, \partial U_2, \dots, \partial U_c, \dots, \partial U_q), & c \in C(\Gamma_k), \\ \widehat{\mathcal{F}} : U_h|_{\Gamma_k} \longrightarrow \widehat{\mathcal{F}}_{\Gamma_k}, & \forall \Gamma_k \in \partial\Omega_k. \end{cases} \quad (1.2)$$

Upwind numerical flux formats are one of the common numerical flux techniques, and this method of constructing numerical fluxes originates from the characteristic line theory of hyperbolic equations: along the reverse direction of the characteristic line (which we refer to as the upwind direction), the state quantities located in the dependency domain of the hyperbolic equation are selected to participate in the calculation of the numerical flux. The upwind direction is easy to determine for scalar hyperbolic equations: since $\frac{\partial f(U)}{\partial x} = \frac{\partial f}{\partial U} \frac{\partial U}{\partial x}$, the upwind direction can be determined based on the sign of $\frac{\partial f}{\partial U}$. Of course, considering variable coefficients and nonlinear cases, the upwind direction itself is also a local concept, meaning that the trend of characteristic lines, and thus the upwind direction, is different in different computational regions. However, for systems, since $\frac{\partial f}{\partial U}$ will be a matrix, which we refer to as the Jacobian matrix, the aforementioned approach of “judging the wind direction based on the sign of $\frac{\partial f}{\partial U}$ ” cannot be directly generalized. Scalar hyperbolic conservation laws involve only a single characteristic wave, but the Jacobian matrix of a hyperbolic conservation system has more than one eigenvalue, corresponding to multiple characteristic waves. Thus, the propagation direction of each characteristic wave can be entirely inconsistent. As long as the Jacobian of the system’s flux function is indefinite over the entire flow field (the eigenvalues of the flux function’s Jacobian at some point in the flow field are not all of the same sign), then from a global flow field perspective, each conservation variable $U^{(i)}$ corresponds to a flux $\mathcal{F}^{(i)}$ that is composed of two scalar sub-fluxes with opposite propagation directions $\left\{ \mathcal{F}^{(i)+}, \mathcal{F}^{(i)-} \right\}_{i=1}^{i=m}$. Here, i is the system component index, and m is the dimension of the system’s state. Therefore, applying the “upwind philosophy” to hyperbolic conservation systems can be achieved by splitting the flux \mathcal{F} into two sub-fluxes with opposite propagation directions, \mathcal{F}^+ and \mathcal{F}^- , and then constructing corresponding upwind schemes for each sub-flux.

Flux splitting techniques, commonly seen in FD schemes and FV schemes, include two categories: flux vector splitting (FVS) and flux difference splitting (FDS) [4, 5]. Flux Vector Splitting (FVS) schemes can be further divided into two categories: those based on Jacobian eigenvalue splitting and those based on Mach number splitting. The former requires the flux function to satisfy homogeneity, i.e., $\mathcal{F} = \frac{\partial \mathcal{F}}{\partial U} \cdot U$. The latter is designed for some practical fluid control equations, such as the compressible Euler equations and shallow water wave equations. Let $A = \frac{\partial \mathcal{F}}{\partial U}$, where A is referred to as the Jacobian matrix of the flux function \mathcal{F} . In the Jacobian-FVS scheme, the Jacobian matrix A is diagonalized by a similarity transformation, and then its eigenvalues are split to obtain A^+ and A^- , which satisfy positive definiteness and negative definiteness, respectively. This allows the flux \mathcal{F} to be split into positive flux F^+ and negative flux F^- [6], and subsequently, numerical fluxes are constructed

for each according to their respective upwind directions. Some hyperbolic conservation systems do not satisfy $\mathcal{F} = \frac{\partial \mathcal{F}}{\partial U} \cdot U$, such as the shallow water wave equations. We forcibly modify the wave speed to $a^* = \sqrt{\frac{1}{2}gh}$ to satisfy the homogeneity requirement of the flux function, and then perform flux splitting based on the modified Jacobian eigenvalues [7]. Regardless of whether the flux function of a hyperbolic conservation system satisfies the homogeneity requirement, the Mach-FVS scheme can be employed. In the Mach-FVS scheme, the characteristic wave propagation direction is no longer determined by the eigenvalues of the Jacobian matrix. Instead, the local Mach number $M_a = \frac{u}{a}$ (where a is the local speed of sound) is used to determine the characteristic propagation direction at that point: when $M_a \geq 1$, the characteristic propagates in the positive direction; when $M_a \leq -1$, it propagates in the negative direction; and when $|M_a| < 1$, both positive and negative characteristic waves coexist. Similar to the Jacobi-FVS splitting of the Jacobian eigenvalues, a natural idea is to achieve flux vector splitting by splitting the Mach number M_a . Depending on whether the pressure term is split separately, the Mach-FVS can be further divided into the van Leer splitting method [4, 5] and the AUSM splitting method [8]. The former includes the pressure term in the flux function for splitting, while the latter separates the pressure term from the flux function for individual splitting.

Reed and Hill [9] proposed the first Discontinuous Galerkin method in 1973, for solving the neutron transport equation. Since then, the DG method has developed rapidly, with the emergence of local discontinuous Galerkin methods (LDG) [10], direct discontinuous Galerkin methods (DDG) [11], and ultra-weak discontinuous Galerkin methods (UWDG) [12] for solving higher-order equations. In addition, semi-Lagrangian discontinuous Galerkin methods (SLDG) [13] and Euler-Lagrange discontinuous Galerkin (ELDG) [14] have been developed based on the characteristic line theory of hyperbolic equations from a Lagrangian perspective. These methods exhibit high-order structure-preserving and unconditional stability (large time-step stability) characteristics when solving transport equations. The DG method discussed in this paper is the Runge-Kutta Discontinuous Galerkin (RKDG) method. RKDG [15, 16, 17, 18, 19] is a commonly used DG scheme for solving hyperbolic conservation systems such as the compressible Euler equations. It employs explicit and nonlinearly stable high-order Runge-Kutta methods for temporal discretization and uses DG methods for spatial discretization. The former ensures the nonlinear stability of the method regardless of accuracy, while the latter integrates the concepts of numerical fluxes and slope limiters from high-resolution FD and FV schemes. The resulting RKDG method is stable, high-order accurate, and highly parallelizable [20], capable of easily handling complex geometries and boundary conditions. The RKDG method has the advantages of local conservation, arbitrary triangular meshing, good parallel efficiency, h-p adaptive capability [21], and certain superconvergence properties [22, 23, 24, 25, 26].

Even if the initial conditions are smooth, hyperbolic conservation law equations may develop discontinuities during their evolution. When calculating discontinuous problems,

high-order schemes such as the RKDG method can exhibit numerical pseudo-oscillations (non-physical oscillations) near discontinuities, which can lead to instability in the numerical scheme. To suppress numerical oscillations, a common practice is to introduce limiters. There are currently two types of common limiters: TVD/TVB-type limiters [1, 27] and ENO/WENO-type limiters [2, 28, 29, 30, 31, 32].

TVD (Total Variation Diminishing) limiters initially decompose a high-order scheme into a first-order upwind scheme and a linear sum of a correction term (the first-order scheme has sufficient dissipation to automatically satisfy the non-oscillation property). Subsequently, according to Harten's lemma [33], the correction term is restricted. The requirement is that in smooth regions, the limiter does not affect the correction term, while near discontinuities, the limiter sets the correction term to zero, thereby automatically degrading to a stable, non-oscillatory first-order upwind scheme. TVD limiters tend to lose accuracy near extremum points, leading to the development of TVB (Total Variation Bounded) limiters, which only require that the average total variation be bounded. WENO (Weighted Essentially Non-Oscillatory) limiters utilize several stencils composed of the troubled cell and its neighboring cells. Based on the cell integral mean, several template polynomials are obtained through interpolation. Subsequently, a template smoothness measurement function is introduced to non-linearly weight and combine the various template polynomials to obtain the final reconstructed polynomial on the troubled cell. The smoother the stencil, the greater the weight of the corresponding template polynomial, thereby achieving a smooth filtering of the pseudo-oscillations near discontinuities [28, 34].

From the above, it is known that limiters should act near discontinuities; using limiters in smooth regions can easily lead to loss of accuracy and increase unnecessary computational time costs. Therefore, a discontinuity indicator can be introduced to identify troubled cells, i.e., those that contain discontinuities. Accordingly, the steps for using the limiter are: (1) The discontinuity indicator identifies troubled cells; (2) The limiter is applied to troubled cells.

It should be noted that some limiters, such as the TVB(D)-minmod limiter, have built-in discontinuity indication functions, while WENO limiters require additional configuration of a discontinuity indicator [35]. An obvious idea is to remove the slope correction part of the TVB(D)-minmod limiter and keep the remaining part as a discontinuity indicator for WENO. In addition, a commonly used discontinuity indicator is the KXRCF discontinuity indicator [36]. This indicator is based on the conclusion that "DG algorithms have strong superconvergence on the outflow boundaries of each cell in smooth areas". If a cell contains a discontinuity, then its strong superconvergence on the outflow boundary will be destroyed, which in turn can determine that the cell is a troubled cell.

When applying limiters to a system of equations, there are typically two implementation methods: (a) reconstruct each component individually; (b) in conjunction with characteristic decomposition [37]. The former is easy to implement in a program; reconstructing directly by component is essentially assuming that the components are already decoupled, but in the

original system of equations, the unknown components are usually coupled, hence the actual effect may not be good. The latter leverages the Jacobian of the flux function to transform the system of equations into the characteristic space to achieve true decoupling, and then reconstructs each component individually.

The main work of this paper is as follows: For the first time, the FVS method is introduced into the DG scheme. By utilizing $\mathcal{F} = \mathcal{F}^+ + \mathcal{F}^-$, an equivalent flux-split form of the original differential equation is obtained: $U_t + \nabla \cdot \mathcal{F}^+ + \nabla \cdot \mathcal{F}^- = 0$. A variation is then performed based on the flux-split form of the original equation, leading to the derivation of a numerical flux format based on flux vector splitting within the DG framework. Specifically, for the two-dimensional hyperbolic conservation system, flux vector splitting is implemented using normal fluxes in the outward normal direction on the interfaces of cells. In terms of limiters, we have introduced the smoothness measurement function IS from the WENO limiter into the TVB(D)-minmod limiter. An optimization problem based on the smoothness factor constraint has been constructed, realizing a TVB(D)-minmod limiter applicable to arbitrary high-order polynomial approximation, which we refer to as the IS-TVB(D)-minmod limiter. Thanks to the constraint of the smoothness factor IS , the IS-TVB(D)-minmod limiter possesses strong capabilities in suppressing numerical oscillations, but correspondingly, it is prone to losing high-order accuracy. To address this issue, we have adopted the approach from the literature [1], introducing the L^2 -error between the reconstructed polynomial and the original high-order scheme into the objective function. We have constructed a “smoothness factor IS- L^2 -error constraint dual-objective optimization” problem to further improve the TVB(D)-minmod limiter, which we refer to as IS- L^2 -TVB(D)-minmod limiter. Furthermore, when reconstructing for the hyperbolic systems, it is often necessary to accompany it with local characteristic decomposition, and the transformation matrix required for characteristic decomposition is determined by local freezing techniques. To this end, we propose to use the integral mean on the cell interfaces instead of the traditional approach of using the cell integral mean to achieve local freezing. After obtaining the characteristic transformation matrix, it is necessary to transform the DG polynomials from the physical space to the characteristic space. The traditional approach is to construct a column vector of modal coefficients for the same degree terms of each system component’s DG polynomial, and then perform characteristic transformation through matrix-vector multiplication, referred to as “moment characteristic transformation”. In contrast, we have constructed an interpolation-based polynomial characteristic transformation to achieve characteristic decomposition, and in the one-dimensional case, we have proven the equivalence of the interpolation-based characteristic transformation with the moment characteristic transformation.

The structure of this paper is organized as follows: In Section 2, a review of the Runge-Kutta Discontinuous Galerkin (RKDG) method is presented. Subsequently, Section 3 introduces two implementations of Flux Vector Splitting (FVS) for one-dimensional hyperbolic conservation systems, exemplified by Jacobian eigenvalue-based splitting and Mach number-based splitting. Considering that characteristic reconstruction necessitates the use of local

freezing techniques, and that local freezing for variable-coefficient or nonlinear equations requires the employment of some form of averaging, Roe's averaging is introduced in Section 4. Section 5 details the flux vector splitting formulation for the RKDG scheme applied to one-dimensional hyperbolic conservation systems, while the construction process for the two-dimensional case is elaborated in Section 6. Section 7 focuses on the improvements made to the TVB(D)-minmod limiter. Section 8 is dedicated to showcasing our unique perspectives on local characteristic decomposition. The effectiveness of the aforementioned work is validated through carefully designed numerical experiments in Section 9. Finally, Section 10 provides a summary of the work presented in this paper.

Note that a detailed explanation of the symbols and operational rules used in the definition of the FVS-DG weak solution process in Section 5 and Section 6 must be referred to Appendix A. Furthermore, Appendix B provides the FVS-DG formulation for the shallow water wave equations. Lastly, Appendix C re-examines the construction process and working principles of the classical Lax-Friedrichs flux scheme from the perspective of Jacobian-FVS. Inspired by the Steger-Warming splitting method for systems, Appendix C presents the Steger-Warming flux scheme for scalar transport equations and the Burgers' equation, and proves that this flux scheme satisfies compatibility, Lipschitz continuity, and monotonicity, thereby ensuring that the weak solutions obtained for scalar transport equations and the Burgers' equation using this flux scheme meet the cell entropy inequality and L^2 -stability.

2. OVERVIEW OF RKDG

Let the computational domain be $\Omega \subset \mathbb{R}^d$, where $d = 1, 2, \dots$. The mesh partitioning is denoted as $\mathcal{T}_h = \Omega_1 \cup \Omega_2 \cup \dots \cup \Omega_N$, with the property that $\Omega_i \cap \Omega_j = \emptyset$, for all $i, j \in \{1, 2, \dots, N\}$.

The definition of the local orthogonal basis functions on an arbitrary domain Ω_i is as follows:

Definition 2.1 (Normative Orthogonal System). *Let $\{\phi_0^{(\mathcal{D})}, \phi_1^{(\mathcal{D})}, \dots, \phi_\ell^{(\mathcal{D})}\}$ (where ℓ is finite or $\ell \in \infty$) be a set of functions defined on a bounded domain \mathcal{D} , with $\phi_i^{(\mathcal{D})} \in L^2(\mathcal{D})$, for all $i = 0, 1, 2, \dots, \ell$. We refer to $\{\phi_i^{(\mathcal{D})}\}_{i=0}^{i=\ell}$ as an normative orthonormal system on the bounded domain \mathcal{D} , if*

$$\left\langle \phi_j^{(\mathcal{D})}, \phi_k^{(\mathcal{D})} \right\rangle_{L^2(\mathcal{D})} = \int_{\mathcal{D}} \phi_j^{(\mathcal{D})} \phi_k^{(\mathcal{D})} d\mathbf{X} = \delta_{jk} = \begin{cases} 0, & j \neq k \\ 1, & j = k \end{cases} \quad (2.1)$$

where, $\mathbf{X} \in \mathbb{R}^d, d = 1, 2, \dots$

Let $P^K(\Omega_i)$ be the space of polynomials of degree K defined on Ω_i . It can be spanned by any normative orthonormal system $\{\phi_m^{(i)}\}_{m=0}^{m=K}$ on it. The set $\{\phi_m^{(i)}\}_{m=0}^{m=K}$ satisfies:

- $\{\phi_m^{(i)}\}_{m=0}^{m=K}$ is an normative orthonormal system in $P^K(\Omega_i)$,
- $\forall q \in P^K(\Omega_i), \exists! \{\alpha_m\}_{m=0}^{m=K} \subset \mathbb{R}, s.t. q = \sum_{m=0}^{m=K} \alpha_m \phi_m^{(i)}$.

On this basis, the space $\mathbb{V}_h^K(\Omega)$, where the DG weak solution resides, can be defined:

Definition 2.2 (DG Weak Solution Space).

$$\mathbb{V}_h^K(\Omega) = \{v \in L^2(\Omega) : v|_{\Omega_i} \in P^K(\Omega_i), \forall \Omega_i \in \mathcal{T}_h\}$$

here, \mathcal{T}_h is a partitioning of $\Omega \subset \mathbb{R}^d$.

Thus, for Equation (1.1), the definition of the DG weak solution is as follows:

Definition 2.3 (DG Weak Solution). *In the DG weak solution space, a unique polynomial function $U_h \in \mathbb{V}_h^K$ is determined such that for all $V \in \mathbb{V}_h^K$ and for all $\Omega_i \in \mathcal{T}_h$, the following is satisfied:*

$$\int_{\Omega_i} V \partial_t(U_h) \, d\mathbf{X} = \int_{\Omega_i} \mathcal{F}(U_h) \cdot (\nabla V) \, d\mathbf{X} - \int_{\partial\Omega_i} \widehat{\mathcal{F}}_\nu V \, d\sigma. \quad (2.2)$$

Here, $\widehat{\mathcal{F}}_\nu$ is the numerical normal flux, which is a single-valued function on $\partial\Omega_i$ and satisfies $\widehat{\mathcal{F}}_\nu \approx \mathbf{n} \cdot \mathcal{F}(U_h)$, where \mathbf{n} is the unit outward normal vector on $\partial\Omega_i$.

Remark 2.1. *The construction of the numerical flux $\widehat{\mathcal{F}}_\nu$ to approximate $\mathbf{n} \cdot \mathcal{F}(U_h)$ is key to the spatial discretization in the DG algorithm. The literature [38, 39] introduces a variety of typical numerical flux formats, while our work involves the introduction of flux splitting methods from FD/FV schemes, especially the flux vector splitting method, to construct $\widehat{\mathcal{F}}_\nu$ in the DG scheme.*

Let $U_h^{(i)} := U_h|_{\Omega_i} \in P^K(\Omega_i)$, then we have

$$U_h^{(i)}(\mathbf{X}, t) = \sum_{\ell=0}^{\ell=K} \alpha_\ell^{(i)}(t) \phi_\ell^{(i)}(\mathbf{X}), \quad (2.3)$$

The $\alpha_\ell^{(i)}$ is referred to as the ℓ -th moment or the ℓ -th modal coefficient of $U_h^{(i)}$, satisfying

$$\alpha_\ell^{(i)} = \left\langle U_h^{(i)}, \phi_\ell^{(i)} \right\rangle_{L^2(\Omega_i)}. \quad (2.4)$$

Note that the set $\{\phi_\ell^{(i)}\}_{\ell=0}^{\ell=K}$ is normative orthonormal.

On Ω_i , take the test function $V = \phi_r^{(i)}$. Note that the basis function $\phi_\ell^{(i)}$ depends only on \mathbf{X} , that is, it does not evolve with time. Thus, we have

$$\int_{\Omega_i} \phi_r^{(i)} \partial_t(U_h) \, d\mathbf{X} = \int_{\Omega_i} \phi_r^{(i)} \partial_t(U_h^{(i)}) \, d\mathbf{X} = \sum_{\ell=0}^{\ell=K} \partial_t(\alpha_\ell^{(i)}) \cdot \left\langle \phi_r^{(i)}, \phi_\ell^{(i)} \right\rangle_{L^2(\Omega_i)}. \quad (2.5)$$

Utilizing the orthogonality (2.1), it immediately follows that

$$\int_{\Omega_i} \phi_r^{(i)} \partial_t(U_h) \, d\mathbf{X} = \partial_t(\alpha_r^{(i)}). \quad (2.6)$$

From equations (2.2) and (2.6), the following semi-discrete spatial format can be derived:

$$\frac{d}{dt}(\alpha_r^{(i)}) = DG(U_h^{(i)}), \quad (2.7)$$

where $i = 1, 2, \dots, N$; $r = 0, 1, \dots, K$.

The following presents the third-order Total Variation Diminishing (TVD) Runge-Kutta (RK) temporal discretization scheme (TVD-RK3), the fourth-order four-stage non-Strong Stability Preserving (non-SSP) RK temporal discretization scheme (RK4), and the 10-stage fourth-order SSP RK temporal discretization scheme (SSPRK(10,4)): Let $\Delta t > 0$, and partition the time domain as $t_n = n \cdot \Delta t$, for $0 \leq n \leq M = T/\Delta t$. Let L_h be some spatial discretization operator, such as the previously mentioned Discontinuous Galerkin spatial discretization scheme $DG(\cdot)$. Then,

- TVD-RK3

$$\left. \begin{aligned} u^{(1)} &= u^n + \Delta t \cdot L_h(u^n; t_n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t \cdot L_h(u^{(1)}; t_n + \Delta t), \\ u^{(3)} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t \cdot L_h\left(u^{(2)}; t_n + \frac{1}{2}\Delta t\right), \\ u^{n+1} &= u^{(3)}. \end{aligned} \right\} \quad (2.8)$$

- RK4

$$\left. \begin{aligned} u^{(1)} &= u^n + \frac{1}{2}\Delta t \cdot L_h(u^n; t_n), \\ u^{(2)} &= u^n + \frac{1}{2}\Delta t \cdot L_h\left(u^{(1)}; t_n + \frac{1}{2}\Delta t\right), \\ u^{(3)} &= u^n + \Delta t \cdot L_h\left(u^{(2)}; t_n + \frac{1}{2}\Delta t\right), \\ u^{(4)} &= \frac{1}{3}\left(-u^n + u^{(1)} + 2u^{(2)} + u^{(3)}\right) + \frac{1}{6}\Delta t \cdot L_h(u^{(3)}; t_n + \Delta t), \\ u^{n+1} &= u^{(4)}. \end{aligned} \right\} \quad (2.9)$$

- SSPRK(10, 4)

$$\left. \begin{aligned}
u^{(1)} &= u^n + \frac{1}{6}\Delta t L_h(u^n; t_n), \\
u^{(2)} &= u^{(1)} + \frac{1}{6}\Delta t L_h\left(u^{(1)}; t_n + \frac{1}{6}\Delta t\right), \\
u^{(3)} &= u^{(2)} + \frac{1}{6}\Delta t L_h\left(u^{(2)}; t_n + \frac{1}{3}\Delta t\right), \\
u^{(4)} &= u^{(3)} + \frac{1}{6}\Delta t L_h\left(u^{(3)}; t_n + \frac{1}{2}\Delta t\right), \\
u^{(5)} &= \frac{3}{5}u^n + \frac{2}{5}u^{(4)} + \frac{1}{15}\Delta t L_h\left(u^{(4)}; t_n + \frac{2}{3}\Delta t\right), \\
u^{(6)} &= u^{(5)} + \frac{1}{6}\Delta t L_h\left(u^{(5)}; t_n + \frac{1}{3}\Delta t\right), \\
u^{(7)} &= u^{(6)} + \frac{1}{6}\Delta t L_h\left(u^{(6)}; t_n + \frac{1}{2}\Delta t\right), \\
u^{(8)} &= u^{(7)} + \frac{1}{6}\Delta t L_h\left(u^{(7)}; t_n + \frac{2}{3}\Delta t\right), \\
u^{(9)} &= u^{(8)} + \frac{1}{6}\Delta t L_h\left(u^{(8)}; t_n + \frac{5}{6}\Delta t\right), \\
u^{n+1} &= \frac{1}{25}u^n + \frac{9}{25}u^{(4)} + \frac{3}{5}u^{(9)} + \frac{3}{50}\Delta t L_h\left(u^{(4)}; t_n + \frac{2}{3}\Delta t\right) + \frac{1}{10}\Delta t L_h\left(u^{(9)}; t_n + \Delta t\right).
\end{aligned} \right\} \quad (2.10)$$

By combining the RK-type temporal discretization schemes such as (2.8), (2.9), or (2.10) with the DG spatial discretization format (2.2), one obtains the fully discrete RKDG scheme, which allows the evolution of U_h^n to U_h^{n+1} . This is succinctly denoted as

$$U_h^{n+1} = RKDG(U_h^n). \quad (2.11)$$

To suppress numerical pseudo-oscillations, it is necessary to introduce a limiter $\Lambda \Pi_h$ to restrict U_h^n before the temporal evolution from t_n to t_{n+1} , yielding \tilde{U}_h^n , that is,

$$\tilde{U}_h^n = \Lambda \Pi_h(U_h^n). \quad (2.12)$$

Using \tilde{U}_h^n in the temporal evolution gives

$$U_h^{n+1} = RKDG(\tilde{U}_h^n). \quad (2.13)$$

Lastly, for convenience, we take the explicit Euler temporal discretization scheme as an example to provide a complete description of the spatial discretization, temporal discretization, and limiter correction processes in the DG algorithm:

Determine a unique polynomial function $U_h^{n+1} \in \mathbb{V}_h^K$ in the DG weak solution space such that for all $V \in \mathbb{V}_h^K$ and for all $\Omega_i \in \mathcal{T}_h^d$, the following is satisfied:

$$\int_{\Omega_i} V \frac{U_h^{n+1} - \tilde{U}_h^n}{\Delta t} d\mathbf{X} = \int_{\Omega_i} \mathcal{F}(\tilde{U}_h^n) \cdot (\nabla V) d\mathbf{X} - \int_{\partial\Omega_i} \widehat{\mathcal{F}}_\nu^n V d\sigma, \quad (2.14)$$

where $\tilde{U}_h^n = \Lambda \Pi_h(U_h^n)$, and $\widehat{\mathcal{F}}_\nu^n$ is the normal numerical flux at time t_n .

3. OVERVIEW OF FLUX VECTOR SPLITTING FOR HYPERBOLIC CONSERVATIVE SYSTEM IN ONE-DIMENSION

This section introduces the flux vector splitting method using the one-dimensional hyperbolic conservation system as an example. In general, homogeneous systems can be split based on the eigenvalues of the Jacobian matrix of the flux function, with the goal of obtaining both a positive definite matrix and a negative definite matrix whose sum equals the original Jacobian matrix. This allows the flux function to be divided into positive and negative flux components, where the positive flux corresponds to characteristic waves propagating in the positive direction of the coordinate system, and the negative flux corresponds to characteristic waves propagating in the negative direction. Consequently, the upwind direction for both can be clearly defined, and a stable upwind scheme can be employed for computation. For practical fluid equations such as the compressible Euler equations or shallow water wave equations, the Mach number M_a (or the Froude number F_r for shallow water waves) can be introduced to rewrite the flux function as a linear function of M_a , thereby achieving flux function splitting through the splitting of M_a .

3.1. Based on Jacobian eigenvalue Splitting. The purpose of splitting the flux function through Jacobian eigenvalue splitting is to construct an upwind-type numerical flux scheme. This method requires that the flux function itself be a homogeneous function. The following provides the definition of a homogeneous function:

Definition 3.1 (Homogeneous Function). *A function f is homogeneous if it satisfies $f(\alpha u) = \alpha f(u)$, for all $u \in D(f)$, where $D(f)$ is the domain of f , and α is any constant such that for a given u , $\alpha u \in D(f)$.*

It is easy to see that homogeneous functions satisfy $f'(\alpha u) \cdot u = f(u)$.

By taking $\alpha = 1$, we obtain

$$f(u) = f'(u) \cdot u. \quad (3.1)$$

Consider the following one-dimensional homogeneous system $\partial_t \mathbf{U} + \partial_x f(\mathbf{U}) = 0$, where $\mathbf{U} = (u_1, u_2, u_3, \dots, u_m)^T$, and $f(\mathbf{U}) = (f_1(\mathbf{U}), f_2(\mathbf{U}), f_3(\mathbf{U}), \dots, f_m(\mathbf{U}))^T$, with the property that $f(\mathbf{U}) = \frac{\partial f}{\partial \mathbf{U}} \cdot \mathbf{U}$. Let $A = \frac{\partial f}{\partial \mathbf{U}}$, then $f = A\mathbf{U}$.

The general process of Jacobian-FVS is as follows:

- Step1. First, the flux function needs to be rewritten as $f(\mathbf{U}) = A(\mathbf{U}) \cdot \mathbf{U}$.

Remark 3.1. *if $f(\mathbf{U})$ is a homogeneous function (e.g., Euler equations), then take $A(\mathbf{U}) = \frac{\partial f}{\partial \mathbf{U}}$.*

Remark 3.2. *The flux function f of the shallow water wave equations is not a homogeneous function and requires special preprocessing. For example, the wave speed $a = \sqrt{gh}$ may be forcibly modified to $a^* = \sqrt{\frac{gh}{2}}$. For more details, see Appendix B.*

- Step2. Diagonalize A by similarity transformation.

$$\begin{aligned} T^{-1}AT &= \Lambda, \\ A &= T\Lambda T^{-1} = R\Lambda L. \end{aligned}$$

Note: $RL = I_d$ (identity matrix), $T = R$, $T^{-1} = L$. A, R, L, Λ are all matrix-valued functions of \mathbf{U} .

- Step3. Split Λ .

$$\Lambda = \Lambda^+ + \Lambda^-,$$

satisfying

$$\lambda(\Lambda^+) \geq 0, \quad \lambda(\Lambda^-) \leq 0,$$

that is, Λ^+ is positive definite, Λ^- is negative definite.

- Step4. Split A .

$$\begin{aligned} A &= R(\Lambda^+ + \Lambda^-)L = R\Lambda^+L + R\Lambda^-L = A^+ + A^-, \\ A^+ &= R\Lambda^+L, \quad A^- = R\Lambda^-L, \end{aligned}$$

satisfying

$$\lambda(A^+) \geq 0, \quad \lambda(A^-) \leq 0,$$

that is, A^+ is positive definite, A^- is negative definite.

- Step5. Split the flux vector f .

$$f = f^+ + f^- \quad , \quad f^+ = A^+\mathbf{U} \quad , \quad f^- = A^-\mathbf{U}.$$

There are two common schemes to implement Step 3, namely the Steger-Warming splitting scheme and the Lax-Friedrichs splitting scheme. They are introduced separately below.

Remark 3.3. *The flux function for the 1D-Euler equations, given by $f = (\rho u, \rho u^2 + P, (E + u)P)^T$, and the normal flux function for the 2D-Euler equations, given by $F_n = n_x \cdot F + n_y \cdot G$, are homogeneous functions of the 1-dimensional conservative vector $\mathbf{U} = (\rho, \rho u, E)^T$ and the 2-dimensional conservative vector $\mathbf{U} = (\rho, \rho u, \rho v, E)^T$, respectively.*

3.1.1. *Steger-Warming Splitting.* The Steger-Warming splitting scheme (S-W Splitting) is as follows:

$$\lambda_k^+ = \frac{\lambda_k + |\lambda_k|}{2} \quad , \quad \lambda_k^- = \frac{\lambda_k - |\lambda_k|}{2} \quad , \quad k = 1, 2, \dots, m. \quad (3.2)$$

Then,

$$\Lambda^+ = \frac{1}{2}(\Lambda + |\Lambda|) \quad , \quad \Lambda^- = \frac{1}{2}(\Lambda - |\Lambda|). \quad (3.3)$$

Hence,

$$A^+ = R \frac{(\Lambda + |\Lambda|)}{2} L = \frac{1}{2}(A + |A|) \quad , \quad A^- = R \frac{(\Lambda - |\Lambda|)}{2} L = \frac{1}{2}(A - |A|). \quad (3.4)$$

Here, $|A| = R|\Lambda|L$, where $|\Lambda|$ denotes taking the absolute value of the elements in Λ . If smoothness is considered, it can be improved as:

$$\lambda_k^+ = \frac{\lambda_k + \sqrt{|\lambda_k|^2 + \delta^2}}{2} \quad , \quad \lambda_k^- = \frac{\lambda_k - \sqrt{|\lambda_k|^2 + \delta^2}}{2}, \quad (3.5)$$

where δ is a small positive constant, for example, δ can be taken as 10^{-8} .

3.1.2. *Lax-Friedrichs Splitting.* The Lax-Friedrichs splitting scheme (L-F Splitting) is as follows:

$$\lambda_k^+ = \frac{\lambda_k + M}{2}, \quad \lambda_k^- = \frac{\lambda_k - M}{2}, \quad M > 0. \quad (3.6)$$

Local L-F splitting:

$$M_j = \max_k \{|\lambda_k(x_j)|\}. \quad (3.7)$$

Global L-F splitting:

$$M = \max_j \max_k \{\lambda_k(x_j)\}. \quad (3.8)$$

Therefore,

$$\Lambda^+ = \frac{1}{2}(\Lambda + MI_d) \quad , \quad \Lambda^- = \frac{1}{2}(\Lambda - MI_d). \quad (3.9)$$

Note that $RL = I_d$, hence

$$A^+ = R \frac{(\Lambda + MI_d)}{2} L = \frac{1}{2}(A + MI_d) \quad , \quad A^- = R \frac{(\Lambda - MI_d)}{2} L = \frac{1}{2}(A - MI_d). \quad (3.10)$$

Remark 3.4. *From this, it can be seen that the L-F flux vector splitting is a “simplified” version of the S-W flux vector splitting, as L-F replaces the individual moduli of all eigenvalues with a common upper bound of their moduli, using this upper bound uniformly in the splitting process. Conversely, S-W provides more precise control over artificial viscosity than L-F (S-W has less dissipation than L-F), because S-W involves the splitting with the modulus of each eigenvalue itself, rather than a simplified treatment using a single upper bound value.*

Remark 3.5. *The S-W flux vector splitting and local L-F flux vector splitting possess adaptivity, meaning that at each cell boundary, the corresponding $|\lambda|_{i+1/2}, M_{i+1/2}$ are taken. In contrast, the global Lax-Friedrichs flux vector splitting scheme abandons this adaptivity, but it is more convenient to implement and introduces greater artificial viscosity, which is beneficial for suppressing numerical oscillations (at the cost of reduced accuracy).*

Remark 3.6. *Noticing the similarity between the Lax-Friedrichs splitting method and the common Lax-Friedrichs flux format in DG schemes, we may also attempt to develop the Steger-Warming splitting method into a flux format for scalar equations. Details can be found in Appendix C.*

3.2. Based on Mach Number Splitting. For practical fluid control equations, such as the compressible Euler equations and shallow water wave equations, a dimensionless parameter, the Mach number, can be defined as the ratio of the fluid velocity to the local speed of sound, denoted as $M_a = \frac{u}{a}$. The Mach number determines the characteristic propagation direction. This section illustrates the concept using the 1D-Euler equations as an example, and for the Mach-FVS related to the shallow water wave equations, please refer to Appendix B.

The eigenvalues of the Jacobian for the one-dimensional (1D) Euler equations are given as follows:

$$\lambda_1 = u \quad \lambda_2 = u - a \quad \lambda_3 = u + a. \quad (3.11)$$

Note that the speed of sound $a = \sqrt{\frac{\gamma P}{\rho}} > 0$, where γ is the specific heat ratio.

- if $M_a > 1$, then $u > a > 0$,

hence $u > 0, u - a > 0, u + a > 0$.

This implies that all three characteristic waves propagate in the positive direction.

Consequently, there is no negative flux in the flux vector \mathbf{F} , that is $\mathbf{F} = \mathbf{F}^+, \mathbf{F}^- = \mathbf{0}$.

- if $M_a < -1$, then $u < -a < 0$,

hence $u < 0, u - a < 0, u + a < 0$.

This implies that all three characteristic waves propagate in the negative direction.

Consequently, there is no positive flux in the flux vector \mathbf{F} , that is $\mathbf{F} = \mathbf{F}^-, \mathbf{F}^+ = \mathbf{0}$.

- If $|M_a| < 1$, the situation can be discussed in the following two cases:

Case 1: $u > 0, u - a < 0, u + a > 0$ (both positive and negative fluxes exist);

Case 2: $u < 0, u - a < 0, u + a > 0$ (both positive and negative fluxes exist).

