

# Finite difference alternative WENO schemes with Riemann invariant-based local characteristic decompositions for compressible Euler equations\*

Yue Wu<sup>†</sup> and Chi-Wang Shu<sup>‡</sup>

**Abstract:** The weighted essentially non-oscillatory (WENO) schemes are widely used for hyperbolic conservation laws due to the ability to resolve discontinuities and maintain high-order accuracy in smooth regions at the same time. For hyperbolic systems, the WENO procedure is usually performed on local characteristic variables that are obtained by local characteristic decompositions to avoid oscillation near shocks. However, such decompositions are often computationally expensive. In this paper, we study a Riemann invariant-based local characteristic decomposition for the compressible Euler equations that reduces the cost. We apply the WENO procedure to the local characteristic fields of the Riemann invariants, where the eigenmatrix is sparse and thus the computational cost can be reduced. It is difficult to obtain the cell averages of Riemann invariants from those of the conserved variables due to the nonlinear relation between them, so we only focus on the finite difference alternative WENO versions. The efficiency and non-oscillatory property of the proposed schemes are well demonstrated by our numerical results.

**Mathematics Subject Classification:** 65M06, 76M20

**Keywords:** hyperbolic conservation laws, compressible flow, Riemann invariants, finite difference alternative WENO schemes

## 1 Introduction

It is well known that nonlinear hyperbolic conservation laws, e.g. the compressible Euler equations for gas dynamics, can generate sharp discontinuities in finite time even with smooth initial conditions. Numerically resolving these discontinuities while maintaining high-order accuracy in smooth regions is a challenging task, because spurious oscillations that appear near the discontinuities may mess/blow up the simulation. Inspired by the essentially non-oscillatory (ENO) schemes [9, 20, 21], the weighted ENO (WENO) schemes, originally proposed in [14] and improved to achieve higher order in [12], perform well in this regard. Both finite volume and finite difference versions of the WENO schemes have been developed and widely used in practice, and many other variants are proposed. For simplicity, we only focus on the WENO-JS version [12] in this paper. Besides the classical finite difference WENO formulation that is applied to point values of split flux functions, there is also one formulation called alternative WENO (A-WENO) [13] that works on those of conserved variables, which roots back to early studies of the ENO method [20].

---

\*Research is supported in part by NSF grant DMS-2309249.

<sup>†</sup>Division of Applied Mathematics, Brown University, Providence, RI 02912, USA (yue.wu3@brown.edu).

<sup>‡</sup>Division of Applied Mathematics, Brown University, Providence, RI 02912, USA (chi-wang\_shu@brown.edu).

For scalar hyperbolic conservation laws, the application of WENO methods is straightforward, and the oscillations are well suppressed. However, the situation is more complicated for systems. The component-wise ENO/WENO schemes often produce oscillatory solutions after discontinuities where multiple shocks interact with each other [9], such as Riemann problems. To avoid this, the local characteristic decomposition (LCD) procedure should be involved. Both the finite volume [9] and finite difference [21] versions include the multiplication of the left and right eigenmatrices with corresponding vectors. For a 1D system consisting of  $m$  components, the extra cost at each cell interface of a  $r$ th-order ENO (or a  $(2r - 1)$ th-order WENO) scheme is  $4r + 2$  multiplications of an  $m \times m$ -matrix and an  $m$ -vector, compared to the component-wise version. This costs  $(4r + 2)m^2$  floating point multiplications in a vanilla implementation. Various experiments have shown that LCD does take a big portion of the computational time [22]. There have been some attempts to alleviate this extra cost, but none of them works well for the compressible Euler equations in high-order simulations. For example, Jiang and Shu [12] proposed to use pressure and entropy to compute the nonlinear WENO weights for compressible Euler equations, but oscillations can occur near strong shocks. The second-order component-wise central ACM-MUSCL scheme [16] and the third-order component-wise central PPM method [15] are both non-oscillatory, whereas the LCD is still necessary to suppress oscillations in higher-order central schemes [17]. Xu and Shu [28] circumvented the LCD procedure by applying the WENO interpolation on the coordinate system of Riemann invariants, if the equations admit such a system. This reduces the per-node cost (other than the WENO part) dependency upon the stencil width  $2r$  from linear to constant, while maintaining non-oscillatory and high-order. Unfortunately, the compressible Euler equations do not belong to this category. In this paper, we propose to use the Riemann invariants to improve the sparsity of the eigenmatrix, so as to reduce the cost of the LCD procedure for the compressible Euler equations. Although we cannot completely eliminate the linear dependency upon  $r$  of the per-node LCD cost, a big improvement is still available by bringing the matrix as close to the identity as possible. Because of the nonlinear relation between the conserved variables and the Riemann invariants, it is difficult to obtain the cell averages of the latter from those of the former, so we only concentrate on the finite difference A-WENO versions, which are based on the point values of the conserved variables.

The rest of the paper is organized as follows. In Section 2, we introduce the Riemann invariants and the compressible Euler equations. In Section 3, we review the A-WENO scheme and propose the Riemann invariant-based LCD. In Section 4, we present our numerical results. Finally, we conclude in Section 5.

## 2 Preliminaries

### 2.1 Riemann invariants

Consider a system of conservation laws of  $m$  components in one space dimension

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = \mathbf{0}, \quad (2.1)$$

where  $\mathbf{u} = (u_1, u_2, \dots, u_m)^T \in \mathbb{R}^m$  are the conserved variables and  $\mathbf{f}(\mathbf{u}) = (f_1(\mathbf{u}), f_2(\mathbf{u}), \dots, f_m(\mathbf{u}))^T \in \mathbb{R}^m$  are the flux functions. We are only interested in the case where  $\mathbf{f}(\mathbf{u})$  is a  $C^1$  function of  $\mathbf{u}$ . Assuming all the functions are differentiable, the system (2.1) can be rewritten into a general quasilinear form,

$$\mathbf{u}_t + \mathbf{A}(\mathbf{u})\mathbf{u}_x = \mathbf{0}, \quad (2.2)$$

where  $\mathbf{A}(\mathbf{u}) = \frac{\partial \mathbf{f}}{\partial \mathbf{u}}$  is the Jacobian matrix.

**Definition 2.1** (hyperbolicity in 1D). The quasilinear system (2.2) is hyperbolic if the Jacobian matrix  $\mathbf{A}(\mathbf{u})$  is diagonalizable for all  $\mathbf{u}$ , i.e. there exists an invertible matrix  $\mathbf{R}(\mathbf{u})$  such that  $\mathbf{A}(\mathbf{u}) = \mathbf{R}(\mathbf{u})\mathbf{\Lambda}(\mathbf{u})\mathbf{R}(\mathbf{u})^{-1}$ , where  $\mathbf{\Lambda}(\mathbf{u}) = \text{diag}(\lambda_1(\mathbf{u}), \lambda_2(\mathbf{u}), \dots, \lambda_m(\mathbf{u}))$  is the diagonal matrix of eigenvalues,  $\mathbf{R}(\mathbf{u}) = (\mathbf{r}_1(\mathbf{u}), \mathbf{r}_2(\mathbf{u}), \dots, \mathbf{r}_m(\mathbf{u}))^T$  is the right eigenmatrix, and  $\mathbf{R}(\mathbf{u})^{-1} = \mathbf{L}(\mathbf{u}) = (\mathbf{l}_1(\mathbf{u})^T, \mathbf{l}_2(\mathbf{u})^T, \dots, \mathbf{l}_m(\mathbf{u})^T)^T$  is the left eigenmatrix.

Correspondingly, (2.1) is called a system of hyperbolic conservation laws, if the Jacobian matrix  $\frac{\partial \mathbf{f}(\mathbf{u})}{\partial \mathbf{u}}$  is diagonalizable for all  $\mathbf{u}$ .

**Definition 2.2** (hyperbolicity in multi-D). For a system of conservation laws in  $d$  spatial dimensions,

$$\mathbf{u}_t + \sum_{i=1}^d \mathbf{f}_i(\mathbf{u})_{x_i} = \mathbf{0}, \quad (2.3)$$

it is hyperbolic if for any unit vector  $\mathbf{n} = (n_1, n_2, \dots, n_d)$ , the following 1D equation

$$\mathbf{u}_t + \sum_{i=1}^d n_i \mathbf{f}_i(\mathbf{u})_x = \mathbf{0} \quad (2.4)$$

is hyperbolic.

*Remark 2.1.* In the following, we will only state the results for 1D systems, but they can be easily extended to multi-D by applying it to each direction.

**Definition 2.3** (Riemann invariant). A scalar-valued function  $w(\mathbf{u})$  is called an  $i$ -Riemann invariant for the hyperbolic system (2.2) if  $\nabla w(\mathbf{u}) \cdot \mathbf{r}_i(\mathbf{u}) = 0$  for any  $\mathbf{u}$ .

**Corollary 2.1** (closedness under composition of differentiable functions). *If  $w(\mathbf{u})$  is an  $i$ -Riemann invariant, then for any  $C^1$  function  $g : \mathbb{R} \mapsto \mathbb{R}$ ,  $g(w(\mathbf{u}))$  is also an  $i$ -Riemann invariant. Moreover, a differentiable function of several  $i$ -Riemann invariants is also an  $i$ -Riemann invariant.*

**Theorem 2.1** (jump of Riemann invariants [23, Theorem 17.16]). *For a system of hyperbolic conservation laws (2.1), the change of an  $i$ -Riemann invariant  $w(\mathbf{u})$  across an isolated  $i$ -rarefaction or  $i$ -contact is 0, whereas the change across an isolated  $i$ -shock is  $\mathcal{O}(\epsilon^3)$  as  $\epsilon \rightarrow 0$ , where  $\epsilon = |\lambda_i(\mathbf{u}_l) - \lambda_i(\mathbf{u}_r)|$  is the shock strength.*

**Proposition 2.1** (change of variables). *Consider a hyperbolic system (2.2) and assume all the functions are differentiable. If there exists a differentiable and invertible transform  $\mathbf{v}(\mathbf{u}) : \mathbb{R}^m \mapsto \mathbb{R}^m$  for all  $\mathbf{u}$ , then it can be rewritten as another hyperbolic system,*

$$\mathbf{v}_t + \tilde{\mathbf{A}}(\mathbf{v})\mathbf{v}_x = \mathbf{0}, \quad (2.5)$$

where  $\tilde{\mathbf{A}} = \frac{\partial \mathbf{v}}{\partial \mathbf{u}} \mathbf{A} \frac{\partial \mathbf{u}}{\partial \mathbf{v}}$ . Moreover,  $\tilde{\mathbf{A}}$  have the same eigenvalues as  $\mathbf{A}$ , and the corresponding eigenmatrices are  $\tilde{\mathbf{R}} = \frac{\partial \mathbf{v}}{\partial \mathbf{u}} \mathbf{R}$  and  $\tilde{\mathbf{L}} = \mathbf{L} \frac{\partial \mathbf{u}}{\partial \mathbf{v}}$ .

**Corollary 2.2** (sparsity of the eigenmatrix). *Assume there exists a differentiable and invertible transform  $\mathbf{v}(\mathbf{u}) = (v_1(\mathbf{u}), v_2(\mathbf{u}), \dots, v_m(\mathbf{u}))^T$  for all  $\mathbf{u}$ , where  $v_i(\mathbf{u})$  is a  $j$ -Riemann invariant, then for the transformed system (2.5), the  $(i, j)$ -th entry of  $\tilde{\mathbf{R}}$  is 0.*

*Remark 2.2* (coordinate system of Riemann invariants [28]). If  $v_i(\mathbf{u})$  is a  $(1, \dots, i-1, i+1, \dots, m)$ -Riemann invariant for all  $i$ 's, the system (2.1) is endowed with a coordinate system of Riemann invariants which is exactly  $\mathbf{v}(\mathbf{u})$ . Then for the transformed system (2.5), the eigenmatrix  $\tilde{\mathbf{R}}$  is diagonal and so is  $\tilde{\mathbf{A}}$ . In this case we can even take  $\tilde{\mathbf{R}} = \mathbf{I}$ .