Remark 3.7. *When $|M_a| > 1$, that is, $M_a > 1$ or $M_a < -1$, it is referred to as supersonic; when $|M_a| < 1$, it is referred to as subsonic.*

3.2.1. van Leer Splitting. The split fluxes f^+ and f^- of the Steger-Warming scheme are not continuously differentiable near the sonic point, which can lead to errors in calculations. In contrast, the van Leer split fluxes are continuously differentiable near the sonic point [4].

Rewrite the flux function \mathbf{F} of the 1D-Euler equations in the van Leer form:

$$\begin{aligned}
\mathbf{F} &= \begin{bmatrix} \rho u \\ \rho u^2 + P \\ (E + P)u \end{bmatrix} = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \left(\frac{1}{2}\rho u^2 + \frac{P}{\gamma-1} + P\right)u \end{bmatrix} = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \left(\frac{1}{2}\rho u^2 + \frac{\gamma P}{\gamma-1}\right)u \end{bmatrix} \\
&= \begin{bmatrix} \rho u \\ \rho u^2 + \frac{\rho a^2}{\gamma} \\ \left(\frac{1}{2}\rho u^2 + \frac{\gamma}{\gamma-1}\frac{\rho a^2}{\gamma}\right)u \end{bmatrix} = \begin{bmatrix} \rho a M_a \\ \rho a M_a u + \frac{\rho a^2}{\gamma} \\ \left(\frac{1}{2}M_a u + \frac{a}{\gamma-1}\right)\rho a u \end{bmatrix} \\
&= \rho a \begin{bmatrix} M_a \\ u M_a + a/\gamma \\ u^2 M_a/2 + au/(\gamma-1) \end{bmatrix}, \tag{3.12}
\end{aligned}$$

Then achieve the splitting of \mathbf{F} by splitting M_a .

- When $M_a \geq 1$ (characteristics propagate to the right, flux is positive),

$$\mathbf{F} = \mathbf{F}^+ (\mathbf{F}^- = 0). \tag{3.13}$$

- When $M_a \leq -1$ (characteristics propagate to the left, flux is negative),

$$\mathbf{F} = \mathbf{F}^- (\mathbf{F}^+ = 0). \tag{3.14}$$

- $|M_a| < 1$, that is, $-1 < M_a < 1$:

$$\mathbf{F} = \mathbf{F}^+ + \mathbf{F}^- \quad , \quad \mathbf{F}^\pm = \begin{bmatrix} 1 \\ [(\gamma-1)u \pm 2a]/\gamma \\ [(\gamma-1)u \pm 2a]^2 / [2(\gamma^2-1)] \end{bmatrix} \cdot \rho a M_a^\pm, \tag{3.15}$$

here,

$$M_a^+ = \frac{(M_a + 1)^2}{4}, \quad M_a^- = -\frac{(M_a - 1)^2}{4}, \tag{3.16}$$

satisfying

$$M_a = M_a^+ + M_a^-. \tag{3.17}$$

3.2.2. Liou-Steffen Splitting (AUSM-type Methods). The fundamental idea of the AUSM (Advection Upstream Splitting Method) scheme is to consider the advection waves (related to the characteristic velocity u , linear) and the acoustic waves (related to the characteristic velocities $u + c$ and $u - c$, nonlinear) as physically distinct processes. Therefore, the inviscid flux is split into advection and pressure flux terms for separate treatment. In terms of scheme construction, the AUSM scheme is an evolution and improvement of the van Leer scheme. However, it distinguishes between linear and nonlinear characteristics, splits the pressure term and the advection flux, obtains the ‘‘upwind’’ property by judging the specially constructed interface Mach number, and accurately captures shocks through the careful design of a unified sound speed in the adjacent cells at the interface. From the analysis of its

dissipation term, this is a composite scheme of FVS (Flux Vector Splitting) and FDS (Flux Difference Splitting). The AUSM scheme theoretically distinguishes between the linear field (related to the characteristic velocity u) and the nonlinear field (related to the characteristic velocities $u \pm c$) in the flow's advection characteristics and splits the pressure term and the advection flux separately.

The flux function \mathbf{F} of the one-dimensional (1D) Euler equations, in the Liou-Stenffen form, is given as follows:

$$\mathbf{F} = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ u(E + P) \end{bmatrix} = \begin{bmatrix} \rho u \\ \rho u^2 + P \\ \rho u \tilde{H} \end{bmatrix} = \begin{bmatrix} \rho a M_a \\ \rho a M_a u + P \\ \rho a M_a \tilde{H} \end{bmatrix} = \underbrace{\rho a M_a \begin{bmatrix} 1 \\ u \\ \tilde{H} \end{bmatrix}}_{\mathbf{F}^a} + \underbrace{\begin{bmatrix} 0 \\ P \\ 0 \end{bmatrix}}_{\mathbf{F}^P}, \quad (3.18)$$

where \mathbf{F}^a is referred to as the advection flux and \mathbf{F}^P as the pressure flux, satisfying $\mathbf{F} = \mathbf{F}^a + \mathbf{F}^P$.

The Mach number M_a is split to thereby split the advection flux:

$$M_a = M_a^+ + M_a^-,$$

where,

$$M_a^+ = \begin{cases} M_a & , \quad M_a > 1 \\ (M_a + 1)^2 / 4 & , \quad |M_a| \leq 1 \\ 0 & , \quad M_a < -1 \end{cases}$$

$$M_a^- = \begin{cases} 0 & , \quad M_a > 1 \\ -(M_a - 1)^2 / 4 & , \quad |M_a| \leq 1 \\ M_a & , \quad M_a < -1 \end{cases}$$

Thus,

$$\mathbf{F}_{\pm}^a = \rho a M_a^{\pm} \begin{bmatrix} 1 \\ u \\ \tilde{H} \end{bmatrix}. \quad (3.19)$$

Splitting the pressure term thereby leads to the splitting of the pressure flux:

$$P = P^+ + P^-, \quad (3.20)$$

where,

$$P^+ = \begin{cases} P & , \quad M_a > 1 \\ \frac{P(1+M_a)}{2} & , \quad |M_a| \leq 1 \\ 0 & , \quad M_a < -1 \end{cases} \quad (3.21a)$$

$$P^- = \begin{cases} 0 & , \quad M_a > 1 \\ \frac{P(1-M_a)}{2} & , \quad |M_a| \leq 1 \\ P & , \quad M_a < -1 \end{cases} \quad (3.21b)$$

Thus, there is

$$\mathbf{F}_{\pm}^P = P^{\pm} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}. \quad (3.22)$$

In summary,

$$\mathbf{F} = \mathbf{F}^+ + \mathbf{F}^- \quad , \quad \mathbf{F}^{\pm} = \mathbf{F}_{\pm}^a + \mathbf{F}_{\pm}^P. \quad (3.23)$$

Remark 3.8. *The van Leer splitting method and the Liou-Stenffen splitting method (a class of AUSM methods) share a consistent approach to the splitting of the Mach number M_a .*

4. ROE AVERAGE

For control equations of a system, the limiters introduced typically need to be used in conjunction with local characteristic decomposition. For variable-coefficient equations or nonlinear equations, local freezing techniques are required to determine the characteristic transformation matrix before performing local characteristic decomposition. A simple approach is naturally to take the arithmetic average of the left and right states (in 1D) or the internal and external states of the cell (in 2D), but a more effective method, or one that is more in line with the physical process, is the Roe average. The approach of Roe averaging is as follows: by utilizing the constant states of the left and right functions, \mathbf{U}^L and \mathbf{U}^R , a reasonable constant matrix $Q(\mathbf{U}^L, \mathbf{U}^R)$ is constructed to approximate the original $A(\mathbf{U})$. This transformation simplifies the complex nonlinear problem into a linear one. Roe [40] achieved this construction by ensuring that $Q(\mathbf{U}_{i+\frac{1}{2}}^L, \mathbf{U}_{i+\frac{1}{2}}^R)$ satisfies

$$\mathbf{F}(\mathbf{U}_{i+\frac{1}{2}}^R) - \mathbf{F}(\mathbf{U}_{i+\frac{1}{2}}^L) = Q(\mathbf{U}_{i+\frac{1}{2}}^L, \mathbf{U}_{i+\frac{1}{2}}^R) (\mathbf{U}_{i+\frac{1}{2}}^R - \mathbf{U}_{i+\frac{1}{2}}^L), \quad (4.1)$$

thereby completing the construction. (Here, $\mathbf{F}(\mathbf{U})$ is flux function in hyperbolic systems.) In the Roe scheme, the matrix $Q(\mathbf{U}_{i+\frac{1}{2}}^L, \mathbf{U}_{i+\frac{1}{2}}^R)$ is a linear approximation of the Jacobian matrix A at a certain state $\mathbf{U}_{i+\frac{1}{2}}^*$. The state $\mathbf{U}_{i+\frac{1}{2}}^*$ is referred to as the Roe average of $\mathbf{U}_{i+\frac{1}{2}}^L$ and $\mathbf{U}_{i+\frac{1}{2}}^R$.

In the subsequent sections, when employing characteristic reconstruction in the FVS-RKDG scheme for the hyperbolic systems, Roe averaging is used to achieve local freezing. Therefore, in this section, we provide the Roe averaging formulations for the compressible Euler equations in 1D and 2D/3D as examples.

The Roe averaging scheme for the 1D-Euler equations is given by:

$$\left. \begin{aligned} \sqrt{\bar{\rho}^{Roe}} &= (\sqrt{\rho_L} + \sqrt{\rho_R}) / 2, \\ \sqrt{\bar{\rho}^{Roe}} \cdot \bar{u}^{Roe} &= (\sqrt{\rho_L} u_L + \sqrt{\rho_R} u_R) / 2, \\ \sqrt{\bar{\rho}^{Roe}} \cdot \bar{H}^{Roe} &= (\sqrt{\rho_L} \tilde{H}_L + \sqrt{\rho_R} \tilde{H}_R) / 2. \end{aligned} \right\} \quad (4.2)$$

Note: \tilde{H} is the specific enthalpy.

The pressure \bar{P}^{Roe} , speed of sound \bar{a}^{Roe} , and total energy \bar{E}^{Roe} are computed indirectly from $\bar{\rho}^{Roe}$, \bar{u}^{Roe} , and \bar{H}^{Roe} as follows:

$$\left. \begin{aligned} \bar{P}^{Roe} &= \frac{\gamma - 1}{\gamma} \left(\bar{\rho}^{Roe} \bar{H}^{Roe} - \frac{1}{2} (\bar{\rho}^{Roe})^2 \right), \\ (\bar{a}^{Roe})^2 &= (\gamma - 1) \left(\bar{H}^{Roe} - \frac{(\bar{u}^{Roe})^2}{2} \right), \\ \bar{E}^{Roe} &= \bar{\rho}^{Roe} \bar{H}^{Roe} - \bar{P}^{Roe}. \end{aligned} \right\} \quad (4.3)$$

Note: \bar{P}^{Roe} , \bar{a}^{Roe} , and \bar{E}^{Roe} are not obtained by directly averaging P , a , and E with density but are computed indirectly.

The Roe matrix A^{Roe} is constructed as:

$$\left. \begin{aligned} \bar{\mathbf{U}}^{Roe} &= \left(\bar{\rho}^{Roe}, (\bar{\rho}^{Roe} \cdot \bar{u}^{Roe}), \bar{E}^{Roe} \right)^T, \\ A^{Roe} &= A(\bar{\mathbf{U}}^{Roe}) = R(\bar{\mathbf{U}}^{Roe}) \cdot \Lambda(\bar{\mathbf{U}}^{Roe}) \cdot L(\bar{\mathbf{U}}^{Roe}) = \bar{R}^{Roe} \cdot \bar{\Lambda}^{Roe} \cdot \bar{L}^{Roe}, \\ |A^{Roe}| &= \bar{R}^{Roe} \cdot |\bar{\Lambda}^{Roe}| \cdot \bar{L}^{Roe}, \quad |\bar{\Lambda}^{Roe}| = \text{diag} \left(|\bar{\lambda}_1^{Roe}|, |\bar{\lambda}_2^{Roe}|, |\bar{\lambda}_3^{Roe}| \right). \end{aligned} \right\} \quad (4.4)$$

For the 2D/3D case, the following additional formulas are included:

$$\left. \begin{aligned} \bar{v}^{Roe} &= \frac{(\sqrt{\rho_L} v_L + \sqrt{\rho_R} v_R)}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \\ \bar{w}^{Roe} &= \frac{(\sqrt{\rho_L} w_L + \sqrt{\rho_R} w_R)}{\sqrt{\rho_L} + \sqrt{\rho_R}}, \\ \bar{\mathbf{U}}^{Roe} &= \left(\bar{\rho}^{Roe}, (\bar{\rho}^{Roe} \cdot \bar{u}^{Roe}), (\bar{\rho}^{Roe} \cdot \bar{v}^{Roe}), (\bar{\rho}^{Roe} \cdot \bar{w}^{Roe}), \bar{E}^{Roe} \right)^T. \end{aligned} \right\} \quad (4.5)$$

Remark 4.1. *In the 2D case, the Roe averaging scheme for the compressible Euler equations presented in this paper is based on the normal flux, that is, the normal velocity $q_n = u \cdot n_x + v \cdot n_y$ is introduced. For details, please refer to Section 6.2 below.*

Remark 4.2. *The specific form of the Roe average may vary for different fluid motion models. For instance, in the case of the shallow water wave equations, the Roe averaging scheme can be referred to in [41]*

5. DG BASED ON FLUX VECTOR SPLITTING IN ONE DIMENSION (1D-FVS-DG)

This section will present the flux vector splitting scheme for one-dimensional hyperbolic conservation systems within the DG framework. Flux vector splitting itself does not introduce any error; therefore, one can first perform a functional split on the flux term in the original control equation and then define the weak solution through variation. Based on the physical interpretation of the split sub-fluxes and considering the characteristic line theory and the stability of the discrete scheme, the corresponding upwind-type numerical flux format can be derived.

Consider the one-dimensional hyperbolic conservation system

$$\partial_t(\mathbf{U}) + \partial_x(\mathbf{F}(\mathbf{U})) = 0. \quad (5.1)$$

We perform the following splitting on the flux vector \mathbf{F} :

$$\mathbf{F}(\mathbf{U}) = \mathbf{F}^+(\mathbf{U}) + \mathbf{F}^-(\mathbf{U}), \quad (5.2)$$

where \mathbf{F}^+ denotes the positive flux, which is the flux carried by characteristic waves propagating in the positive direction of the one-dimensional coordinate system; \mathbf{F}^- denotes the negative flux, which is the flux carried by characteristic waves propagating in the negative direction of the one-dimensional coordinate system. The construction methods for \mathbf{F}^+ and \mathbf{F}^- are discussed in Section 3.

Thus, we have

$$\partial_x(\mathbf{F}(\mathbf{U})) = \partial_x(\mathbf{F}^+(\mathbf{U})) + \partial_x(\mathbf{F}^-(\mathbf{U})), \quad (5.3)$$

$$\partial_t(\mathbf{U}) + \partial_x(\mathbf{F}^+(\mathbf{U})) + \partial_x(\mathbf{F}^-(\mathbf{U})) = 0. \quad (5.4)$$

Multiplying both sides of equation (5.4) by an arbitrary test function $\mathbf{V} \in \mathbb{V}_h^K$ and integrating by parts over the cell I_i yields

$$\begin{aligned} & \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(I_i)} + \langle \partial_x(\mathbf{F}^+(\mathbf{U})), \mathbf{V} \rangle_{L^2(I_i)} + \langle \partial_x(\mathbf{F}^-(\mathbf{U})), \mathbf{V} \rangle_{L^2(I_i)} \\ &= \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(I_i)} - \langle \mathbf{F}^+(\mathbf{U}), \partial_x(\mathbf{V}) \rangle_{L^2(I_i)} - \langle \mathbf{F}^-(\mathbf{U}), \partial_x(\mathbf{V}) \rangle_{L^2(I_i)} + (\mathbf{F}^+ \odot \mathbf{V})_{i-1/2}^{i+1/2} + (\mathbf{F}^- \odot \mathbf{V})_{i-1/2}^{i+1/2} \\ &= \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(I_i)} - \langle \mathbf{F}^+(\mathbf{U}) + \mathbf{F}^-(\mathbf{U}), \partial_x(\mathbf{V}) \rangle_{L^2(I_i)} + (\mathbf{F}^+ \odot \mathbf{V})_{i-1/2}^{i+1/2} + (\mathbf{F}^- \odot \mathbf{V})_{i-1/2}^{i+1/2} \\ &= \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(I_i)} - \langle \mathbf{F}(\mathbf{U}), \partial_x(\mathbf{V}) \rangle_{L^2(I_i)} + (\mathbf{F}^+ \odot \mathbf{V})_{i-1/2}^{i+1/2} + (\mathbf{F}^- \odot \mathbf{V})_{i-1/2}^{i+1/2} = 0. \end{aligned} \quad (5.5)$$

Thus, the definition of the DG weak solution \mathbf{U}_h based on flux vector splitting can be given:

Definition 5.1 (Weak Solution \mathbf{U}_h for 1D-FVS-DG). *If $\mathbf{U}_h \in \mathbb{V}_h^K$ and for all $\mathbf{V}_h \in \mathbb{V}_h^K$, it satisfies*

$$\begin{aligned} & \langle \partial_t(\mathbf{U}_h), \mathbf{V}_h \rangle_{L^2(I_i)} - \langle \mathbf{F}(\mathbf{U}_h), \partial_x(\mathbf{V}_h) \rangle_{L^2(I_i)} \\ &+ \left[\left(\widehat{\mathbf{F}}^+_{i+1/2} \odot (\mathbf{V}_h^L)_{i+1/2} \right) - \left(\widehat{\mathbf{F}}^+_{i-1/2} \odot (\mathbf{V}_h^R)_{i-1/2} \right) \right] \\ &+ \left[\left(\widehat{\mathbf{F}}^-_{i+1/2} \odot (\mathbf{V}_h^L)_{i+1/2} \right) - \left(\widehat{\mathbf{F}}^-_{i-1/2} \odot (\mathbf{V}_h^R)_{i-1/2} \right) \right] = 0, \end{aligned} \quad (5.6)$$

where $\widehat{\mathbf{F}}^\pm$ are the numerical fluxes, then \mathbf{U}_h is referred to as the weak solution of the original equation (5.1) in the sense of FVS-DG.

Let

$$\widehat{\mathbf{F}}_{i+1/2} := \widehat{\mathbf{F}}^+_{i+1/2} + \widehat{\mathbf{F}}^-_{i+1/2}, \quad (5.7)$$

then we have

$$\langle \partial_t(\mathbf{U}_h), \mathbf{V}_h \rangle_{L^2(I_i)} - \langle \mathbf{F}(\mathbf{U}_h), \partial_x(\mathbf{V}_h) \rangle_{L^2(I_i)} + \left(\widehat{\mathbf{F}}_{i+1/2} \odot (\mathbf{V}_h^L)_{i+1/2} \right) - \left(\widehat{\mathbf{F}}_{i-1/2} \odot (\mathbf{V}_h^R)_{i-1/2} \right) = 0. \quad (5.8)$$

In actual calculations, after obtaining $\widehat{\mathbf{F}}^+$ and $\widehat{\mathbf{F}}^-$ at the same interface, the two are superimposed scalarly to obtain $\widehat{\mathbf{F}}$, which is then substituted into equation (5.8) for computation.

Based on the characteristic line theory of hyperbolic equations and the stability of the discrete scheme, an upwind numerical flux can be employed, which takes the interface state on the reverse side of the characteristic propagation direction as the numerical state. This numerical state can be split into two numerical sub-fluxes with opposite directions, and the sub-flux that is consistent with the characteristic propagation direction under consideration is taken as the final numerical flux for this interface in that characteristic propagation direction. From the definitions of \mathbf{F}^+ and \mathbf{F}^- , the following can be immediately obtained:

$$\widehat{\mathbf{F}}_{i+1/2}^+ = \mathbf{F}_{i+1/2}^{L,+}, \quad \widehat{\mathbf{F}}_{i-1/2}^+ = \mathbf{F}_{i-1/2}^{L,+}, \quad (5.9a)$$

$$\widehat{\mathbf{F}}_{i+1/2}^- = \mathbf{F}_{i+1/2}^{R,-}, \quad \widehat{\mathbf{F}}_{i-1/2}^- = \mathbf{F}_{i-1/2}^{R,-}. \quad (5.9b)$$

Remark 5.1 (DG Based on Jacobian Matrix Eigenvalues Splitting). *In the Steger-Warming and Lax-Friedrichs splitting schemes, we have $\mathbf{F}^+ = A^+ \mathbf{U}$ and $\mathbf{F}^- = A^- \mathbf{U}$, so we obtain*

$$\widehat{\mathbf{F}}_{i+1/2}^+ = \mathbf{A}^{L,+}(\mathbf{U}_{i+1/2}^L) \mathbf{U}_{i+1/2}^L, \quad \widehat{\mathbf{F}}_{i-1/2}^+ = \mathbf{A}^{L,+}(\mathbf{U}_{i-1/2}^L) \mathbf{U}_{i-1/2}^L, \quad (5.10a)$$

$$\widehat{\mathbf{F}}_{i+1/2}^- = \mathbf{A}^{R,-}(\mathbf{U}_{i+1/2}^R) \mathbf{U}_{i+1/2}^R, \quad \widehat{\mathbf{F}}_{i-1/2}^- = \mathbf{A}^{R,-}(\mathbf{U}_{i-1/2}^R) \mathbf{U}_{i-1/2}^R. \quad (5.10b)$$

6. DG BASED ON FLUX VECTOR SPLITTING IN TWO-DIMENSION (2D-FVS-DG)

Consider the two-dimensional hyperbolic conservation system

$$\partial_t(\mathbf{U}) + \partial_x(\mathbf{F}(\mathbf{U})) + \partial_y(\mathbf{G}(\mathbf{U})) = 0. \quad (6.1)$$

Let

$$\mathbb{H} = (\mathbf{F}, \mathbf{G}), \quad (6.2)$$

the original equation can be written as

$$\partial_t(\mathbf{U}) + \nabla \cdot \mathbb{H} = 0. \quad (6.3)$$

Now, introduce the normal flux of the two-dimensional hyperbolic conservation system as

$$\mathcal{F}_\nu := \mathbf{n} \cdot \mathbb{H} \quad (6.4)$$

that is

$$\mathcal{F}_\nu = n_x \cdot \mathbf{F} + n_y \cdot \mathbf{G}. \quad (6.5)$$

Remark 6.1. *For high-dimensional problems, the DG scheme cannot discretize space dimension by dimension like FD or FV schemes, especially on triangular meshes, where the concept of “dimension-by-dimension discretization” is invalid. Therefore, we introduce the outward normal flux on the cell interfaces, transforming the multi-dimensional problem into several one-dimensional problems along the outward normal directions of the mesh interfaces (the number of normal one-dimensional problems depends on the number of numerical integration points arranged on the interfaces).*

6.1. DG Based on Flux Vector Splitting in Two-dimension. In the two-dimensional hyperbolic conservation system, there are flux functions \mathbf{F} and \mathbf{G} in the x and y directions, respectively. By using the outward normal vector on the cell interfaces, \mathbf{F} and \mathbf{G} are combined into a single flux function, namely the normal flux \mathcal{F}_ν . On the basis of the normal flux function \mathcal{F}_ν , the Jacobian matrix eigenvalue splitting or Mach number splitting is achieved.

Consider the two-dimensional hyperbolic conservation system

$$\partial_t(\mathbf{U}) + \partial_x(\mathbf{F}(\mathbf{U})) + \partial_y(\mathbf{G}(\mathbf{U})) = 0,$$

following the one-dimensional case, split the flux vectors in each dimension (\mathbf{F} is the flux vector in the x direction, \mathbf{G} is the flux vector in the y direction):

$$\mathbf{F} = \mathbf{F}^+ + \mathbf{F}^- \quad , \quad \mathbf{G} = \mathbf{G}^+ + \mathbf{G}^- . \quad (6.6)$$

\mathbf{F}^+ represents the flux component in the x dimension along the $+x$ direction; \mathbf{F}^- represents the flux component in the x dimension along the $-x$ direction; \mathbf{G}^+ represents the flux component in the y dimension along the $+y$ direction; \mathbf{G}^- represents the flux component in the y dimension along the $-y$ direction. The construction methods for $\mathbf{F}^\pm, \mathbf{G}^\pm$ are discussed in Section 3.

Let

$$\mathbb{H}^+ := (\mathbf{F}^+, \mathbf{G}^+)^\text{T} \quad , \quad \mathbb{H}^- := (\mathbf{F}^-, \mathbf{G}^-)^\text{T}, \quad (6.7)$$

hence

$$\mathbb{H} = \mathbb{H}^+ + \mathbb{H}^-, \quad (6.8)$$

then

$$\partial_t(\mathbf{U}) + \nabla \cdot \mathbb{H}^+ + \nabla \cdot \mathbb{H}^- = 0. \quad (6.9)$$

Multiply both sides of equation (6.9) by an arbitrary test function $\mathbf{V} \in \mathbb{V}_h^K$, and integrate by parts over Ω_i to obtain

$$\begin{aligned} & \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(\Omega_i)} + \langle \nabla \cdot \mathbb{H}^+(\mathbf{U}), \mathbf{V} \rangle_{L^2(\Omega_i)} + \langle \nabla \cdot \mathbb{H}^-(\mathbf{U}), \mathbf{V} \rangle_{L^2(\Omega_i)} \\ &= \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}^+(\mathbf{U}) : \nabla \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}^-(\mathbf{U}) : \nabla \mathbf{V} \rangle_{L^2(\Omega_i)} + \int_{\Omega_i} \nabla \cdot (\mathbb{H}^+(\mathbf{U}) \odot \mathbf{V}) d\mathbf{X} + \int_{\Omega_i} \nabla \cdot (\mathbb{H}^-(\mathbf{U}) \odot \mathbf{V}) d\mathbf{X} \\ &= \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}^+(\mathbf{U}) : \nabla \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}^-(\mathbf{U}) : \nabla \mathbf{V} \rangle_{L^2(\Omega_i)} + \int_{\partial\Omega_i} (\mathbb{H}^+(\mathbf{U}) \odot \mathbf{V}) \cdot \mathbf{n} dl + \int_{\partial\Omega_i} (\mathbb{H}^-(\mathbf{U}) \odot \mathbf{V}) \cdot \mathbf{n} dl \\ &= \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}^+(\mathbf{U}) : \nabla \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}^-(\mathbf{U}) : \nabla \mathbf{V} \rangle_{L^2(\Omega_i)} + \int_{\partial\Omega_i} (\mathbb{H}^+(\mathbf{U}) \cdot \mathbf{n}) \odot \mathbf{V} dl + \int_{\partial\Omega_i} (\mathbb{H}^-(\mathbf{U}) \cdot \mathbf{n}) \odot \mathbf{V} dl \end{aligned}$$

$$\begin{aligned}
&= \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}^+ + \mathbb{H}^-(\mathbf{U}) : \nabla \mathbf{V} \rangle_{L^2(\Omega_i)} + \int_{\partial\Omega_i} (\mathbb{H}^+(\mathbf{U}) \cdot \mathbf{n}) \odot \mathbf{V} dl + \int_{\partial\Omega_i} (\mathbb{H}^-(\mathbf{U}) \cdot \mathbf{n}) \odot \mathbf{V} dl \\
&= \langle \partial_t(\mathbf{U}), \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}(\mathbf{U}) : \nabla \mathbf{V} \rangle_{L^2(\Omega_i)} + \int_{\partial\Omega_i} (\mathbb{H}^+(\mathbf{U}) \cdot \mathbf{n}) \odot \mathbf{V} dl + \int_{\partial\Omega_i} (\mathbb{H}^-(\mathbf{U}) \cdot \mathbf{n}) \odot \mathbf{V} dl = 0. \quad (6.10)
\end{aligned}$$

Let

$$\mathcal{F}_\nu^+ := \mathbb{H}^+ \cdot \mathbf{n} \quad , \quad \mathcal{F}_\nu^- := \mathbb{H}^- \cdot \mathbf{n}. \quad (6.11)$$

Note that the outward normal vector has been incorporated into the numerical fluxes.

Obviously,

$$\mathcal{F}_\nu^+ = n_x \mathbf{F}^+ + n_y \mathbf{G}^+ \quad , \quad \mathcal{F}_\nu^- = n_x \mathbf{F}^- + n_y \mathbf{G}^-, \quad (6.12)$$

and it holds that

$$\mathcal{F}_\nu = \mathcal{F}_\nu^+ + \mathcal{F}_\nu^-. \quad (6.13)$$

Remark 6.2. Given the physical significance of $\mathbf{F}^\pm, \mathbf{G}^\pm$, it is understood that \mathcal{F}_ν^+ represents the flux flowing from the cell interior to the cell exterior in the direction of the unit outward normal vector \mathbf{n} . Conversely, \mathcal{F}_ν^- represents the flux flowing from the cell exterior to the cell interior in the direction of the unit outward normal vector \mathbf{n} .

Now, the definition of the DG weak solution \mathbf{U}_h based on flux vector splitting is given:

Definition 6.1 (The weak solution of the 2D-FVS-DG). If $\mathbf{U}_h \in \mathbb{V}_h^K$ and for all $\mathbf{V}_h \in \mathbb{V}_h^K$, it satisfies

$$\langle \partial_t(\mathbf{U}_h), \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}(\mathbf{U}_h) : \nabla \mathbf{V}_h \rangle_{L^2(\Omega_i)} + \int_{\partial\Omega_i} \widehat{\mathcal{F}}_\nu^+ \cdot \mathbf{V}_h dl + \int_{\partial\Omega_i} \widehat{\mathcal{F}}_\nu^- \cdot \mathbf{V}_h dl = 0, \quad (6.14)$$

where $\widehat{\mathcal{F}}_\nu^\pm$ are the normal numerical fluxes, then \mathbf{U}_h is referred to as the weak solution of the original equation (6.1) in the sense of FVS-DG.

Let

$$\widehat{\mathcal{F}}_\nu := \widehat{\mathcal{F}}_\nu^+ + \widehat{\mathcal{F}}_\nu^-, \quad (6.15)$$

In actual calculations, the following form is used:

$$\langle \partial_t(\mathbf{U}_h), \mathbf{V} \rangle_{L^2(\Omega_i)} - \langle \mathbb{H}(\mathbf{U}_h) : \nabla \mathbf{V}_h \rangle_{L^2(\Omega_i)} + \int_{\partial\Omega_i} \widehat{\mathcal{F}}_\nu \cdot \mathbf{V}_h dl = 0. \quad (6.16)$$

Similar to the one-dimensional problem, and still due to considerations of characteristic lines and stability, based on the physical meaning of \mathcal{F}_ν^\pm , it can be obtained that

$$\widehat{\mathcal{F}}_\nu^+ = n_x \mathbf{F}^+(\mathbf{U}^{int}) + n_y \mathbf{G}^+(\mathbf{U}^{int}), \quad (6.17a)$$

$$\widehat{\mathcal{F}}_\nu^- = n_x \mathbf{F}^-(\mathbf{U}^{ext}) + n_y \mathbf{G}^-(\mathbf{U}^{ext}). \quad (6.17b)$$

Here, ‘‘int’’ denotes the interior of the cell, and ‘‘ext’’ denotes the exterior of the cell.

6.2. Normal Roe Average Employed by DG in Two-dimension. Roe average can be adopted for local-freezing when local characteristic decomposition discussed in Section 8. Similar to previous approach in Subsection 6.1, Roe average along the outward normal direction on the cell interfaces is utilized to handle multi-dimensional cases. Here, we provides the specific implementation details of the normal Roe averaging.

Firstly, review that the Roe average along the coordinate vectors:

$$\begin{aligned}\sqrt{\bar{\rho}^{Roe}} &= (\sqrt{\rho_{int}} + \sqrt{\rho_{ext}}) / 2, \\ \sqrt{\bar{\rho}^{Roe}} \cdot \bar{u}^{Roe} &= (\sqrt{\rho_{int}} \cdot u_{int} + \sqrt{\rho_{ext}} \cdot u_{ext}) / 2, \\ \sqrt{\bar{\rho}^{Roe}} \cdot \bar{v}^{Roe} &= (\sqrt{\rho_{int}} \cdot v_{int} + \sqrt{\rho_{ext}} \cdot v_{ext}) / 2, \\ \sqrt{\bar{\rho}^{Roe}} \cdot \bar{H}^{Roe} &= (\sqrt{\rho_{int}} \cdot \tilde{H}_{int} + \sqrt{\rho_{ext}} \cdot \tilde{H}_{ext}) / 2.\end{aligned}$$

Note: \tilde{H} is the specific enthalpy.

$$\begin{aligned}\bar{P}^{Roe} &= \frac{\gamma - 1}{\gamma} \left(\bar{\rho}^{Roe} \bar{H}^{Roe} - \frac{1}{2} (\bar{\rho}^{Roe})^2 \right), \\ (\bar{a}^{Roe})^2 &= (\gamma - 1) \left(\bar{H}^{Roe} - \frac{1}{2} (\bar{u}^{Roe})^2 - \frac{1}{2} (\bar{v}^{Roe})^2 \right), \\ \bar{E}^{Roe} &= \bar{\rho}^{Roe} \bar{H}^{Roe} - \bar{P}^{Roe}.\end{aligned}$$

Note that \bar{P}^{Roe} , \bar{a}^{Roe} , and \bar{E}^{Roe} are not obtained by directly averaging P , a , and E with density but are computed indirectly.

Then, the normal Jacobian and its eigenstructure are substituted as follows:

$$\left. \begin{aligned}\bar{q}_n^{Roe} &= \bar{u}^{Roe} n_x + \bar{v}^{Roe} n_y, \\ \bar{\lambda}_1^{Roe} = \bar{\lambda}_2^{Roe} = \bar{q}_n^{Roe}, \quad \bar{\lambda}_3^{Roe} = \bar{q}_n^{Roe} - \bar{a}^{Roe}, \quad \bar{\lambda}_4^{Roe} = \bar{q}_n^{Roe} + \bar{a}^{Roe}, \\ \bar{A}_n^{Roe} = A_n(\bar{\mathbf{U}}^{Roe}) = R_n(\bar{\mathbf{U}}^{Roe}) \cdot \Lambda_n(\bar{\mathbf{U}}^{Roe}) \cdot L_n(\bar{\mathbf{U}}^{Roe}) = \bar{R}_n^{Roe} \cdot \bar{\Lambda}_n^{Roe} \cdot \bar{L}_n^{Roe}, \\ \left| \bar{A}_n^{Roe} \right| = \bar{R}_n^{Roe} \cdot \left| \bar{\Lambda}_n^{Roe} \right| \cdot \bar{L}_n^{Roe}, \quad \left| \bar{\Lambda}_n^{Roe} \right| = \text{diag} \left(\left| \bar{\lambda}_1^{Roe} \right|, \left| \bar{\lambda}_2^{Roe} \right|, \left| \bar{\lambda}_3^{Roe} \right|, \left| \bar{\lambda}_4^{Roe} \right| \right).\end{aligned}\right\} \quad (6.18)$$

Remark 6.3. The normal Roe flux for the 2D-Euler equations is given by:

$$\hat{\mathbf{F}}_n^{Roe} = \frac{1}{2} (\mathbf{F}_n^{ext} + \mathbf{F}_n^{int}) - \frac{1}{2} |\bar{A}_n| (\mathbf{U}^{ext} - \mathbf{U}^{int}), \quad (6.19)$$

where

$$\mathbf{F}_n^{ext} = \mathbf{F}_n(\mathbf{U}^{ext}) = \mathbf{F}(\mathbf{U}^{ext}) \cdot n_x + \mathbf{G}(\mathbf{U}^{ext}) \cdot n_y, \quad (6.20a)$$

$$\mathbf{F}_n^{int} = \mathbf{F}_n(\mathbf{U}^{int}) = \mathbf{F}(\mathbf{U}^{int}) \cdot n_x + \mathbf{G}(\mathbf{U}^{int}) \cdot n_y. \quad (6.20b)$$

Roe flux is a kind of Riemann solvers, which is different from our FVS-based flux schemes introduced in Section 5 and Subsection 6.1.