## 2.2 The compressible Euler equations

The 2D compressible Euler equations are

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x + \mathbf{g}(\mathbf{u})_y = \mathbf{0}, \quad (2.6a)$$

where the conserved variables are

$$\mathbf{u} = (\rho, \rho u, \rho v, E)^T, \quad (2.6b)$$

and the fluxes are

$$\begin{aligned} \mathbf{f}(\mathbf{u}) &= (\rho u, \rho u^2 + p, \rho uv, u(E + p))^T, \\ \mathbf{g}(\mathbf{u}) &= (\rho v, \rho vu, \rho v^2 + p, v(E + p))^T. \end{aligned} \quad (2.6c)$$

Here,  $\rho$  is the density,  $\vec{u} = (u, v)$  is the velocity,  $E = \frac{\rho}{2} |\vec{u}|^2 + \rho e$  is the total energy,  $e$  is the specific internal energy, and  $p = p(\rho, e)$  is the pressure determined by the equation of state (EOS). Additional derived variables include the total enthalpy  $H = \frac{E+p}{\rho}$ , the sonic speed  $c = \sqrt{\left. \frac{\partial p}{\partial \rho} \right|_s}$  and the specific entropy  $s$ . We consider the simplest  $\gamma$ -law EOS for ideal polytropic gas,

$$p = \rho e(\gamma - 1), \quad (2.7)$$

with  $\gamma = \frac{c_p}{c_v} > 1$  being the specific heat capacity ratio which is constant. Under this EOS, we have  $c = \sqrt{\frac{\gamma p}{\rho}}$  and  $s = \ln(p\rho^{-\gamma})$ . We use  $S = \exp(s) = p\rho^{-\gamma}$  to denote the entropy.

We list the eigenstructure of the Jacobian matrix in (A.1), (A.2) and (A.3). For the 1D version of the equations, the eigenmatrices can be obtained by first restricting  $v = 0$  and then removing the third row and column of the 2D eigenmatrices. By simple algebraic manipulations, it is easy to check that the equations have the following Riemann invariants corresponding to the eigenmatrix in (A.2).

**Proposition 2.2** (Riemann invariants of the 2D compressible Euler equations). *The 2D compressible Euler equations (2.6) with the  $\gamma$ -law EOS (2.7) has the following Riemann invariants in the  $x$ -direction.*

- 1-Riemann invariants:  $S, u + \frac{2}{\gamma-1}c, v$ .
- 2-Riemann invariants:  $u, p, v$ .
- 3-Riemann invariants:  $S, u, p$ .
- 4-Riemann invariants:  $S, u - \frac{2}{\gamma-1}c, v$ .

## 3 Numerical methods

### 3.1 The A-WENO scheme with transform variables

Consider the 1D hyperbolic system (2.1) on a finite interval  $[a, b]$ . We use the uniform grid  $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$  for  $j = 1, 2, \dots, N$ , where  $x_j = a + (j - \frac{1}{2})\Delta x$  and  $\Delta x = \frac{b-a}{N}$ . The semi-discrete  $(2r - 1)$ th-order finite difference A-WENO scheme is

$$\frac{d}{dt} \mathbf{u}_j + \frac{1}{\Delta x} (\hat{\mathbf{f}}_{j+\frac{1}{2}} - \hat{\mathbf{f}}_{j-\frac{1}{2}}) = \mathbf{0}. \quad (3.1)$$



Here,  $\mathbf{u}_j$  is the approximation to  $\mathbf{u}(t, x_j)$  and  $\hat{\mathbf{f}}_{j+\frac{1}{2}}$  is a numerical flux at the cell interface  $x_{j+\frac{1}{2}}$ .

Given a set of transform variables  $\mathbf{v}(\mathbf{u})$ , the procedure to compute  $\hat{\mathbf{f}}_{j+\frac{1}{2}}$ 's is as follows.

1. Obtain ghost point values using the boundary conditions.
2. Compute all the flux function values  $\mathbf{f}_j = \mathbf{f}(\mathbf{u}_j)$  and all the transform variables  $\mathbf{v}_j = \mathbf{v}(\mathbf{u}_j)$ .
3. For each cell interface  $x_{j+\frac{1}{2}}$ :
  - (a) Compute the Roe average  $\mathbf{u}_{j+\frac{1}{2}}^{\text{Roe}}$  using  $\mathbf{u}_j$  and  $\mathbf{u}_{j+1}$ , then obtain the transformed eigenmatrices  $\tilde{\mathbf{R}}_{j+\frac{1}{2}} = \tilde{\mathbf{R}}(\mathbf{u}_{j+\frac{1}{2}}^{\text{Roe}})$  and  $\tilde{\mathbf{L}}_{j+\frac{1}{2}} = \tilde{\mathbf{L}}(\mathbf{u}_{j+\frac{1}{2}}^{\text{Roe}})$  at the cell interface.
  - (b) Compute the local characteristic variables  $\mathbf{w}_{j-r+s} = \tilde{\mathbf{L}}_{j+\frac{1}{2}} \mathbf{v}_{j-r+s}$  in the whole stencil  $s = 1, 2, \dots, 2r$ .
  - (c) Use WENO interpolation (see Section B for details) to obtain approximations  $\mathbf{w}_{j+\frac{1}{2}}^- = \text{WENO}(\mathbf{w}_{j-r+1}, \dots, \mathbf{w}_{j+r-1})$  and  $\mathbf{w}_{j+\frac{1}{2}}^+ = \text{WENO}(\mathbf{w}_{j+r}, \dots, \mathbf{w}_{j-r+2})$ .
  - (d) Transform back to  $\mathbf{v}_{j+\frac{1}{2}}^\pm = \tilde{\mathbf{R}}_{j+\frac{1}{2}} \mathbf{w}_{j+\frac{1}{2}}^\pm$  and then to the conserved variables  $\mathbf{u}_{j+\frac{1}{2}}^\pm = \mathbf{u}(\mathbf{v}_{j+\frac{1}{2}}^\pm)$ .
  - (e) Compute the lowest-order flux  $\hat{\mathbf{f}}_{j+\frac{1}{2}}^{\text{low}} = \hat{\mathbf{f}}(\mathbf{u}_{j+\frac{1}{2}}^-, \mathbf{u}_{j+\frac{1}{2}}^+)$  using some approximate Riemann solver, e.g. the HLL-family flux.
  - (f) Compute the high-order flux correction terms (see Section C for details)  $\hat{\mathbf{f}}_{j+\frac{1}{2}}^{\text{cor}}$  using flux function values  $\mathbf{f}_{j-r+s}$  for  $s = 1, 2, \dots, 2r$ . Obtain  $\hat{\mathbf{f}}_{j+\frac{1}{2}} = \hat{\mathbf{f}}_{j+\frac{1}{2}}^{\text{low}} + \hat{\mathbf{f}}_{j+\frac{1}{2}}^{\text{cor}}$ .

Time is advanced using the strong stability-preserving Runge-Kutta (SSP-RK) methods [6]. The time step is determined by the CFL condition,

$$\Delta t = \text{CFL} \min_j \frac{1}{\frac{\alpha_{j+\frac{1}{2}}}{\Delta x}}, \quad (3.2)$$

where  $\alpha_{j+\frac{1}{2}}$  is the estimate of the maximum wave speed at the cell interface  $x_{j+\frac{1}{2}}$  by the selected approximate Riemann solver.

For the 2D case, the semi-discrete A-WENO scheme is performed in a dimension-by-dimension manner. The time step should be taken as

$$\Delta t = \text{CFL} \min_{i,j} \frac{1}{\frac{\alpha_{i+\frac{1}{2},j}}{\Delta x} + \frac{\beta_{i,j+\frac{1}{2}}}{\Delta y}}, \quad (3.3)$$

where  $\alpha_{i+\frac{1}{2},j}$  is the estimate of the maximum wave speed in the  $x$ -direction at the cell interface  $(x_{i+\frac{1}{2}}, y_j)$  and  $\beta_{i,j+\frac{1}{2}}$  is the estimate in the  $y$ -direction at the cell interface  $(x_i, y_{j+\frac{1}{2}})$ .

### 3.2 Reducing the cost of LCD

Besides the WENO interpolation part, the most costly part that grows with the stencil width  $2r$  or the scheme order  $k = 2r - 1$  should be the LCD procedure, which is documented to take a big portion of the total computational time [22]. For a system of conservation laws consisting of  $m$  components, the A-WENO LCD procedure at *every* cell interface  $x_{j+\frac{1}{2}}$  has the following costs: (1)  $4r$  multiplications between the left eigenmatrix  $\tilde{\mathbf{L}}_{j+\frac{1}{2}}$  and vectors of transform variables  $\mathbf{v}_{j-r+s}$ , (2) 2 multiplications between the right eigenmatrix  $\tilde{\mathbf{R}}_{j+\frac{1}{2}}$  and vectors of characteristic vectors  $\mathbf{w}_{j+\frac{1}{2}}^\pm$ . A vanilla implementation of the matrix-vector multiplication will cost  $m^2$  floating point multiplications. With these in mind, one could easily see that the major part that grows with the stencil width is the left eigenmatrix-vector multiplication. Therefore, reducing this cost is the key to improve the efficiency of the A-WENO scheme at high-order.

One attempt for some hyperbolic systems so far is to use the coordinate system of Riemann invariants [28] as the transform variables to completely avoid computing such multiplications, because in this case both the left and right eigenmatrices are diagonal and thus can be replaced by the identity matrix. Unfortunately, the compressible Euler equations (2.6) with the  $\gamma$ -law EOS (2.7) is *not* endowed with such a coordinate system. (See detailed proof in Proposition A.1.) We have to seek other ways to reduce the cost, and our main freedom lies in choosing “good” transform variables  $\mathbf{v}(\mathbf{u})$  and designing efficient left eigenmatrix-vector multiplication subroutines.

If we stick with the trivial transform variables  $\mathbf{v} = \mathbf{u}$ , there are still some possibilities to improve the efficiency. Two main ideas are to suitably normalize the eigenvectors to increase the number of unit elements in the eigenvectors so that to avoid multiplications, and to reuse intermediate results as possible. Take the 2D case as an example. The left eigenvectors are split as  $\mathbf{l}_1 = \boldsymbol{\xi}_1 - \boldsymbol{\xi}_2$ ,  $\mathbf{l}_2 = \boldsymbol{\xi}_2 - \boldsymbol{\xi}_3$ ,  $\mathbf{l}_3 = \boldsymbol{\xi}_4$  and  $\mathbf{l}_4 = \boldsymbol{\xi}_1 + \boldsymbol{\xi}_2$  as in (A.3). When computing  $\mathbf{w} = \mathbf{L}\mathbf{u}$ , we first calculate the intermediate results  $\boldsymbol{\xi}_i \cdot \mathbf{u}$  and then sum them back to get the final results. Although there are still 4 vectors, cost can be saved because of their sparsity. The inner product between a general vector and the  $\boldsymbol{\xi}_i$ ’s require only 1, 4, 1 and 1 “true” multiplications, so the total number of multiplications can go down from 16 to 7.

Inspired by Corollary 2.2, it is possible to improve the sparsity of the eigenmatrix so as to reduce the cost further. We use the following Riemann variables as transform variables for the 2D compressible Euler equations in  $x$ -direction,  $\mathbf{v}(\mathbf{u}) = (u - \frac{2}{\gamma-1}c, S^{\frac{1}{2\gamma}}, v, u + \frac{2}{\gamma-1}c)$ , whose related eigenmatrices are given in (A.4) and (A.5). Each of the eigenmatrix has only two non-zero off-diagonals. Moreover, these two off-diagonals are opposite to each other and all lie in the second column. Let us denote the positive off-diagonal element by  $\mu$ . We can implement the left eigenmatrix-vector multiplication  $\mathbf{w} = \tilde{\mathbf{L}}\mathbf{v}$  using only 1 multiplications: compute the intermediate results  $a = \mu v_2$ , then the final result is just  $\mathbf{w} = (v_1 + a, v_2, v_3, v_1 - a)$ . It’s same for both 1D and 2D versions of the equations.

We list the cost of the original and our new method in Table 3.1. In terms of the whole scheme, the cost reduction will not be as significant as the reduction of solely the matrix-vector multiplications, because possible additional work is required to compute the forward and backward transforms and the coefficients in the new eigenmatrices. In this particular case, the cost of forward and backward transforms is only constant per-node, but such constant may become relatively large because of the exponential part in  $S^{\frac{1}{2\gamma}} = p^{\frac{1}{2\gamma}} \rho^{-\frac{1}{2}}$  which is expensive for CPUs. It might exceed the reduction of cost gained from our new LCD if not treated properly. Highly optimized SIMD instructions are needed to overcome this issue.

Table 3.1: Number of floating point multiplications of one left eigenmatrix-vector multiplication for the compressible Euler equations of different dimensions.

	original method	our new method
1D	9	1
2D	16	1

### 3.3 Properties of the new scheme

We emphasize here that with any smooth transform variables, the A-WENO scheme is still conservative, and the convergence rates for smooth solutions will remain the same. Moreover, with our selected Riemann invariant-based transform variables, the scheme has the following property.

In 1D simulations, the E-property [7] refers to the ability to maintain constant velocity  $u$  and pressure  $p$  in the fully discrete solution if they are initially constant, regardless of the initial condition of  $\rho$ . Schemes that satisfy the E-property, such as the primitive variable-based and the classical characteristic-wise A-WENO schemes, do not produce oscillations near contact discontinuities. In contrast, schemes that do not satisfy the E-property, such as the component-wise A-WENO scheme, may produce oscillations in these regions. By carefully analyzing the evolution of every component of the variable in a fully discretized scheme, we have the following proposition.

**Proposition 3.1** (E-property). *Assume the WENO procedure is affine-invariant, i.e. the WENO procedure is commutative with affine transformations. Then for the 1D case, both the original and the Riemann invariant-based A-WENO schemes satisfy the E-property.*

*Proof.* See [7] for the proof for the original scheme. The proof for our Riemann invariant-based version is similar. The key point is to notice that when  $u_j$  and  $p_j$  are constants, the WENO interpolation for each component of  $v_{j+\frac{1}{2}}^-$  actually gives the same result as that for  $(\rho_{j+\frac{1}{2}}^-)^{-\frac{1}{2}}$ , up to affine transformations.  $\square$

*Remark 3.1.* Actually, the WENO-JS procedure is not strictly affine-invariant in computation because of the artificial  $\epsilon$  to avoid division by zero. However, such impact is very small because  $\epsilon$  can be taken near machine zero. We refer the readers to the strictly affine-invariant WENO procedure [26, 5] for more details.

*Remark 3.2.* The exponential term  $S^{\frac{1}{2\gamma}}$  may seem arbitrary at first glance, because Corollary 2.1 implies that any smooth and invertible function of  $S$  can take the second position in the transform variables while preserving the sparsity of the eigenmatrix. However, it is actually the *only* choice to satisfy the E-property, and it is less oscillatory near contact discontinuities than other choices.

### 3.4 Positivity-preserving limiters

During the computation, it is crucial to ensure that certain variables, which are both mathematically and physically required to be positive, do not take negative values so that the computation might break down. This can be achieved by implementing two positivity-preserving (PP) limiters in [4, Appendix D]. These limiters only interfere when certain negativity is detected, so in most of the time they contribute little to the computational cost.

The first one is a PP limiter on the WENO interpolation results. Let us only consider the 1D case, as generalizations to 2D are straightforward. The admissible set of the transform variables  $\mathbf{v} = (u - \frac{2}{\gamma-1}c, S^{\frac{1}{2\gamma}}, u + \frac{2}{\gamma-1}c)$ ,  $\mathcal{G} = \{\mathbf{v} \in \mathbb{R}^3 : v_1 < v_3, v_2 > 0\}$ , is convex, so the convex combination-based limiter [29] can be directly adopted to the interpolated transform variables  $\mathbf{v}_{j+\frac{1}{2}}^\pm$  in our framework.

The second one is the so-called PP flux limiter, which requires a stricter CFL condition to maintain accuracy when it is triggered. The readers are referred to [11] for details.

## 4 Numerical results

In this section, we perform numerical tests using 5th-, 7th- and 9th-order A-WENO schemes. The time is discretized by the explicit 3rd-order SSP-RK method. The classical conserved variable based characteristic decomposition is abbreviated as CH-CON, the component-wise version is abbreviated as CP-CON and the Riemann invariant-based one is abbreviated as CH-RI. We use the HLL numerical flux [10]. For the lowest-order flux, the positivity-preserving Einfeldt's wave speed estimate [2, 3] is used. For the PP flux in the flux limiter, we adopt the more dissipative two-rarefaction wave speed estimate [24, Section 10.5.2] that proves to bound the physical wave speeds for  $\gamma \in (1, \frac{5}{3}]$  [8]. Unless otherwise specified, the CFL number is set to 0.5, and the specific heat capacity ratio  $\gamma$  is set to 1.4 for air.

First of all we list the per-time step CPU time cost of the three versions of the A-WENO scheme. Following that, different test examples are presented to verify the accuracy and robustness of our CH-RI version in contrast to the classical CH-CON version.

### 4.1 Efficiency test

We use  $N = 2000$  nodes for the 1D tests and  $(Nx, Ny) = (400, 400)$  nodes for the 2D tests. The tests are conducted on Intel(R) Core(TM) i7-9750H CPU @ 2.60GHz with Intel Turbo Boost disabled. The compiler is Intel Fortran IFX 2025.0.0 20241008, and the compiling options are `-O3 -xHost -r8 -qopenmp-stubs`. The Intel SVML library is triggered by OpenMP SIMD directives to accelerate floating point exponentials. The results are shown in Table 4.1. The LCD time saving refers to the percentage of LCD time reduction of the CH-RI version compared to the CH-CON version.

Table 4.1: Efficiency test. Per-time step CPU time cost (in seconds) of the A-WENO scheme.

	order	CH-RI	CH-CON	CP-CON	LCD time saving
1D	3	1.58E-03	1.98E-03	1.33E-03	62%
	5	1.83E-03	2.30E-03	1.61E-03	68%
	7	2.23E-03	2.72E-03	2.01E-03	69%
	9	2.26E-03	2.82E-03	2.10E-03	78%
2D	3	4.98E-01	6.36E-01	4.69E-01	82%
	5	5.20E-01	6.66E-01	5.07E-01	92%
	7	5.39E-01	6.84E-01	5.20E-01	88%
	9	6.21E-01	7.67E-01	5.99E-01	86%

## 4.2 1D tests

### 4.2.1 Accuracy test 1

We consider the simple periodic density-transport problem whose exact solution is  $(\rho, u, p) = (1 + 0.2 \sin(\pi(x - t)), 1, 1)$ . The computation domain is  $[0, 2]$  with periodic boundaries. Solve up to  $T = 2$ . We use  $N_0 = 20$  as the coarsest mesh and use  $h_0 = \frac{2}{N_0}$  to denote the coarsest mesh size. Using SSP-RK(3,3), the time steps in the following tests are taken to be  $\Delta t = \text{CFL} \frac{\Delta x}{\alpha} \left( \frac{\Delta x}{h_0} \right)^{\frac{k}{3}-1}$ , which makes the time discretization error  $k$ th-order while maintaining stability. Here,  $\alpha$  is the maximum wave speed. The errors are shown by Tables 4.2 to 4.4.

Table 4.2: Accuracy test 1. Error table of the density.  $k = 5$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	8.02E-04	-	1.10E-03	-	4.66E-04	-	5.06E-04	-
40	3.53E-05	4.507	5.80E-05	4.251	1.44E-05	5.015	1.78E-05	4.830
60	4.95E-06	4.842	9.03E-06	4.587	1.85E-06	5.056	2.44E-06	4.900
80	1.19E-06	4.961	2.29E-06	4.766	4.34E-07	5.050	5.78E-07	5.000
100	3.88E-07	5.018	7.75E-07	4.858	1.41E-07	5.043	1.86E-07	5.088
120	1.54E-07	5.059	3.04E-07	5.131	5.62E-08	5.037	7.32E-08	5.113
140	7.03E-08	5.098	1.39E-07	5.094	2.59E-08	5.034	3.36E-08	5.045
160	3.54E-08	5.137	6.83E-08	5.310	1.32E-08	5.035	1.68E-08	5.180

Table 4.3: Accuracy test 1. Error table of the density.  $k = 7$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	1.28E-04	-	2.59E-04	-	3.52E-05	-	5.74E-05	-
40	2.07E-06	5.945	5.88E-06	5.460	4.99E-07	6.142	1.16E-06	5.636
60	1.68E-07	6.202	5.79E-07	5.714	4.56E-08	5.902	1.28E-07	5.417
80	2.54E-08	6.562	9.97E-08	6.118	7.91E-09	6.088	2.60E-08	5.548
100	5.20E-09	7.107	2.21E-08	6.743	1.86E-09	6.481	6.98E-09	5.902
120	1.30E-09	7.597	5.78E-09	7.369	5.26E-10	6.937	2.18E-09	6.389
140	3.84E-10	7.923	1.72E-09	7.876	1.69E-10	7.361	7.50E-10	6.911
160	1.30E-10	8.093	5.72E-10	8.233	6.04E-11	7.704	2.79E-10	7.406

Table 4.4: Accuracy test 1. Error table of the density.  $k = 9$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	1.95E-05	-	3.23E-05	-	1.63E-05	-	1.60E-05	-
40	4.70E-08	8.695	8.18E-08	8.624	3.20E-08	8.992	3.13E-08	8.994
60	1.38E-09	8.702	2.79E-09	8.332	8.32E-10	8.998	8.23E-10	8.973
80	1.09E-10	8.839	2.32E-10	8.651	6.24E-11	9.002	6.22E-11	8.976
100	1.48E-11	8.930	3.17E-11	8.916	8.31E-12	9.035	8.87E-12	8.729
120	3.25E-12	8.309	4.98E-12	10.143	2.18E-12	7.337	2.60E-12	6.731

### 4.2.2 Accuracy test 2

Notice that any isentropic solution are completely determined by the initial entropy value  $S_0 = p\rho^{-\gamma}$  and the left- and right- Riemann invariants,  $J^\pm = u \pm \frac{2c}{\gamma-1}$ . When the specific heat ratio is  $\gamma = 3$  and all functions are smooth,  $J^\pm$  satisfy two disjoint Burgers' equations [25], so now the exact solution can be constructed from the exact solution of Burgers' equation. Let  $\Omega = [-1, 1]$  with periodic boundaries. Choose  $S_0 = 1$  and  $J^+ \equiv 2$ , and let  $J^-$  be the solution with the exact solution of the Burgers' equation with initial condition  $J^-|_{t=0} = \sin(\pi x)$ . The initial condition in primitive variables is then  $\rho = \frac{1 - \frac{1}{2}\sin(\pi x)}{\sqrt{3}}$ ,  $u = 1 + \frac{1}{2}\sin(\pi x)$ ,  $p = \rho^3$ . We solve the problem to  $T = 0.2$ . The time step is taken to be the same as the previous example.

The errors are shown in Tables 4.5 to 4.7. This is a genuinely nontrivial test since all variables have nontrivial variations. We can see from the error tables that the CH-RI version is as accurate as the CH-CON version in terms of the convergence rate.

Table 4.5: Accuracy test 2. Error table of the density.  $k = 5$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	2.99E-03	-	6.84E-03	-	9.13E-03	-	2.21E-02	-
40	2.98E-04	3.324	1.05E-03	2.701	8.91E-04	3.358	2.45E-03	3.172
60	5.18E-05	4.319	2.07E-04	4.006	2.87E-04	2.791	6.53E-04	3.265
80	1.27E-05	4.875	5.26E-05	4.768	1.00E-04	3.670	2.60E-04	3.201
100	4.00E-06	5.192	1.66E-05	5.168	2.90E-05	5.547	1.03E-04	4.158
120	1.50E-06	5.367	6.20E-06	5.395	1.28E-05	4.493	4.43E-05	4.616
140	6.49E-07	5.451	2.65E-06	5.516	6.33E-06	4.558	2.06E-05	4.971
160	3.12E-07	5.481	1.32E-06	5.212	3.38E-06	4.707	1.07E-05	4.871

Table 4.6: Accuracy test 2. Error table of the density.  $k = 7$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	2.50E-03	-	5.60E-03	-	5.24E-03	-	1.29E-02	-
40	1.53E-04	4.032	5.13E-04	3.449	6.72E-04	2.963	1.90E-03	2.768
60	1.82E-05	5.242	7.43E-05	4.766	1.14E-04	4.376	3.94E-04	3.878
80	3.24E-06	6.003	1.42E-05	5.764	1.88E-05	6.265	8.01E-05	5.541
100	7.56E-07	6.523	3.37E-06	6.425	5.27E-06	5.698	2.22E-05	5.746
120	2.15E-07	6.890	9.79E-07	6.789	1.79E-06	5.930	7.87E-06	5.692
140	7.17E-08	7.137	3.42E-07	6.823	6.96E-07	6.118	3.03E-06	6.195
160	2.70E-08	7.303	1.28E-07	7.351	3.02E-07	6.257	1.26E-06	6.544

### 4.2.3 The Sod shock tube problem

The initial condition is  $(\rho, u, p) = (1, 0, 1)$  for  $x < 0$  and  $(\rho, u, p) = (0.125, 0, 0.1)$  for  $x > 0$ . The computation domain is  $[-5, 5]$ . We solve the problem to  $T = 2$ . The results are shown in Figure 4.1. For this example, both CH-RI and CH-CON have similarly good performance.