7. A NOVEL TVB(D)-MINMOD LIMITER FOR NUMERICAL PSEUDO-OSCILLATION TREATMENT

This section will introduce the discontinuity indicators firstly and then focuses on some improvements we have made to the TVB(D)-minmod limiter.

7.1. Commonly Used Discontinuity Indicators. In this subsection, we introduce two types of discontinuity indicators used in the subsequent numerical experiments, namely the TVB(D)-minmod discontinuity indicator and the KXRFC discontinuity indicator. The TVB(D)-minmod discontinuity indicator is applicable to 1D/2D scalar equations as well as 1D/2D systems. We employ this type of limiter solely on rectangular (arbitrary quadrilateral) meshes; The KXRFC discontinuity indicator is used exclusively on triangular meshes for the control equations of fluid systems. It is important to note that when discontinuity indicators are applied to system control equations, each component's troubled cells should be inspected individually.

7.1.1. TVB(D)-minmod Discontinuity Indicator. The TVB(D)-minmod limiter is a slope limiter that compares the jump between the average of the troubled cell and the averages of the neighboring cells with the jump between the average of the troubled cell and the average of the boundary integrals of the troubled cell. The smaller of these absolute values is taken as the revised average of the boundary integrals of the troubled cell, thereby reducing the average slope of the approximate solution curve within the troubled cell. When used as a discontinuity indicator, the working principle of the TVB(D)-minmod limiter is as follows:

- 1D-TV(B(D))-minmod Discontinuity Indicator

The deviation of the cell left boundary value from the cell average (left deviation) is given by

$$\widehat{U}_i := \bar{U}_i - U_{i-\frac{1}{2}}^+,$$

hence,

$$U_{i-\frac{1}{2}}^+ = \bar{U}_i - \widehat{U}_i.$$

The deviation of the cell right boundary value from the cell average (right deviation) is

$$\widetilde{U}_i := U_{i+\frac{1}{2}}^- - \bar{U}_i,$$

hence,

$$U_{i+\frac{1}{2}}^- = \bar{U}_i + \widetilde{U}_i.$$

The forward difference between neighboring cell averages is

$$\Delta_- = \bar{U}_i - \bar{U}_{i-1}.$$

The backward difference between neighboring cell averages is

$$\Delta_+ = \bar{U}_{i+1} - \bar{U}_i.$$

The modified left deviation is calculated as

$$\widehat{U}_i^{mod} = \text{minmod} \left(\widehat{U}_i, \Delta_-, \Delta_+ \right).$$

The modified left boundary value is then

$$U_{i-\frac{1}{2}}^{+,mod} = \bar{U}_i - \widehat{U}_i^{mod}.$$

The modified right deviation is calculated as

$$\widetilde{U}_i^{mod} = \text{minmod} \left(\widetilde{U}_i, \Delta_-, \Delta_+ \right).$$

The modified right boundary value is then

$$U_{i+\frac{1}{2}}^{-,mod} = \bar{U}_i + \widetilde{U}_i^{mod}.$$

If the modification is not equal to the original, it indicates that the cell is a troubled cell, that is:

$$\text{if } \widehat{U}_i^{mod} \neq \widehat{U}_i \text{ or } \widetilde{U}_i^{mod} \neq \widetilde{U}_i \text{ then } I_i \text{ is a troubled cell.}$$

Remark 7.1 (Correct the end-point values of the troubled interval and implement the TVB(D)-minmod limiter). *According to $U_{i-\frac{1}{2}}^{+,mod}$ and $U_{i+\frac{1}{2}}^{-,mod}$, we can correct the DG polynomial on the I_i cell. Executing this correction step can implement the TVB(D)-minmod limiter (if not, then it just work as a discontinuity indicator).*

- 2D-TVB(D)-minmod Discontinuity Indicator

The 2D rectangular cell integral average is given by

$$\bar{U}_{ij} = \bar{U}_{\Omega_{ij}} = \frac{1}{|\Omega_{ij}|} \int_{\Omega_{ij}} u_{ij}(x, y) dx dy.$$

The left boundary (L) integral average is

$$\bar{U}_{ij}^L = \bar{U}_{\partial\Omega_{ij}^L} = \frac{1}{|\partial\Omega_{ij}^L|} \int_{\partial\Omega_{ij}^L} u_{ij}(x, y) dl.$$

The right boundary (R) integral average is

$$\bar{U}_{ij}^R = \bar{U}_{\partial\Omega_{ij}^R} = \frac{1}{|\partial\Omega_{ij}^R|} \int_{\partial\Omega_{ij}^R} u_{ij}(x, y) dl.$$

The bottom boundary (B) integral average is

$$\bar{U}_{ij}^B = \bar{U}_{\partial\Omega_{ij}^B} = \frac{1}{|\partial\Omega_{ij}^B|} \int_{\partial\Omega_{ij}^B} u_{ij}(x, y) dl.$$

The top boundary (T) integral average is

$$\bar{U}_{ij}^T = \bar{U}_{\partial\Omega_{ij}^T} = \frac{1}{|\partial\Omega_{ij}^T|} \int_{\partial\Omega_{ij}^T} u_{ij}(x, y) dl.$$

The left difference between adjacent cell averages is

$$\Delta_L = \bar{U}_{ij} - \bar{U}_{i-1,j}.$$

The right difference between adjacent cell averages is

$$\Delta_R = \bar{U}_{i+1,j} - \bar{U}_{i,j}.$$

The bottom difference between adjacent cell averages is

$$\Delta_B = \bar{U}_{ij} - \bar{U}_{i,j-1}.$$

The top difference between adjacent cell averages is

$$\Delta_T = \bar{U}_{i,j+1} - \bar{U}_{ij}.$$

The left deviation of the boundary average from the cell average is

$$\hat{U}_{ij}^L = \bar{U}_{ij} - \bar{U}_{ij}^L,$$

hence,

$$\bar{U}_{ij}^L = \bar{U}_{ij} - \hat{U}_{ij}^L.$$

The right deviation of the boundary average from the cell average is

$$\tilde{U}_{ij}^R = \bar{U}_{ij}^R - \bar{U}_{ij},$$

hence,

$$\bar{U}_{ij}^R = \bar{U}_{ij} + \tilde{U}_{ij}^R.$$

The bottom deviation of the boundary average from the cell average is

$$\hat{U}_{ij}^B = \bar{U}_{ij} - \bar{U}_{ij}^B,$$

hence,

$$\bar{U}_{ij}^B = \bar{U}_{ij} - \hat{U}_{ij}^B.$$

The top deviation of the boundary average from the cell average is

$$\tilde{U}_{ij}^T = \bar{U}_{ij}^T - \bar{U}_{ij},$$

hence,

$$\bar{U}_{ij}^T = \bar{U}_{ij} + \tilde{U}_{ij}^T.$$

The modified left deviation is

$$\hat{U}_{ij}^{L,mod} = \text{minmod} \left(\hat{U}_{ij}^L, \Delta_L, \Delta_R \right).$$

The modified right deviation is

$$\tilde{U}_{ij}^{R,mod} = \text{minmod} \left(\tilde{U}_{ij}^R, \Delta_L, \Delta_R \right).$$

The modified bottom deviation is

$$\hat{U}_{ij}^{B,mod} = \text{minmod} \left(\hat{U}_{ij}^B, \Delta_B, \Delta_T \right).$$

The modified top deviation is

$$\tilde{U}_{ij}^{T,mod} = \text{minmod} \left(\tilde{U}_{ij}^T, \Delta_B, \Delta_T \right).$$

The 2D-rectangular mesh TVB(D)-minmod discontinuity indicator is as follows:

$$\text{if } \widehat{U}_{ij}^{L,mod} \neq \widehat{U}_{ij}^L \quad \text{or} \quad \widetilde{U}_{ij}^{R,mod} \neq \widetilde{U}_{ij}^R \quad \text{or} \quad \widehat{U}_{ij}^{B,mod} \neq \widehat{U}_{ij}^B \quad \text{or} \quad \widetilde{U}_{ij}^{T,mod} \neq \widetilde{U}_{ij}^T,$$

then Ω_{ij} is a troubled cell.

Remark 7.2 (Correction of Boundary Integral Averages using the 2D Rectangular Cell TVB(D)-minmod Limiter).

$$\begin{aligned} \overline{U}_{ij}^{L,mod} &= \overline{U}_{ij} + \widehat{U}_{ij}^{L,mod}, \\ \overline{U}_{ij}^{R,mod} &= \overline{U}_{ij} + \widetilde{U}_{ij}^{R,mod}, \\ \overline{U}_{ij}^{B,mod} &= \overline{U}_{ij} + \widehat{U}_{ij}^{B,mod}, \\ \overline{U}_{ij}^{T,mod} &= \overline{U}_{ij} + \widetilde{U}_{ij}^{T,mod}. \end{aligned}$$

Executing this correction step can implement the TVB(D)-minmod limiter (if not, then it just work as a discontinuity indicator).

As mentioned in the ‘‘Introduction’’, the TVB(D)-minmod discontinuity indicator is obtained by omitting the slope correction step from the the TVB(D)-minmod limiter. All computational processes of the TVB(D)-minmod discontinuity indicator are naturally also the main steps of the TVB(D)-minmod limiter.

7.1.2. *KXRCF Discontinuity Indicator.* Let the inflow boundary of the solution be denoted as $\partial\Omega^-$. The *KXRCF* indicator is defined as follows:

$$J_\Omega = \frac{\left| \int_{\partial\Omega^-} (u|_\Omega - u|_{\Omega_{nb}}) dS_x \right|}{h^{\frac{K+1}{2}} |\partial\Omega^-| \cdot \|u|_\Omega\|_{L^\infty}}.$$

If $J_\Omega > 1$, then Ω is identified as a troubled cell. Here, Ω_{nb} refers to the neighboring cell sharing the inflow boundary $\partial\Omega^-$ (not all neighboring cells), K is the highest degree of the piecewise polynomial, and h is the radius of the cell.

- 1D-KXRCF Discontinuity Indicator: For the 1D fluid systems, if $v \cdot \mathbf{n} < 0$, it is considered an inflow boundary; otherwise, it is an outflow boundary. Here \mathbf{n} is the unit outward normal vector; For 1D scalar case, we define v , taking its value from inside the cell I_j as $f'(u)$ and take u as the indicator variable.

- 2D-KXRCF Discontinuity Indicator: For the 2D fluid systems, if $(v_x, v_y) \cdot \mathbf{n} < 0$, it is classified as an inflow boundary; otherwise, it is an outflow boundary. Here \mathbf{n} is the unit outward normal vector; For 2D scalar case, we define v_1, v_2 , taking their values from inside the cell Ω_j as $f'(u), g'(u)$ respectively and still take u as the indicator variable.

7.2. Constrained Optimization-based TVB(D)-minmod Limiter Compatible with High-Order Polynomial Approximation. In the context of high-order P^K -polynomial approximation ($K \geq 3$), the original TVB(D)-minmod limiter results in indeterminate correction equations with non-unique solutions. *Direct use of the least squares solution for the above indeterminate correction equations will result in noticeable numerical oscillations*

when using higher-order polynomial approximation (Please refer to Figure 1). The primary goal of employing the TVB(D)-minmod limiter is to suppress numerical oscillations. To better achieve this goal, we select the solution from the set of indeterminate solutions that is “the smoothest,” serving as the DG solution modified by the TVB(D)-minmod limiter. To address the ill-posed problems between the original TVB(D)-minmod limiter and high-order polynomial approximation, we introduce the smoothness measurement function IS from WENO reconstruction as the objective function to construct an optimization problem. Additionally, while suppressing numerical oscillations, it is essential to maintain the solution’s high accuracy. The approach taken in [1] is to introduce an L^2 -error term to characterize the difference between the corrected solution and the original high-order scheme. By minimizing the “ L^2 -error between the corrected solution and the original high-order scheme,” we ensure the accuracy of the solution modified by the limiter. Consequently, a natural idea is to combine the aforementioned constraints to form a bi-objective optimization problem that balances oscillation suppression and precision.

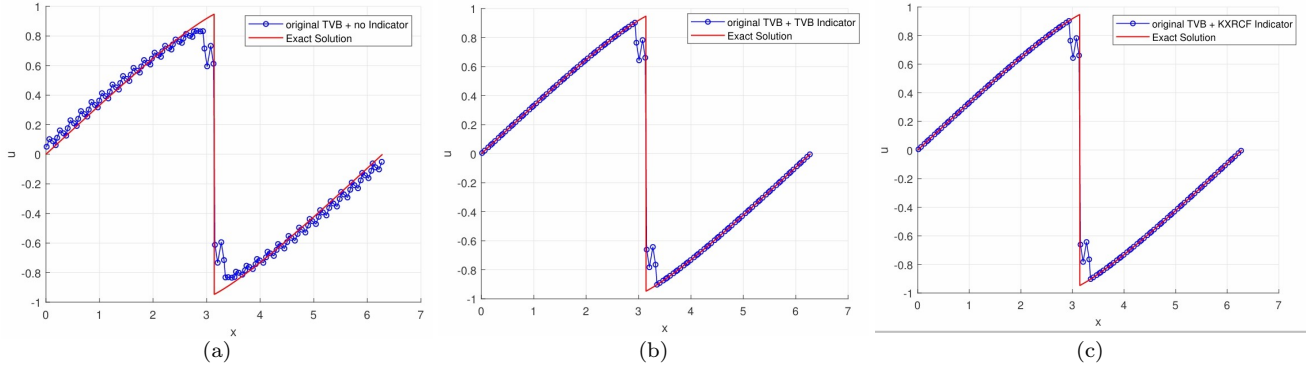


FIGURE 1. 1D-Burgers’ equation $u_t + (\frac{1}{2}u^2)_x = 0$ with initial condition $u_0(x) = \sin(x)$. The simulation is performed up to time $t = 2.0$. P^3 -polynomial approximations and uniquely spaced 32 cells. TVB parameter $M = 1$. The v for KXRCF indicator in scalar case is taken as $u_{i\pm 1/2}$ from inside the cell I_i : (a) classical TVB-minmod limiter without discontinuity indicator; (b) classical TVB-minmod limiter with TVB discontinuity indicator; (c) classical TVB-minmod limiter with KXRCF discontinuity indicator

7.2.1. Smoothness Factor IS Constrained TVB(D)-minmod Limiter: One-Dimensional Case.

The one-dimensional smoothness measurement function is defined as:

$$IS(Q_i^K(x)) = \sum_{l=1}^{l=K} \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta x_i^{2l-1} \left(\frac{\partial^l Q_i^K(x)}{\partial x^l} \right)^2 dx. \quad (7.1)$$

Let the original DG weak solution on interval I_i without the limiter be denoted as u_i^{old} , with modal coefficients given by $\mathbf{a}^{i,old} = [\alpha_0^{i,old}, \alpha_1^{i,old}, \alpha_2^{i,old}, \alpha_3^{i,old}, \dots, \alpha_{K-1}^{i,old}, \alpha_K^{i,old}]^T$. An optimization problem for the DG weak solution u_i on interval I_i is constructed as follows:

$$\begin{aligned} & \min IS(u_i) \\ \text{s.t.} \quad & \begin{cases} \frac{1}{|I_i|} \int_{I_i} u_i(x) dx = \bar{U}_i \\ u_i(x_{i-1/2}) = U_{i-1/2}^{+,mod} \\ u_i(x_{i+1/2}) = U_{i+1/2}^{-,mod} \end{cases} \end{aligned} \quad (7.2)$$

Let the solution to problem (7.2) be u_i^{mod} , with modal coefficients

$$\mathbf{a}^{i,mod} = \left[\alpha_0^{i,mod}, \alpha_1^{i,mod}, \alpha_2^{i,mod}, \alpha_3^{i,mod}, \dots, \alpha_{K-1}^{i,mod}, \alpha_K^{i,mod} \right]^T, \quad (7.3)$$

such that

$$u_i^{mod} = \sum_{\ell=0}^{\ell=K} \alpha_\ell^{i,mod} \phi_\ell(I_i). \quad (7.4)$$

The integral mean remains unchanged, i.e., $\alpha_0^{i,mod} = \alpha_0^{i,old}$, which allows further simplification to

$$\begin{aligned} & \min IS(u_i) \\ \text{s.t.} \quad & \begin{cases} u_i(x_{i-1/2}) = U_{i-1/2}^{+,mod} \\ u_i(x_{i+1/2}) = U_{i+1/2}^{-,mod} \end{cases} \end{aligned} \quad (7.5)$$

For the sake of brevity in notation, we omit the cell indices for the modal coefficients and the basis functions. It is known that

$$\begin{aligned} u_i' &= \alpha_1 \phi_1'(x) + \alpha_2 \phi_2'(x) + \alpha_3 \phi_3'(x) + \dots + \alpha_K \phi_K'(x), \\ u_i'' &= \alpha_2 \phi_2''(x) + \alpha_3 \phi_3''(x) + \dots + \alpha_K \phi_K''(x), \\ u_i^{(3)} &= \alpha_3 \phi_3^{(3)}(x) + \dots + \alpha_K \phi_K^{(3)}(x), \\ & \vdots \\ u_i^{(K-1)} &= \alpha_{K-1} \phi_{K-1}^{(K-1)}(x) + \alpha_K \phi_K^{(K-1)}(x), \\ u_i^{(K)} &= \alpha_K \phi_K^{(K)}(x). \end{aligned}$$

Define

$$M_{jk}^d := \Delta x^{2d-1} \int_{I_i} \frac{\partial^d \phi_j}{\partial x^d} \cdot \frac{\partial^d \phi_k}{\partial x^d} dx = \Delta x^{2d-1} \left\langle \phi_j^{(d)}, \phi_k^{(d)} \right\rangle_{L^2(I_i)}, \quad (7.6)$$

then the smoothness factor is given by

$$IS_i = \sum_{d=1}^{d=K} \int_{x_{i-1/2}}^{x_{i+1/2}} \Delta x_i^{2d-1} \left(u_i^{(d)} \right)^2 dx = \sum_{d=1}^{d=K} \sum_{j=1}^K \sum_{k=1}^K M_{jk}^d \alpha_j \alpha_k = \sum_{d=1}^{d=K} \left(\sum_{j=1}^K M_{jj}^d \alpha_j^2 + \sum_{j=1}^{K-1} \sum_{k=j+1}^K 2M_{jk}^d \alpha_j \alpha_k \right). \quad (7.7)$$

The corresponding relationship between coefficients is as follows:

$$\begin{bmatrix} \alpha_1^2 & & & & & \\ \alpha_1\alpha_2 & \alpha_2^2 & & & & \\ \alpha_1\alpha_3 & \alpha_2\alpha_3 & \alpha_3^2 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ \alpha_1\alpha_{K-1} & \alpha_2\alpha_{K-1} & \alpha_3\alpha_{K-1} & \cdots & \alpha_{K-1}^2 & \\ \alpha_1\alpha_K & \alpha_2\alpha_K & \alpha_3\alpha_K & \cdots & \alpha_{K-1}\alpha_K & \alpha_K^2 \end{bmatrix}, \quad (7.8a)$$

$$\begin{bmatrix} M_{11}^1 & & & & & \\ 2M_{12}^1 & M_{22}^1 + M_{22}^2 & & & & \\ 2M_{13}^1 & 2(M_{23}^1 + M_{23}^2) & M_{33}^1 + M_{33}^2 + M_{33}^3 & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ 2M_{1K-1}^1 & 2(M_{2K-1}^1 + M_{2K-1}^2) & 2(M_{3K-1}^1 + M_{3K-1}^2 + M_{3K-1}^3) & \cdots & M_{K-1K-1}^1 + M_{K-1K-1}^2 + \cdots + M_{K-1K-1}^{K-1} & \\ 2M_{1K}^1 & 2(M_{2K}^1 + M_{2K}^2) & 2(M_{3K}^1 + M_{3K}^2 + M_{3K}^3) & \cdots & 2(M_{K-1K}^1 + M_{K-1K}^2 + \cdots + M_{K-1K}^{K-1}) & \Delta \end{bmatrix}, \quad (7.8b)$$

$$\Delta = M_{KK}^1 + M_{KK}^2 + \cdots + M_{KK}^K. \quad (7.8c)$$

From this point on, we restore the cell indices. Define

$\mathbb{M}_i :=$

$$\begin{bmatrix} 2M_{11}^1 & & & & & \\ 2M_{12}^1 & 2(M_{22}^1 + M_{22}^2) & & & & \\ 2M_{13}^1 & 2(M_{23}^1 + M_{23}^2) & 2(M_{33}^1 + M_{33}^2 + M_{33}^3) & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ 2M_{1K-1}^1 & 2(M_{2K-1}^1 + M_{2K-1}^2) & 2(M_{3K-1}^1 + M_{3K-1}^2 + M_{3K-1}^3) & \cdots & 2(M_{K-1K-1}^1 + M_{K-1K-1}^2 + \cdots + M_{K-1K-1}^{K-1}) & \\ 2M_{1K}^1 & 2(M_{2K}^1 + M_{2K}^2) & 2(M_{3K}^1 + M_{3K}^2 + M_{3K}^3) & \cdots & 2(M_{K-1K}^1 + M_{K-1K}^2 + \cdots + M_{K-1K}^{K-1}) & 2\Delta \end{bmatrix}. \quad (7.9)$$

Note that each M_{jk}^d of \mathbb{M}_i belongs to the cell I_i . The matrix \mathbb{M}_i is decomposed as follows:

$$\mathbb{M}_i = \mathbb{L}_i + \mathbb{D}_i, \quad \mathbb{D}_i = \text{diag}(\mathbb{M}_i(1, 1), \mathbb{M}_i(2, 2), \mathbb{M}_i(3, 3), \dots, \mathbb{M}_i(K, K)). \quad (7.10)$$

Let

$$\mathbb{U}_i = \mathbb{L}_i^T. \quad (7.11)$$

Define the matrix \mathbb{M}_i^{IS} as

$$\mathbb{M}_i^{IS} := \mathbb{L}_i + \mathbb{D}_i + \mathbb{U}_i. \quad (7.12)$$

Additionally, define

$$\tilde{\mathbf{a}}^{i,mod} := \left[\alpha_1^{i,mod}, \alpha_2^{i,mod}, \alpha_3^{i,mod}, \dots, \alpha_{K-1}^{i,mod}, \alpha_K^{i,mod} \right]^T, \quad (7.13)$$

$$[\Phi]_i := \begin{bmatrix} \phi_1(x_{i-1/2}) & \phi_2(x_{i-1/2}) & \cdots & \phi_K(x_{i-1/2}) \\ \phi_1(x_{i+1/2}) & \phi_2(x_{i+1/2}) & \cdots & \phi_K(x_{i+1/2}) \end{bmatrix}^T, \quad (7.14)$$

$$[\Lambda^*] := [\lambda_1^*, \lambda_2^*]^T. \quad (7.15)$$

To solve the constrained optimization problem (7.5), the Lagrangian is constructed as follows:

$$L_i(\alpha_1^i, \alpha_2^i, \dots, \alpha_K^i, \lambda_1, \lambda_2) = IS_i + \lambda_1 \cdot \left(u_i(x_{i-1/2}) - U_{i-1/2}^{+,mod} \right) + \lambda_2 \cdot \left(u_i(x_{i+1/2}) - U_{i+1/2}^{-,mod} \right). \quad (7.16)$$

The solution to problem (7.5), denoted as $u_i^{mod}(x; \alpha_0^{i,old}, \tilde{\mathbf{a}}^{i,mod})$, should satisfy the saddle-point equation, that is:

$$\nabla L_i = \begin{bmatrix} \partial_{\alpha_1} L_i \\ \partial_{\alpha_2} L_i \\ \vdots \\ \partial_{\alpha_K} L_i \\ \partial_{\lambda_1} L_i \\ \partial_{\lambda_2} L_i \end{bmatrix}_{(\tilde{\mathbf{a}}^{i,mod}, \lambda_1^*, \lambda_2^*)} = \begin{bmatrix} \mathbb{M}_i^{IS} \tilde{\mathbf{a}}^{i,mod} + [\Phi]_i [\Lambda^*] \\ \sum_{k=0}^K \alpha_k^{i,mod} \phi_k(x_{i-1/2}) - U_{i-1/2}^{+,mod} \\ \sum_{k=0}^K \alpha_k^{i,mod} \phi_k(x_{i+1/2}) - U_{i+1/2}^{-,mod} \end{bmatrix} = 0. \quad (7.17)$$

Taking into account that $\alpha_0^{i,mod} = \alpha_0^{i,old}$ is known, the matrix form of the saddle-point equation is given by:

$$\mathbb{A}_i^{IS} \mathbf{X}_i^{IS} = \mathbf{b}_i^{IS}, \quad (7.18)$$

where

$$\mathbb{A}_i^{IS} = \begin{bmatrix} \mathbb{M}_i^{IS} & [\Phi]_i \\ [\Phi]_i^T & \mathbf{0}_{2 \times 2} \end{bmatrix}, \quad (7.19a)$$

$$\mathbf{b}_i^{IS} = \begin{bmatrix} \mathbf{0}_{K \times 1} \\ U_{i-1/2}^{+,mod} - \alpha_0^{i,old} \phi_0^{(I_i)}(x_{i-1/2}) \\ U_{i+1/2}^{-,mod} - \alpha_0^{i,old} \phi_0^{(I_i)}(x_{i+1/2}) \end{bmatrix}, \quad (7.19b)$$

$$\mathbf{X}_i^{IS} = \begin{bmatrix} \tilde{\mathbf{a}}^{i,mod} \\ [\Lambda^*] \end{bmatrix}. \quad (7.19c)$$

Thus, the DG modal coefficients modified by the IS-TVB(D)-minmod limiter can be obtained $\mathbf{a}_i^{mod} = \left[\alpha_0^{i,old}, [\tilde{\mathbf{a}}_i^{mod}]^T \right]^T$.

7.2.2. Smoothness Factor IS Constrained TVB(D)-minmod Limiter: Two-Dimensional Case.

We consider the DG weak solution on the rectangular element $\Omega_{\mathcal{K}}$:

$$U_h(x, y, t) = \sum_{k=0}^K \alpha_k(t) \phi_k(x, y), \quad (x, y) \in \Omega_{\mathcal{K}}. \quad (7.20)$$

Let \mathbf{D} be a multi-index of spatial partial derivatives, then

$$U_h^{(\mathbf{D})}(x, y, t) = \sum_{k=0}^K \alpha_k(t) \phi_k^{(\mathbf{D})}(x, y). \quad (7.21)$$

In the two-dimensional case, the computation of the smoothness measurement function IS is as follows:

$$IS(Q_K(x, y)) = \sum_{1 \leq |\mathbf{D}| \leq K} |\Omega_{\mathcal{K}}|^{2|\mathbf{D}|-1} \int_{\Omega_{\mathcal{K}}} (\partial^{\mathbf{D}} Q_K(x, y))^2 \, dx dy. \quad (7.22)$$

We denote

$$M_{kl}^{\mathbf{D}} := |\Omega_{\mathcal{K}}|^{2|\mathbf{d}|-1} \sum_{|\mathbf{d}|=|\mathbf{D}|} \langle \phi_k^{(\mathbf{d})}, \phi_l^{(\mathbf{d})} \rangle_{L^2(\Omega_{\mathcal{K}})}, \quad (7.23)$$

where \mathbf{d} is a multi-index of spatial partial derivatives.

Thus, we have

$$\begin{aligned} IS(U_h) &= \sum_{1 \leq |\mathbf{D}| \leq K} |\Omega_{\mathcal{K}}|^{2|\mathbf{D}|-1} \int_{\Omega_{\mathcal{K}}} (\partial^{\mathbf{D}} U_h)^2 \, dx dy \\ &= \sum_{1 \leq |\mathbf{D}| \leq K} |\Omega_{\mathcal{K}}|^{2|\mathbf{D}|-1} \int_{\Omega_{\mathcal{K}}} \left(\sum_{k=1}^K \alpha_k \phi_k^{(\mathbf{D})} \right)^2 \, dx dy \\ &= \sum_{1 \leq |\mathbf{D}| \leq K} |\Omega_{\mathcal{K}}|^{2|\mathbf{D}|-1} \sum_{k=1}^K \sum_{l=1}^K \alpha_k \alpha_l \langle \phi_k^{(\mathbf{D})}, \phi_l^{(\mathbf{D})} \rangle_{L^2(\Omega_{\mathcal{K}})} \\ &= \sum_{1 \leq |\mathbf{D}| \leq K} \sum_{k=1}^K \sum_{l=1}^K \alpha_k \alpha_l M_{kl}^{\mathbf{D}} \\ &= \sum_{1 \leq |\mathbf{D}| \leq K} \left(\sum_{k=1}^K \alpha_k^2 M_{kk}^{\mathbf{D}} + \sum_{k=1}^{K-1} \sum_{l=k+1}^K 2\alpha_k \alpha_l M_{kl}^{\mathbf{D}} \right) \\ &= \sum_{1 \leq |\mathbf{D}| \leq K} \sum_{k=1}^K \alpha_k^2 M_{kk}^{\mathbf{D}} + \sum_{1 \leq |\mathbf{D}| \leq K} \sum_{k=1}^{K-1} \sum_{l=k+1}^K 2\alpha_k \alpha_l M_{kl}^{\mathbf{D}} \\ &= \sum_{k=1}^K \alpha_k^2 \cdot \left(\sum_{1 \leq |\mathbf{D}| \leq K} M_{kk}^{\mathbf{D}} \right) + \sum_{k=1}^{K-1} \sum_{l=k+1}^K 2\alpha_k \alpha_l \cdot \left(\sum_{1 \leq |\mathbf{D}| \leq K} M_{kl}^{\mathbf{D}} \right). \end{aligned} \quad (7.24)$$

Note that when $|\mathbf{d}| > \min(k, l)$, it holds that $\phi_{\min(k, l)}^{(\mathbf{d})} = 0$, we have

$$\sum_{1 \leq |\mathbf{D}| \leq K} M_{kl}^{\mathbf{D}} = \sum_{1 \leq |\mathbf{D}| \leq \min(k, l)} M_{kl}^{\mathbf{D}}, \quad (7.25)$$

therefore,

$$\begin{aligned} IS(U_h) &= \sum_{k=1}^K \alpha_k^2 \cdot \left(\sum_{1 \leq |\mathbf{D}| \leq k} M_{kk}^{\mathbf{D}} \right) + \sum_{k=1}^{K-1} \sum_{l=k+1}^K 2\alpha_k \alpha_l \cdot \left(\sum_{1 \leq |\mathbf{D}| \leq \min(k, l)} M_{kl}^{\mathbf{D}} \right) \\ &= \underbrace{\sum_{k=1}^K \alpha_k^2 \cdot \left(\sum_{1 \leq |\mathbf{D}| \leq k} M_{kk}^{\mathbf{D}} \right)}_{\sim \mathbb{D}} + \underbrace{\sum_{k=1}^{K-1} \sum_{l=k+1}^K 2\alpha_k \alpha_l \cdot \left(\sum_{1 \leq |\mathbf{D}| \leq k} M_{kl}^{\mathbf{D}} \right)}_{\sim (\mathbb{L} + \mathbb{U}), \mathbb{L} = \mathbb{U}^T}. \end{aligned} \quad (7.26)$$

Note that unlike the 1D case,

$$M_{kl}^{\mathbf{D}} = \sum_{|\mathbf{d}|=|\mathbf{D}|} \langle \phi_k^{(\mathbf{d})}, \phi_l^{(\mathbf{d})} \rangle_{L^2(\Omega_{\mathcal{K}})}. \quad (7.27)$$

The remaining derivation is similar to the 1D case.

7.2.3. TVB(D)-minmod Limiter under Dual Constraints of Smoothness Factor IS and L^2 -Error. The TVB(D)-minmod limiter provides three (or two) correction conditions that inherently suppress numerical oscillations, optimizing for the *smoothness factor* still only serves to further suppress numerical oscillations, but *over-suppression* of numerical oscillations is clearly detrimental to accuracy: the smoothness measurement function is in the form of a *sum of squares*, with 0 being its optimization ideal point (extreme value). Obviously, 0 is quite unfavorable for accuracy (large dissipation, with the numerical solution curve tending towards a horizontal line).

Numerical oscillations are generally more prevalent in high-order schemes, *Considering that the TVB(D)-minmod limiter's three (or two) correction conditions have already played a role in suppressing numerical oscillations*, we should select the solution with higher precision from the weakly oscillatory solution set provided by the original TVB(D)-minmod limiter. Therefore, it is appropriate to make the *corrected numerical solution as close as possible to the oscillating numerical solution of the original high-order scheme under the premise of weak oscillations*. Hence, one can choose the L^2 -norm to characterize the “closeness” of the corrected solution to the original high-order scheme’s solution. This is the approach taken in the literature [1].

It is worth noting the following observations:

- For the computation of the Burgers’ equation using P^3 -polynomial approximation, the “smoothness factor constraint” is superior to the “high-order scheme L^2 -error constraint.” (Please refer to Example 9.4 in Subsection 9.2.)
- In the 1D-Euler Lax shock tube problem, the “smoothness factor constraint” is superior

to the “high-order scheme L^2 -error constraint” (the “high-order scheme L^2 -error constraint” may lead to a “blow-up”).

The shock tube problem is an example that is relatively extreme and demanding, hence stability (suppression of oscillations) is more important than accuracy. The “smoothness factor constraint” introduces additional conditions to suppress oscillations beyond those provided by the TVB(D)-minmod limiter (equivalent to additional artificial viscosity), thus yielding better results. On the other hand, the “high-order scheme L^2 -error constraint” may lead to computational divergence due to insufficient suppression of oscillations.

Therefore, we can further consider the combination of the two and construct and solve the bi-objective optimization problem according to “Linear Weighted Sum Method” from optimization theory.

■ One-dimensional case

A common approach to multi-objective optimization is to form a new objective function by taking a linear convex combination of several objective functions, i.e.,

$$\begin{aligned} & \min \left\{ \omega_{IS} \cdot IS(u_i^{mod}) + \omega_{L^2} \cdot \|u_i^{mod} - u_i^{old}\|_{L^2(I_i)} \right\} \\ \text{s.t.} \quad & \begin{cases} \frac{1}{|I_i|} \int_{I_i} u_i^{mod}(x) dx = \bar{U}_i \\ u_i^{mod}(x_{i-1/2}) = U_{i-1/2}^{+,mod} \\ u_i^{mod}(x_{i+1/2}) = U_{i+1/2}^{-,mod} \end{cases} \end{aligned} \quad (7.28)$$

The integral mean remains unchanged, hence $\alpha_0^{i,mod} = \alpha_0^{i,old}$, which allows further simplification as follows:

$$\begin{aligned} & \min \left\{ \omega_{IS} \cdot IS(u_i^{mod}) + \omega_{L^2} \cdot \|u_i^{mod} - u_i^{old}\|_{L^2(I_i)} \right\} \\ \text{s.t.} \quad & \begin{cases} u_i^{mod}(x_{i-1/2}) = U_{i-1/2}^{+,mod} \\ u_i^{mod}(x_{i+1/2}) = U_{i+1/2}^{-,mod} \end{cases} \end{aligned} \quad (7.29)$$

We continue to use the notations from the previous Section 7.2.1:

$$\begin{aligned} \tilde{\mathbf{a}}_i^{mod} &:= \left[\alpha_1^{i,mod}, \alpha_2^{i,mod}, \alpha_3^{i,mod}, \dots, \alpha_{K-1}^{i,mod}, \alpha_K^{i,mod} \right]^T, \\ [\Phi]_i &:= \begin{bmatrix} \phi_1^{(I_i)}(x_{i-1/2}) & \phi_2^{(I_i)}(x_{i-1/2}) & \cdots & \phi_K^{(I_i)}(x_{i-1/2}) \\ \phi_1^{(I_i)}(x_{i+1/2}) & \phi_2^{(I_i)}(x_{i+1/2}) & \cdots & \phi_K^{(I_i)}(x_{i+1/2}) \end{bmatrix}^T, \\ [\Lambda^*] &:= [\lambda_1^*, \lambda_2^*]^T. \end{aligned}$$

The solution u_i^{mod} to problem (7.29) satisfies the following saddle-point equation:

$$\mathbb{A}_i^{IS+L^2} \mathbf{X}_i^{IS+L^2} = \mathbf{b}_i^{IS+L^2}, \quad (7.30)$$

where

$$\mathbb{A}_i^{IS+L^2} = \left[\begin{array}{c|c} \omega_{IS}\mathbb{M}_i^{IS} + \omega_{L^2}\mathbb{M}_i^{L^2} & [\Phi]_i \\ \hline [\Phi]_i^T & \mathbf{0}_{2 \times 2} \end{array} \right], \quad (7.31a)$$

$$\mathbf{b}_i^{IS+L^2} = \left[\begin{array}{c} \omega_{IS} \cdot \mathbf{0}_{K \times 1} + \omega_{L^2} \cdot 2\tilde{\mathbf{a}}^{i,old} \\ \hline U_{i-1/2}^{+,mod} - \alpha_0^{i,old} \phi_0^{(I_i)}(x_{i-1/2}) \\ U_{i+1/2}^{-,mod} - \alpha_0^{i,old} \phi_0^{(I_i)}(x_{i+1/2}) \end{array} \right], \quad (7.31b)$$

$$\mathbf{X}_i^{IS+L^2} = \left[\begin{array}{c} \tilde{\mathbf{a}}_i^{mod} \\ \hline [\Lambda^*] \end{array} \right]. \quad (7.31c)$$

Note:

$$\mathbb{M}_i^{L^2} = \text{diag}(2)_{K \times K}, \quad (7.32)$$

$$\tilde{\mathbf{a}}^{i,old} = \left[\alpha_1^{i,old}, \alpha_2^{i,old}, \alpha_3^{i,old}, \dots, \alpha_{K-1}^{i,old}, \alpha_K^{i,old} \right]^T. \quad (7.33)$$

Finally, the optimal solution to problem (7.29) can be obtained $\mathbf{a}_i^{mod} = \left[\alpha_0^{i,old}, [\tilde{\mathbf{a}}_i^{mod}]^T \right]^T$.

■ Two-dimensional case

For the DG weak solution U_{ij} on the two-dimensional rectangular element Ω_{ij} , the following bi-objective optimization problem is constructed:

$$\begin{aligned} & \min \left\{ \omega_{IS} \cdot IS(U_{ij}^{mod}) + \omega_{L^2} \cdot \|U_{ij}^{mod} - U_{ij}^{old}\|_{L^2(\Omega_{ij})} \right\} \\ & \text{s.t.} \quad \left\{ \begin{array}{l} \int_{\partial\Omega_{ij}^L} U_{ij}^{mod}(x, y) dl = \overline{U}_{\partial\Omega_{ij}^L}^{L,mod} = \overline{U}_{ij}^{L,mod} \\ \int_{\partial\Omega_{ij}^R} U_{ij}^{mod}(x, y) dl = \overline{U}_{\partial\Omega_{ij}^R}^{R,mod} = \overline{U}_{ij}^{R,mod} \\ \int_{\partial\Omega_{ij}^B} U_{ij}^{mod}(x, y) dl = \overline{U}_{\partial\Omega_{ij}^B}^{B,mod} = \overline{U}_{ij}^{B,mod} \\ \int_{\partial\Omega_{ij}^T} U_{ij}^{mod}(x, y) dl = \overline{U}_{\partial\Omega_{ij}^T}^{T,mod} = \overline{U}_{ij}^{T,mod} \end{array} \right. \quad (7.34) \end{aligned}$$

For conservation requirement, it still holds that $\alpha_0^{(ij),mod} = \alpha_0^{(ij),old}$.

Here, we denote

$$\tilde{\mathbf{a}}_{(ij)}^{old} := \left[\alpha_1^{(ij),old}, \alpha_2^{(ij),old}, \dots, \alpha_{(K+1)(K+2)/2-1}^{(ij),old} \right]^T, \quad (7.35)$$

$$\tilde{\mathbf{a}}_{(ij)}^{mod} := \left[\alpha_1^{(ij),mod}, \alpha_2^{(ij),mod}, \dots, \alpha_{(K+1)(K+2)/2-1}^{(ij),mod} \right]^T, \quad (7.36)$$

$$\left. \begin{aligned} \Gamma_k^{L,(ij)} &:= \int_{\partial\Omega_{ij}^L} \phi_k^{(ij)}(x, y) dl, & \Gamma_k^{R,(ij)} &:= \int_{\partial\Omega_{ij}^R} \phi_k^{(ij)}(x, y) dl \\ \Gamma_k^{B,(ij)} &:= \int_{\partial\Omega_{ij}^B} \phi_k^{(ij)}(x, y) dl, & \Gamma_k^{T,(ij)} &:= \int_{\partial\Omega_{ij}^T} \phi_k^{(ij)}(x, y) dl \end{aligned} \right\}, \quad (7.37)$$

$$k = 0, 1, 2, 3, \dots, (K+1)(K+2)/2 - 1$$

$$[\Gamma]_{(ij)} := \begin{bmatrix} \Gamma_{\mathbf{1}}^{L,(ij)} & \Gamma_2^{L,(ij)} & \dots & \Gamma_{(K+1)(K+2)/2-1}^{L,(ij)} \\ \Gamma_{\mathbf{1}}^{R,(ij)} & \Gamma_2^{R,(ij)} & \dots & \Gamma_{(K+1)(K+2)/2-1}^{R,(ij)} \\ \Gamma_{\mathbf{1}}^{B,(ij)} & \Gamma_2^{B,(ij)} & \dots & \Gamma_{(K+1)(K+2)/2-1}^{B,(ij)} \\ \Gamma_{\mathbf{1}}^{T,(ij)} & \Gamma_2^{T,(ij)} & \dots & \Gamma_{(K+1)(K+2)/2-1}^{T,(ij)} \end{bmatrix}, \quad (7.38)$$

$$[\Lambda^*] := [\lambda_1^*, \lambda_2^*, \lambda_3^*, \lambda_4^*]^T. \quad (7.39)$$

The saddle-point equation is given by:

$$\mathbb{A}_{(ij)}^{IS+L^2} \mathbf{X}_{(ij)}^{IS+L^2} = \mathbf{b}_{(ij)}^{IS+L^2}, \quad (7.40)$$

where

$$\mathbb{A}_{(ij)}^{IS+L^2} = \begin{bmatrix} \omega_{IS} \mathbb{M}_{(ij)}^{IS} + \omega_{L^2} \mathbb{M}_{(ij)}^{L^2} & \vdots & [\Gamma]_{(ij)}^T \\ \hline [\Gamma]_{(ij)} & \vdots & \mathbf{0}_{4 \times 4} \end{bmatrix}, \quad (7.41a)$$

$$\mathbf{b}_{(ij)}^{IS+L^2} = \begin{bmatrix} \omega_{IS} \cdot \mathbf{0}_{(K+1)(K+2)/2-1 \times 1} + \omega_{L^2} \cdot 2\tilde{\mathbf{a}}_{(ij)}^{old} \\ \hline \bar{U}_{ij}^{L,mod} - \alpha_0^{(ij),old} \Gamma_0^{L,(ij)} \\ \bar{U}_{ij}^{R,mod} - \alpha_0^{(ij),old} \Gamma_0^{R,(ij)} \\ \bar{U}_{ij}^{B,mod} - \alpha_0^{(ij),old} \Gamma_0^{B,(ij)} \\ \bar{U}_{ij}^{T,mod} - \alpha_0^{(ij),old} \Gamma_0^{T,(ij)} \end{bmatrix}, \quad (7.41b)$$

$$\mathbf{X}_{(ij)}^{IS+L^2} = \begin{bmatrix} \tilde{\mathbf{a}}_{(ij)}^{mod} \\ \hline [\Lambda^*] \end{bmatrix}. \quad (7.41c)$$

Note that in the two-dimensional case, $\mathbb{M}_{(ij)}^{L^2} = \text{diag}(2)_{(K+1)(K+2)/2-1 \times (K+1)(K+2)/2-1}$, while $\mathbb{M}_{(ij)}^{IS}$ must be derived based on the conclusions *equations (7.26) and (7.27) in Section 7.2.2*, following the process *equations (7.8) to (7.12) outlined in Section 7.2.1*.

Then, $\mathbf{a}_{(ij)}^{mod} = \left[\alpha_0^{(ij),old}, [\tilde{\mathbf{a}}_{(ij)}^{mod}]^T \right]^T$ is just the optimal solution to problem (7.34)

We denote the limiter based on the bi-objective optimization problem concerning the IS function and L^2 -error as the IS - L^2 -TVB(D)-minmod limiter. It is worth noting that when setting $\omega_{L^2} = 0$ and $\omega_{IS} = 1$, the IS - L^2 -TVB(D)-minmod limiter degenerates into the IS -TVB(D)-minmod limiter discussed in the previous sections [7.2.1](#) and [7.2.2](#); On the other

hand, if we take $\omega_{L^2} = 1$ and $\omega_{IS} = 0$, then the IS- L^2 -TVB(D)-minmod limiter degenerates into the L^2 -TVB(D)-minmod limiter originated from literature [1].

8. RECONSTRUCTION IN CHARACTERISTIC FIELD

When applying limiters to a system of equations, there are typically two implementation methods: (a) directly reconstructing each component in the original physical space; (b) performing local characteristic decomposition in conjunction to transform into the characteristic space and then reconstructing each component individually.

The former is easier to implement in a program. Directly reconstructing by component is essentially assuming that the components are already decoupled. However, in the original system of equations, the unknown components are usually coupled, so the actual effect may not be satisfactory. The latter method uses the Jacobian of the flux function to transform the system into the characteristic space, achieving true decoupling, and then proceeds to reconstruct each component individually.

The basic principle of characteristic reconstruction is as follows (taking a one-dimensional hyperbolic conservation system as an example):

Consider the equation

$$\partial_t \mathbf{U} + \partial_x (f(\mathbf{U})) = 0,$$

where $A = \frac{\partial f(\mathbf{U})}{\partial \mathbf{U}}$, and assume it is diagonalizable, that is, $A = R\Lambda L$ and $RL = I_d$. Then the above equation can be rewritten as

$$\partial_t \mathbf{U} + R\Lambda L \partial_x \mathbf{U} = 0. \quad (8.1)$$

When A is a constant matrix (R, Λ, L are all constant matrices), introduce the characteristic variable $\mathbf{V} = L\mathbf{U}$, and multiply both sides of the equation by L to obtain

$$\partial_t \mathbf{V} + \Lambda \partial_x \mathbf{V} = 0. \quad (8.2)$$

Since Λ is a constant diagonal matrix, the above equation is a completely decoupled set of N independent constant-coefficient single-wave equations ($\mathbf{U} \in \mathbb{R}^d$). Thus, \mathbf{V} can be reconstructed component by component to obtain $\tilde{\mathbf{V}}$, and then the inverse transformation using R is applied to return to the original physical space, that is, $\tilde{\mathbf{U}} = R\tilde{\mathbf{V}}$, which completes the characteristic reconstruction.

Considering the variable coefficients and nonlinear characteristics of the equations (where A, R, Λ , and L are function matrices), local freezing must be introduced to fix A, R, Λ , and L . Similar to the point values in the FD method and the cell-averaged integrals in the FV schemes, the DG scheme also requires determining the “values at which” to substitute into the Jacobian of the flux function and its characteristic structure, i.e., the “values at which” to implement local freezing.

Furthermore, in the DG scheme, each component of \mathbf{U} is a polynomial function; therefore, special attention is needed on how to use the eigenstructure of the Jacobian (L, R) to transform the polynomial into the corresponding characteristic space.

Currently, regarding the two aforementioned issues, for 2D systems, the existing practice is to use the arithmetic mean of the area-weighted average of the troubled cell and the area-weighted average of its adjacent cells as the local freezing value to determine the characteristic transformation for (L, R) . Note that for each edge of the troubled cell, there is an adjacent cell, thus it is necessary to traverse all edges of the troubled cell, perform multiple characteristic reconstructions, obtain multiple characteristic reconstruction polynomials, and finally synthesize the final characteristic reconstruction result according to certain weights. Of course, the Jacobian of a 2D hyperbolic conservation system on a certain edge can still be implemented using the normal Jacobian following the idea of FVS as discussed earlier. As for the characteristic transformation of the polynomial, although matrix transformation cannot be applied to a single polynomial, since the overall transformation of \mathbf{U} to the characteristic space involves transforming multiple polynomials simultaneously, considering that polynomial functions can be uniquely determined by their degree and coefficients, one can form a column vector of coefficients in front of each like term (by traversing all degrees, a coefficient matrix can be obtained). By sequentially applying the left eigenvector matrix L to these like term coefficient column vectors, the characteristic transformation of the polynomial can be completed. Since the coefficients of the DG polynomial are usually referred to as “moments” (also known as “modal coefficients”), this paper refers to the aforementioned characteristic transformation process as “moment characteristic transformation” (or “modal coefficient transformation”).

Specifically, the general procedure for characteristic reconstruction in 2D systems is as follows:

Let the troubled cell be denoted as Ω_0 , with its boundary composed of $\partial\Omega_0 = \cup_{\ell=1}^{\mathcal{L}} \Gamma_0^\ell$, and the neighboring cells adjacent to Ω_0 across Γ_0^ℓ are denoted as Ω_ℓ , $\ell = 1, 2, \dots, \mathcal{L}$. The outward normal vector on Γ_0^ℓ is $\mathbf{n}^\ell = (n_x^\ell, n_y^\ell)$, and the Jacobian in this direction (the normal Jacobian) is $A_n^\ell = n_x^\ell \cdot \frac{\partial \mathbf{F}}{\partial \mathbf{U}} + n_y^\ell \cdot \frac{\partial \mathbf{G}}{\partial \mathbf{U}}$, with its eigenstructure being $\mathbb{R}_n^\ell, \mathbb{L}_n^\ell$.

- step1. Generate A_n^ℓ and its eigenstructure $(\mathbb{L}_n^\ell, \mathbb{R}_n^\ell)$ using the arithmetic mean of the cell integral means of Ω_0, Ω_ℓ , i.e., $(\bar{u}_0 + \bar{u}_\ell)/2$.
- step2. Perform the characteristic transformation of the template polynomials using \mathbb{L}_n^ℓ for each edge.
- step3. In the characteristic space, perform reconstruction component by component.
- step4. Use the corresponding \mathbb{R}_n^ℓ to inverse transform back to the physical space, obtaining \mathcal{L} characteristic reconstruction polynomials $u_0^{mod, \ell}$, $\ell = 1, 2, \dots, \mathcal{L}$.
- step5. The final result of the characteristic reconstruction is $u_0^{mod} = \sum_{\ell} w_{\ell} \cdot u_0^{mod, \ell}$. If it is a uniform grid, then $w_i = 1/\mathcal{L}$; if it is a non-uniform grid, then $w_i = \frac{|\Omega_i|}{\sum_{\ell=1}^{\mathcal{L}} |\Omega_\ell|}$.

Remark 8.1 (The process of characteristic reconstruction for a one-dimensional system). *Let the troubled cell be I_j , with its left and right nodes at $x_{j-1/2}$ and $x_{j+1/2}$, respectively. The adjacent cells are I_{j-1} and I_{j+1} . At the left and right nodes, the arithmetic mean of the integral averages of the troubled cell and its neighboring cell sharing the node is used to*

generate the locally frozen Jacobian and its eigenstructure:

$$\left\{ \mathbb{L}_1 \left(\frac{\bar{u}_{j-1} + \bar{u}_j}{2} \right), \mathbb{R}_1 \left(\frac{\bar{u}_{j-1} + \bar{u}_j}{2} \right) \right\}; \quad \left\{ \mathbb{L}_2 \left(\frac{\bar{u}_j + \bar{u}_{j+1}}{2} \right), \mathbb{R}_2 \left(\frac{\bar{u}_j + \bar{u}_{j+1}}{2} \right) \right\}.$$

Two characteristic transformations are then performed using \mathbb{L}_1 and \mathbb{L}_2 , i.e., $\tilde{P}_s^i = \mathbb{L}_i P_s$, $i = 1, 2$, where P_s is the stencil polynomial used for the reconstruction. Reconstruction is executed twice in the characteristic space, and the inverse transformations via \mathbb{R}_1 and \mathbb{R}_2 are applied to return to the original physical space, yielding $u_j^{\text{mod},1}$ and $u_j^{\text{mod},2}$. The final reconstruction result is given by $u_j^{\text{mod}} = w_1 \cdot u_j^{\text{mod},1} + w_2 \cdot u_j^{\text{mod},2}$. For uniform grids, $w_1 = w_2 = 1/2$; for non-uniform grids, the weights are proportional to the lengths of the stencil intervals. For example, in SimpleWENO, $w_1 = \frac{|I_{j-1}|}{|I_{j-1}| + |I_{j+1}|}$, $w_2 = \frac{|I_{j+1}|}{|I_{j-1}| + |I_{j+1}|}$.

Below, the one-dimensional hyperbolic conservation system is taken as an example to introduce the ‘‘moment characteristic transformation’’ (‘‘modal coefficient transformation’’) process:

Consider $\partial_t \mathbf{U} + \partial_x \mathbf{F}(\mathbf{U}) = 0$, $\mathbf{U} = (u_1, u_2, u_3, \dots, u_m)^\top$.

Let $\mathbb{J} = \frac{\partial \mathbf{F}(\mathbf{U})}{\partial \mathbf{U}} = \mathbb{R} \Lambda \mathbb{L}$. Note: $\mathbb{R} \mathbb{L} = \mathbb{I}_d$ (identity matrix); $\mathbb{J}, \mathbb{R}, \Lambda, \mathbb{L}$ are all matrix-valued functions of \mathbf{U} .

The approximate solution for the i -th component u_i is given by:

$$u_i = \sum_{k=0}^{k=K} \alpha_k^{(i)} \phi_{i,k},$$

where i denotes the i -th component, K represents the highest degree of the basis functions (polynomials), $\phi_{i,k}$ represents the k -th basis function (polynomial) of the i -th component, and $\alpha_k^{(i)}$ is the k -th moment (or k -th modal coefficient) of the i -th component. Form a column vector with the basis functions of all orders for the i -th system component

$$\left[\Phi^{(i)} \right] = [\phi_{i,0}, \phi_{i,1}, \phi_{i,2}, \dots, \phi_{i,K}]^\top. \quad (8.3)$$

Now, partition the left eigenvector matrix \mathbb{L} column-wise, such that

$$\mathbb{L} = \begin{bmatrix} l_{11} & l_{12} & l_{13} & \cdots & l_{1m} \\ l_{21} & l_{22} & l_{23} & \cdots & l_{2m} \\ l_{31} & l_{32} & l_{33} & \cdots & l_{3m} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ l_{m1} & l_{m2} & l_{m3} & \cdots & l_{mm} \end{bmatrix} = (\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3, \dots, \mathbf{L}_m). \quad (8.4)$$

Form a row vector consisting of the modal coefficients of the i -th system component, denoted as $[\mathbf{a}^{(i)}]^\top$. The modal coefficients of all components constitute the following modal coefficient

matrix

$$\mathbb{A} = \begin{bmatrix} \alpha_0^{(1)} & \alpha_1^{(1)} & \alpha_2^{(1)} & \cdots & \alpha_K^{(1)} \\ \alpha_0^{(2)} & \alpha_1^{(2)} & \alpha_2^{(2)} & \cdots & \alpha_K^{(2)} \\ \alpha_0^{(3)} & \alpha_1^{(3)} & \alpha_2^{(3)} & \cdots & \alpha_K^{(3)} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \alpha_0^{(m)} & \alpha_1^{(m)} & \alpha_2^{(m)} & \cdots & \alpha_K^{(m)} \end{bmatrix} = \begin{pmatrix} [\mathbf{a}^{(1)}]^\top \\ [\mathbf{a}^{(2)}]^\top \\ [\mathbf{a}^{(3)}]^\top \\ \vdots \\ [\mathbf{a}^{(m)}]^\top \end{pmatrix}. \quad (8.5)$$

When the left eigenvector matrix \mathbb{L} acts on the modal coefficient matrix \mathbb{A} , it produces the matrix of characteristic modal coefficients \mathbb{B} , given by

$$\begin{aligned} \mathbb{B} = \mathbb{L}\mathbb{A} &= (\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3, \dots, \mathbf{L}_m) \begin{pmatrix} [\mathbf{a}^{(1)}]^\top \\ [\mathbf{a}^{(2)}]^\top \\ [\mathbf{a}^{(3)}]^\top \\ \vdots \\ [\mathbf{a}^{(m)}]^\top \end{pmatrix} \\ &= \mathbf{L}_1 [\mathbf{a}^{(1)}]^\top + \mathbf{L}_2 [\mathbf{a}^{(2)}]^\top + \mathbf{L}_3 [\mathbf{a}^{(3)}]^\top + \cdots + \mathbf{L}_m [\mathbf{a}^{(m)}]^\top. \end{aligned} \quad (8.6)$$

The characteristic modal coefficient matrix \mathbb{B} has its i -th row denoted as $\mathbb{B}(i, :)$, satisfying

$$\mathbb{B}(i, :) = \mathbf{L}_1(i) \cdot [\mathbf{a}^{(1)}]^\top + \mathbf{L}_2(i) \cdot [\mathbf{a}^{(2)}]^\top + \mathbf{L}_3(i) \cdot [\mathbf{a}^{(3)}]^\top + \cdots + \mathbf{L}_m(i) \cdot [\mathbf{a}^{(m)}]^\top. \quad (8.7)$$

Let

$$[\mathbf{b}^{(i)}]^\top = \mathbb{B}(i, :), \quad (8.8a)$$

$$\beta_k^{(i)} = [\mathbf{b}^{(i)}]^\top(k) = \mathbb{B}(i, k). \quad (8.8b)$$

Define

$$v_i = [\mathbf{b}^{(i)}]^\top [\Phi^{(i)}] = \sum_{k=0}^{k=K} \beta_k^{(i)} \phi_{i,k}, \quad (8.9)$$

then the polynomial v_i is the projection of the polynomial u_i in the original physical space onto the characteristic space.

Remark 8.2. *The procedure for the inverse characteristic transformation is analogous to that of the characteristic transformation; it only requires substituting \mathbb{L} with \mathbb{R} .*

8.1. Local Freezing Based on Cell Interface Integral Mean. It is noted that limiters ultimately serve to correct the interface fluxes, which are calculated on the boundaries of cells. Therefore, it is reasonable to choose the integral average values of the field functions on the grid edges as candidate states (the cells on either side of an edge can each provide an edge integral average value). In fact, in the 1D problem FD framework, characteristic reconstruction uses the arithmetic average or Roe average of the left and right states at the grid midpoint to achieve local freezing. In one-dimensional problems, grid boundaries collapse to nodes, so in higher dimensions, they should revert to curve (surface) integral average values. Based on this, we make the following modifications to the local freezing scheme during the characteristic reconstruction process for high-dimensional systems: each adjacent cell calculates the line (surface) integral average value of the field function on the common edge (face), and then takes the arithmetic average or Roe average as the local freezing value.

Consider two adjacent cells, Ω_0 and Ω_ℓ , sharing a common boundary Γ_0^ℓ , with DG solutions u_0 and u_ℓ , respectively. The integral averages of u_0 and u_ℓ on Γ_0^ℓ are computed component-wise as follows:

$$\begin{aligned}\bar{\Gamma}_0^\ell(i, \Omega_0) &= \frac{1}{|\Gamma_0^\ell|} \int_{\Gamma_0^\ell} u_0^{(i)} dl, \quad i = 1, 2, \dots, m. \\ \bar{\Gamma}_0^\ell(i, \Omega_\ell) &= \frac{1}{|\Gamma_0^\ell|} \int_{\Gamma_0^\ell} u_\ell^{(i)} dl, \quad i = 1, 2, \dots, m.\end{aligned}$$

Here, m denotes the total number of system components (the total number of unknown field functions). Next, compute the arithmetic or Roe average of $\bar{\Gamma}_0^\ell(i, \Omega_0)$ and $\bar{\Gamma}_0^\ell(i, \Omega_\ell)$ component-wise, and use these averages to generate the normal Jacobian.

Remark 8.3 (Improved Scheme for Local Freezing in 1D Systems). *Consider the troubled cell I_j with its left and right nodes at $x_{j-1/2}$ and $x_{j+1/2}$, respectively. The adjacent cells are I_{j-1} and I_{j+1} . At each of these nodes, the local freezing Jacobian and its eigenstructure are generated by taking the arithmetic average of the left and right state values at the node:*

$$\left\{ \mathbb{L}_1 \left(\frac{U_{j-1/2}^L + U_{j-1/2}^R}{2} \right), \mathbb{R}_1 \left(\frac{U_{j-1/2}^L + U_{j-1/2}^R}{2} \right) \right\}; \quad \left\{ \mathbb{L}_2 \left(\frac{U_{j+1/2}^L + U_{j+1/2}^R}{2} \right), \mathbb{R}_2 \left(\frac{U_{j+1/2}^L + U_{j+1/2}^R}{2} \right) \right\}.$$

Alternatively, Roe averages of the left and right states can also be used.

8.2. Interpolation-based Characteristic Transformation. Distinct from the ‘‘moment characteristic transformation’’ mentioned earlier, this paper proposes an ‘‘interpolation-based characteristic transformation’’, and illustrates the process using a one-dimensional hyperbolic conservation system as an example.

Let the set of sampling points be $\mathbf{X} = (x_0, x_1, x_2, \dots, x_K)^\top$. Correspondingly, the sampled values for the i -th component are $\mathbf{Y}^{(i)} = (y_0^{(i)}, y_1^{(i)}, y_2^{(i)}, \dots, y_K^{(i)})^\top$, where $y_k^{(i)} = u_i(x_k)$. The

sampled values of all components together form the sampling point matrix \mathbb{Y} :

$$\mathbb{Y} = \begin{pmatrix} [\mathbf{Y}^{(1)}]^\top \\ [\mathbf{Y}^{(2)}]^\top \\ [\mathbf{Y}^{(3)}]^\top \\ \vdots \\ [\mathbf{Y}^{(m)}]^\top \end{pmatrix}. \quad (8.10)$$

Transform the sampling point matrix to the characteristic space

$$\begin{aligned} \tilde{\mathbb{Y}} = \mathbb{L}\mathbb{Y} &= (\mathbf{L}_1, \mathbf{L}_2, \mathbf{L}_3, \dots, \mathbf{L}_m) \begin{pmatrix} [\mathbf{Y}^{(1)}]^\top \\ [\mathbf{Y}^{(2)}]^\top \\ [\mathbf{Y}^{(3)}]^\top \\ \vdots \\ [\mathbf{Y}^{(m)}]^\top \end{pmatrix} \\ &= \mathbf{L}_1 [\mathbf{Y}^{(1)}]^\top + \mathbf{L}_2 [\mathbf{Y}^{(2)}]^\top + \mathbf{L}_3 [\mathbf{Y}^{(3)}]^\top + \dots + \mathbf{L}_m [\mathbf{Y}^{(m)}]^\top. \end{aligned} \quad (8.11)$$

The i -th row of the characteristic sampling point matrix $\tilde{\mathbb{Y}}$ is denoted as $\tilde{\mathbb{Y}}(i, :)$, satisfying

$$\tilde{\mathbb{Y}}(i, :) = \mathbf{L}_1(i) \cdot [\mathbf{Y}^{(1)}]^\top + \mathbf{L}_2(i) \cdot [\mathbf{Y}^{(2)}]^\top + \mathbf{L}_3(i) \cdot [\mathbf{Y}^{(3)}]^\top + \dots + \mathbf{L}_m(i) \cdot [\mathbf{Y}^{(m)}]^\top. \quad (8.12)$$

Let

$$\tilde{\mathbf{Y}}^{(i)} := [\tilde{\mathbb{Y}}(i, :)]^\top = \mathbf{L}_1(i) \cdot \mathbf{Y}^{(1)} + \mathbf{L}_2(i) \cdot \mathbf{Y}^{(2)} + \mathbf{L}_3(i) \cdot \mathbf{Y}^{(3)} + \dots + \mathbf{L}_m(i) \cdot \mathbf{Y}^{(m)}, \quad (8.13)$$

then $\tilde{\mathbf{Y}}^{(i)}$ is the projection of the original physical space's $\mathbf{Y}^{(i)}$ onto the characteristic space. In other words, $\tilde{\mathbf{Y}}^{(i)}$ represents the values of the characteristic projection \tilde{u}_i of the original physical space's u_i at the same set of sampling points $\mathbf{X} = (x_0, x_1, x_2, \dots, x_K)^\top$.

At this point, the polynomial \tilde{u}_i in the characteristic space that satisfies the interpolation data set $(\mathbf{X}, \tilde{\mathbf{Y}}^{(i)})$ is determined using the method of undetermined coefficients.

Let the modal coefficient set of the characteristic polynomial \tilde{u}_i be $\tilde{\mathbf{a}}^{(i)}$, so that

$$\tilde{u}_i = \sum_{k=0}^K \tilde{\alpha}_k^{(i)} \phi_{i,k} \quad \text{with} \quad \tilde{\alpha}_k^{(i)} = \tilde{\mathbf{a}}^{(i)}(k).$$

Let

$$\mathbb{P}^{(i)} := \begin{bmatrix} \phi_{i,0}(x_0) & \phi_{i,1}(x_0) & \phi_{i,2}(x_0) & \cdots & \phi_{i,K}(x_0) \\ \phi_{i,0}(x_1) & \phi_{i,1}(x_1) & \phi_{i,2}(x_1) & \cdots & \phi_{i,K}(x_1) \\ \phi_{i,0}(x_2) & \phi_{i,1}(x_2) & \phi_{i,2}(x_2) & \cdots & \phi_{i,K}(x_2) \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ \phi_{i,0}(x_K) & \phi_{i,1}(x_K) & \phi_{i,2}(x_K) & \cdots & \phi_{i,K}(x_K) \end{bmatrix}. \quad (8.14)$$

The characteristic polynomial \tilde{u}_i should satisfy

$$\tilde{u}_i(\mathbf{X}) = \tilde{\mathbf{Y}}^{(i)}, \quad (8.15)$$

hence

$$\mathbb{P}^{(i)} \tilde{\mathbf{a}}^{(i)} = \tilde{\mathbf{Y}}^{(i)}, \quad (8.16)$$

and thus

$$\tilde{\mathbf{a}}^{(i)} = [\mathbb{P}^{(i)}]^{-1} \tilde{\mathbf{Y}}^{(i)}. \quad (8.17)$$

With this, the projection of the polynomial u_i from the original physical space onto the characteristic space, \tilde{u}_i , can be obtained.

8.3. Equivalence of Interpolation-based Characteristic Transformation and Moment Characteristic Transformation.

Proposition 8.1 (Equivalence of Interpolation-based Characteristic Transformation and Moment Characteristic Transformation when Basis Functions are Identical). *When all components of the system share the same set of basis functions, that is, $K_1 = K_2 = K_3 = \cdots = K_m = K$ and $\phi_{1,k}(\cdot) \equiv \phi_{2,k}(\cdot) \equiv \phi_{3,k}(\cdot) \equiv \cdots \equiv \phi_{m,k}(\cdot)$, $\forall k \in \{1, 2, 3, \dots, K\}$, it follows that $\mathbb{P}^{(1)} = \mathbb{P}^{(2)} = \mathbb{P}^{(3)} = \cdots = \mathbb{P}^{(m)}$, thus the interpolation-based characteristic transformation is equivalent to the moment characteristic transformation, i.e., $\tilde{\mathbf{a}}^{(i)} = \mathbf{b}^{(i)}$, $\forall i \in \{1, 2, 3, \dots, m\}$.*

Taking the one-dimensional hyperbolic conservation system as an example, Proposition 8.1 is proven.

Proof. By substituting equation (8.13) into equation (8.17), we obtain

$$\begin{aligned} [\tilde{\mathbf{a}}^{(i)}]^\top &= \left([\mathbb{P}^{(i)}]^{-1} \tilde{\mathbf{Y}}^{(i)} \right)^\top = [\tilde{\mathbf{Y}}^{(i)}]^\top \left([\mathbb{P}^{(i)}]^{-1} \right)^\top = [\tilde{\mathbf{Y}}^{(i)}]^\top \left([\mathbb{P}^{(i)}]^\top \right)^{-1} \\ &= (\mathbf{L}_1(i) \cdot \mathbf{Y}^{(1)} + \mathbf{L}_2(i) \cdot \mathbf{Y}^{(2)} + \cdots + \mathbf{L}_m(i) \cdot \mathbf{Y}^{(m)})^\top \left([\mathbb{P}^{(i)}]^{-1} \right)^\top \\ &= \mathbf{L}_1(i) \cdot [\mathbf{Y}^{(1)}]^\top \left([\mathbb{P}^{(i)}]^{-1} \right)^\top + \mathbf{L}_2(i) \cdot [\mathbf{Y}^{(2)}]^\top \left([\mathbb{P}^{(i)}]^{-1} \right)^\top + \cdots + \mathbf{L}_m(i) \cdot [\mathbf{Y}^{(m)}]^\top \left([\mathbb{P}^{(i)}]^{-1} \right)^\top. \end{aligned} \quad (8.18)$$

From equations (8.7) and (8.8), it is easy to know

$$[\mathbf{b}^{(i)}]^\top = \mathbf{L}_1(i) \cdot [\mathbf{a}^{(1)}]^\top + \mathbf{L}_2(i) \cdot [\mathbf{a}^{(2)}]^\top + \mathbf{L}_3(i) \cdot [\mathbf{a}^{(3)}]^\top + \cdots + \mathbf{L}_m(i) \cdot [\mathbf{a}^{(m)}]^\top. \quad (8.19)$$

Note that in the original physical space, it holds that $u_i(\mathbf{X}) = \mathbf{Y}^{(i)}$, similar to equations (8.15) and (8.16), we have

$$\mathbb{P}^{(i)} \mathbf{a}^{(i)} = \mathbf{Y}^{(i)}, \quad i = 1, 2, \dots, m; \quad (8.20a)$$

$$\mathbf{a}^{(i)} = [\mathbb{P}^{(i)}]^{-1} \mathbf{Y}^{(i)}, \quad i = 1, 2, \dots, m. \quad (8.20b)$$

By substituting equation (8.20b) into equation (8.19), we get

$$[\mathbf{b}^{(i)}]^\top = \mathbf{L}_1(i) \cdot [\mathbf{Y}^{(1)}]^\top \left([\mathbb{P}^{(1)}]^{-1} \right)^\top + \mathbf{L}_2(i) \cdot [\mathbf{Y}^{(2)}]^\top \left([\mathbb{P}^{(2)}]^{-1} \right)^\top + \cdots + \mathbf{L}_m(i) \cdot [\mathbf{Y}^{(m)}]^\top \left([\mathbb{P}^{(m)}]^{-1} \right)^\top. \quad (8.21)$$

Comparing $[\tilde{\mathbf{a}}^{(i)}]^\top$ and $[\mathbf{b}^{(i)}]^\top$, and noting the superscripts of the matrix \mathbb{P} , it can be seen that

if $\phi_{1,k} = \phi_{2,k} = \phi_{3,k} = \cdots = \phi_{m,k}, \forall k \in \{1, 2, 3, \dots, K\}$,

then $\mathbb{P}^{(1)} = \mathbb{P}^{(2)} = \mathbb{P}^{(3)} = \cdots = \mathbb{P}^{(m)}$,

and thus $\tilde{\mathbf{a}}^{(i)} = \mathbf{b}^{(i)}, i = 1, 2, 3, \dots, m$.