Table 4.7: Accuracy test 2. Error table of the density.  $k = 9$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	2.00E-03	-	4.53E-03	-	2.67E-03	-	7.13E-03	-
40	1.11E-04	4.167	3.65E-04	3.636	3.12E-04	3.093	9.76E-04	2.869
60	1.11E-05	5.685	4.49E-05	5.164	2.99E-05	5.789	1.06E-04	5.484
80	1.60E-06	6.721	7.21E-06	6.358	4.28E-06	6.754	1.63E-05	6.506
100	3.07E-07	7.411	1.43E-06	7.244	7.17E-07	8.004	3.01E-06	7.552
120	7.18E-08	7.969	3.38E-07	7.916	1.38E-07	9.036	6.33E-07	8.563
140	1.97E-08	8.390	9.35E-08	8.343	3.31E-08	9.271	1.61E-07	8.873
160	6.17E-09	8.697	3.17E-08	8.089	1.03E-08	8.724	4.93E-08	8.875

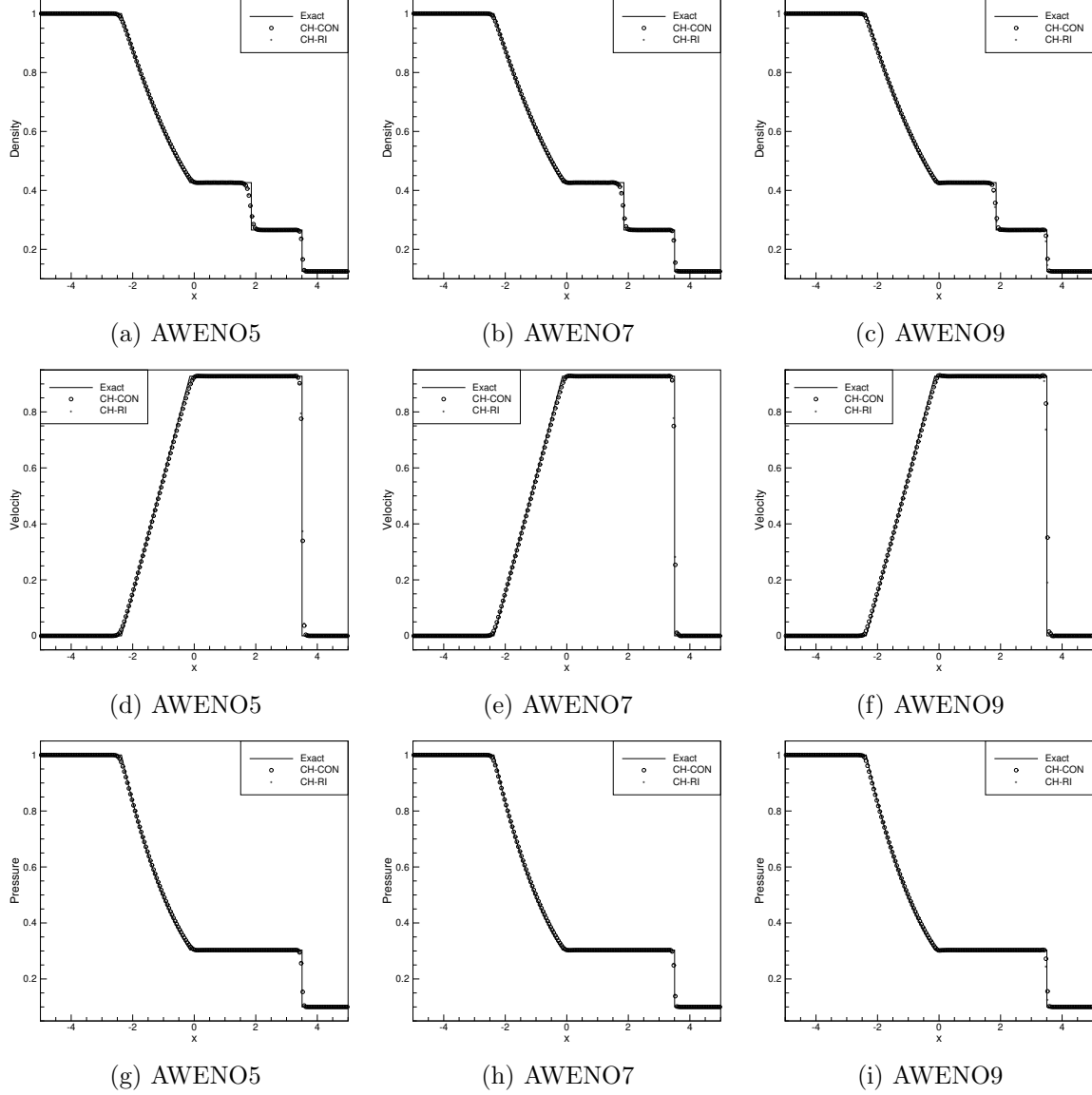


Figure 4.1: The Sod shock tube problem at  $T = 2$  with  $N = 200$  grids.

#### 4.2.4 The Lax shock tube problem

The initial condition is  $(\rho, u, p) = (0.445, 0.698, 3.528)$  for  $x < 0$  and  $(\rho, u, p) = (0.5, 0, 0.571)$  for  $x > 0$ . The computation domain is  $[-5, 5]$ . We solve the problem to  $T = 1.3$ . The results are shown in Figure 4.2. The CH-RI and CH-CON agree with each other well.

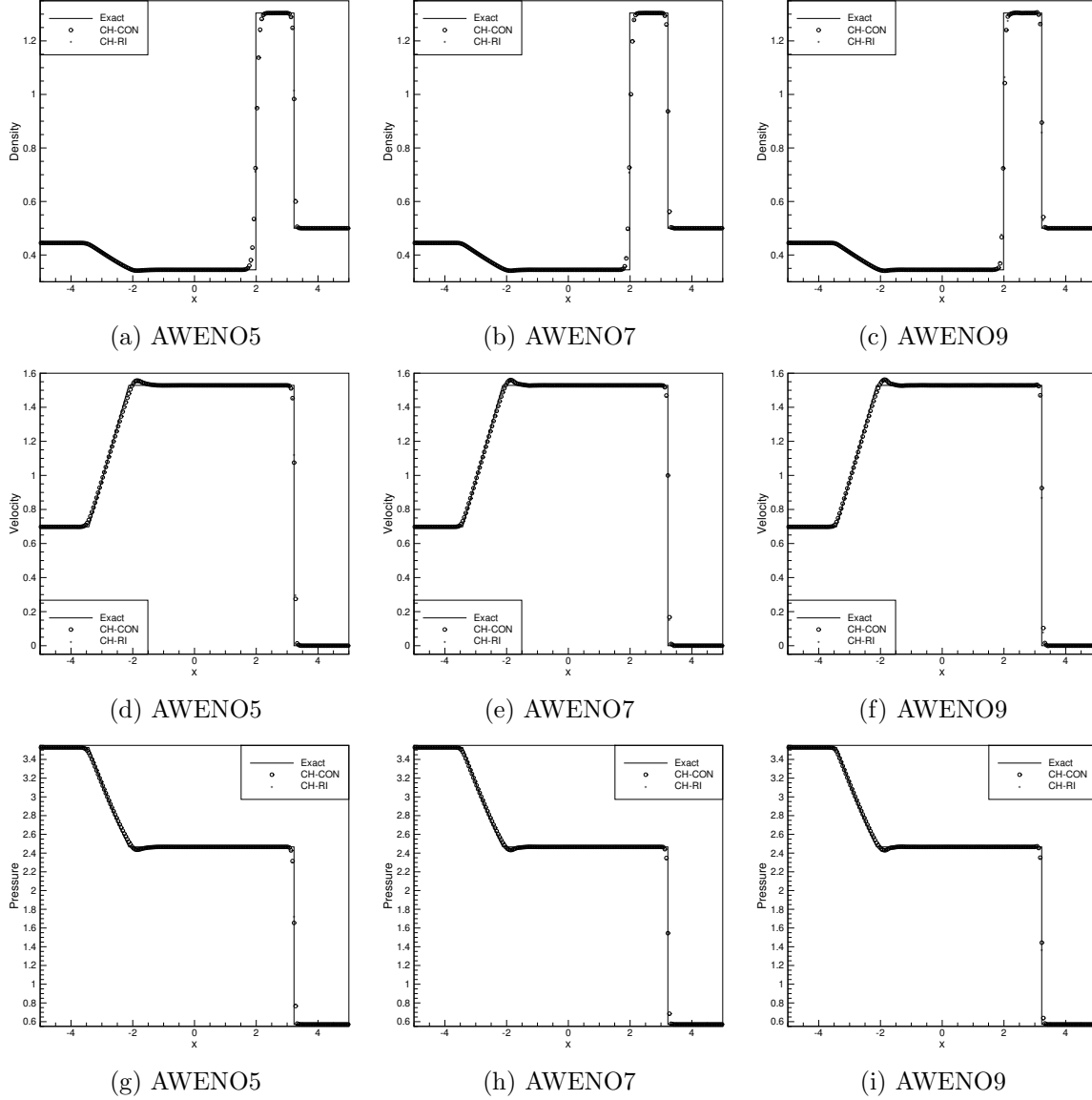


Figure 4.2: The Lax shock tube problem at  $T = 1.3$  with  $N = 200$  grids.

#### 4.2.5 The LeBlanc shock tube problem

The initial condition is  $(\rho, u, p) = (2, 0, 10^9)$  for  $x < 0$  and  $(\rho, u, p) = (10^{-3}, 0, 1)$  for  $x > 0$ . The computation domain is  $[-5, 5]$ . We solve the problem to  $T = 5 \times 10^{-5}$ . The results are shown in Figure 4.3. The extreme pressure ratio makes the problem very stiff and challenges the PP property of the scheme. With our PP limiters, both CH-RI and CH-CON perform quite well.



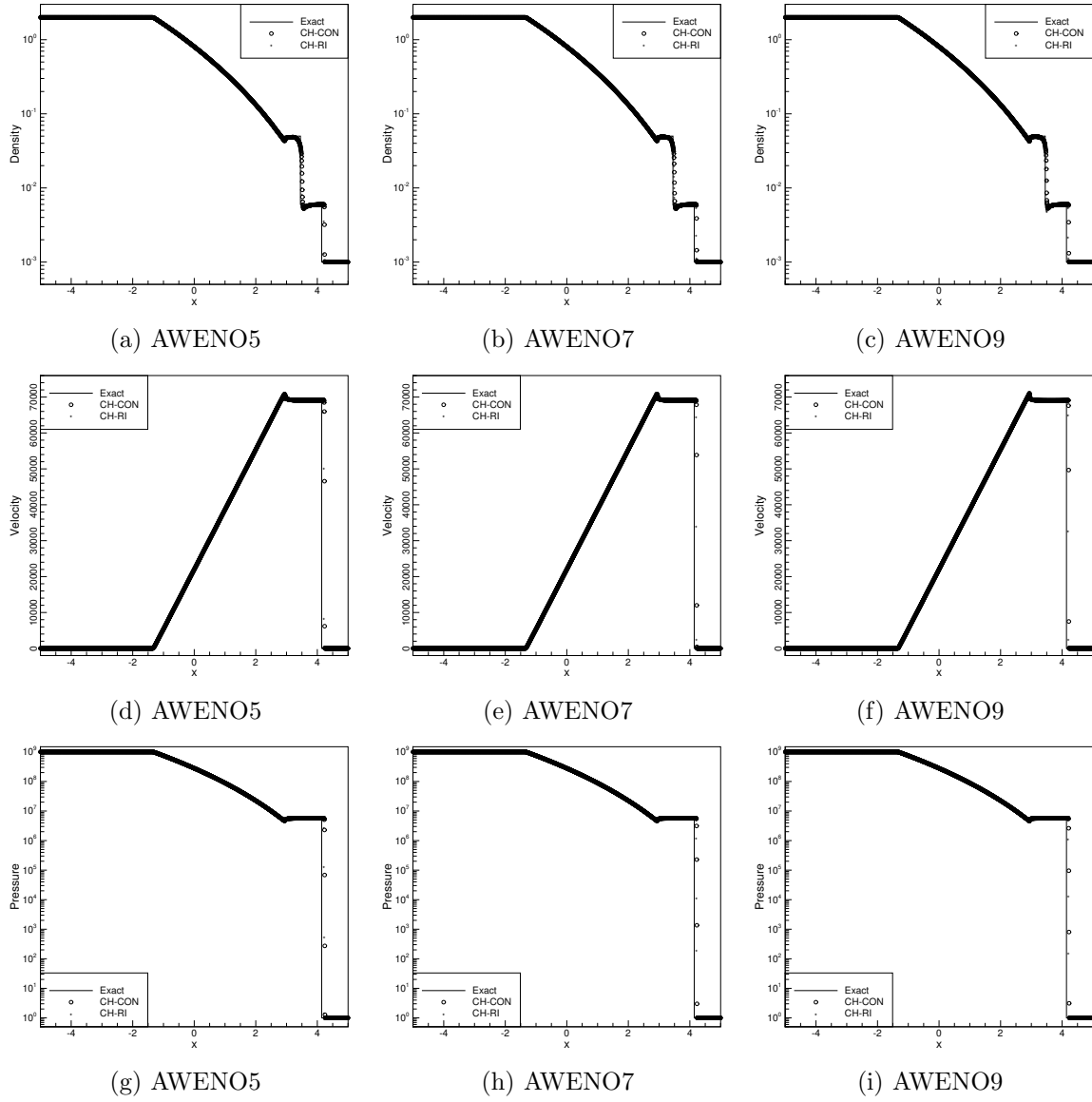


Figure 4.3: The LeBlanc shock tube problem at  $T = 5 \times 10^{-5}$  with  $N = 2000$  grids.