That is, at this point, the ‘‘moment characteristic transformation’’ and the ‘‘interpolation-based characteristic transformation’’ are equivalent. \square

9. NUMERICAL RESULTS

This section first conducts a accuracy test on the FVS-DG method for the compressible Euler equations and the shallow water wave equations, including one-dimensional and two-dimensional test cases. Subsequently, it demonstrates the advantage of the IS- L^2 -TVB(D)-minmod limiter in suppressing numerical oscillations near discontinuities on the linear scalar transport equation, the nonlinear Burgers’ equation, and the non-convex Buckley-Leverett equation. Finally, we solve several classical Riemann problems for the compressible Euler equations and the shallow water wave equations using the FVS-DG method coupled with the IS- L^2 -TVB(D)-minmod limiter, including the Sod problem, Lax problem, Shu-Osher problem, Blast problem, three typical Riemann test cases from the literature [42], and the Dam Break problem for the shallow water wave equations [43].

In all numerical experiments of this section, the adiabatic index for the compressible Euler equations is set to $\gamma = 1.4$ (simulating the dynamical process of air), and the acceleration due to gravity for the shallow water wave equations is set to $g = 9.8120 \text{ m/s}^2$.

All numerical experiments in this section use a uniform mesh partitioning, where the two-dimensional test cases include both rectangular and structured triangular meshes.

9.1. Accuracy Tests for FVS-DG.

9.1.1. One-dimensional Test Cases.

Example 9.1. 1D-Euler Compressible Equations.

Control Eqs:

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E + p) \end{pmatrix} = 0;$$

Computational domain:

$$\Omega \times [0, T_{end}] = [0, 1] \times [0, 1];$$

ICs:

$$\rho_0(x) = 1 + 0.2 \cos(\pi x), \quad u_0(x) = -0.7, \quad P_0(x) = 1;$$

BCs: *periodic boundary conditions*;

True solutions:

$$\rho(x, t) = 1 + 0.2 \cos(\pi(x + 0.7t)), \quad u(x, t) = -0.7, \quad P(x, t) = 1.$$

- Numerical flux format: AUSM;
Temporal discretization format: TVD-RK3;
CFL=0.1.
 L^∞ , L^2 , L^1 numerical errors and convergence orders for conservative variables ρ , ρu , E with P^2 - and P^3 -polynomial approximations are summarized respectively in Table 1 and Table 2.
- Numerical flux format: Steger-Warming;
Temporal discretization format: TVD-RK3;
CFL=0.1.
 L^∞ , L^2 , L^1 numerical errors and convergence orders for conservative variables ρ , ρu , E with P^2 - and P^3 -polynomial approximations are summarized respectively in Table 3 and Table 4.

Example 9.2. 1D-Shallow Water Wave Equations.

Control Eqs:

$$\partial_t \begin{bmatrix} h \\ hu \end{bmatrix} + \partial_x \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} 0 \\ -gh(z_0)_x \end{bmatrix};$$

Bottom elevation:

$$Z_0 = \sin^2(\pi x);$$

Computational domain:

$$\Omega \times [0, T_{end}] = [0, 1] \times [0, 0.075];$$

Note: The equation will develop a discontinuity near $t = 0.1$. To test for accuracy, we set the simulation time to $t = 0.075$, at which point the solution is still smooth. This example does

TABLE 1. P^2 -FVS(AUSM)-DG using equally spaced cells. 1D-Compressible Euler Equations with smooth initial conditions: $\rho_0(x) = 1 + 0.2 \cos(\pi x)$, $u_0(x) = -0.7$, $P_0(x) = 1$. L^∞ , L^2 , L^1 errors and convergence orders for conservative variables ρ , ρu , E .

<i>Mesh</i>	10	20	40	80	160
L^∞ -error	1.1024E-03	1.4025E-04	1.7693E-05	2.2156E-06	2.7701E-07
L^∞ -order	—	2.9746	2.9868	2.9974	2.9997
ρ L^2 -error	2.4737E-04	3.1521E-05	3.9607E-06	4.9575E-07	6.1989E-08
L^2 -order	—	2.9723	2.9925	2.9981	2.9995
L^1 -error	1.8559E-04	2.3295E-05	2.9126E-06	3.6395E-07	4.5479E-08
L^1 -order	—	2.9940	2.9997	3.0005	3.0005
L^∞ -error	5.3443E-04	6.6924E-05	8.4195E-06	1.0543E-06	1.3184E-07
L^∞ -order	—	2.9974	2.9907	2.9975	2.9994
ρu L^2 -error	2.4737E-04	3.1521E-05	3.9607E-06	4.9575E-07	6.1989E-08
L^2 -order	—	2.9723	2.9925	2.9981	2.9995
L^1 -error	1.8559E-04	2.3295E-05	2.9126E-06	3.6395E-07	4.5479E-08
L^1 -order	—	2.9940	2.9997	3.0005	3.0005
L^∞ -error	8.2911E-04	1.0494E-04	1.3268E-05	1.6616E-06	2.0772E-07
L^∞ -order	—	2.9819	2.9836	2.9972	2.9999
E L^2 -error	2.4737E-04	3.1521E-05	3.9607E-06	4.9575E-07	6.1989E-08
L^2 -order	—	2.9723	2.9925	2.9981	2.9995
L^1 -error	1.8559E-04	2.3295E-05	2.9126E-06	3.6395E-07	4.5479E-08
L^1 -order	—	2.9940	2.9997	3.0005	3.0005

not have an analytical true solution; instead, we use numerical solutions on finer meshes as references to obtain the errors and convergence orders of the algorithm.

ICs:

$$h_0(x) = 5 + e^{\cos(2\pi x)}, \quad u_0(x) = \frac{\sin(\cos(2\pi x))}{5 + e^{\cos(2\pi x)}};$$

BCs: *periodic boundary conditions*;

Numerical flux format: *van Leer*;

Temporal discretization format: *TVD-RK3*;

L^∞ , L^2 , L^1 numerical errors and convergence orders for conservative variables h , hu with P^2 -polynomial approximation are summarized respectively in Table 5.

9.1.2. Two-dimensional Test Cases.

TABLE 2. P^3 -FVS(AUSM)-DG using equally spaced cells. 1D-Compressible Euler Equations with smooth initial conditions: $\rho_0(x) = 1 + 0.2 \cos(\pi x)$, $u_0(x) = -0.7$, $P_0(x) = 1$. L^∞ , L^2 , L^1 errors and convergence orders for conservative variables ρ , ρu , E .

<i>Mesh</i>	10	20	40	80	160
L^∞ -error	4.1850E-05	2.7313E-06	1.7220E-07	1.0778E-08	6.7472E-10
L^∞ -order	—	3.9376	3.9875	3.9978	3.9977
ρ L^2 -error	7.8755E-06	4.9119E-07	3.0563E-08	1.9097E-09	1.1935E-10
L^2 -order	—	4.0030	4.0064	4.0004	4.0001
L^1 -error	5.4508E-06	3.3906E-07	2.0980E-08	1.3095E-09	8.1857E-11
L^1 -order	—	4.0069	4.0145	4.0019	3.9998
L^∞ -error	2.3771E-05	1.5271E-06	9.6370E-08	6.0382E-09	3.7818E-10
L^∞ -order	—	3.9604	3.9861	3.9964	3.9970
ρu L^2 -error	7.8755E-06	4.9119E-07	3.0563E-08	1.9097E-09	1.1935E-10
L^2 -order	—	4.0030	4.0064	4.0004	4.0001
L^1 -error	5.4508E-06	3.3906E-07	2.0980E-08	1.3095E-09	8.1857E-11
L^1 -order	—	4.0069	4.0145	4.0019	3.9998
L^∞ -error	3.2236E-05	2.0668E-06	1.2930E-07	8.0974E-09	5.0454E-10
L^∞ -order	—	3.9632	3.9986	3.9971	4.0044
E L^2 -error	7.8755E-06	4.9119E-07	3.0563E-08	1.9097E-09	1.1935E-10
L^2 -order	—	4.0030	4.0064	4.0004	4.0001
L^1 -error	5.4508E-06	3.3906E-07	2.0980E-08	1.3095E-09	8.1857E-11
L^1 -order	—	4.0069	4.0145	4.0019	3.9998

Example 9.3. 2D-Compressible Euler Equations.

Control Eqs:

$$\frac{\partial}{\partial t} \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix} + \frac{\partial}{\partial x} \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{pmatrix} + \frac{\partial}{\partial y} \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{pmatrix} = 0;$$

Computational domain:

$$\Omega \times [0, T_{end}] = \{[0, 2] \times [-1, 1]\} \times [0, 1];$$

ICs:

$$\rho_0(x, y) = 1 + 0.2 \cos(\pi x + \pi y), \quad u_0(x, y) = -0.7, \quad v_0(x, y) = 0.3, \quad P_0(x, y) = 1;$$

TABLE 3. P^2 -FVS(Steger-Warming)-DG using equally spaced cells. 1D-Compressible Euler Equations with smooth initial conditions: $\rho_0(x) = 1 + 0.2 \cos(\pi x)$, $u_0(x) = -0.7$, $P_0(x) = 1$. L^∞ , L^2 , L^1 errors and convergence orders for conservative variables ρ , ρu , E .

	Mesh	10	20	40	80	160
ρ	L^∞ -error	1.1728E-03	1.5873E-04	2.0237E-05	2.5390E-06	3.1747E-07
	L^∞ -order	—	2.8853	2.9716	2.9947	2.9996
	L^2 -error	2.7279E-04	3.6698E-05	4.6856E-06	5.8895E-07	7.3721E-08
	L^2 -order	—	2.8940	2.9694	2.9920	2.9980
	L^1 -error	2.0508E-04	2.7383E-05	3.4908E-06	4.3859E-07	5.4894E-08
	L^1 -order	—	2.9048	2.9717	2.9926	2.9981
ρu	L^∞ -error	5.2307E-04	6.8032E-05	8.5503E-06	1.0695E-06	1.3369E-07
	L^∞ -order	—	2.9427	2.9922	2.9991	3.0000
	L^2 -error	2.7279E-04	3.6698E-05	4.6856E-06	5.8895E-07	7.3721E-08
	L^2 -order	—	2.8940	2.9694	2.9920	2.9980
	L^1 -error	2.0508E-04	2.7383E-05	3.4908E-06	4.3859E-07	5.4894E-08
	L^1 -order	—	2.9048	2.9717	2.9926	2.9981
E	L^∞ -error	8.8839E-04	1.2148E-04	1.5573E-05	1.9591E-06	2.4527E-07
	L^∞ -order	—	2.8705	2.9636	2.9908	2.9978
	L^2 -error	2.7279E-04	3.6698E-05	4.6856E-06	5.8895E-07	7.3721E-08
	L^2 -order	—	2.8940	2.9694	2.9920	2.9980
	L^1 -error	2.0508E-04	2.7383E-05	3.4908E-06	4.3859E-07	5.4894E-08
	L^1 -order	—	2.9048	2.9717	2.9926	2.9981

BCs: *periodic boundary conditions*;

True solutions:

$$\begin{cases} \rho(x, y, t) = 1 + 0.2 \cos(\pi(x + 0.7t) + \pi(y - 0.3t)), \\ u(x, y, t) = -0.7, \\ v(x, y, t) = 0.3, \\ P(x, y, t) = 1. \end{cases}$$

- Numerical flux format: AUSM;

Temporal discretization format: TVD-RK3;

CFL=0.025;

L^∞ , L^2 , L^1 numerical errors and convergence orders for conservative variables ρ , ρu , ρv , E with P^2 -polynomial approximation are shown in Table 6.

TABLE 4. P^3 -FVS(Steger-Warming)-DG using equally spaced cells. 1D-Compressible Euler Equations with smooth initial conditions: $\rho_0(x) = 1 + 0.2 \cos(\pi x)$, $u_0(x) = -0.7$, $P_0(x) = 1$. L^∞ , L^2 , L^1 errors and convergence orders for conservative variables ρ , ρu , E .

<i>Mesh</i>	10	20	40	80	160
L^∞ -error	3.6110E-05	2.3683E-06	1.4967E-07	9.3550E-09	5.8578E-10
L^∞ -order	—	3.9304	3.9840	3.9999	3.9973
ρ L^2 -error	6.6707E-06	4.1281E-07	2.5553E-08	1.5954E-09	9.9687E-11
L^2 -order	—	4.0143	4.0139	4.0015	4.0004
L^1 -error	4.6767E-06	2.8866E-07	1.7743E-08	1.1087E-09	6.9260E-11
L^1 -order	—	4.0180	4.0240	4.0003	4.0008
L^∞ -error	2.1895E-05	1.4053E-06	8.8435E-08	5.5424E-09	3.4715E-10
L^∞ -order	—	3.9616	3.9901	3.9960	3.9969
ρu L^2 -error	6.6707E-06	4.1281E-07	2.5553E-08	1.5954E-09	9.9687E-11
L^2 -order	—	4.0143	4.0139	4.0015	4.0004
L^1 -error	4.6767E-06	2.8866E-07	1.7743E-08	1.1087E-09	6.9260E-11
L^1 -order	—	4.0180	4.0240	4.0003	4.0008
L^∞ -error	2.4615E-05	1.5544E-06	9.6544E-08	6.0418E-09	3.7597E-10
L^∞ -order	—	3.9851	4.0090	3.9981	4.0063
E L^2 -error	6.6707E-06	4.1281E-07	2.5553E-08	1.5954E-09	9.9687E-11
L^2 -order	—	4.0143	4.0139	4.0015	4.0004
L^1 -error	4.6767E-06	2.8866E-07	1.7743E-08	1.1087E-09	6.9260E-11
L^1 -order	—	4.0180	4.0240	4.0003	4.0008

- Numerical flux format: AUSM;
 Temporal discretization format: RK4;
 CFL=0.01;
 L^∞ , L^2 , L^1 numerical errors and convergence orders for conservative variables ρ , ρu , ρv , E with P^3 -polynomial approximation are shown in Table 7.
- Numerical flux format: Steger-Warming;
 Temporal discretization format: TVD-RK3;
 CFL=0.05;
 L^∞ , L^2 , L^1 numerical errors and convergence orders for conservative variables ρ , ρu , ρv , E with P^2 -polynomial approximation are shown in Table 8.

From the accuracy test results of the aforementioned one-dimensional and two-dimensional examples, it can be seen that both the P^K -FVS-DG scheme based on the Jacobian eigenvalue splitting and the P^K -FVS-DG scheme based on the Mach number splitting can achieve the

TABLE 5. P^2 -FVS(vanLeer)-DG using equally spaced cells with TVD-RK3 and CFL=0.01. 1D-Shallow Water Equations with smooth initial conditions: $h_0(x) = 5 + e^{\cos(2\pi x)}$, $u_0(x) = \frac{\sin(\cos(2\pi x))}{5 + e^{\cos(2\pi x)}}$. L^∞ , L^2 , L^1 errors and convergence orders for conservative variables h , hu .

	Mesh	50	100	200	400	800
h	L^∞ -error	2.0111E-02	6.3551E-03	2.0769E-03	2.9931E-04	4.3458E-05
	L^∞ -order	—	1.6620	1.6135	2.7947	2.7840
	L^2 -error	2.9563E-03	5.3791E-04	9.2442E-05	1.1241E-05	1.4936E-06
	L^2 -order	—	2.4584	2.5407	3.0397	2.9119
	L^1 -error	1.0235E-03	1.3895E-04	2.0637E-05	2.6006E-06	3.5516E-07
	L^1 -order	—	2.8809	2.7513	2.9883	2.8723
hu	L^∞ -error	7.4667E-02	2.5399E-02	8.3407E-03	1.2216E-03	1.7540E-04
	L^∞ -order	—	1.5557	1.6065	2.7714	2.8000
	L^2 -error	1.3649E-02	2.4733E-03	4.0608E-04	4.7905E-05	7.1252E-06
	L^2 -order	—	2.4643	2.6066	3.0835	2.7492
	L^1 -error	5.6059E-03	7.4207E-04	1.0431E-04	1.3226E-05	1.8956E-06
	L^1 -order	—	2.9173	2.8307	2.9795	2.8026

optimal spatial convergence order of $(K + 1)$. In addition, The examples for the Euler system and the shallow water wave system demonstrate the universality of the FVS-DG for general hyperbolic conservation laws.

9.2. Performance of the IS- L^2 -TVB(D)-minmod Limiter for Scalar Conservation Law. In this subsection, IS- L^2 -TVB(D)-minmod Limiter is applied to 1D/2D-scalar equations, including inviscid Burgers' equation (Example 9.4 and Example 9.6), non-convex scalar Buckley-Leverett problem (Example 9.5) and linear variable coefficient transport equation (swirling deformation flow in Example 9.7).

9.2.1. *One-dimensional Test Cases for Scalar Equations.* To compare IS- L^2 -TVB(D)-minmod Limiter with other classical limiters, we choose local Lax-Friedrichs (LLF) flux format for all of them. The LLF flux format is given as follows:

$$\hat{f}^{LF} = \frac{1}{2} \cdot (f(U^L) + f(U^R) - \alpha \cdot (U^R - U^L)),$$

$$\alpha = \max_{u \in I(u^L, u^R)} \{|f'(u)|\}, \quad I(u^L, u^R) = (\min\{u^L, u^R\}, \max\{u^L, u^R\}).$$

- If $f(u) = c \cdot u$ then $f'(u) = c$, $\alpha = |c|$;
- If $f(u) = c \cdot \frac{1}{2}u^2$ then $f'(u) = c \cdot u$, $\alpha = |c| \cdot \max\{|u^L|, |u^R|\}$.

The relationship between Local Lax-Friedrichs splitting method previously mentioned and LLF flux format here should be referred to Appendix C.1.

TABLE 6. P^2 -FVS(AUSM)-DG using uniform structured triangular meshes. 2D-Compressible Euler Equations with smooth initial conditions: $\rho_0(x, y) = 1 + 0.2 \cos(\pi x + \pi y)$, $u_0(x, y) = -0.7$, $v_0(x, y) = 0.3$, $P_0(x, y) = 1$. L^∞ , L^2 , L^1 errors and convergence orders for conservative variables ρ , ρu , ρv , E .

	Mesh	8×8	16×16	32×32	64×64
ρ	L^∞ -error	8.9167E-04	1.2342E-04	1.5657E-05	1.9642E-06
	L^∞ -order	—	2.8530	2.9786	2.9948
	L^2 -error	7.0277E-04	9.0059E-05	1.1425E-05	1.4334E-06
	L^2 -order	—	2.9641	2.9787	2.9947
	L^1 -error	1.1249E-03	1.4124E-04	1.7661E-05	2.2048E-06
	L^1 -order	—	2.9936	2.9995	3.0018
ρu	L^∞ -error	5.5158E-04	7.6466E-05	9.7164E-06	1.2195E-06
	L^∞ -order	—	2.8507	2.9763	2.9942
	L^2 -error	4.5169E-04	5.7971E-05	7.3577E-06	9.2326E-07
	L^2 -order	—	2.9619	2.9780	2.9944
	L^1 -error	7.2709E-04	9.1365E-05	1.1470E-05	1.4332E-06
	L^1 -order	—	2.9924	2.9938	3.0005
ρv	L^∞ -error	3.4939E-04	4.5836E-05	5.8122E-06	7.3063E-07
	L^∞ -order	—	2.9303	2.9793	2.9919
	L^2 -error	2.6454E-04	3.4709E-05	4.4310E-06	5.5688E-07
	L^2 -order	—	2.9301	2.9696	2.9922
	L^1 -error	4.2513E-04	5.4671E-05	6.8958E-06	8.6307E-07
	L^1 -order	—	2.9590	2.9870	2.9982
E	L^∞ -error	4.1011E-04	5.2523E-05	6.6514E-06	8.3368E-07
	L^∞ -order	—	2.9650	2.9812	2.9961
	L^2 -error	3.2336E-04	4.2374E-05	5.4071E-06	6.7950E-07
	L^2 -order	—	2.9319	2.9702	2.9923
	L^1 -error	4.9739E-04	6.5928E-05	8.3778E-06	1.0506E-06
	L^1 -order	—	2.9154	2.9762	2.9954

Example 9.4. 1D inviscid Burgers' problem with smooth initial conditions and a shock during the evolution.

Control Eq: $u_t + (\frac{1}{2}u^2)_x = 0$;

Computational domain: $\Omega \times [0, T_{end}] = [0, 2\pi] \times [0, 2.0]$;

ICs: $u_0(x) = \sin(x)$;

Note: a discontinuity occurs at $t = 1$ and $x = \pi$ under this given initial condition;

BCs: *periodic boundary conditions*;

True solutions: $u(x, t) = u_0(x^*)$, x^* satisfies $x^* + u_0(x^*) \cdot t = x$;

TABLE 7. P^3 -FVS(AUSM)-DG using [uniform structured rectangular meshes](#). 2D-Compressible Euler Equations with smooth initial conditions: $\rho_0(x, y) = 1 + 0.2 \cos(\pi x + \pi y)$, $u_0(x, y) = -0.7$, $v_0(x, y) = 0.3$, $P_0(x, y) = 1$. L^∞ , L^2 , L^1 errors and convergence orders for conservative variables ρ , ρu , ρv , E .

	Mesh	10×10	20×20	40×40	80×80
ρ	L^∞ -error	1.5314E-04	9.0975E-06	5.6476E-07	3.2868E-08
	L^∞ -order	—	4.0732	4.0098	4.1029
	L^2 -error	1.2555E-04	8.8175E-06	5.9757E-07	2.8805E-08
	L^2 -order	—	3.8317	3.8832	4.3747
	L^1 -error	2.0791E-04	1.4274E-05	1.0451E-06	4.8381E-08
	L^1 -order	—	3.8645	3.7717	4.4331
ρu	L^∞ -error	9.2719E-05	5.8423E-06	3.6815E-07	1.9127E-08
	L^∞ -order	—	3.9883	3.9882	4.2666
	L^2 -error	8.0222E-05	5.7501E-06	3.9938E-07	1.8463E-08
	L^2 -order	—	3.8024	3.8477	4.4351
	L^1 -error	1.3091E-04	9.3984E-06	6.9878E-07	3.0859E-08
	L^1 -order	—	3.8000	3.7495	4.5010
ρv	L^∞ -error	5.6795E-05	3.1587E-06	1.7577E-07	1.1372E-08
	L^∞ -order	—	4.1684	4.1675	3.9502
	L^2 -error	4.2407E-05	2.7470E-06	1.8497E-07	9.3140E-09
	L^2 -order	—	3.9484	3.8925	4.3118
	L^1 -error	7.0188E-05	4.3302E-06	3.2201E-07	1.5252E-08
	L^1 -order	—	4.0187	3.7493	4.4000
E	L^∞ -error	7.6232E-05	4.8337E-06	2.8666E-07	1.6498E-08
	L^∞ -order	—	3.9792	4.0757	4.1190
	L^2 -error	5.7506E-05	3.8493E-06	2.3815E-07	1.2923E-08
	L^2 -order	—	3.9011	4.0146	4.2038
	L^1 -error	9.5498E-05	6.1273E-06	4.1033E-07	2.1644E-08
	L^1 -order	—	3.9621	3.9004	4.2448

Flux format: local Lax-Friedrichs flux;

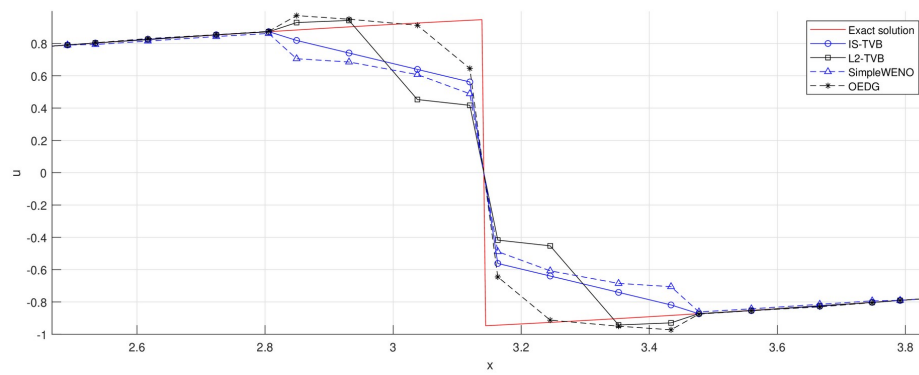
Temporal discretization format: TVD-RK3;

CFL=0.1;

P^3 and P^5 -polynomial approximations and uniquely spaced cells are utilized;

Note that in order to fully demonstrate the influence of various limiters on numerical results, [no discontinuity indicators are used in this Example 9.4](#).

Numerical results based on different limiters including IS-TVB-minmod limiter, L^2 -TVB-minmod limiter, SimpleWENO and OEDG [44] are demonstrated in [Figure 2](#). Note: when



(a) 20 cells, P^3 -polynomial approximation with different limiters

TABLE 8. P^2 -FVS(Steger-Warming)-DG using **uniform structured triangular meshes**. 2D-Compressible Euler Equations with smooth initial conditions: $\rho_0(x, y) = 1 + 0.2 \cos(\pi x + \pi y)$, $u_0(x, y) = -0.7$, $v_0(x, y) = 0.3$, $P_0(x, y) = 1$. L^∞ , L^2 , L^1 errors and convergence orders for conservative variables ρ , ρu , ρv , E .

	Mesh	8×8	16×16	32×32	64×64
ρ	L^∞ -error	9.3377E-04	1.3671E-04	1.7590E-05	2.1995E-06
	L^∞ -order	—	2.7719	2.9583	2.9996
	L^2 -error	8.1292E-04	1.1096E-04	1.4344E-05	1.8091E-06
	L^2 -order	—	2.8730	2.9515	2.9872
	L^1 -error	1.3097E-03	1.7368E-04	2.1929E-05	2.7399E-06
	L^1 -order	—	2.9147	2.9856	3.0006
ρu	L^∞ -error	5.9550E-04	8.5540E-05	1.1053E-05	1.3838E-06
	L^∞ -order	—	2.7994	2.9522	2.9977
	L^2 -error	5.1786E-04	7.0930E-05	9.1898E-06	1.1599E-06
	L^2 -order	—	2.8681	2.9483	2.9861
	L^1 -error	8.4123E-04	1.1217E-04	1.4225E-05	1.7796E-06
	L^1 -order	—	2.9069	2.9791	2.9988
ρv	L^∞ -error	3.7102E-04	5.5142E-05	7.1893E-06	9.1021E-07
	L^∞ -order	—	2.7503	2.9392	2.9816
	L^2 -error	3.0768E-04	4.3852E-05	5.7438E-06	7.2709E-07
	L^2 -order	—	2.8107	2.9326	2.9818
	L^1 -error	4.9562E-04	6.8821E-05	8.8443E-06	1.1098E-06
	L^1 -order	—	2.8483	2.9600	2.9945
E	L^∞ -error	4.2025E-04	6.0872E-05	7.8618E-06	9.9419E-07
	L^∞ -order	—	2.7874	2.9528	2.9833
	L^2 -error	3.4478E-04	4.9342E-05	6.4757E-06	8.2032E-07
	L^2 -order	—	2.8048	2.9297	2.9808
	L^1 -error	5.4155E-04	7.7310E-05	1.0117E-05	1.2801E-06
	L^1 -order	—	2.8084	2.9339	2.9824

using P^5 -polynomial approximation, SimpleWENO doesn't work with 20 cells. (the numerical results are "NaN")

According to the Figure 2, we can see that when approximating with high-order polynomials, the IS-TVB-minmod limiter exhibits no overshoot near discontinuities, whereas both the L^2 -TVB-minmod limiter and OEDG show overshoot phenomena; When approximating with P^3 -polynomial, SimpleWENO, although not overshoot near discontinuities, has slight oscillations and is not as smooth as the IS-TVB-minmod limiter; Under the same CFL number (0.1) and the same coarse mesh (20 cells) conditions, when approximating with P^5 -polynomial,

SimpleWENO does not work (in fact, even if adding up cells to 500, SimpleWENO still doesn't work), while the IS-TVB-minmod limiter remains effective, demonstrating that the mesh resolution of the IS-TVB-minmod limiter is superior to that of SimpleWENO and IS-TVB-minmod limiter is more suitable for high-order approximation than SimpleWENO. Compared with the IS-TVB-minmod limiter, OEDG requires more cells to mitigate overshoot phenomena, in fact, OEDG still has overshoot even with 500 cells, while IS-TVB-minmod limiter consistently maintains no overshoot during mesh refinement, even on a very coarse mesh (20 cells), there is no overshoot phenomenon with IS-TVB-minmod limiter. Therefore, the artificial viscosity and the mesh resolution of the IS-TVB-minmod limiter is also superior to that of OEDG. Finally, it needs to be emphasized that as the mesh is refined, the discontinuities captured by IS-TVB-minmod limiter become increasingly clear and sharp, yet there is still no overshoot.

Example 9.5. *nonlinear non-convex scalar Buckley-Leverett problem.*

Control Eq: $u_t + \left(\frac{4u^2}{4u^2+(1-u)^2}\right)_x = 0;$

Computational domain: $\Omega \times [0, T_{end}] = [-1, 1] \times [0, 0.4];$

$$ICs: u_0(x) = \begin{cases} 1, & -\frac{1}{2} \leq x \leq 0, \\ 0, & \text{elsewhere;} \end{cases}$$

BCs: *periodic boundary conditions;*

Flux format: *global Lax-Friedrichs flux with fixed $\alpha = 2.4$;*

Temporal discretization format: *TVD-RK3;*

CFL=0.1;

P^3 -polynomial approximation and uniquely spaced 80 cells are utilized;

TVB-minmod discontinuity indicator with parameter $M = 1$ is utilized for all limiters in this Example 9.5.

Numerical results based on different limiters including IS-TVB-minmod limiter, L^2 -TVB-minmod limiter, WENO5-JS, SimpleWENO and OEDG are demonstrated in Figure 3.

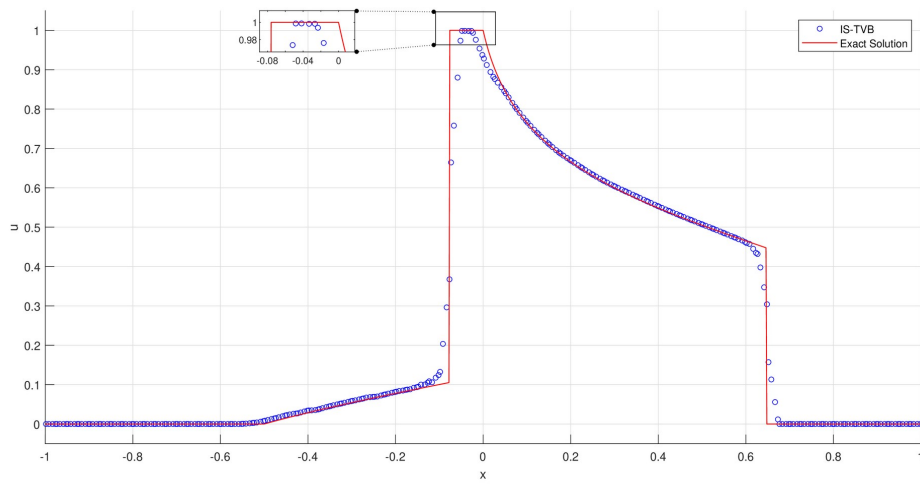
The IS-TVB exhibits no overshoot behavior, while the SimpleWENO and OEDG exhibit noticeable overshoot phenomena; the IS-TVB has no numerical spurious oscillations, whereas the L^2 -TVB shows noticeable numerical oscillations within the range $-0.2 \leq x \leq 0$; although the WENO5-JS does not have numerical oscillations, its dissipation is significantly larger and its precision is inferior to that of the IS-TVB within the range $-0.06 \leq x \leq -0.02$.

9.2.2. *Two-dimensional Test Cases for Scalar Equations.* In this subsection, in addition to LLF flux, we used “**scalar Steger-Warming flux**” inspired by **Steger-Warming splitting for systems**. More details about “scalar Steger-Warming flux”, please refer to Appendix C.2.2.

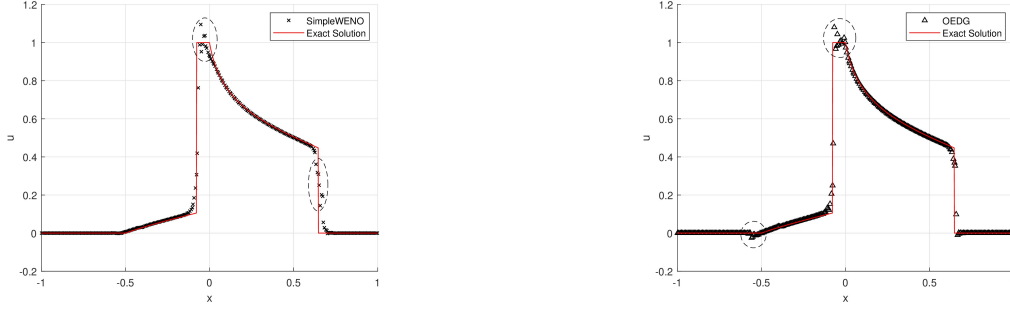
Example 9.6. *2D nonlinear inviscid Burgers' problem.*

Control Eq: $U_t + \left(\frac{1}{2}U\right)_x + \left(\frac{1}{2}U\right)_y = 0;$

- *2D-Burgers' problem with smooth initial conditions and a shock during the evolution*



(a) 80 cells, P^3 -polynomial approximation with IS-TVB



(d) 80 cells, P^3 -polynomial approximation with SimpleWENO (e) 80 cells, P^3 -polynomial approximation with OEDG

FIGURE 3. 1D-Buckley-Leverett problem $u_t + (\frac{4u^2}{4u^2+(1-u)^2})x = 0$ with initial condition $u = 1$ when $-\frac{1}{2} \leq x \leq 0$ and $u = 0$ elsewhere. The simulation is performed up to time $t = 0.4$. P^3 -polynomial approximation and uniquely spaced 80 cells. Numerical results based on different limiters including IS-TVb-minmod limiter, L^2 -TVb-minmod limiter, WENO5-JS, SimpleWENO and OEDG are compared with each other.