#### 4.2.6 The critical double rarefaction problem

The initial condition is  $(\rho, u, p) = (7, -1, 0.2)$  for  $x < 0$  and  $(\rho, u, p) = (7, 1, 0.2)$  for  $x > 0$ . The computation domain is  $[-5, 5]$ . We solve the problem to  $T = 3.3$ . The results are shown in Figure 4.4. The point  $x = 0$  is the only vacuum point in the exact solution for  $t > 0$ , so it causes difficulty for the scheme. Because the PP interpolation limiters for the two versions are different, there seems to be some difference near the vacuum point in the numerical solution.

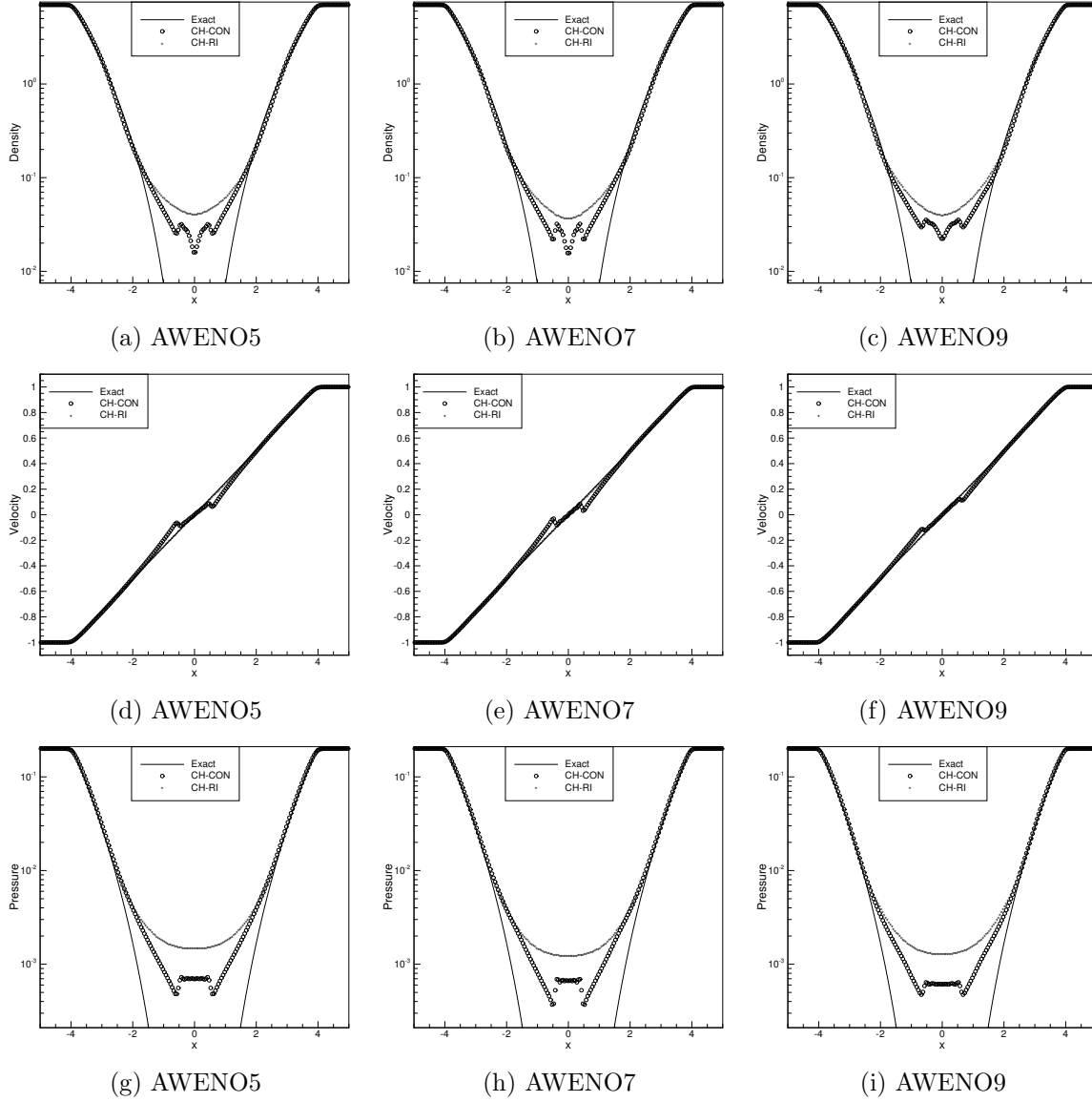


Figure 4.4: The critical double rarefaction problem at  $T = 3.3$  with  $N = 200$  grids.

#### 4.2.7 The blast wave interaction problem

The initial condition is  $\rho = 1$  and  $u = 0$  with  $p = 1000$  for  $x \in (0, 0.1)$ ,  $p = 0.01$  for  $x \in (0.1, 0.9)$  and  $p = 100$  for  $x \in (0.9, 1)$ . The computation domain is  $[0, 1]$ , and two reflective walls are set at both ends. We solve the problem to  $T = 0.038$ . This model problem describes

the interaction of two blast waves initialized near the boundaries. The results are shown in Figure 4.5. Because there is no exact solutions for this problem, we use the solution with  $N = 40000$  grid points as the reference solution. Both versions perform well. The CH-RI seems less dissipative at the density peak than the CH-CON.

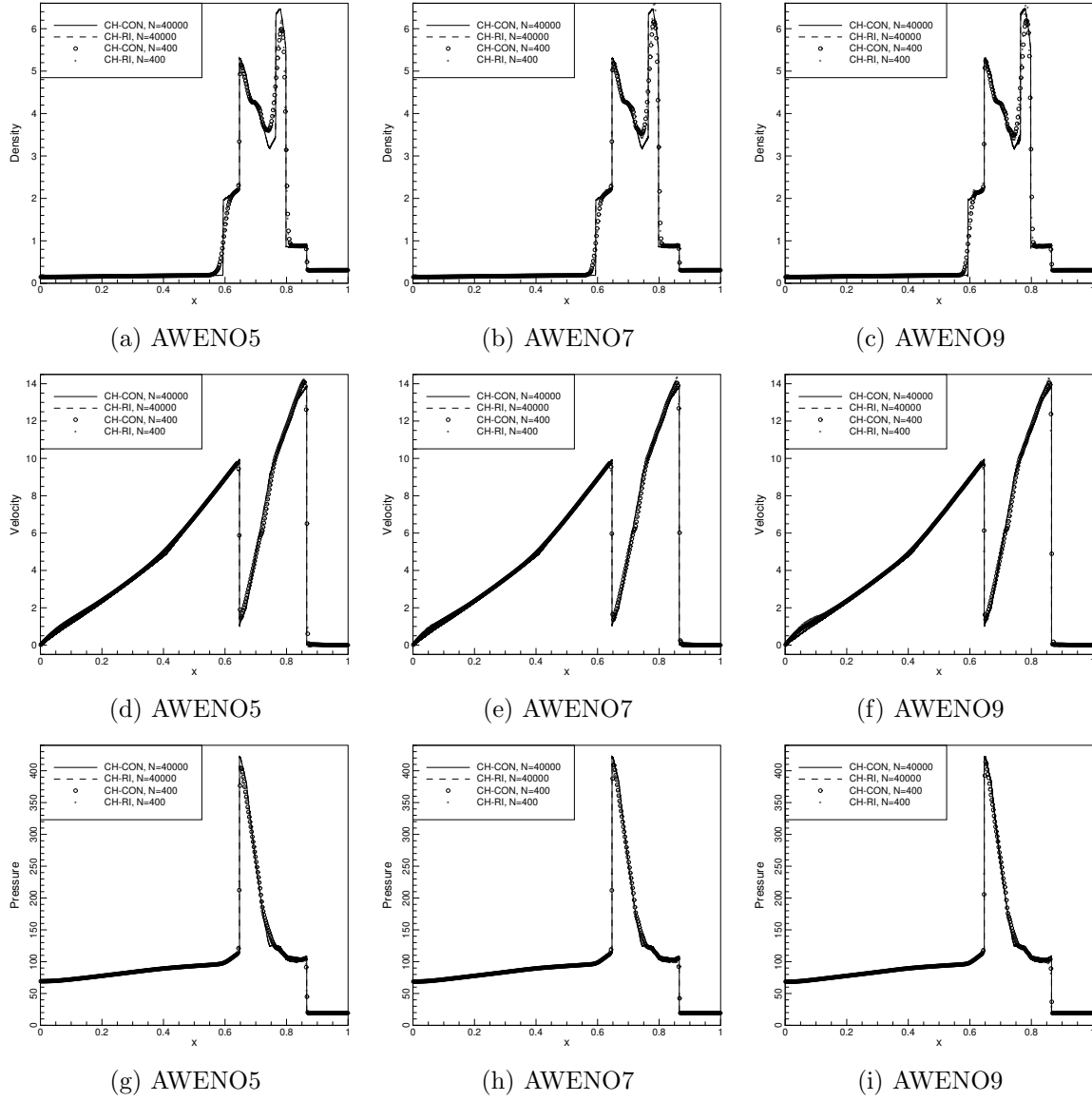


Figure 4.5: The blast wave interaction problem at  $T = 0.038$ .

#### 4.2.8 The Sedov blast wave problem

For the initial condition, the background state is  $(\rho, u, p) = (1, 0, 10^{-12})$ , and a singular delta total energy source of strength  $E_0$  is added at the origin. The computation domain is  $[-2, 2]$ . We solve the problem to  $T = 0.001$ . This problem is a simple model of point explosion. It is an extreme test for scheme stability and has been used to test PP limiters of DG schemes [29]. We choose  $E_0 = 3.2 \times 10^6$ . For initialization, we use odd number of cells and set the

additional total energy at the central node to be  $\frac{E_0}{h}$ . The results are shown in Figure 4.6. The  $N = 4001$  solution is used for reference. The CH-RI seems less dissipative near the central point in the sense that it produces a smaller minimum density. Besides this region, both methods agree well.