Computational domain: $\Omega \times [0, T_{end}] = \{[0, 4] \times [0, 4]\} \times [0, \frac{1.5}{\pi}]$;

ICs: $U_0(x, y) = \sin(\frac{\pi}{2}(x + y))$

BCs: *periodic boundary conditions*;

Flux format: *local Lax-Friedrichs flux*;

Temporal discretization format: *TVD-RK3*;

$CFL=0.1$;

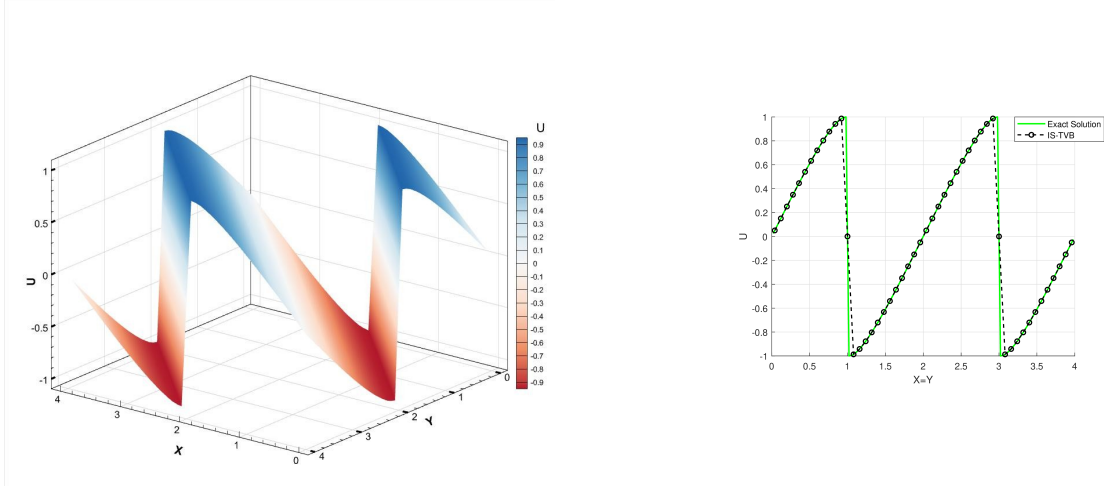
P^3 -polynomial approximation and uniform rectangular 50×50 cells are utilized;

TVb-minmod discontinuity indicator with parameter $M = 1$ is utilized in this test case.

Numerical results based on IS-TVb-minmod limiter ($\omega_{IS} = 1, \omega_{L^2} = 0$) are demonstrated in Figure 4.

- 2D-Burgers' problem with discontinuous initial conditions

Computational domain: $\Omega \times [0, T_{end}] = \{[0, 0.1] \times [0, 0.1]\} \times [0, 0.05]$;



(a) 3D view for IS-TVB numerical solution

(b) IS-TVB solution vs. exact solution along the cut $y = x$

FIGURE 4. 2D-Burgers' problem $U_t + (\frac{1}{2}U)_x + (\frac{1}{2}U)_y = 0$ with smooth initial conditions $U_0(x, y) = \sin(\frac{\pi}{2}(x + y))$ while a shock during the evolution. The simulation is performed up to time $t = \frac{1.5}{\pi}$ when a shock has already appeared. P^3 -polynomial approximation and uniquely spaced rectangular 50×50 cells. Discontinuity is captured based on IS-TVB-minmod limiter ($\omega_{IS} = 1$, $\omega_{L2} = 0$).

$$ICs: u_0(x, y) = \begin{cases} 0.5, & x < 0.05, y < 0.05 \\ 0.8, & x > 0.05, y < 0.05 \\ -1, & x > 0.05, y > 0.05 \\ -0.2, & x < 0.05, y > 0.05 \end{cases}$$

BCs: *free boundary conditions are imposed on all edges of Ω* ;

Flux format: *Steger-Warming flux (please refer to Appendix C.2.2)*;

Temporal discretization format: *TVD-RK3*;

CFL=0.1;

P^3 -polynomial approximation and uniform rectangular 50×50 cells are utilized;

TVB-minmod discontinuity indicator with parameter $M = 1$ is utilized in this test case.

Numerical results based on IS-TVB-minmod limiter ($\omega_{IS} = 1$, $\omega_{L2} = 0$) are demonstrated in Figure 5.

Example 9.7. *2D linear advection equation.*

• *Swirling deformation flow [45].*

*Control Eqs: $U_t - (\cos^2(\frac{x}{2}) \sin(y)g(t)U)_x + (\sin(x) \cos^2(\frac{y}{2})g(t)U)_y = 0$,
where $g(t) = 2\pi \cos(\pi t/T)$, $T = 0.75$.*

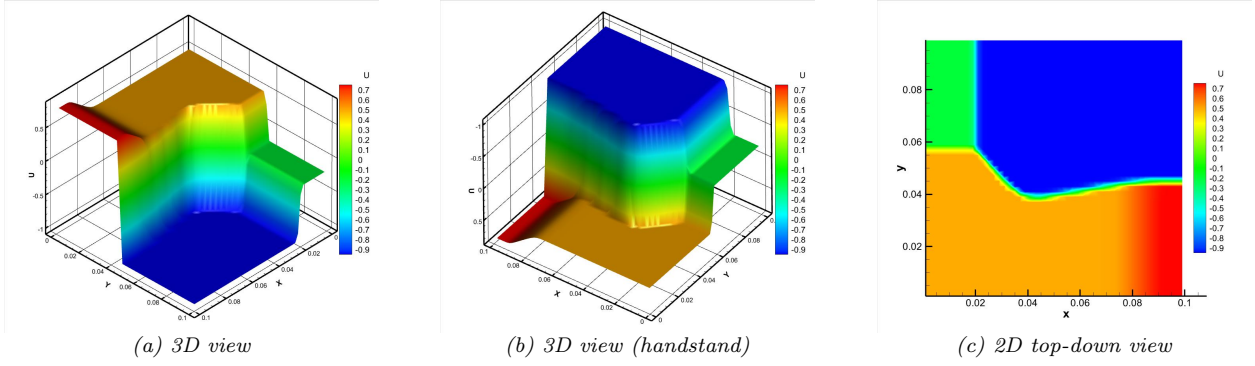


FIGURE 5. 2D-Burgers' problem with discontinuous initial conditions. $U_t + (\frac{1}{2}U)_x + (\frac{1}{2}U)_y = 0$. At $t = 0$, $U_0(x, y) = 0.5$ when $x < 0.05$ and $y < 0.05$, $U_0(x, y) = 0.8$ when $x > 0.05$ and $y < 0.05$, $U_0(x, y) = -1$ when $x > 0.05$ and $y > 0.05$, $U_0(x, y) = -0.2$ when $x < 0.05$ and $y > 0.05$. The simulation is performed up to time $t = 0.05$. P^3 -polynomial approximation and uniform rectangular 50×50 cells. Discontinuity is dealt with IS-TVB-minmod limiter ($\omega_{IS} = 1$, $\omega_{L^2} = 0$).

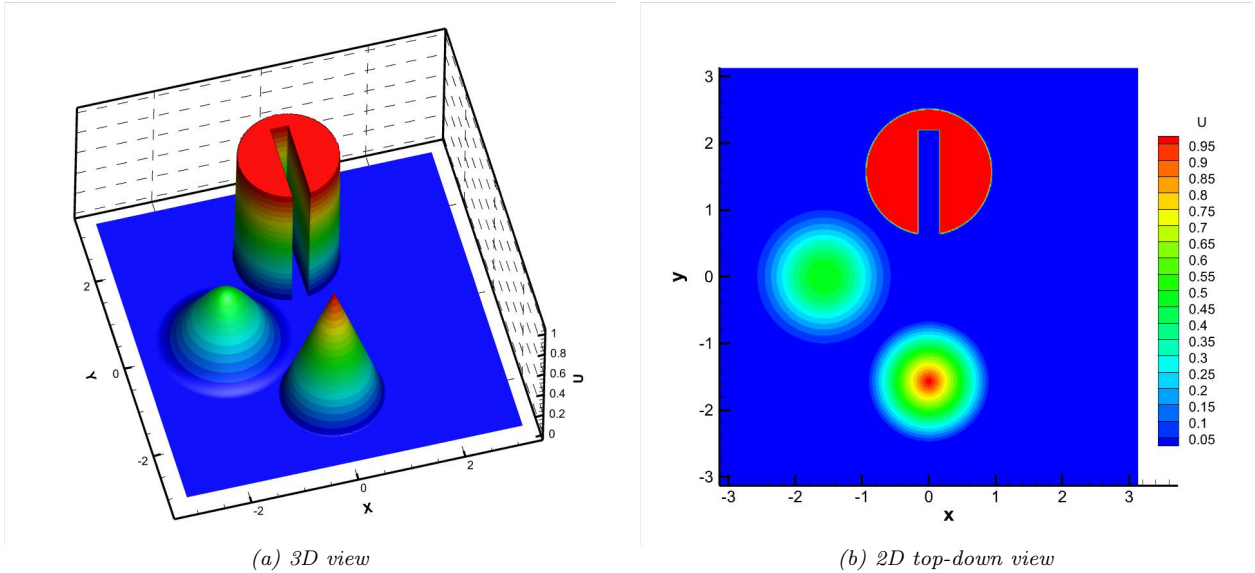


FIGURE 6. initial profile for rigid body rotation and swirling deformation flow

Computational Domain: $\Omega \times [0, T_{end}] = \{[-\pi, \pi] \times [-\pi, \pi]\} \times [0, 0.75]$

ICs: initial condition plotted in Figure.6 which consists of a slotted disk, a cone as well as a smooth hump, similar to the one used in [45].

BCs: *periodic boundary conditions;*

Flux format: Steger-Warming flux (please refer to Appendix C.2.2);

Temporal discretization format: TVD-RK3;

CFL=0.1;

P^3 -polynomial approximation and uniform rectangular 120×120 cells are utilized;

TVB-minmod discontinuity indicator with parameter $M = 1$ is utilized in this test case.

Numerical results based on IS- L^2 -TVB-minmod limiter ($\omega_{IS} = 0.75$, $\omega_{L^2} = 0.25$) are demonstrated in Figure 7 and Figure 8.

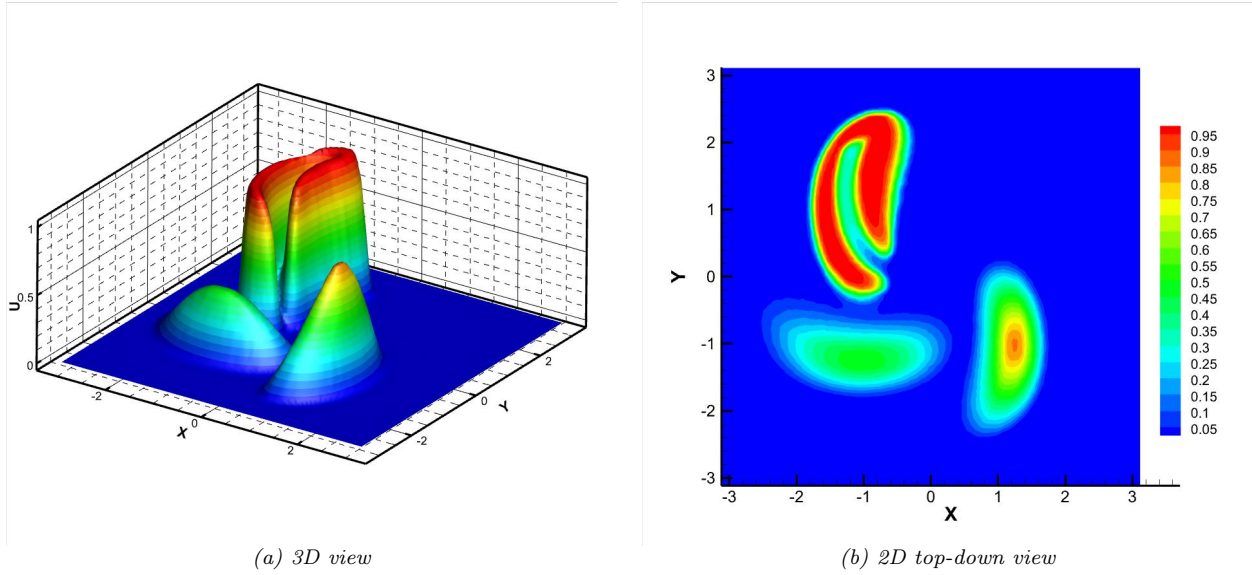


FIGURE 7. 2D-swirling deformation flow $U_t - (\cos^2(\frac{x}{2}) \sin(y)g(t)U)_x + (\sin(x) \cos^2(\frac{y}{2})g(t)U)_y = 0$, $g(t) = 2\pi \cos(\pi t/T)$ with initial data in Fig.6: **the period $T = 0.75$ and the final integration time 0.375** . P^3 -polynomial approximation with IS- L^2 -TVB-minmod limiter ($\omega_{IS} = 0.75$, $\omega_{L^2} = 0.25$). Uniform rectangular 120×120 cells.

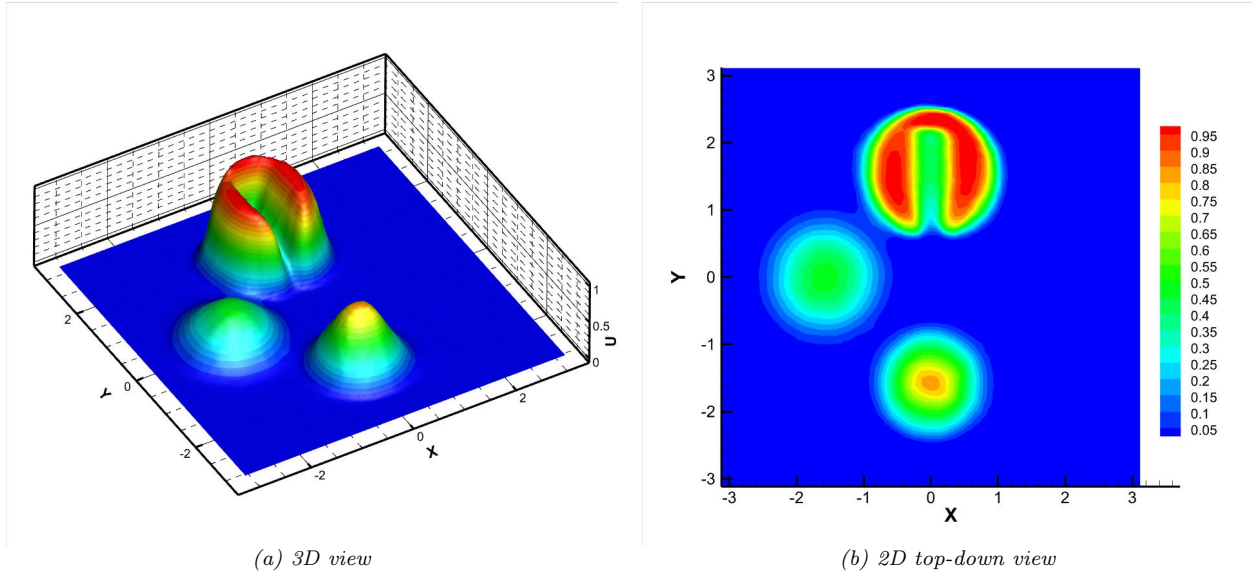


FIGURE 8. 2D-swirling deformation flow $U_t - (\cos^2(\frac{x}{2}) \sin(y)g(t)U)_x + (\sin(x) \cos^2(\frac{y}{2})g(t)U)_y = 0$, $g(t) = 2\pi \cos(\pi t/T)$ with initial data in Fig.6: **the period $T = 0.75$ and the final integration time 0.75** . P^3 -polynomial approximation with IS- L^2 -TVB-minmod limiter ($\omega_{IS} = 0.75$, $\omega_{L^2} = 0.25$). Uniform rectangular 120×120 cells.

9.3. Solving the Riemann Problems for Hyperbolic Conservative Systems Using FVS-DG with IS- L^2 -TVB(D)-minmod Limiter.

9.3.1. *One-dimensional Test Cases for Hyperbolic Systems.* In this subsection, several classical one-dimensional Riemann problems related to the compressible Euler equations and

shallow water wave equations are numerically simulated to test the suitability of the IS- L^2 -TVB(D)-minmod limiter with high-order polynomial approximations (P^K , $K \geq 3$) and its ability to suppress numerical oscillations.

- Riemann Problems for one-dimensional compressible Euler system.

Example 9.8. *Sod problem with compressible Euler equations.*

ICs:

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

Free boundary conditions are applied on both the left and right boundaries;

Computational Domain:

$$\Omega \times [0, T_{end}] = [-1, 1] \times [0, 0.2];$$

Limiter parameters: IS-TVB-minmod limiter, $\omega_{IS} = 1.0$, $\omega_{L^2} = 0.0$;

TroubledCell-Indicator: TVB-minmod discontinuous indicator and take $M = 1$ into effect;

P^3 -polynomial approximation was applied;

Numerical flux format: Steger-Warming;

Temporal discretization format: TVD-RK3;

CFL=0.05;

Mesh with 400 cells was applied.

At $t = 0.2$, the density ρ , velocity u , pressure P and total energy E are plotted in Figure 9.

Example 9.9. *Lax problem with compressible Euler equations.*

ICs:

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528), & -5 < x < 0 \\ (0.5, 0, 0.571), & 0 < x < 5 \end{cases}$$

Free boundary conditions are applied on both the left and right boundaries;

Computational Domain:

$$\Omega \times [0, T_{end}] = [-5, 5] \times [0, 1.3];$$

Limiter parameters: IS-TVB-minmod limiter, $\omega_{IS} = 1.0$, $\omega_{L^2} = 0.0$;

TroubledCell-Indicator: TVB-minmod discontinuous indicator and take $M = 1$ into effect;

P^3 -polynomial approximation was applied;

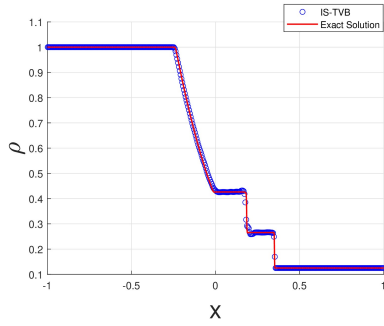
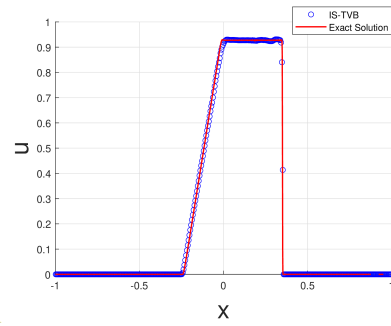
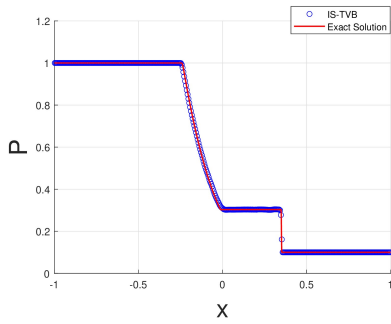
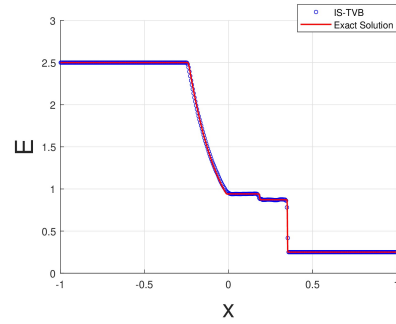
Numerical flux format: van Leer;

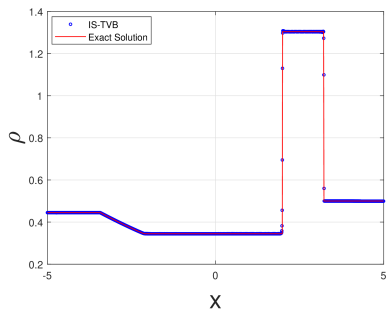
Temporal discretization format: TVD-RK3;

CFL=0.1;

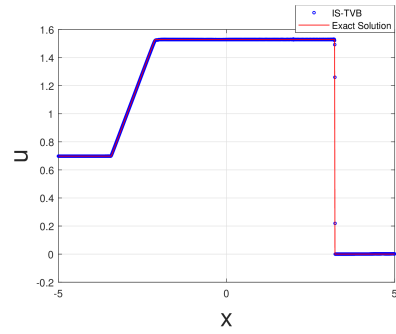
Uniform mesh with 2000 cells was applied.

At $t = 1.3$, the density ρ , velocity u , pressure P and total energy E are plotted in Figure 10.

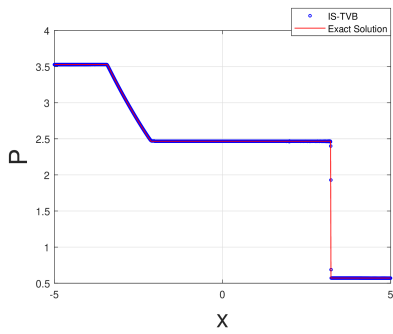
(a) Density ρ (b) Velocity u (c) Pressure P (d) Total energy E



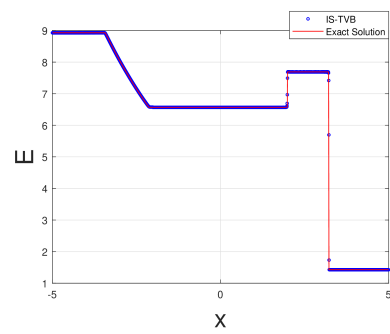
(a) Density ρ



(b) Velocity u



(c) Pressure P



(d) Total energy E

Example 9.10. *Shu-Osher problem with compressible Euler equations.*

ICs:

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333), & -5 < x < -4 \\ (1 + 0.2 \sin(5x), 0, 1), & -4 < x < 5 \end{cases}$$

Free boundary conditions are applied on both the left and right boundaries;

Computational Domain:

$$\Omega \times [0, T_{end}] = [-5, 5] \times [0, 1.8];$$

Limiter parameters: IS-L²-TVB-minmod limiter, $\omega_{IS} = 0.75$, $\omega_{L^2} = 0.25$;

TroubledCell-Indicator: TVB-minmod discontinuous indicator and take $M = 1$ into effect;

P⁵-polynomial approximation was applied;

Numerical flux format: Steger-Warming;

Temporal discretization format: TVD-RK3;

CFL=0.1;

Uniform mesh with 500 cells was applied.

At $t = 1.8$, the density ρ , velocity u , pressure P and total energy E are plotted in Figure 11.

Example 9.11. *Blast problem with compressible Euler equations.*

ICs:

$$(\rho, u, p) = \begin{cases} (1, 0, 10^3), & 0 \leq x < 0.1 \\ (1, 0, 10^{-2}), & 0.1 < x \leq 0.9 \\ (1, 0, 10^2), & 0.9 < x \leq 1 \end{cases}$$

Reflect boundary conditions are applied on both the left and right boundaries;

Computational Domain:

$$\Omega \times [0, T_{end}] = [0, 1] \times [0, 0.026];$$

Limiter parameters: IS-L²-TVB-minmod limiter, $\omega_{IS} = 0.8$, $\omega_{L^2} = 0.2$;

TroubledCell-Indicator: TVB-minmod discontinuity indicator and take $M = 1$ into effect;

P²-polynomial approximation was applied;

Numerical flux format: AUSM;

Temporal discretization format: TVD-RK3;

CFL=0.005;

uniform mesh with 800 cells was applied.

At $t = 0.026$, the density ρ is plotted in Figure 12.

- Riemann Problems for one-dimensional shallow water wave system.

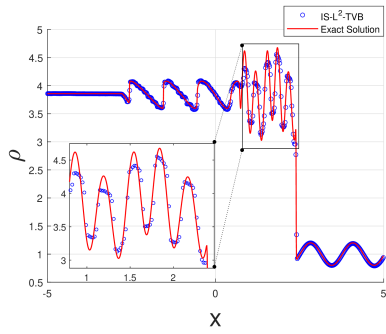
Example 9.12. *Dam break on a flat bed with shallow water wave equations.*

Bottom elevation:

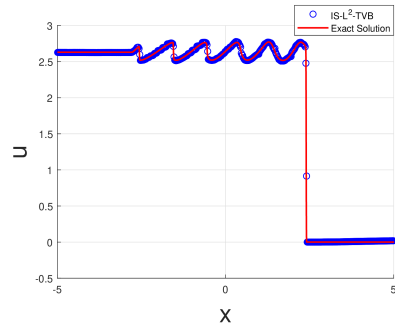
$$Z_0 \equiv 0;$$

Computational domain:

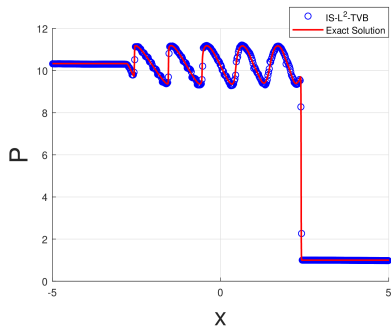
$$\Omega \times [0, T_{end}] = [-1, 1] \times [0, 0.2];$$



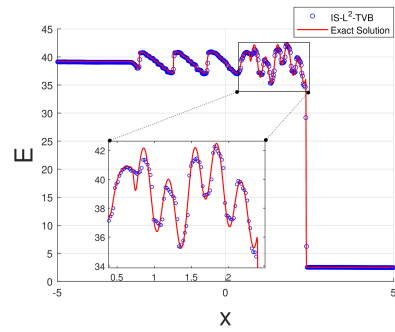
(a) Density ρ



(b) Velocity u



(c) Pressure P



(d) Total energy E

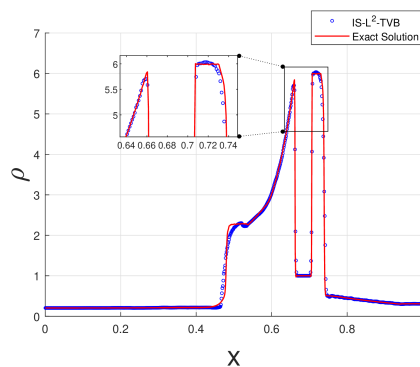


FIGURE 12. Blast problem for 1D-Compressible Euler Equations. P^2 -FVS(AUSM)-DG with IS- L^2 -TVB-minmod limiter ($\omega_{IS} = 0.8$, $\omega_{L^2} = 0.2$) using equally spaced 800 cells. At $t = 0.026$, physical variable ρ is demonstrated.

Riemann ICs:

$$(h, u) = \begin{cases} (1, 0), & -1 < x < 0, \\ (0.1, 0), & 0 < x < 1; \end{cases}$$

BCs: free boundary conditions are applied on both the left and right boundaries;

P^4 -polynomial approximation was applied;

Limiter parameters: IS- L^2 -TVB-minmod limiter, $\omega_{IS} = 0.75$, $\omega_{L^2} = 0.25$;

TroubledCell-Indicator: TVD-minmod discontinuous indicator;

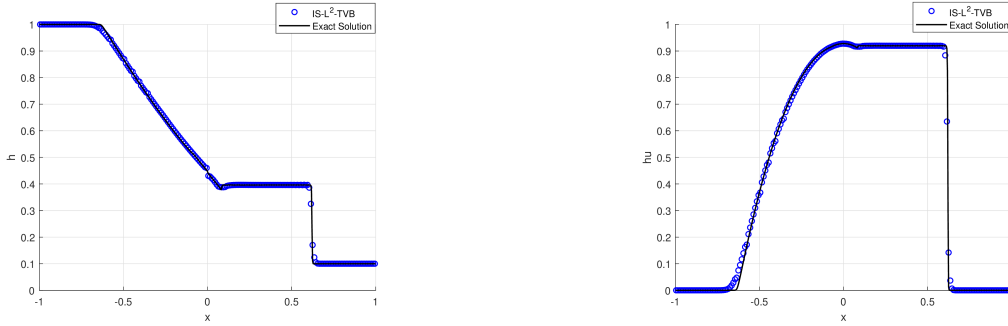
Numerical flux format: van Leer;

Temporal discretization format: TVD-RK3;

CFL=0.1;

Uniform mesh with 200 cells was applied.

At $t = 0.2$, the surface level h and discharge hu are plotted in Figure 13.



(a) DamBreak: surface level h at $t = 0.2$

(b) DamBreak: discharge hu at $t = 0.2$

FIGURE 13. Dam break at a flat bed with shallow water wave equations. P^4 -FVS(vanLeer)-DG with IS- L^2 -TVB-minmod limiter using equally spaced 200 cells. At $t = 0.2$, conservative variables h , hu are demonstrated.

9.3.2. *Two-dimensional Test Cases for Hyperbolic Systems.* Three Riemann problems in 2D case for compressible Euler equations are simulated here to demonstrate the performance of IS- L^2 -TVB(D)-Limiter in conjunction with FVS-DG.

Example 9.13.

Computational Domain: $\Omega \times [0, T_{end}] = \{[0, 0.1] \times [0, 0.1]\} \times [0, 0.022]$;

ICs(1):

$$(\rho, u, v, p)^T = \begin{cases} (0.5313, 0, 0, 0.4)^T, & x > 0.05, y > 0.05, \\ (1, 0.7276, 0, 1)^T, & x < 0.05, y > 0.05, \\ (0.8, 0, 0, 1)^T, & x < 0.05, y < 0.05, \\ (1, 0, 0.7276, 1)^T, & x > 0.05, y < 0.05, \end{cases}$$

BCs: free boundary conditions are imposed on all edges of Ω ;

P^3 -polynomial approximation was applied;

Limiter parameters: IS- L^2 -TVB-minmod limiter, $\omega_{IS} = 0.8$, $\omega_{L^2} = 0.2$;

TroubledCell-Indicator: TVB-minmod discontinuity indicator with $M = 1$;

Numerical Flux Format: Steger-Warming;

Temporal Discretization: TVD-RK3;

CFL: 0.2;

Uniform rectangular mesh with 40×40 cells.

The numerical result of density at $t = 0.022$ is plotted in Figure 14.

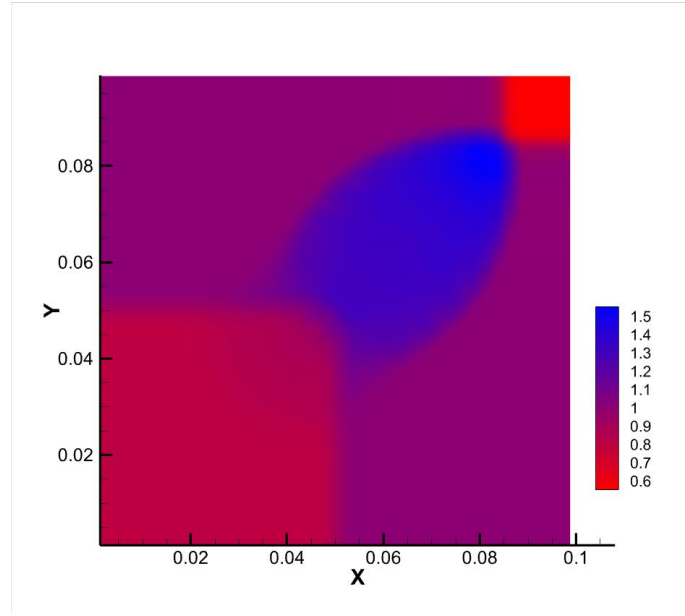


FIGURE 14. 2D Euler equations for Riemann problem with ICs(1). P^3 -FVS(Steger-Warming)-DG with IS- L^2 -TVB-minmod limiter ($\omega_{IS} = 0.8$, $\omega_{L^2} = 0.2$). Uniform rectangular mesh with 40×40 cells. The density distribution at $t = 0.022$

Example 9.14.

Computational Domain: $\Omega \times [0, T_{end}] = \{[0, 1] \times [0, 1]\} \times [0, 0.25]$;

ICs(2):

$$(\rho, u, v, p)^T = \begin{cases} (1.1, 0, 0, 1.1)^T, & x > 0.5, y > 0.5 \\ (0.5065, 0.8939, 0, 0.35)^T, & x < 0.5, y > 0.5 \\ (1.1, 0.8939, 0.8939, 1.1)^T, & x < 0.5, y < 0.5 \\ (0.5065, 0, 0.8939, 0.35)^T, & x > 0.5, y < 0.5 \end{cases}$$

BCs: free boundary conditions are imposed on all edges of Ω ;

P^2 -polynomial approximation was applied;

Limiter parameters: IS-TVb-minmod limiter, $\omega_{IS} = 1.0$, $\omega_{L^2} = 0.0$;

TroubledCell-Indicator: TVB-minmod discontinuity indicator with $M = 1$;

Numerical Flux Format: AUSM;

Temporal Discretization: TVD-RK3;

CFL: 0.2;

Uniform rectangular mesh with 100×100 cells.

The numerical result of density at $t = 0.25$ is plotted in Figure 15.

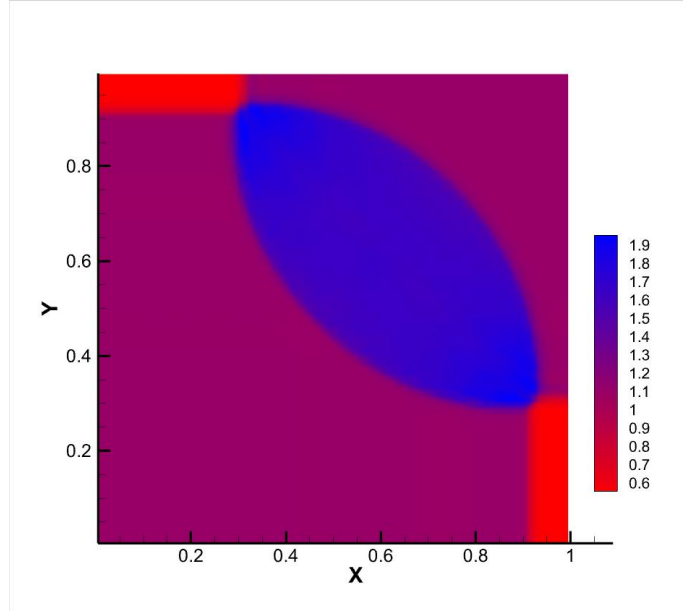


FIGURE 15. 2D Euler equations for Riemann problem with ICs(2). P^2 -FVS(AUSM)-DG with IS-TVb-minmod limiter ($\omega_{IS} = 1.0$, $\omega_{L^2} = 0.0$). Uniform rectangular mesh with 100×100 cells. The density distribution at $t = 0.25$

Example 9.15.

Computational Domain: $\Omega \times [0, T_{end}] = \{[0, 1] \times [0, 1]\} \times [0, 0.3]$;

ICs(3):

$$(\rho, u, v, p)^T = \begin{cases} (0.5313, 0, 0, 0.4)^T, & x > 0.5, y > 0.5, \\ (1, 0.7276, 0, 1)^T, & x < 0.5, y > 0.5, \\ (0.8, 0, 0, 1)^T, & x < 0.5, y < 0.5, \\ (1, 0, 0.7276, 1)^T, & x > 0.5, y < 0.5, \end{cases}$$

BCs: free boundary conditions are imposed on all edges of Ω ;

P^2 -polynomial approximation was applied;

Limiting parameters: IS- L^2 -TVb-minmod limiter, $\omega_{IS} = 0.8$, $\omega_{L^2} = 0.2$;

TroubledCell-Indicator: TVb-minmod discontinuity indicator with $M = 1$;

Numerical Flux Format: Steger-Warming;

Temporal Discretization: TVD-RK3;

CFL: 0.2;

Uniform rectangular mesh with 100×100 cells.

The numerical result of density at $t = 0.3$ is plotted in Figure 16.