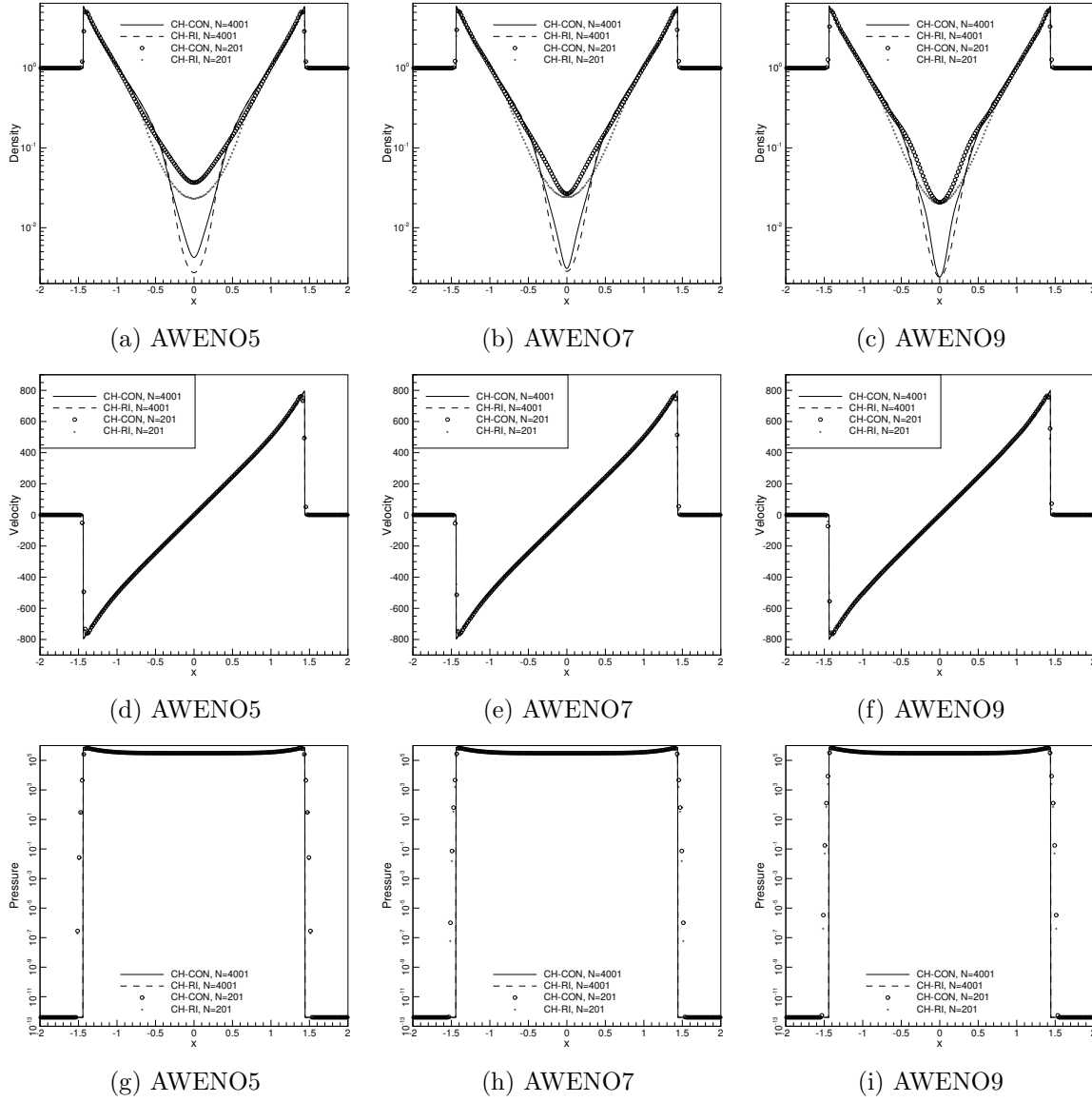


Figure 4.6: The Sedov blast wave problem at  $T = 0.001$ .

#### 4.2.9 The shock-density wave interaction problem

The initial condition is  $(\rho, u, p) = (\frac{27}{7}, \frac{4}{9}\sqrt{35}, \frac{31}{3})$  for  $x < -4$  and  $(\rho, u, p) = (1 + \epsilon \sin(5x), 0, 1)$  for  $x > -4$ . The computation domain is  $[-5, 5]$ . We solve the problem to  $T = 1.8$ . It describes the interaction between a right-moving Mach 3 shock and a stationary density wave [21]. We choose the parameter  $\epsilon = 0.2$ . The results are shown in Figure 4.7. We use  $N = 40000$  as references. The CH-RI and CH-CON versions agree well with each other and capture the high

frequency part after the shock well.

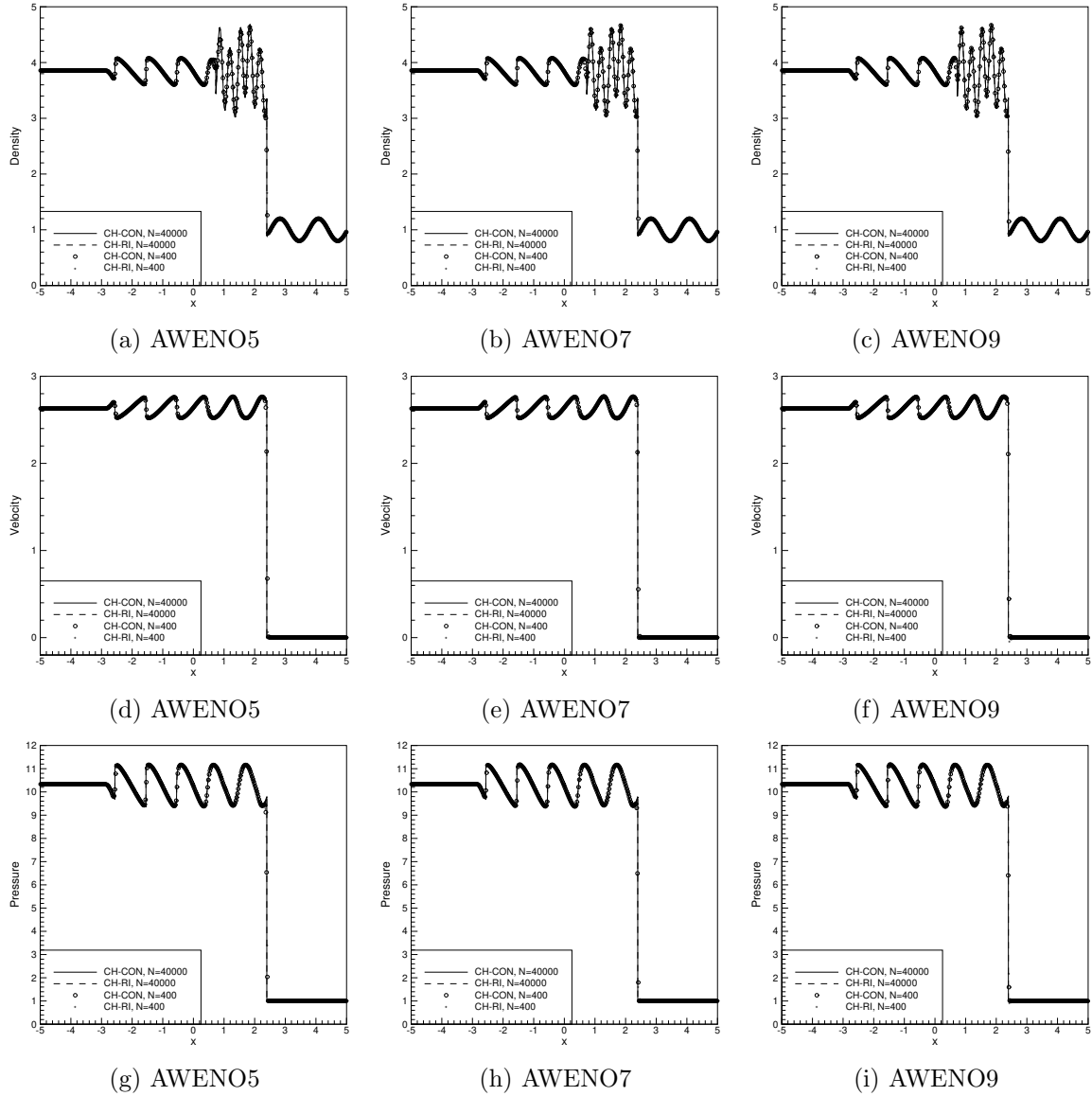


Figure 4.7: The shock-density wave interaction problem at  $T = 1.8$ .

### 4.3 2D tests

#### 4.3.1 Accuracy test

We consider the rotated simple density-transport problem, whose exact solution is  $(\rho, u, v, p) = (1 + 0.2 \sin(\pi(x + y)), 1, 1, 1)$ . The computational domain is  $[0, 2] \times [0, 2]$  with periodic boundaries. We solve up to  $T = 2$ . We use  $(N_{x,0}, N_{y,0}) = (20, 20)$  as the coarsest mesh and use  $h_0 = \frac{2}{N_0}$  to denote the coarsest mesh size. Using SSP-RK(3,3), the time steps in the following tests with uniform mesh are taken to be  $\Delta t = \text{CFL} \frac{\text{CFL}}{\frac{\alpha}{\Delta x} + \frac{\beta}{\Delta y}} \left( \frac{h}{h_0} \right)^{\frac{k}{3}-1}$ , which makes the time

discretization error  $k$ th-order while maintaining stability. Here,  $\alpha$  and  $\beta$  are the maximum wave speeds in the  $x$ - and  $y$ -direction respectively. The results are shown in Tables 4.8 to 4.10.

Table 4.8: Accuracy test. Error table of the density.  $k = 5$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	2.08E-03	-	1.86E-03	-	1.20E-03	-	8.48E-04	-
40	9.55E-05	4.446	1.00E-04	4.216	3.88E-05	4.956	3.27E-05	4.696
60	1.37E-05	4.795	1.66E-05	4.442	5.03E-06	5.038	4.46E-06	4.917
80	3.31E-06	4.931	4.22E-06	4.750	1.18E-06	5.039	1.07E-06	4.955
100	1.08E-06	4.999	1.46E-06	4.754	3.84E-07	5.034	3.48E-07	5.038
120	4.32E-07	5.045	5.87E-07	5.005	1.53E-07	5.030	1.39E-07	5.043
140	1.97E-07	5.087	2.71E-07	5.013	7.07E-08	5.028	6.41E-08	5.008
160	9.95E-08	5.128	1.35E-07	5.235	3.61E-08	5.030	3.24E-08	5.123

Table 4.9: Accuracy test. Error table of the density.  $k = 7$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	3.01E-04	-	4.26E-04	-	7.07E-05	-	7.82E-05	-
40	5.29E-06	5.833	1.04E-05	5.361	1.21E-06	5.874	1.89E-06	5.368
60	4.36E-07	6.153	1.04E-06	5.667	1.15E-07	5.797	2.21E-07	5.292
80	6.72E-08	6.500	1.82E-07	6.068	2.03E-08	6.017	4.59E-08	5.472
100	1.40E-08	7.020	4.10E-08	6.671	4.87E-09	6.405	1.25E-08	5.838
120	3.57E-09	7.506	1.09E-08	7.279	1.40E-09	6.852	3.94E-09	6.319
140	1.06E-09	7.852	3.28E-09	7.782	4.55E-10	7.280	1.37E-09	6.846
160	3.63E-10	8.050	1.10E-09	8.161	1.64E-10	7.634	5.16E-10	7.322

Table 4.10: Accuracy test. Error table of the density.  $k = 9$ .

$N$	CH-RI				CH-CON			
	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order	$\ell_h^2$ error	order	$\ell_h^\infty$ error	order
20	2.72E-05	-	3.39E-05	-	6.30E-06	-	4.11E-06	-
40	9.20E-08	8.205	1.00E-07	8.402	1.26E-08	8.968	8.93E-09	8.847
60	3.00E-09	8.440	3.92E-09	7.996	3.29E-10	8.990	2.39E-10	8.929
80	2.45E-10	8.715	3.41E-10	8.484	2.48E-11	8.977	1.83E-11	8.937
100	3.40E-11	8.851	4.75E-11	8.836	4.20E-12	7.961	3.61E-12	7.271

### 4.3.2 Double Mach reflection

We consider the double Mach reflection (DMR) problem [27]. The computational domain is  $[0, 4] \times [0, 1]$ . Let  $x_0 = \frac{1}{6}$  denote the start of the reflective wall and  $\theta = \frac{\pi}{6}$  denote the angle between the incoming flow and the reflective wall. The reflective wall lies at the bottom boundary starting from  $x_0$ . Initially, a Mach 10 shock is positioned at  $(x, y) = (x_0, 0)$ , and it is perpendicular to the incoming flow direction. The post-shock (to the left of the shock)

primitive variables are  $(\rho, u, v, p) = (8, 8.5 \cos(-\theta), 8.5 \sin(-\theta), 116.5)$  and the pre-shock (to the right of the shock) primitive variables are  $(\rho, u, v, p) = (1.4, 0, 0, 1)$ . The velocity of the shock front is  $(u_s, v_s) = (10 \cos(-\theta), 10 \sin(-\theta))$ . Except for the reflective wall, all other boundaries are posed with the exact motion of the shock. We compute to  $T = 0.2$  and plot the density profile using 30 equally spaced contours from 1.5 to 22.7. The results are shown in Figures 4.8 and 4.9. We can see that both methods agree well.

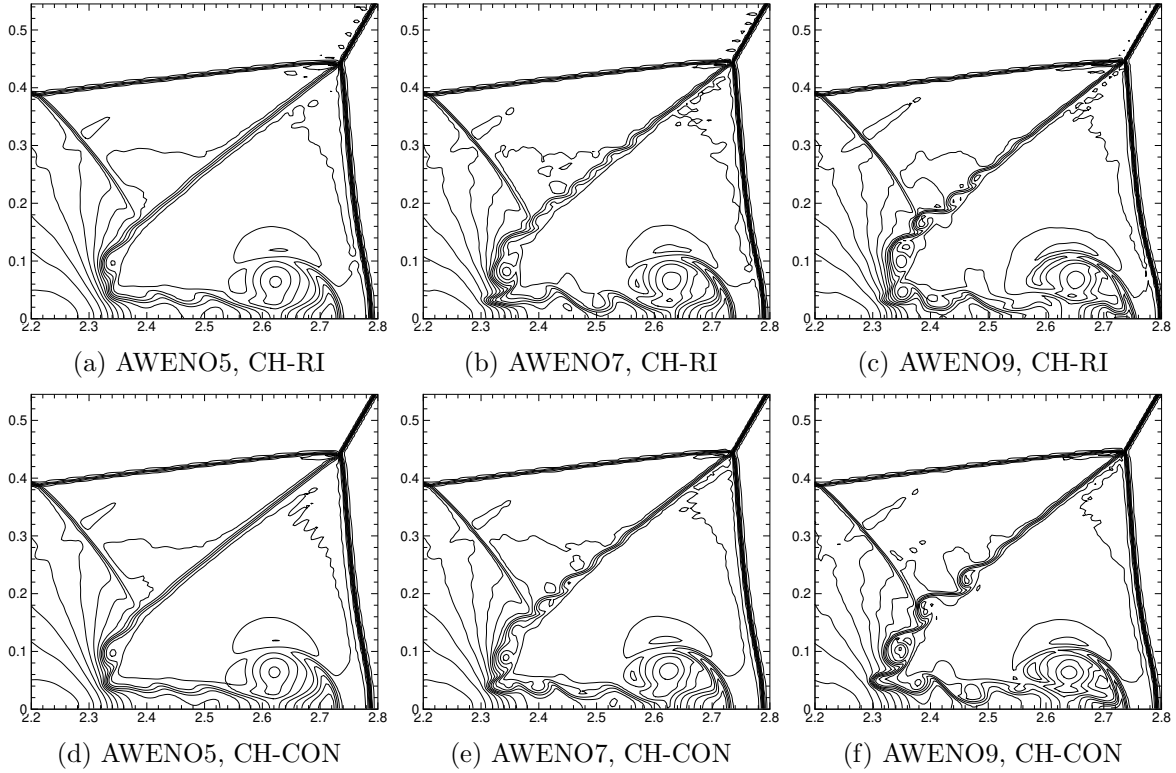


Figure 4.8: On the coarse mesh  $(N_x, N_y) = (960, 240)$ .

### 4.3.3 Rayleigh–Taylor instability

We consider the Rayleigh–Taylor instability (RTI) problem [19]. The computational domain is  $[0, \frac{1}{4}] \times [0, 1]$ . Initially, the interface is located at  $y = \frac{1}{2}$ . The heavy fluid with density  $\rho = 2$  is below the interface, and the light fluid with density  $\rho = 1$  is above the interface, with acceleration in the positive  $y$ -direction, i.e.  $(g_x, g_y) = (0, 1)$ . The pressure  $p$  is continuous across the interface. A small perturbation is given to the  $y$ -direction fluid speed: for  $0 \leq y < \frac{1}{2}$ ,  $(\rho, u, v, p) = (2, 0, -0.025c \cdot \cos(8\pi x), 2y + 1)$ , and for  $\frac{1}{2} \leq y \leq 1$ ,  $(\rho, u, v, p) = (1, 0, -0.025c \cdot \cos(8\pi x), y + \frac{3}{2})$ , where  $c = \sqrt{\frac{\gamma p}{\rho}}$  is the sonic speed. The specific heat capacity ratio is set to  $\gamma = \frac{5}{3}$ . Reflective wall boundary conditions are imposed for both left and right boundaries. For the top boundary, the flow values are set to be  $(\rho, u, v, p) = (1, 0, 0, 2.5)$ . For the bottom boundary, the flow values are set to be  $(\rho, u, v, p) = (2, 0, 0, 1)$ . The source terms  $\rho g_y$  and  $\rho v g_y$  are added to the right hand side of the third and fourth equations. The final time is  $T = 1.95$ . The results are shown in Figure 4.10 using 15 equally spaced contours from 0.9 to 2.2.

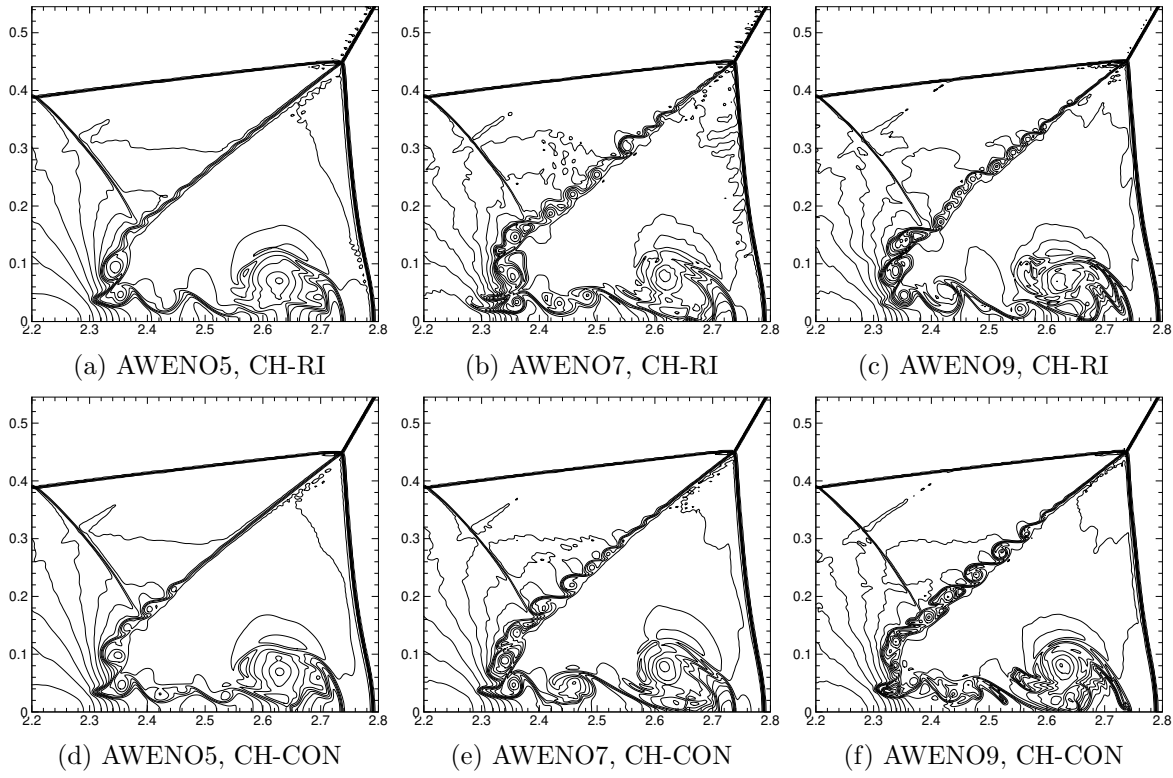


Figure 4.9: On the fine mesh  $(Nx, Ny) = (1920, 480)$ .

#### 4.3.4 Kelvin–Helmholtz instability

We consider the Kelvin–Helmholtz instability (KHI) problem [18]. The computational domain is  $[-0.5, 0.5] \times [-0.5, 0.5]$  with periodic boundary conditions in all directions. For  $|y| \leq 0.25$ , the initial condition is  $(\rho, u, v, p) = (2, -0.5, 0.01 \sin(2\pi x), 2.5)$ , and for  $|y| > 0.25$ , the initial condition is  $(\rho, u, v, p) = (1, 0.5, 0.01 \sin(2\pi x), 2.5)$ . The final simulation time is  $T = 1.0$ . The results are shown in Figures 4.11 and 4.12 using 20 equally spaced contours from 0.7 to 2.2. The CH-RI is more diffusive than the CH-CON in this example.

## 5 Conclusion

In this study, we have developed a finite difference A-WENO scheme with a Riemann invariant-based LCD procedure. We use the Riemann invariants as the transform variables, perform the LCD procedure on these variables and conduct WENO interpolation on the associated characteristic field. This change in the variable improves the sparsity of the eigenmatrix and thus greatly reduces the part of per-node computational cost that scales linearly with the stencil width or the order of accuracy increases. In the meanwhile, the scheme still maintains the convergence order of smooth solutions, satisfies the E-property and is free of spurious oscillations near discontinuities. Moreover, a combination of interpolation and flux PP limiters can make it as robust as those classical WENO schemes in the vicinity of low density or pressure. The scheme is tested on a variety of 1D and 2D problems to demonstrate its accuracy, efficiency and robustness. Our experiment shows that it can reduce 23% of the total time or 88% of the LCD time in comparison to the original characteristic-wise version at 9th-order. With our cost



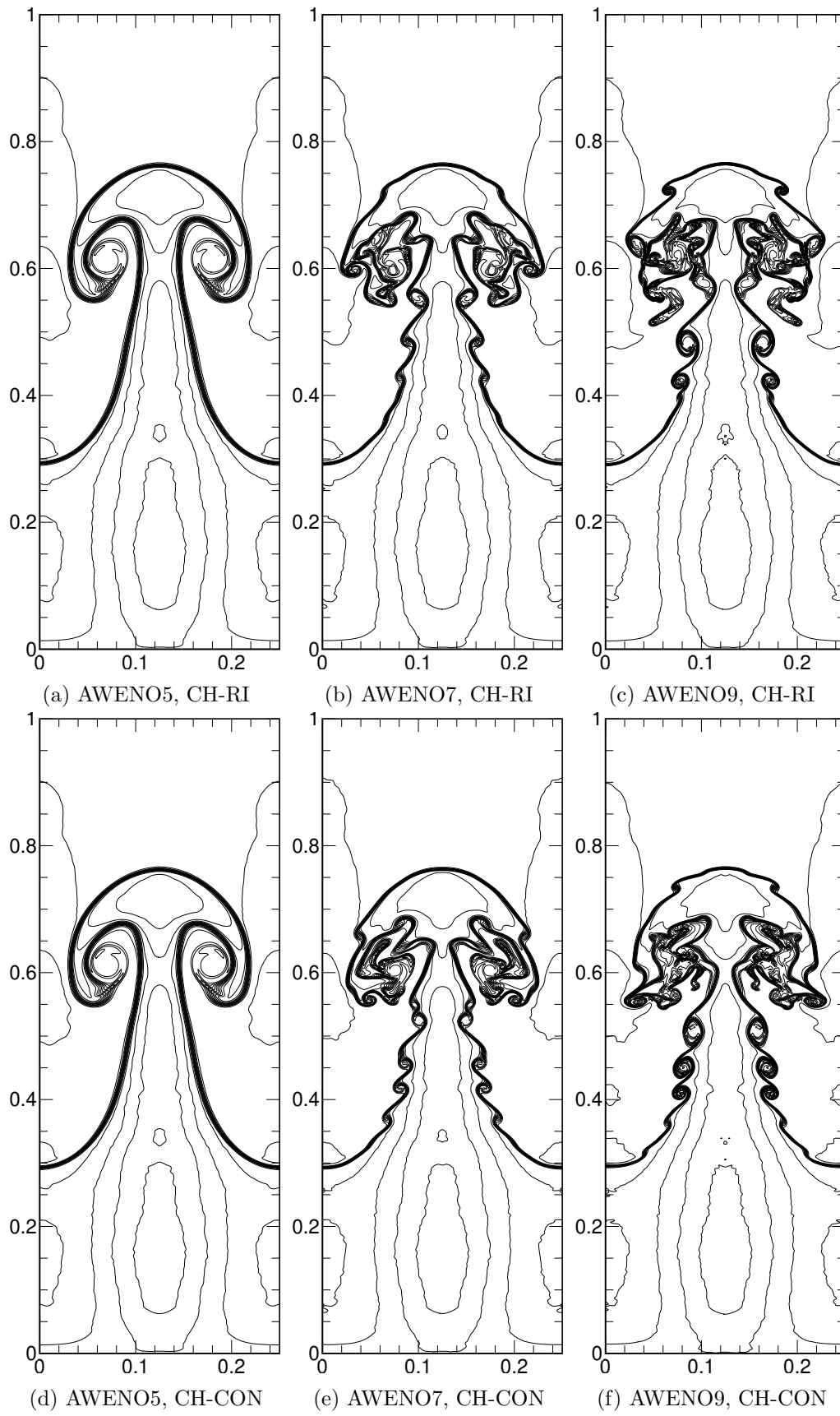


Figure 4.10: On the mesh  $(Nx, Ny) = (240, 960)$ .