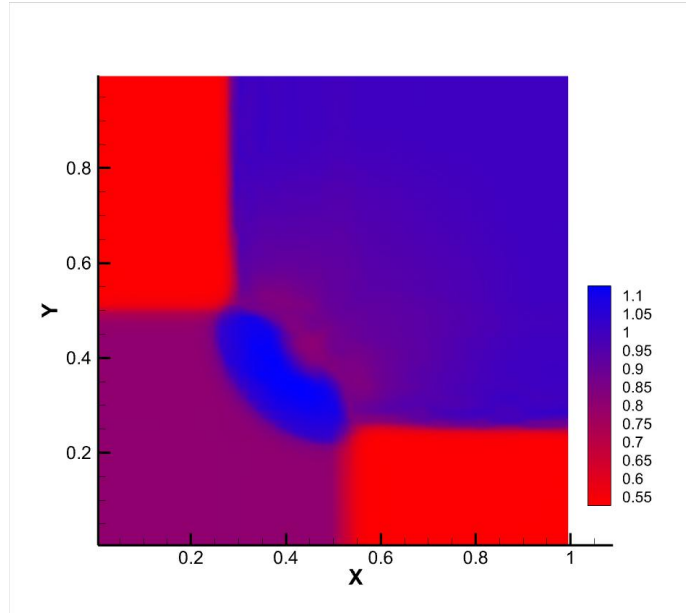


FIGURE 16. 2D Euler equations for Riemann problem with ICs(3). P^2 -FVS(Steger-Warming)-DG with IS- L^2 -TVB-minmod limiter ($\omega_{IS} = 0.8$, $\omega_{L^2} = 0.2$). Uniform rectangular mesh with 100×100 cells. The density distribution at $t = 0.3$

10. CONCLUSION

This paper constructs the numerical flux required for the DG spatial discretization scheme based on the flux vector splitting method, which is implemented in two-dimensional problems by introducing the normal flux on cell interfaces. The TVB(D) limiter and the WENO limiter are currently the two main streams of limiters; we utilize the smoothness factor IS from the WENO limiter in the TVB(D)-minmod limiter, successfully overcoming the ill-posed problem of the TVB(D)-minmod limiter in the case of high-order polynomial approximation by constructing an optimization problem based on the smoothness factor constraint. Furthermore, by drawing on existing practices and introducing the L^2 -error constraint, the TVB(D)-minmod limiter is able to balance the suppression of oscillations and the maintenance of high precision. For local characteristic decomposition, we use an arithmetic average or Roe average of the integral mean on the common interface of adjacent cells for local freezing, which is different from the usual practice. We also use an interpolation-based characteristic transformation to achieve the mutual conversion between the one-dimensional model in physical space and characteristic space. Numerical experiments have confirmed the effectiveness of the aforementioned work.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous referees for their very valuable comments and suggestions.

REFERENCES

- [1] HOTEIT H, ACKERER P, MOSE R, et al. New two-dimensional slope limiters for discontinuous Galerkin methods on arbitrary meshes [J]. *International Journal for Numerical Methods in Engineering*, 2004, 61: 2566-93. (document), 1, 7.2, 7.2.3, 7.2.3
- [2] ZHONG X, SHU C-W. A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods [J]. *Journal of Computational Physics*, 2013, 232(1): 397-415. 1
- [3] DAFERMOS C M. *Hyperbolic Conservation Laws in Continuum Physics* [M]. 4th ed. Springer. 2016: XII Introduction. 1
- [4] VAN LEER B. Flux-vector splitting for the Euler equations; proceedings of the Eighth International Conference on Numerical Methods in Fluid Dynamics: Proceedings of the Conference, Rheinisch-Westfälische Technische Hochschule Aachen, Germany, June 28–July 2, 1982, F, 2005 [C]. Springer. 1, 3.2.1
- [5] VAN LEER B, THOMAS J L, ROE P L, et al. A comparison of numerical flux formulas for the Euler and Navier-Stokes equations [J]. 1987. 1
- [6] STEGER J L, WARMING R. Flux vector splitting of the inviscid gasdynamic equations with application to finite-difference methods [J]. *Journal of computational physics*, 1981, 40(2): 263-93. 1
- [7] BERMÚDEZ A, VÁZQUEZ M. Flux-vector and flux-difference splitting methods for the shallow water equations in a domain with variable depth [M]. *Computer Modelling of Seas and Coastal Regions*. Springer. 1992: 255-67. 1, B, B.1.2, B.2.2
- [8] LIOU M-S. Ten years in the making-AUSM-family; proceedings of the 15th AIAA Computational Fluid Dynamics Conference, F, 2001 [C]. 1
- [9] REED W H, HILL T R. Triangular mesh methods for the neutron transport equation [Z]. Conference: National topical meeting on mathematical models and computational techniques for analysis of nuclear systems, Ann Arbor, Michigan, USA, 8 Apr 1973; Other Information: Orig Receipt Date: 31-DEC-73. United States. 1973: Medium: ED; Size: Pages: 23 1
- [10] COCKBURN B, SHU C-W. The Local Discontinuous Galerkin Method for Time-Dependent Convection-Diffusion Systems [J]. *SIAM Journal on Numerical Analysis*, 1998, 35: 2440-63. 1
- [11] YI N, HUANG Y, LIU H. A direct discontinuous Galerkin method for the generalized Korteweg–de Vries equation: Energy conservation and boundary effect [J]. *Journal of Computational Physics*, 2013, 242: 351-66. 1
- [12] CHENG Y, SHU C-W. A discontinuous Galerkin finite element method for time dependent partial differential equations with higher order derivatives [J]. *Math Comput*, 2008, 77: 699-730. 1
- [13] CAI X, BOSCARINO S, QIU J. High order semi-Lagrangian discontinuous Galerkin method coupled with Runge-Kutta exponential integrators for nonlinear Vlasov dynamics [J]. *Journal of Computational Physics*, 2020, 427: 110036. 1
- [14] CAI X, QIU J-M, YANG Y. An Eulerian-Lagrangian discontinuous Galerkin method for transport problems and its application to nonlinear dynamics [J]. *Journal of Computational Physics*, 2021, 439: 110392 1
- [15] COCKBURN B, SHU C-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. II. General framework [J]. *Mathematics of computation*, 1989, 52(186): 411-35. 1
- [16] COCKBURN B, LIN S-Y, SHU C-W. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems [J]. *Journal of computational Physics*, 1989, 84(1): 90-113. 1

- [17] COCKBURN B, HOU S, SHU C-W. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. IV. The multidimensional case [J]. *Mathematics of Computation*, 1990, 54(190): 545-81. [1](#)
- [18] COCKBURN B, SHU C-W. The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems [J]. *Journal of computational physics*, 1998, 141(2): 199-224. [1](#)
- [19] COCKBURN B, SHU C-W. The Runge-Kutta local projection-discontinuous-Galerkin finite element method for scalar conservation laws [J]. *ESAIM: Mathematical Modelling and Numerical Analysis*, 1991, 25(3): 337-61. [1](#)
- [20] BISWAS R, DEVINE K D, FLAHERTY J E. Parallel, adaptive finite element methods for conservation laws [J]. *Applied Numerical Mathematics*, 1994, 14(1-3): 255-83. [1](#)
- [21] REMACLE J-F, FLAHERTY J E, SHEPHARD M S. An adaptive discontinuous Galerkin technique with an orthogonal basis applied to compressible flow problems [J]. *SIAM review*, 2003, 45(1): 53-72. [1](#)
- [22] CAO W, ZHANG Z. Point-wise and cell average error estimates of the DG and LDG methods for 1D hyperbolic and parabolic equations [J]. *SCIENTIA SINICA Mathematica*, 2015, 45(8): 1115-32. [1](#)
- [23] CAO W, HUANG Q. Superconvergence of local discontinuous Galerkin methods for partial differential equations with higher order derivatives [J]. *Journal of Scientific Computing*, 2017, 72: 761-91. [1](#)
- [24] XU X, HUANG Q. Superconvergence of discontinuous Galerkin methods for nonlinear delay differential equations with vanishing delay [J]. *Journal of Computational and Applied Mathematics*, 2019, 348: 314-27. [1](#)
- [25] MENG X, SHU C-W, YANG Y. Superconvergence of discontinuous Galerkin methods for time-dependent partial differential equations [J]. *SCIENTIA SINICA Mathematica*, 2015, 45(7): 1041-60. [1](#)
- [26] CHENG Y, SHU C-W. Superconvergence of local discontinuous Galerkin methods for one-dimensional convection-diffusion equations [J]. *Computers and Structures*, 2009, 87(11): 630-41. [1](#)
- [27] SWEBY P K. High Resolution Schemes Using Flux Limiters for Hyperbolic Conservation Laws [J]. *SIAM Journal on Numerical Analysis*, 1984, 21: 995-1011. [1](#)
- [28] JIANG G-S, SHU C-W. Efficient Implementation of Weighted ENO Schemes [J]. *Journal of Computational Physics*, 1996, 126(1): 202-28. [1](#)
- [29] ZHU J, ZHONG X, SHU C-W, et al. Runge-Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes [J]. *Journal of Computational Physics*, 2013, 248: 200-20. [1](#)
- [30] QIU J, SHU C-W. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case [J]. *Journal of Computational Physics*, 2004, 193(1): 115-35. [1](#)
- [31] QIU J, SHU C-W. Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: Two dimensional case [J]. *Computers and Fluids*, 2005, 34(6): 642-63. [1](#)
- [32] QIU J, SHU C-W. Hermite WENO schemes for Hamilton-Jacobi equations [J]. *Journal of Computational Physics*, 2005, 204(1): 82-99. [1](#)
- [33] HARTEN A. High Resolution Schemes for Hyperbolic Conservation Laws [J]. *Journal of Computational Physics*, 1997, 135(2): 260-78. [1](#)
- [34] SHU C-W. Essentially non-oscillatory and weighted essentially non-oscillatory schemes [J]. *Acta Numerica*, 2020, 29: 701-62. [1](#)
- [35] QIU J, SHU C-W. A Comparison of Troubled-Cell Indicators for Runge-Kutta Discontinuous Galerkin Methods Using Weighted Essentially Nonoscillatory Limiters [J]. *SIAM Journal on Scientific Computing*, 2005, 27(3): 995-1013. [1](#)

- [36] KRIVODONOVA L, XIN J, REMACLE J F, et al. Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws [J]. Applied Numerical Mathematics, 2004, 48(3): 323-38. [1](#)
- [37] SHU C-W, ZANG T A, ERLEBACHER G, et al. High-order ENO schemes applied to two- and three-dimensional compressible flow [J]. Applied Numerical Mathematics, 1992, 9(1): 45-71. [1](#)
- [38] LU C, QIU J, WANG R. A NUMERICAL STUDY FOR THE PERFORMANCE OF THE WENO SCHEMES BASED ON DIFFERENT NUMERICAL FLUXES FOR THE SHALLOW WATER EQUATIONS [J]. Journal of Computational Mathematics, 2010: 807-25. [2.1](#)
- [39] QIU J, KHOO B C, SHU C-W. A numerical study for the performance of the Runge-Kutta discontinuous Galerkin method based on different numerical fluxes [J]. Journal of Computational Physics, 2006, 212(2): 540-65. [2.1](#)
- [40] ROE P L. Approximate Riemann solvers, parameter vectors, and difference schemes [J]. Journal of computational physics, 1981, 43(2): 357-72. [4](#)
- [41] BURGERJON L, KOREN B. Numerical Methods for the Shallow Water Equations [D]. the Netherlands: Eindhoven University of Technology, Department of Mathematics and Computer Science, 2021. [4.2](#)
- [42] CAI X, ZHU J, QIU J. Hermite Weno Schemes with Strong Stability Preserving Multi-Step Temporal Discretization Methods for Conservation Laws [J]. Journal of Computational Mathematics, 2017, 35: 53-74. [9](#)
- [43] LU C, QIU J, WANG R. A NUMERICAL STUDY FOR THE PERFORMANCE OF THE WENO SCHEMES BASED ON DIFFERENT NUMERICAL FLUXES FOR THE SHALLOW WATER EQUATIONS [J]. Journal of Computational Mathematics, 2010: 807-25. [9](#)
- [44] PENG M, SUN Z, WU K. OEDG: Oscillation-eliminating discontinuous Galerkin method for hyperbolic conservation laws [J]. arXiv preprint arXiv:231004807, 2023. [9.4](#)
- [45] CAI X, GUO W, QIU J-M. A High Order Conservative Semi-Lagrangian Discontinuous Galerkin Method for Two-Dimensional Transport Simulations [J]. J Sci Comput, 2017, 73(2-3): 514-42. [9.7](#)
- [46] MASATSUKA K. I do like CFD, VOL.1, Second Edition [M]. Lulu.com, 2009. [B](#)
- [47] SHU C-W. Discontinuous Galerkin methods: General approach and stability [J]. Numerical Solutions of Partial Differential Equations, 2009. [C.2.3](#)
- [48] JIANG G, SHU C-W. On a cell entropy inequality for discontinuous Galerkin methods [J]. Math Comput, 1994, 62(206): 531-8. [C.2.3](#)

APPENDIX A. VECTOR, MATRIX, AND TENSOR OPERATIONS IN FVS-DG

The following notations and operational rules are used for the derivation of the FVS-DG spatial discretization scheme (definition of the FVS-DG weak solution).

- The Hadamard product between two vector-valued functions: $\mathbf{a} \odot \mathbf{b} = (a_1 b_1, a_2 b_2, \dots, a_m b_m)^T$.
- The quasi-Hadamard product between a matrix-valued function and a vector-valued function is defined as

$$\mathbb{H} \odot \mathbf{V} = \begin{bmatrix} \mathbf{H}^{(1)} \cdot v_1 \\ \mathbf{H}^{(2)} \cdot v_2 \\ \vdots \\ \mathbf{H}^{(m)} \cdot v_m \end{bmatrix},$$

where $\mathbf{H}^{(i)} = (h_{i1}, h_{i2}, \dots, h_{id})$, that is, the matrix \mathbb{H} is partitioned row-wise.

- The inner product between vector-valued functions acts component-wise, i.e.,

$$\langle \mathbf{U}, \mathbf{V} \rangle = \begin{bmatrix} \langle u_1, v_1 \rangle \\ \langle u_2, v_2 \rangle \\ \vdots \\ \langle u_m, v_m \rangle \end{bmatrix}.$$

- The Frobenius inner product between vector functions is defined as $\langle \mathbf{U} : \mathbf{V} \rangle = \sum_{i=1}^m \langle u_i, v_i \rangle$.

- Let the matrix-valued function $\mathbb{H} = \begin{bmatrix} \mathbf{H}^{(1)} \\ \mathbf{H}^{(2)} \\ \vdots \\ \mathbf{H}^{(m)} \end{bmatrix}$, where $\mathbf{H}^{(i)} = (h_{i1}, h_{i2}, \dots, h_{id})$, that is,

the matrix \mathbb{H} is partitioned into row blocks.

- Let the matrix-valued function $\mathbb{V} = \begin{bmatrix} \mathbf{V}^{(1)} \\ \mathbf{V}^{(2)} \\ \vdots \\ \mathbf{V}^{(m)} \end{bmatrix}$, where $\mathbf{V}^{(i)} = (v_{i1}, v_{i2}, \dots, v_{id})$, that is,

the matrix \mathbb{V} is partitioned into row blocks.

The Frobenius inner product between the matrix-valued functions is defined to act

component-wise along the rows, i.e., $\langle \mathbb{H} : \mathbb{V} \rangle = \begin{bmatrix} \langle \mathbf{H}^{(1)} : \mathbf{V}^{(1)} \rangle \\ \langle \mathbf{H}^{(2)} : \mathbf{V}^{(2)} \rangle \\ \vdots \\ \langle \mathbf{H}^{(m)} : \mathbf{V}^{(m)} \rangle \end{bmatrix}.$

- When the divergence operator acts on a matrix-valued function, the matrix-valued function is partitioned column-wise for the computation, such that

$$\nabla \cdot \mathbb{H} = \partial_{x_1} \mathbf{H}_{(1)} + \partial_{x_2} \mathbf{H}_{(2)} + \dots + \partial_{x_d} \mathbf{H}_{(d)},$$

where $\mathbf{H}_{(j)} = (h_{1j}, h_{2j}, \dots, h_{mj})^T$ represents the j -th column of matrix \mathbb{H} .

- In this paper, Kronecker product between two vector-valued functions, still denoted by \otimes , is redefined as:

$$\mathbf{a} \otimes \mathbf{b} = \mathbf{a}\mathbf{b}^T = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix} [b_1, b_2, b_3] = \begin{bmatrix} a_1b_1 & a_1b_2 & a_1b_3 \\ a_2b_1 & a_2b_2 & a_2b_3 \\ a_3b_1 & a_3b_2 & a_3b_3 \end{bmatrix}.$$

In the dummy index summation notation, the new Kronecker product between two vector-valued functions is expressed as: $\mathbf{a} \otimes \mathbf{b} = a_i \mathbf{e}_i \otimes b_j \mathbf{e}_j = a_i b_j \mathbf{e}_i \mathbf{e}_j$, where \mathbf{e}_i and \mathbf{e}_j are the standard basis vectors in their respective spaces.

- The gradient of a vector function is a second-order tensor function. Let $\mathbf{V} = (v_1, v_2, \dots, v_m)^T$, then we have

$$\nabla \mathbf{V} = \begin{bmatrix} \partial_{x_1} v_1 & \partial_{x_2} v_1 & \cdots & \partial_{x_d} v_1 \\ \partial_{x_1} v_2 & \partial_{x_2} v_2 & \cdots & \partial_{x_d} v_2 \\ \vdots & \vdots & \vdots & \vdots \\ \partial_{x_1} v_m & \partial_{x_2} v_m & \cdots & \partial_{x_d} v_m \end{bmatrix},$$

$$\nabla \mathbf{V} = (\partial_i v_j) \mathbf{e}_i \mathbf{e}_j.$$

APPENDIX B. FLUX VECTOR SPLITTING METHOD FOR SHALLOW WATER SYSTEM

- 1D-Shallow Water Equations:

$$\left. \begin{aligned} \mathbf{U}_t + \mathbf{F}_x &= \mathbf{S}, \\ \mathbf{U} &= \begin{bmatrix} h \\ hu \end{bmatrix}, \mathbf{F} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 0 \\ -gh(z_0)_x \end{bmatrix} \end{aligned} \right\}$$

Jacobian matrix of flux function:

$$\mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 1 \\ gh - u^2 & 2u \end{bmatrix}.$$

The eigenstructure of the Jacobian matrix is given by

$$\left. \begin{aligned} \mathbf{A} &= \mathbf{R}\mathbf{\Lambda}\mathbf{L}, \\ \mathbf{\Lambda} &= \begin{bmatrix} u - \sqrt{gh} & 0 \\ 0 & u + \sqrt{gh} \end{bmatrix}, \mathbf{R} = \begin{bmatrix} 1 & 1 \\ u - \sqrt{gh} & u + \sqrt{gh} \end{bmatrix}, \mathbf{L} = \begin{bmatrix} \frac{u + \sqrt{gh}}{2\sqrt{gh}} & -\frac{1}{2\sqrt{gh}} \\ -\frac{u - \sqrt{gh}}{2\sqrt{gh}} & \frac{1}{2\sqrt{gh}} \end{bmatrix} \end{aligned} \right\}$$

- 2D-Shallow Water Equations:

$$\left. \begin{aligned} &\mathbf{U}_t + \mathbf{F}_x + \mathbf{G}_y = \mathbf{S}, \\ &\mathbf{U} = \begin{bmatrix} h \\ hu \\ hv \end{bmatrix}, \mathbf{F} = \begin{bmatrix} hu \\ p + hu^2 \\ huv \end{bmatrix}, \mathbf{G} = \begin{bmatrix} hv \\ huv \\ p + hv^2 \end{bmatrix}, \mathbf{S} = \begin{bmatrix} 0 \\ -gh(z_0)_x \\ -gh(z_0)_y \end{bmatrix}. \end{aligned} \right\}$$

Let $\mathbf{n} = (n_x, n_y)$ be the unit outward normal vector; then the normal flux is defined as

$$\mathcal{F}_n = n_x \cdot \mathbf{F} + n_y \cdot \mathbf{G}.$$

The Jacobians in the x -direction, y -direction, and normal direction are given by

$$\begin{aligned} \mathbf{A} = \frac{\partial \mathbf{F}}{\partial \mathbf{U}} &= \begin{bmatrix} 0 & 1 & 0 \\ gh - u^2 & 2u & 0 \\ -uv & v & u \end{bmatrix}, \mathbf{B} = \frac{\partial \mathbf{G}}{\partial \mathbf{U}} = \begin{bmatrix} 0 & 0 & 1 \\ -uv & v & u \\ gh - v^2 & 0 & 2v \end{bmatrix}, \\ \mathbf{A}_n = \mathbf{A}n_x + \mathbf{B}n_y &= \begin{bmatrix} 0 & n_x & n_y \\ gh n_x - u q_n & q_n + u n_x & u n_y \\ gh n_y - v q_n & v n_x & q_n + v n_y \end{bmatrix}, \end{aligned}$$

where q_n is the normal velocity, defined as

$$q_n = u n_x + v n_y.$$

The eigenstructure of the outward normal Jacobian is

$$\begin{aligned} \mathbf{A}_n &= \mathbf{R}_n \mathbf{\Lambda}_n \mathbf{L}_n, \\ \mathbf{\Lambda}_n &= \begin{bmatrix} q_n - \sqrt{gh} & 0 & 0 \\ 0 & q_n & 0 \\ 0 & 0 & q_n + \sqrt{gh} \end{bmatrix}, \\ \mathbf{R}_n &= \begin{bmatrix} 1 & 0 & 1 \\ u - \sqrt{gh} n_x & \ell_x & u + \sqrt{gh} n_x \\ v - \sqrt{gh} n_y & \ell_y & v + \sqrt{gh} n_y \end{bmatrix}, \\ \mathbf{L}_n &= \begin{bmatrix} \frac{\sqrt{gh} + q_n}{2\sqrt{gh}} & -\frac{n_x}{2\sqrt{gh}} & -\frac{n_y}{2\sqrt{gh}} \\ -q_\ell & \ell_x & \ell_y \\ \frac{\sqrt{gh} - q_n}{2\sqrt{gh}} & \frac{n_x}{2\sqrt{gh}} & \frac{n_y}{2\sqrt{gh}} \end{bmatrix}, \end{aligned}$$

where $\boldsymbol{\tau} = (\ell_x, \ell_y)$ is the unit tangent vector, and q_ℓ is the tangential velocity, defined as

$$\begin{aligned} [\ell_x, \ell_y] &= [-n_y, n_x], \\ q_\ell &= u\ell_x + v\ell_y. \end{aligned}$$

In shallow water equations

$$\begin{aligned} a &= \sqrt{gh}, \\ P &= \frac{1}{2}gh^2. \end{aligned}$$

Note that in the shallow water wave equations, the conservation of mass is described by the water level equation, namely

$$\rho^{(SW)} = h.$$

Further, combining this with the speed of sound calculation formula $a = \sqrt{\frac{\gamma P}{\rho}}$, it can be deduced that

$$\gamma^{(SW)} = 2.$$

It should be noted that the source-free shallow water wave equations do not have an energy conservation equation. In summary, the source-free shallow water wave equations are isentropic Euler equations with the thermodynamic property parameter $\gamma = 2$.

To implement the flux calculation based on Mach number splitting for the shallow water wave equations, simply set the parameter γ to 2 in the van Leer and AUSM splitting sub-routines of the compressible Euler equations, replace the density ρ with the water level h , and use $P = \frac{1}{2}gh^2$.

It is emphasized once again, the Jacobian of the shallow water wave equations does not possess homogeneity, hence it does not satisfy $f(u) = \frac{\partial f}{\partial u} \cdot u$. Reference [7] reconstructs a homogeneous Jacobian by forcibly modifying the wave speed and then applies the eigenvalue splitting method.

For a more detailed introduction to the shallow water wave equations, readers are referred to [46].

B.1. Flux Vector Splitting for Shallow Water Wave Equations in One-dimension.

B.1.1. Mach Number Splitting Based on True Wave Speed.

True wave speed: $a = \sqrt{gh}$.

True local Mach number: $M_a = u/a$.

Pressure: $P = \frac{1}{2}gh^2$; Adiabatic index: $\gamma = 2$.

$$(vanLeer) \quad \mathbf{F} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = ha \begin{bmatrix} M_a \\ uM_a + a/\gamma \end{bmatrix}.$$

$$(AUSM) \quad \mathbf{F} = \begin{bmatrix} hu \\ hu^2 + \frac{1}{2}gh^2 \end{bmatrix} = \begin{bmatrix} haM_a \\ haM_a u + P \end{bmatrix} = \underbrace{haM_a}_{\mathbf{F}^a} \begin{bmatrix} 1 \\ u \end{bmatrix} + \underbrace{P}_{\mathbf{F}^P} \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

The splitting method for M_a and P is the same as for the compressible Euler equations.

B.1.2. Jacobian Eigenvalue Splitting Based on Modified Wave Speed[7].

True wave speed: $a = \sqrt{gh}$.

Modified wave speed: $a^* = \sqrt{gh/2}$.

The Jacobian and its eigenstructure after forced modification are:

$$\begin{aligned} \mathbf{A}^* &= \begin{bmatrix} 0 & 1 \\ \frac{gh}{2} - u^2 & 2u \end{bmatrix}, \\ \mathbf{\Lambda}^* &= \begin{bmatrix} u - \sqrt{gh/2} & 0 \\ 0 & u + \sqrt{gh/2} \end{bmatrix}, \\ \mathbf{R}^* &= \begin{bmatrix} 1 & 1 \\ u - \sqrt{gh/2} & u + \sqrt{gh/2} \end{bmatrix}, \\ \mathbf{L}^* &= \begin{bmatrix} \frac{u + \sqrt{gh/2}}{2\sqrt{gh/2}} & -\frac{1}{2\sqrt{gh/2}} \\ -\frac{u - \sqrt{gh/2}}{2\sqrt{gh/2}} & \frac{1}{2\sqrt{gh/2}} \end{bmatrix}. \end{aligned}$$

After the modification of the wave speed, it holds that $\mathbf{F} = \mathbf{A}^* \mathbf{U}$.

Subsequently, the Steger-Warming splitting or Lax-Friedrichs splitting can be applied.

B.2. Flux Vector Splitting for Shallow Water Wave Equations in Two-dimension.

B.2.1. Mach Number Splitting Based on True Wave Speed.

True wave speed: $a = \sqrt{gh}$.

True local normal Mach number: $M_a^n = q_n/a$.

Pressure: $P = \frac{1}{2}gh^2$; Adiabatic index: $\gamma = 2$.

$$(AUSM) \quad \mathcal{F}_n = \begin{bmatrix} hq_n \\ huq_n + Pn_x \\ hvq_n + Pn_y \end{bmatrix} = \begin{bmatrix} haM_a \\ haM_a u + Pn_x \\ haM_a v + Pn_y \end{bmatrix} = \underbrace{haM_a}_{\mathbf{F}^a} \begin{bmatrix} 1 \\ u \\ v \end{bmatrix} + \underbrace{P}_{\mathbf{F}^P} \begin{bmatrix} 0 \\ n_x \\ n_y \end{bmatrix}.$$

The splitting method for M_a and P is the same as for the compressible Euler equations.

B.2.2. *Jacobian Eigenvalue Splitting Based on Modified Wave Speed*[7].

True wave speed: $a = \sqrt{gh}$.

Modified wave speed: $a^* = \sqrt{gh/2}$.

The Jacobian and its eigenstructure after forced modification are:

$$\mathbf{\Lambda}_n^* = \begin{bmatrix} q_n - \sqrt{gh/2} & 0 & 0 \\ 0 & q_n & 0 \\ 0 & 0 & q_n + \sqrt{gh/2} \end{bmatrix},$$

$$\mathbf{R}_n^* = \begin{bmatrix} 1 & 0 & 1 \\ u - \sqrt{gh/2}n_x & \ell_x & u + \sqrt{gh/2}n_x \\ v - \sqrt{gh/2}n_y & \ell_y & v + \sqrt{gh/2}n_y \end{bmatrix}, \mathbf{L}_n^* = \begin{bmatrix} \frac{\sqrt{gh/2}+q_n}{2\sqrt{gh/2}} & -\frac{n_x}{2\sqrt{gh/2}} & -\frac{n_y}{2\sqrt{gh/2}} \\ -q_\ell & \ell_x & \ell_y \\ \frac{\sqrt{gh/2}-q_n}{2\sqrt{gh/2}} & \frac{n_x}{2\sqrt{gh/2}} & \frac{n_y}{2\sqrt{gh/2}} \end{bmatrix},$$

$$\mathbf{A}_n^* = \mathbf{R}_n^* \mathbf{\Lambda}_n^* \mathbf{L}_n^*.$$

After the modification of the wave speed, it holds that $\mathcal{F}_n = \mathbf{A}_n^* \mathbf{U}$.

Subsequently, the Steger-Warming splitting or Lax-Friedrichs splitting can be applied.

APPENDIX C. DEVELOPMENT OF NUMERICAL FLUX FORMATS FOR SCALAR EQUATIONS BASED ON THE SYSTEM'S JACOBIAN-FVS METHOD

C.1. The Classical Lax-Friedrichs Flux Format from the Jacobian-FVS Perspective. This section attempts to derive the classical Lax-Friedrichs flux format from the L-F splitting method within the Jacobian-FVS.

Consider a one-dimensional hyperbolic conservation system as an example. The L-F splitting process is shown below:

Take

$$A^{L,+} = \frac{1}{2}(A^L + M^L I_d) = \frac{1}{2}(A(U^L) + M^L I_d),$$

$$A^{R,-} = \frac{1}{2}(A^R - M^R I_d) = \frac{1}{2}(A(U^R) - M^R I_d),$$

where M^L, M^R are positive constants that should ensure

$$\left. \begin{aligned} \lambda(A^{L,+}) &= \lambda(A^L + M^L I_d) \geq 0, \\ \lambda(A^{R,-}) &= \lambda(A^R - M^R I_d) \leq 0, \end{aligned} \right\}$$

hold at every cell interface.

At $x_{i+1/2}$, there is

$$\begin{aligned}
\hat{f}_{i+1/2} &= A_{i+1/2}^{L,+} \mathbf{U}_{i+1/2}^L + A_{i+1/2}^{R,-} \mathbf{U}_{i+1/2}^R \\
&= \frac{1}{2} (A(\mathbf{U}_{i+1/2}^L) + M_{i+1/2}^L I) \mathbf{U}_{i+1/2}^L + \frac{1}{2} (A(\mathbf{U}_{i+1/2}^R) - M_{i+1/2}^R I) \mathbf{U}_{i+1/2}^R \\
&= \frac{1}{2} (A(\mathbf{U}_{i+1/2}^L) \mathbf{U}_{i+1/2}^L + M_{i+1/2}^L \mathbf{U}_{i+1/2}^L) + \frac{1}{2} (A(\mathbf{U}_{i+1/2}^R) \mathbf{U}_{i+1/2}^R - M_{i+1/2}^R \mathbf{U}_{i+1/2}^R) \\
&= \frac{1}{2} (f(\mathbf{U}_{i+1/2}^L) + M_{i+1/2}^L \mathbf{U}_{i+1/2}^L) + \frac{1}{2} (f(\mathbf{U}_{i+1/2}^R) - M_{i+1/2}^R \mathbf{U}_{i+1/2}^R) \\
&= \frac{1}{2} (f(\mathbf{U}_{i+1/2}^R) + f(\mathbf{U}_{i+1/2}^L) - (M_{i+1/2}^R \mathbf{U}_{i+1/2}^R - M_{i+1/2}^L \mathbf{U}_{i+1/2}^L)).
\end{aligned}$$

The classic L-F flux format is given by

$$\hat{f}_{i+1/2} = \frac{1}{2} (f(\mathbf{U}_{i+1/2}^R) + f(\mathbf{U}_{i+1/2}^L) - \alpha_{i+1/2} (\mathbf{U}_{i+1/2}^R - \mathbf{U}_{i+1/2}^L)),$$

where $\alpha_{i+1/2}$ is usually taken as

$$\alpha_{i+1/2} = \rho \left(\left. \frac{\partial f}{\partial u} \right|_{x_{i+1/2}} \right),$$

which means $\alpha_{i+1/2}$ is the spectral radius of the Jacobian of the flux function f with respect to the field variable u evaluated at $x_{i+1/2}$.

It is noted that if we take

$$M_{i+1/2} = \max\{M_{i+1/2}^L, M_{i+1/2}^R\},$$

then $M_{i+1/2}$ also satisfies

$$\left. \begin{aligned}
\lambda(A_{i+1/2}^{L,+}) &= \lambda(A_{i+1/2}^L + M_{i+1/2} I_d) \geq 0, \\
\lambda(A_{i+1/2}^{R,-}) &= \lambda(A_{i+1/2}^R - M_{i+1/2} I_d) \leq 0.
\end{aligned} \right\}$$

Thus, $M_{i+1/2}$ can be used to simultaneously replace $M_{i+1/2}^L$ and $M_{i+1/2}^R$ in the numerical flux \hat{f} to obtain

$$\begin{aligned}
\tilde{f}_{i+1/2} &= \frac{1}{2} (f(\mathbf{U}_{i+1/2}^R) + f(\mathbf{U}_{i+1/2}^L) - (M_{i+1/2} \mathbf{U}_{i+1/2}^R - M_{i+1/2} \mathbf{U}_{i+1/2}^L)) \\
&= \frac{1}{2} (f(\mathbf{U}_{i+1/2}^R) + f(\mathbf{U}_{i+1/2}^L) - M_{i+1/2} (\mathbf{U}_{i+1/2}^R - \mathbf{U}_{i+1/2}^L)).
\end{aligned}$$

By comparing \tilde{f} and \hat{f} , it can be observed that the $\alpha_{i+1/2}$ in the classical Lax-Friedrichs flux formula is a specific choice for the $M_{i+1/2}$ in the Lax-Friedrichs splitting method.

C.2. Constructing Steger-Warming flux for the scalar equation based on Steger-Warming splitting for system.

From C.1, it is known that the Lax-Friedrichs splitting method can derive the classical Lax-Friedrichs flux format, which is not only used for system control equations but also commonly applied to scalar equations. Inspired by this, this section attempts to apply the

Steger-Warming splitting method to scalar equations in order to derive a new numerical flux format suitable for scalar equations.

C.2.1. *Steger-Warming Flux for One-Dimensional Scalar Equations.* The Jacobian-FVS for a system of equations first requires rewriting the flux function in the form $F(\mathbf{U}) = A(\mathbf{U}) \cdot \mathbf{U}$. Similarly, for the 1D scalar equation, we make a similar rewrite:

Let

$$a = K f'(u), \quad (\text{C.1})$$

satisfy

$$f(u) = au = K f'(u)u, \quad (\text{C.2})$$

where K is an undetermined constant that varies with different equations (different $f(u)$). Generally, $a = a(x, t, u)$.

It should be emphasized that in scalar equations, A , Λ , and λ are “the same”, all “equal to a ”. Thus, the scalar form of the Steger-Warming splitting method is as follows:

$$a^+ = \frac{a + |a|}{2}, \quad a^- = \frac{a - |a|}{2}. \quad (\text{C.3})$$

If smoothness is considered:

$$a^+ = \frac{a + \sqrt{|a|^2 + \delta^2}}{2}, \quad a^- = \frac{a - \sqrt{|a|^2 + \delta^2}}{2}, \quad (\text{C.4})$$

where δ is a small positive quantity, for example, δ can be taken as 10^{-8} .

Taking the interface $x_{i+1/2}$ as an example, due to the multi-valuedness of u at the interface $(u_{i+1/2}^L, u_{i+1/2}^R)$, a is also multi-valued at the interface, i.e., $(a_{i+1/2}^L, a_{i+1/2}^R)$. Accordingly, the Steger-Warming flux format for the 1D scalar equation is as follows:

$$\begin{aligned} \hat{f}_{i+1/2}^{SW} &= a_{i+1/2}^{L,+} u_{i+1/2}^L + a_{i+1/2}^{R,-} u_{i+1/2}^R \\ &= \left(\frac{a_{i+1/2}^L + |a_{i+1/2}^L|}{2} \right) u_{i+1/2}^L + \left(\frac{a_{i+1/2}^R - |a_{i+1/2}^R|}{2} \right) u_{i+1/2}^R \\ &= \frac{1}{2} (f(u_{i+1/2}^R) + f(u_{i+1/2}^L) - (|a_{i+1/2}^R| u_{i+1/2}^R - |a_{i+1/2}^L| u_{i+1/2}^L)). \end{aligned} \quad (\text{C.5})$$

The following discussion will focus on the linear advection equation and the nonlinear Burgers' equation in detail:

- Case 1: $u_t + (c(x, t)u)_x = 0$.

$$f(u) = c(x, t)u, \quad f'(u) = c(x, t),$$

thus we take

$$K = 1, \quad a = c(x, t).$$

At this point, with $a_{i+1/2}^L = a_{i+1/2}^R = c(x_{i+1/2}, \cdot)$, the Steger-Warming flux splitting scheme for the linear scalar advection equation is given by:

$$\hat{f}_{i+1/2}^{SW} = \frac{1}{2} \left(c(x_{i+1/2}, t) (u_{i+1/2}^R + u_{i+1/2}^L) - |c(x_{i+1/2}, t)| (u_{i+1/2}^R - u_{i+1/2}^L) \right). \quad (\text{C.6})$$

Let's examine the time step t^n :

- If $c(x_{i+1/2}, t^n) > 0$, abbreviated as $c_{i+1/2}^n > 0$,
then $a_{i+1/2} = c_{i+1/2}^n > 0$, and thus $|a_{i+1/2}| = a_{i+1/2}$.
Consequently, $a_{i+1/2}^{L,+} = a_{i+1/2}$, $a_{i+1/2}^{R,-} = 0$,
and the Steger-Warming numerical flux at the interface is given by

$$\hat{f}_{i+1/2}^{SW} = a_{i+1/2}^{L,+} u_{i+1/2}^L = c_{i+1/2}^n u_{i+1/2}^L.$$

Given that $c_{i+1/2}^n > 0$, the local upwind direction at $x_{i+1/2}$ should be in the direction of $-x$, meaning that the numerical flux should be calculated using the left state of the field function. That is, take $\hat{f}_{i+1/2} = f(u_{i+1/2}^L) = c_{i+1/2}^n u_{i+1/2}^L$.
It can be seen that $\hat{f}_{i+1/2}^{SW}$ satisfies the upwind property.

- If $c(x_{i+1/2}, t^n) < 0$, abbreviated as $c_{i+1/2}^n < 0$,
then $a_{i+1/2} = c_{i+1/2}^n < 0$, and thus $|a_{i+1/2}| = -a_{i+1/2}$.
Consequently, $a_{i+1/2}^{L,+} = 0$, $a_{i+1/2}^{R,-} = a_{i+1/2}$,
and the Steger-Warming numerical flux at the interface is given by

$$\hat{f}_{i+1/2}^{SW} = a_{i+1/2}^{R,-} u_{i+1/2}^R = c_{i+1/2}^n u_{i+1/2}^R.$$

Given that $c_{i+1/2}^n < 0$, the local upwind direction at $x_{i+1/2}$ should be in the direction of $+x$, meaning that the numerical flux should be calculated using the right state of the field function. That is, take $\hat{f}_{i+1/2} = f(u_{i+1/2}^R) = c_{i+1/2}^n u_{i+1/2}^R$.
It can be seen that $\hat{f}_{i+1/2}^{SW}$ satisfies the upwind property.

Remark C.1. *The fact that c is a function of x and t implies that the upwind direction in different regions of the flow field is not consistent and may change over time. When c is only a function of t , or more so when $c = \text{const}$, the above discussion results are still applicable, but the entire flow field will share the same upwind direction (if $c = c(t)$, this upwind direction will change over time).*

- Case 2: $u_t + \left(\frac{1}{2}u^2\right)_x = 0$.

$$f(u) = \frac{1}{2}u^2, \quad f'(u) = u,$$

thus we take

$$K = \frac{1}{2}, \quad a = K f'(u) = \frac{1}{2}u.$$

The Steger-Warming flux for the Burgers' equation is given by

$$\begin{aligned}
 \hat{f}_{i+1/2}^{SW} &= \frac{1}{2} \left(\frac{1}{2} |u_{i+1/2}^R|^2 + \frac{1}{2} |u_{i+1/2}^L|^2 - \left(\frac{1}{2} |u_{i+1/2}^R| u_{i+1/2}^R - \frac{1}{2} |u_{i+1/2}^L| u_{i+1/2}^L \right) \right) \\
 &= \frac{1}{4} \left(|u_{i+1/2}^R|^2 + |u_{i+1/2}^L|^2 - (|u_{i+1/2}^R| u_{i+1/2}^R - |u_{i+1/2}^L| u_{i+1/2}^L) \right) \\
 &= \frac{1}{4} (|u_{i+1/2}^R| (|u_{i+1/2}^R| - u_{i+1/2}^R) + |u_{i+1/2}^L| (|u_{i+1/2}^L| + u_{i+1/2}^L)). \tag{C.7}
 \end{aligned}$$

When applied to the nonlinear Burgers' equation, the Steger-Warming flux satisfies ‘‘compatibility, Lipschitz continuity, and monotonicity.’’ This ensures that the spatial semi-discrete weak solution of the Burgers' equation, based on the Steger-Warming flux, satisfies the cell entropy inequality and L^2 stability, as detailed below in section [C.2.3](#).

C.2.2. Steger-Warming Flux for Two-Dimensional Scalar Equations. Similar to the techniques used in Section [6](#), the 2D problem is converted into several 1D problems along the outward normal direction by introducing the normal flux.

Consider the following two-dimensional scalar equation:

$$U_t + (f(U))_x + (g(U))_y = 0.$$

Let $\mathbf{F}(U) = (f(U), g(U))$, and suppose the unit outward normal vector at a point on the cell boundary is $\mathbf{n} = (n_x, n_y)$, The normal flux is then defined as

$$F_n(U) = \mathbf{n} \cdot \mathbf{F} = n_x f(U) + n_y g(U).$$

To use the Jacobian-FVS, it is necessary to construct $a_n(x, t, U)$ such that

$$F_n(U) = a_n \cdot U.$$

Similar to the one-dimensional case, a_n is chosen in the following form:

$$a_n = K F_n'(U),$$

where K is a constant to be determined.

The splitting method for a_n is consistent with the previously described one-dimensional case. Consequently, the Steger-Warming flux for the two-dimensional scalar equation at a specific point on the cell interface, denoted by $\mathbf{G}_{\mathbf{k}}$, is delineated as follows:

$$\begin{aligned}
 \hat{F}_n^{SW}(U^{int}(\mathbf{G}_{\mathbf{k}}, t), U^{ext}(\mathbf{G}_{\mathbf{k}}, t)) &= a_n^{int,+} \cdot U^{int}(\mathbf{G}_{\mathbf{k}}, t) + a_n^{ext,-} \cdot U^{ext}(\mathbf{G}_{\mathbf{k}}, t) \\
 &= \left(\frac{a_n^{int} + |a_n^{int}|}{2} \right) U^{int}(\mathbf{G}_{\mathbf{k}}, t) + \left(\frac{a_n^{ext} - |a_n^{ext}|}{2} \right) U^{ext}(\mathbf{G}_{\mathbf{k}}, t) \\
 &= \frac{1}{2} \left(F_n(U(\mathbf{G}_{\mathbf{k}}, t)^{ext}) + F_n(U^{int}(\mathbf{G}_{\mathbf{k}}, t)) - (|a_n^{ext}| U^{ext}(\mathbf{G}_{\mathbf{k}}, t) - |a_n^{int}| U^{int}(\mathbf{G}_{\mathbf{k}}, t)) \right). \tag{C.8}
 \end{aligned}$$

- The two-dimensional conservative linear scalar transport equation

$$U_t + (\alpha(x, y, t)U)_x + (\beta(x, y, t)U)_y = 0.$$

$$F_n(U) = n_x \cdot \alpha(x, y, t)U + n_y \cdot \beta(x, y, t)U,$$

$$F_n'(U) = n_x \cdot \alpha(x, y, t) + n_y \cdot \beta(x, y, t),$$

hence we take

$$K = 1, \quad a_n = n_x \cdot \alpha(x, y, t) + n_y \cdot \beta(x, y, t).$$

Therefore, $a^{int}(\mathbf{G}_k, \cdot) = a^{ext}(\mathbf{G}_k, \cdot) = n_x \cdot \alpha(\mathbf{G}_k, \cdot) + n_y \cdot \beta(\mathbf{G}_k, \cdot)$,

the Steger-Warming flux for the two-dimensional linear scalar transport equation is given by

$$\begin{aligned} & \hat{F}_n^{SW}(U^{int}(\mathbf{G}_k, t), U^{ext}(\mathbf{G}_k, t)) \\ &= \frac{1}{2} \left((n_x \cdot \alpha(\mathbf{G}_k, t) + n_y \cdot \beta(\mathbf{G}_k, t)) \cdot (U^{ext}(\mathbf{G}_k, t) + U^{int}(\mathbf{G}_k, t)) \right. \\ & \quad \left. - |n_x \cdot \alpha(\mathbf{G}_k, t) + n_y \cdot \beta(\mathbf{G}_k, t)| (U^{ext}(\mathbf{G}_k, t) - U^{int}(\mathbf{G}_k, t)) \right). \end{aligned} \quad (\text{C.9})$$

- The two-dimensional nonlinear Burgers' equation $U_t + (\frac{1}{2}U^2)_x + (\frac{1}{2}U^2)_y = 0$.

$$\begin{aligned} F_n(U) &= (n_x + n_y) \cdot \frac{1}{2}U^2, \\ F'_n(U) &= (n_x + n_y) \cdot U, \end{aligned}$$

hence we take

$$K = \frac{1}{2}, \quad a_n = \frac{1}{2}(n_x + n_y) \cdot U.$$

the Steger-Warming flux for the 2D Burgers' equation is given by

$$\hat{F}_n^{SW}(U^{int}(\mathbf{G}_k, t), U^{ext}(\mathbf{G}_k, t)) \quad (\text{C.10})$$

$$\begin{aligned} &= \frac{1}{4}(n_x + n_y) \left(((U^{ext}(\mathbf{G}_k, t))^2 + (U^{int}(\mathbf{G}_k, t))^2) \right. \\ & \quad \left. - (|U^{ext}(\mathbf{G}_k, t)| U^{ext}(\mathbf{G}_k, t) - |U^{int}(\mathbf{G}_k, t)| U^{int}(\mathbf{G}_k, t)) \right). \end{aligned} \quad (\text{C.11})$$

C.2.3. Properties of the Steger-Warming Flux for the Scalar Equation. This section discusses the properties of the Steger-Warming flux within the scope of one-dimensional conservative linear scalar transport equations and nonlinear Burgers' equations.

The one-dimensional scalar equation Steger-Warming flux is given by

$$\hat{f}_{i+1/2}^{SW} = \frac{1}{2} (f(u_{i+1/2}^R) + f(u_{i+1/2}^L) - (|a_{i+1/2}^R| u_{i+1/2}^R - |a_{i+1/2}^L| u_{i+1/2}^L)).$$

- Conservative linear scalar transport equation $u_t + (c(x, t)u)_x = 0$,

$$a = c(x, t);$$

- Nonlinear Burgers' equation $u_t + (\frac{1}{2}u^2)_x = 0$,

$$a = \frac{1}{2}u.$$

When applied to conservative linear scalar transport equations and nonlinear Burgers' equations, the Steger-Warming flux scheme satisfies “consistency, Lipschitz continuity, and monotonicity”, which will be introduced and proven below:

Since all discussions are at $x_{i+1/2}$, the subscript “ $i + 1/2$ ” is omitted for all variables in the following processes.

- Consistency, that is, it satisfies $\hat{f}^{SW}(u, u) = f(u)$.

Proof. Let

$$u^L = u^R = u^*.$$

Since $a = a(x, t, u)$, then

$$a^L(\cdot, \cdot, u^L) = a^R(\cdot, \cdot, u^R) = a(\cdot, \cdot, u^*).$$

Let us denote $a^* = a(\cdot, \cdot, u^*)$, thus we have

$$\begin{aligned} \hat{f}^{SW}(u^*, u^*) &= \frac{1}{2} (f(u^*) + f(u^*) - (|a^*| u^* - |a^*| u^*)) \\ &= \frac{1}{2} \cdot 2f(u^*) \\ &= f(u^*). \end{aligned}$$

□

- When $f(u; x, t)$ is Lipschitz continuous function of u , the Steger-Warming numerical flux $\hat{f}^{SW}(u^L, u^R)$ is Lipschitz continuous with respect to both u^L and u^R .

Proof.

We first establish the Lipschitz continuity of $\hat{f}^{SW}(u^L, u^R)$ with respect to u^L .

Let Δ is a constant that can be either positive or negative, and define

$$a^{L+\Delta} := a(\cdot, \cdot, u^L + \Delta), \quad (\text{C.12})$$

then

$$\hat{f}^{SW}(u^L + \Delta, u^R) = \frac{1}{2} (f(u^R) + f(u^L + \Delta) - (|a^R| u^R - |a^{L+\Delta}| (u^L + \Delta))), \quad (\text{C.13})$$

hence

$$\begin{aligned} &\hat{f}^{SW}(u^L + \Delta, u^R) - \hat{f}^{SW}(u^L, u^R) \\ &= \frac{1}{2} (f(u^L + \Delta) + |a^{L+\Delta}| (u^L + \Delta) - f(u^L) - |a^L| u^L). \end{aligned} \quad (\text{C.14})$$

Perform the following simple decomposition:

$$\begin{aligned} &\hat{f}^{SW}(u^L + \Delta, u^R) - \hat{f}^{SW}(u^L, u^R) \\ &= \frac{1}{2} (f(u^L + \Delta) - f(u^L) + |a^{L+\Delta}| (u^L + \Delta) - |a^L| (u^L + \Delta) + |a^L| (u^L + \Delta) - |a^L| u^L) \end{aligned}$$

$$= \frac{1}{2} \left(\underbrace{[f(u^L + \Delta) - f(u^L)]}_{T_1} + \underbrace{(|a^{L+\Delta}| - |a^L|) \cdot (u^L + \Delta)}_{T_2} + \underbrace{|a^L| \cdot \Delta}_{T_3} \right). \quad (\text{C.15})$$

Since $f(u; x, t)$ is Lipschitz continuous with respect to u , then

$$\exists L_1 > 0, \text{ s.t. } |T_1| \leq L_1 |\Delta|.$$

f being Lipschitz continuous implies that its derivative f' is bounded. Consequently, a is bounded as well (given $a = K f'(u)$, where K is a constant), and thus we have

$$\exists L_3 > 0, \text{ s.t. } |T_3| \leq L_3 |\Delta|.$$

Both $a^{L+\Delta}$ and a^L are bounded, which implies that the difference $|a^{L+\Delta}| - |a^L|$ is also bounded. Noting that u^L is a finite state, we deduce that

$$\exists L_2 > 0, \text{ s.t. } |T_2| \leq L_2 |\Delta|.$$

Hence, there exists a constant $M > 0$ (e.g. take $M = \frac{3}{2} \cdot \max\{L_1, L_2, L_3\}$) such that

$$\left| \hat{f}^{SW}(u^L + \Delta, u^R) - \hat{f}^{SW}(u^L, u^R) \right| \leq \frac{1}{2} (|T_1| + |T_2| + |T_3|) \leq M |\Delta|, \quad (\text{C.16})$$

which implies that $\hat{f}^{SW}(u^L, u^R)$ is Lipschitz continuous with respect to u^L when $f(u; x, t)$ is Lipschitz continuous function of u .

Similarly, it can be shown that $\hat{f}^{SW}(u^L, u^R)$ is Lipschitz continuous with respect to u^R under the same conditions. \square

- **Monotonicity:** The Steger-Warming numerical flux $\hat{f}^{SW}(u^L, u^R)$ is non-decreasing with respect to the first variable u^L and non-increasing with respect to the second variable u^R . This property is succinctly denoted as $\hat{f}^{SW}(\uparrow, \downarrow)$.

Proof.

We first prove that $\hat{f}^{SW}(u^L, u^R)$ is non-decreasing with respect to the first variable u^L .

Following the proof of Lipschitz continuity, we have

$$\begin{aligned} & \hat{f}^{SW}(u^L + \Delta, u^R) - \hat{f}^{SW}(u^L, u^R) \\ &= \frac{1}{2} (f(u^L + \Delta) + |a^{L+\Delta}| (u^L + \Delta) - f(u^L) - |a^L| u^L). \end{aligned}$$

However, here we set $\Delta > 0$.

- Conservative linear scalar transport equation: $u_t + (c(x, t)u)_x = 0$, $f(u) = c(x, t)u$, $a = c(x, t)$.

Since a is independent of u , it follows that $a^{L+\Delta} = a^L$. Consequently,

$$\begin{aligned} & \hat{f}^{SW}(u^L + \Delta, u^R) - \hat{f}^{SW}(u^L, u^R) \\ &= \frac{1}{2} (c \cdot (u^L + \Delta) + |c| (u^L + \Delta) - c \cdot (u^L) - |c| u^L) \end{aligned}$$

$$= \frac{1}{2} (c + |c|) \Delta \geq 0.$$

This implies that $\hat{f}^{SW}(u^L, u^R)$ is non-decreasing with respect to the first variable u^L when applied to the conservative linear scalar transport equation.

– Nonlinear Burgers’ equation: $u_t + (\frac{1}{2}u^2)_x = 0$, $f(u) = \frac{1}{2}u^2$, $a = \frac{1}{2}u$.

$$\begin{aligned} & \hat{f}^{SW}(u^L + \Delta, u^R) - \hat{f}^{SW}(u^L, u^R) \\ &= \frac{1}{2} \left(\frac{1}{2} (u^L + \Delta)^2 + \frac{1}{2} |u^L + \Delta| (u^L + \Delta) - \frac{1}{2} (u^L)^2 - \frac{1}{2} |u^L| u^L \right) \\ &= \frac{1}{4} (|u^L + \Delta| (|u^L + \Delta| + u^L + \Delta) - |u^L| (|u^L| + u^L)). \end{aligned}$$

We introduce an auxiliary function $g(u) = u^2 + |u|u$. Then, we have

$$\hat{f}^{SW}(u^L + \Delta, u^R) - \hat{f}^{SW}(u^L, u^R) = \frac{1}{4} (g(u^L + \Delta) - g(u^L)).$$

It is clear that $g(u)$ is non-decreasing with respect to u .

Given that $\Delta > 0$ and $u^L + \Delta > u^L$,

it follows that $\hat{f}^{SW}(u^L + \Delta, u^R) \geq \hat{f}^{SW}(u^L, u^R)$,

which demonstrates the non-decreasing nature of $\hat{f}^{SW}(u^L, u^R)$ with respect to the first variable u^L .

By similar reasoning, it can be proven that the Steger-Warming numerical flux $\hat{f}^{SW}(u^L, u^R)$, when applied to either the conservative linear scalar transport equation or the nonlinear Burgers’ equation, is non-increasing with respect to the second variable u^R . \square

When a numerical flux scheme satisfies the aforementioned properties of “consistency, Lipschitz continuity, and monotonicity,” it can be proven that the DG (Discontinuous Galerkin) weak solution based on this flux scheme satisfies the cell entropy inequality and L^2 stability. This proof process is standardized, and readers are referred to the literature [47, 48] for details.

C.2.4. Numerical Experiments. We conducted numerical experiments to compare the Steger-Warming flux scheme discussed in this appendix with the classical Lax-Friedrichs flux scheme within the context of conservative linear scalar transport equations and nonlinear Burgers’ equation.

- Accuracy test

Example C.1. *Linear transport equations.*

- *Control Eqs:* $u_t + (\sin(\omega t)u)_x = 0$, $\omega = \pi$;
Computational domain: $\Omega \times [0, T_{end}] = [0, 2\pi] \times [0, 20]$;
ICs: $u_0(x) = \sin(x)$;
BCs: *periodic boundary conditions*;
True solutions: $u(x, t) = u_0(x + \frac{1}{\omega}(\cos(\omega t) - 1))$;

Note: The smallest positive period of the solution u is $T^* = \frac{2\pi}{\omega} = 2$, hence, it implies simulating up to 10 periods that T_{end} is taken as 20.

Temporal discretization format: TVD-RK3;

CFL=0.1.

L^2 , L^1 numerical errors and convergence orders with P^2 -polynomial approximation are summarized respectively in Table 9 and Figure 17.

TABLE 9. P^2 -Steger-Warming-scalar-DG and P^2 -Lax-Friedrichs-scalar-DG both using equally spaced cells. $u_t + (\sin(t)u)_x = 0$ with smooth initial conditions: $u_0(x) = \sin(x)$. L^2 , L^1 errors for u .

	Mesh	20	40	80	160	320
S-W	L^2 -error	5.0283E-04	6.3601E-05	0.7973E-05	0.9975E-06	1.2471E-07
L-F	L^2 -error	6.1941E-04	8.8553E-05	1.2804E-05	1.8863E-06	2.8289E-07
S-W	L^1 -error	1.1090E-03	1.4070E-04	1.7664E-05	2.2117E-06	2.7663E-07
L-F	L^1 -error	1.3487E-03	1.9093E-04	2.7256E-05	3.9575E-06	5.8503E-07

- Control Eqs: $u_t + (\sin(x)u)_x = 0$;

Computational domain: $\Omega \times [0, T_{end}] = [0, 2\pi] \times [0, 1]$;

ICs: $u_0(x) = 1$;

BCs: *periodic boundary conditions*;

True solutions: $u(x, t) = \frac{\sin(2 \arctan(e^{-t} \tan(\frac{x}{2})))}{\sin(x)}$;

Temporal discretization format: RK4;

CFL=0.05.

L^2 , L^1 numerical errors with P^5 -polynomial approximation are summarized respectively in Table 10 and Figure 18 .

TABLE 10. P^5 -Steger-Warming-scalar-DG and P^5 -Lax-Friedrichs-scalar-DG both using equally spaced cells. $u_t + (\sin(x)u)_x = 0$ with smooth initial conditions: $u_0(x) = 1$. L^2 , L^1 errors and convergence orders for u .

	Mesh	20	40	80	160	320
S-W	L^2 -error	1.1214E-06	3.1954E-08	5.3956E-10	9.4229E-12	5.7994E-13
L-F	L^2 -error	1.2660E-06	3.3854E-08	4.7961E-10	7.8348E-12	7.5041E-13
S-W	L^1 -error	1.1410E-06	3.0824E-08	4.8678E-10	1.0050E-11	0.8858E-12
L-F	L^1 -error	1.3414E-06	3.2669E-08	4.6412E-10	0.9267E-11	1.0540E-12

Example C.2. Nonlinear Burgers' equation.

- 1D-Control Eqs: $u_t + (\frac{1}{2}u^2)_x = 0$;

Computational domain: $\Omega \times [0, T_{end}] = [0, 2\pi] \times [0, 0.6]$;

Note: the exact solution remains smooth during $t \in [0, 0.6]$;

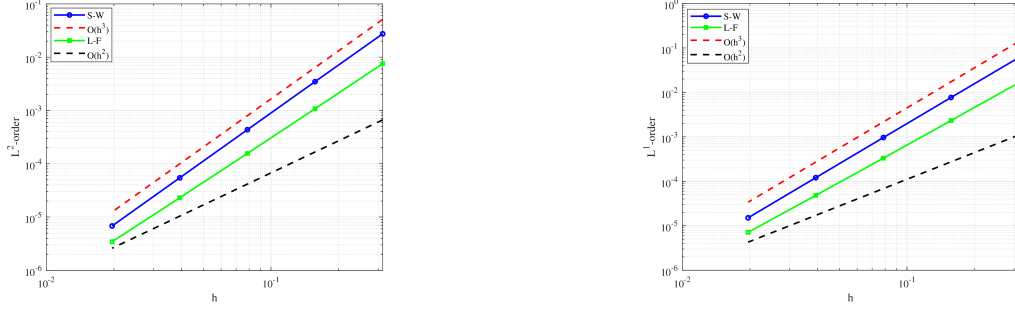
ICs: $u_0(x) = \sin(x)$;

BCs: *periodic boundary conditions*;

True solutions: $u(x, t) = u_0(x^*)$, x^* satisfies $x^* + u_0(x^*) \cdot t = x$;

Temporal discretization format: RK4;

CFL=0.05.



(a) L^2 -convergence order

(b) L^1 -convergence order

FIGURE 17. P^2 -Steger-Warming-scalar-DG and P^2 -Lax-Friedrichs-scalar-DG both using equally spaced cells. $u_t + (\sin(t)u)_x = 0$ with smooth initial conditions: $u_0(x) = \sin(x)$. L^2 , L^1 convergence orders for u .

L^2 , L^1 numerical errors and convergence orders with P^3 -polynomial approximation are summarized respectively in Table 11 and Figure 19.

TABLE 11. P^5 -Steger-Warming-scalar-DG and P^5 -Lax-Friedrichs-scalar-DG both using equally spaced cells. $u_t + (\frac{1}{2}u^2)_x = 0$ with smooth initial conditions: $u_0(x) = \sin(x)$. L^2 , L^1 errors and convergence orders for u .

	Mesh	20	40	80	160	320
S-W	L^2 -error	5.342170E-06	1.62220980E-07	2.285972076E-09	3.9838927E-11	8.2960E-13
L-F	L^2 -error	5.342175E-06	1.62220986E-07	2.285972077E-09	3.9838921E-11	8.2980E-13
S-W	L^1 -error	4.25502E-06	1.0785780E-07	1.4120418E-09	2.2865E-11	1.2853E-12
L-F	L^1 -error	4.25507E-06	1.0785784E-07	1.4120415E-09	2.2864E-11	1.2861E-12

- 2D-Control Eqs: $U_t + (\frac{1}{2}U^2)_x + (\frac{1}{2}U^2)_y = 0$;
 Computational domain: $\Omega \times [0, T_{end}] = \{[0, 4] \times [0, 4]\} \times [0, \frac{0.5}{\pi}]$;
 Note: the exact solution remains smooth during $t \in [0, \frac{0.5}{\pi}]$;

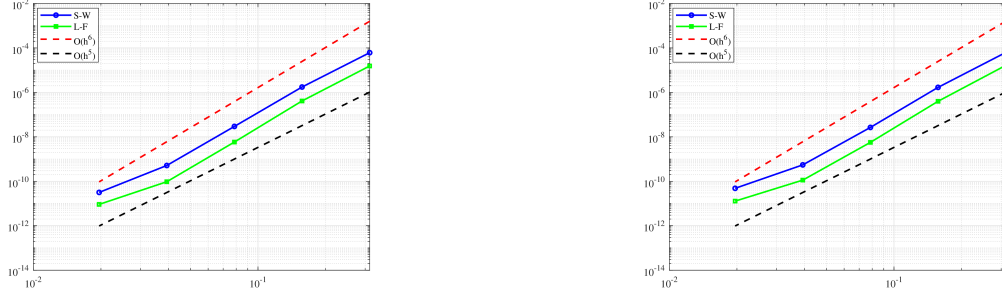
(a) L^2 -convergence order(b) L^1 -convergence order

FIGURE 18. P^5 -Steger-Warming-scalar-DG and P^5 -Lax-Friedrichs-scalar-DG both using equally spaced cells. $u_t + (\sin(x)u)_x = 0$ with smooth initial conditions: $u_0(x) = 1$. L^2 , L^1 convergence orders for u .

ICs: $U_0(x) = \sin(\frac{\pi}{2}x + \frac{\pi}{2}y)$;

BCs: *periodic boundary conditions*;

By performing the variable substitution $\xi = x + y$ on $U_0(x, y)$, we obtain $\tilde{u}_0(\xi) = \sin(\frac{\pi}{2}\xi)$, then the true solution is given as:

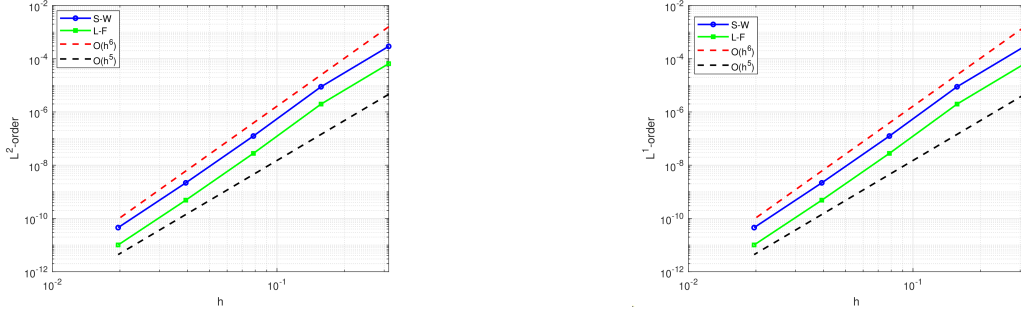
$U(x, y, t) = \tilde{u}_0(\xi^*)$, ξ^* satisfies $\xi = \xi^* + 2\tilde{u}_0(\xi^*) \cdot t$, where $\xi = x + y$;

Temporal discretization format: TVD-RK3;

CFL=0.05.

L^∞ , L^2 , L^1 numerical errors and convergence orders with P^2 -polynomial approximation based on scalar Steger-Warming flux are summarized in Table 12.

The one-dimensional and two-dimensional numerical results indicate that the P^K -DG method, based on the Steger-Warming flux scheme, is capable of achieving the optimal convergence order of $K + 1$. Furthermore, the accuracy test results in one-dimensional case indicate that for scalar equations, the dissipation of the Steger-Warming flux is slightly lower



(a) L^2 -convergence order

(b) L^1 -convergence order

FIGURE 19. P^5 -Steger-Warming-scalar-DG and P^5 -Lax-Friedrichs-scalar-DG both using equally spaced cells. $u_t + (\frac{1}{2}u^2)_x = 0$ with smooth initial conditions: $u_0(x) = \sin(x)$. L^2 , L^1 convergence orders for u .

TABLE 12. P^2 -Steger-Warming-scalar-DG using uniform rectangular meshes. 2D-Burgers' Equation with smooth initial condition: $U_0(x) = \sin(\frac{\pi}{2}x + \frac{\pi}{2}y)$. L^∞ , L^2 , L^1 errors and convergence orders for U .

Mesh	15×15	30×30	60×60	120×120	240×240
L^∞ -error	6.5224E-02	1.1657E-02	1.5466E-03	1.8680E-04	3.0712E-05
L^∞ -order	—	2.4843	2.9140	3.0495	2.6046
S-W L^2 -error	5.1671E-02	6.5497E-03	8.5550E-04	1.1426E-04	1.5193E-05
L^2 -order	—	2.9798	2.9366	2.9044	2.9109
L^1 -error	1.0390E-01	1.1668E-02	1.6231E-03	2.1695E-04	2.7951E-05
L^1 -order	—	3.1546	2.8456	2.9034	2.9564

than that of the classical Lax-Friedrichs flux. This is consistent with the statement that “in system control equations, the dissipation of the Steger-Warming splitting is less than that of the Lax-Friedrichs splitting”.

Remark C.2. *Due to the excessive dissipation of the Lax-Friedrichs flux, it paradoxically results in a slightly smaller L^∞ -error compared to the Steger-Warming flux. This is primarily because the L^∞ -error is more sensitive to peaks than L^2 -error and L^1 -error; hence, a scheme with higher dissipation may often achieve a smaller L^∞ -error. The degree of dissipation of numerical fluxes becomes more pronounced in high-order schemes, which is why this paper selects the P^5 -DG method for testing, in order to more clearly reflect the dissipative characteristics of the L-F and S-W fluxes respectively. Furthermore, the dissipative properties of the flux scheme become evident in long-time numerical simulations. At such times, using lower-order polynomial approximation, such as P^2 -polynomial approximation, can distinguish the degree of dissipation between the S-W flux and the L-F flux, as we have done in the test case “ $u_t + (\sin(\omega t)u)_x = 0, t \in [0, 20]$ ”.*

- Numerical oscillation test

Example C.3. *We continue to utilize the one-dimensional test case presented in Example C.2, with the simulation duration extended to $T_{end} = 2.6$, at which point discontinuities have developed. No additional limiters are incorporated to amend the numerical solutions, allowing for observing the inherent performance differences of the Steger-Warming and Lax-Friedrichs numerical fluxes near discontinuities. Contrastive tests were performed under four distinct conditions: coarse and fine meshes, as well as low- and high-order polynomial approximations (P^1 and P^3). The results are demonstrated in Figure 20.*

Without the introduction of any limiters, near discontinuities, the number of oscillations and the degree of overshoot of the L-F flux are both greater than those of the S-W flux. Additionally, by observing the jump in the DG approximation solution u_h at the cell interfaces, it is found that $[[u_h]]_{x_{i+1/2}}^{L-F} > [[u_h]]_{x_{i+1/2}}^{S-W}$. In summary, the S-W flux scheme inherently has a better capability for handling discontinuities than the L-F flux, and the DG approximation based on the S-W flux is more “smooth” (with smaller jumps at interfaces).

(Z. Xie) SCHOOL OF MATHEMATICAL SCIENCES, EAST CHINA NORMAL UNIVERSITY, SHANGHAI 200241, CHINA

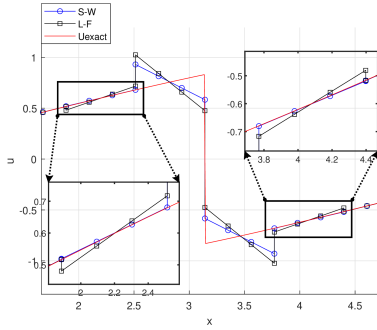
Email address: xzr_nature@163.com; 52265500018@stu.ecnu.edu.cn

(X. Cai) RESEARCH CENTER FOR MATHEMATICS, BEIJING NORMAL UNIVERSITY, DIVISION OF SCIENCE AND TECHNOLOGY, BNU-HKBU UNITED INTERNATIONAL COLLEGE, ZHUHAI 519087, CHINA

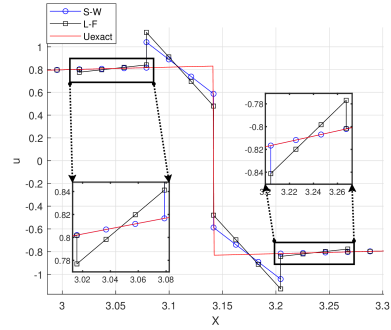
Email address: xfcai@bnu.edu.cn

(H. Zheng) SCHOOL OF MATHEMATICAL SCIENCES, EAST CHINA NORMAL UNIVERSITY, SHANGHAI 200241, CHINA

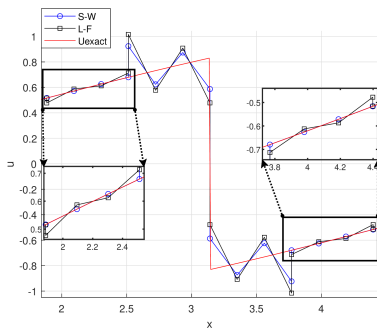
Email address: hbzheng@math.ecnu.edu.cn



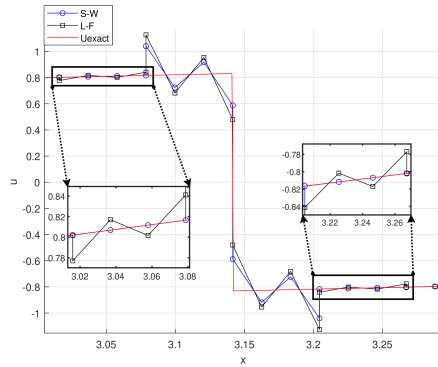
(a) P^1 , Mesh=10



(b) P^1 , Mesh=100



(c) P^3 , Mesh=10



(d) P^3 , Mesh=100

FIGURE 20. Steger-Warming scalar DG and Lax-Friedrichs scalar DG, both