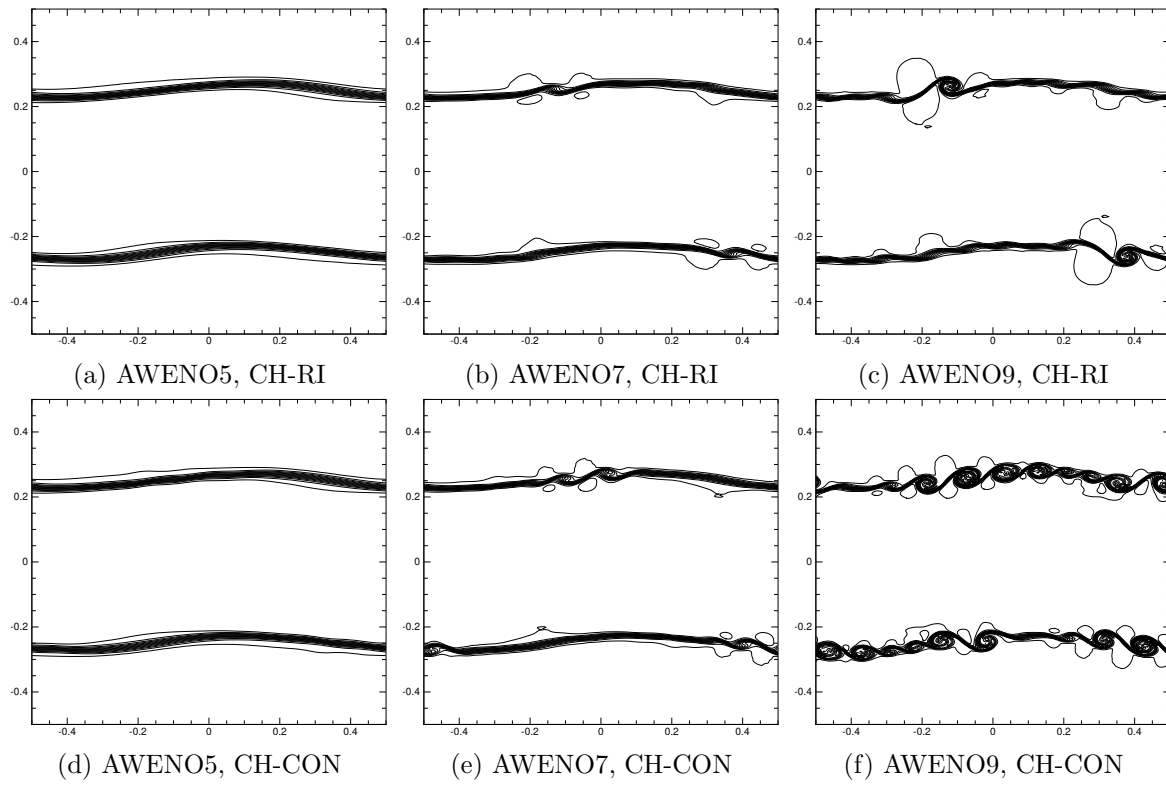


Figure 4.11: On the coarse mesh  $(Nx, Ny) = (256, 256)$ .

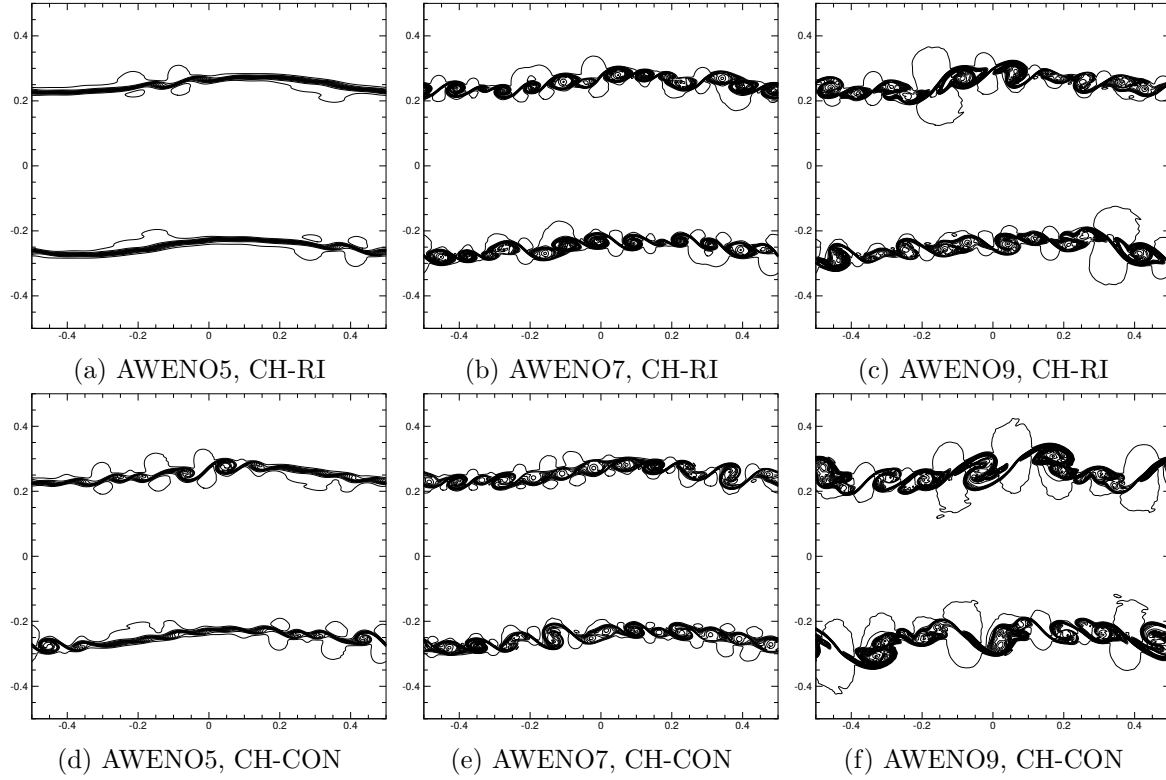


Figure 4.12: On the fine mesh  $(Nx, Ny) = (512, 512)$ .

analysis, greater acceleration can be expected for higher-order and 3D cases. This work merely focuses on the compressible Euler equations with the  $\gamma$ -law EOS. Generalizations to other EOS will probably change the Riemann invariants, but the sparsity pattern of the eigenmatrices will remain the same as long as the physical entropy exists, so such acceleration is still possible, and similar efficiency can be expected. Besides the compressible Euler equations, using Riemann invariants to reduce the LCD cost is also possible for other hyperbolic systems, such as the multi-component Euler equations, the magnetohydrodynamics (MHD) equations and the nonlinear elasticity equations.

## Appendix A Supplemental material for the compressible Euler equations

Following the notation in Definition 2.1, the 2D compressible Euler equations (2.6) with the  $\gamma$ -law EOS (2.7) is a hyperbolic system with the following eigenstructure in the  $x$ -direction.

$$\mathbf{\Lambda} = \text{diag} (u - c, u, u, u + c), \quad (\text{A.1})$$

$$\mathbf{r}_1 = \frac{1}{2} (0, 1, 0, u)^T - \frac{1}{2c} (1, u, v, H)^T, \quad (\text{A.2a})$$

$$\mathbf{r}_2 = -\frac{1}{c} \left( 1, u, v, \frac{|\vec{u}|^2}{2} \right)^T, \quad (\text{A.2b})$$

$$\mathbf{r}_3 = (0, 0, 1, v)^T, \quad (\text{A.2c})$$

$$\mathbf{r}_4 = \frac{1}{2} (0, 1, 0, u)^T + \frac{1}{2c} (1, u, v, H)^T. \quad (\text{A.2d})$$

$$\mathbf{l}_1 = (-u, 1, 0, 0) - \frac{\gamma - 1}{c} \left( \frac{|\vec{u}|^2}{2}, -u, -v, 1 \right), \quad (\text{A.3a})$$

$$\mathbf{l}_2 = \frac{\gamma - 1}{c} \left( \frac{|\vec{u}|^2}{2}, -u, -v, 1 \right) - (c, 0, 0, 0), \quad (\text{A.3b})$$

$$\mathbf{l}_3 = (-v, 0, 1, 0), \quad (\text{A.3c})$$

$$\mathbf{l}_4 = (-u, 1, 0, 0) + \frac{\gamma - 1}{c} \left( \frac{|\vec{u}|^2}{2}, -u, -v, 1 \right). \quad (\text{A.3d})$$

Using the transform variables  $\mathbf{v}(\mathbf{u}) = (u - \frac{2}{\gamma-1}c, S^{\frac{1}{2\gamma}}, v, u + \frac{2}{\gamma-1}c)$ , the transformed 2D compressible Euler equations (2.6) with the  $\gamma$ -law EOS (2.7) have the following normalized eigenvectors in the  $x$ -direction:

$$\tilde{\mathbf{r}}_1 = (1, 0, 0, 0)^T, \quad (\text{A.4a})$$

$$\tilde{\mathbf{r}}_2 = \left( -\frac{2\sqrt{\gamma}p^{\frac{\gamma-1}{2\gamma}}}{\gamma-1}, 1, 0, \frac{2\sqrt{\gamma}p^{\frac{\gamma-1}{2\gamma}}}{\gamma-1} \right)^T, \quad (\text{A.4b})$$

$$\tilde{\mathbf{r}}_3 = (0, 0, 1, 0)^T, \quad (\text{A.4c})$$

$$\tilde{\mathbf{r}}_4 = (0, 0, 0, 1)^T. \quad (\text{A.4d})$$

and

$$\tilde{\mathbf{l}}_1 = \left( 1, \frac{2\sqrt{\gamma}p^{\frac{\gamma-1}{2\gamma}}}{\gamma-1}, 0, 0 \right), \quad (\text{A.5a})$$

$$\tilde{\mathbf{l}}_2 = (0, 1, 0, 0), \quad (\text{A.5b})$$

$$\tilde{\mathbf{l}}_3 = (0, 0, 1, 0), \quad (\text{A.5c})$$

$$\tilde{\mathbf{l}}_4 = \left( 0, -\frac{2\sqrt{\gamma}p^{\frac{\gamma-1}{2\gamma}}}{\gamma-1}, 0, 1 \right). \quad (\text{A.5d})$$

**Proposition A.1.** *The compressible Euler equations (2.6) with the  $\gamma$ -law EOS (2.7) is not endowed with any coordinate system of Riemann invariants.*

*Proof.* Using the primitive variable  $\mathbf{v} = (\rho, u, v, p)$  as the transform variable, the normalized left eigenvectors are

$$\begin{aligned} \tilde{\mathbf{l}}_1 &= \left( 0, 1, 0, -\frac{1}{\rho c} \right), \\ \tilde{\mathbf{l}}_2 &= \left( -\frac{\gamma}{\rho}, 0, 0, \frac{\gamma}{\rho c^2} \right), \\ \tilde{\mathbf{l}}_3 &= (0, 0, 1, 0), \\ \tilde{\mathbf{l}}_4 &= \left( 0, 1, 0, \frac{1}{\rho c} \right). \end{aligned}$$

Suppose there exists a coordinate system of Riemann invariants, then there has to be 2, 3, 4-Riemann invariant  $w(\mathbf{v})$  that satisfies  $\frac{\partial w}{\partial \mathbf{v}} \cdot \tilde{\mathbf{r}}_i = 0$  for  $i = 2, 3, 4$ . This yields that there is a non-zero function  $\varphi(\mathbf{v})$  such that  $\frac{\partial w}{\partial \mathbf{v}} = \varphi(\mathbf{v})\tilde{\mathbf{l}}_1$ . By the expression we deduce that  $w(\mathbf{v})$  is independent of the density  $\rho$  and so is  $\varphi(\mathbf{v})$  and  $\frac{\partial w}{\partial \mathbf{v}} = \varphi(\mathbf{v})\tilde{\mathbf{l}}_1$ . This is a contradiction since the last component of  $\varphi(\mathbf{v})\tilde{\mathbf{l}}_1$  will depend on  $\rho$  in this case.  $\square$

## Appendix B WENO-JS interpolation

The  $k = (2r - 1)$ th-order WENO-JS interpolation procedure for  $u_{j+\frac{1}{2}}^-$  is

$$u_{j+\frac{1}{2}}^- = \sum_{s=0}^{r-1} \omega_s u_{j+\frac{1}{2}}^{-,(s)}, \quad (\text{B.1})$$

where

$$\omega_s = \frac{\alpha_s}{\sum_{\tau=0}^{r-1} \alpha_\tau}, \quad (\text{B.2})$$

and

$$\alpha_s = \frac{d_s}{(\epsilon + \beta_s)^2}. \quad (\text{B.3})$$

Here, the small constant  $\epsilon$  is usually taken to be  $10^{-6}$ . Other parameters are given below.

For  $k = 3$  ( $r = 2$ ):

$$d_0 = \frac{3}{4}, \quad u_{j+\frac{1}{2}}^{-(0)} = \frac{1}{2}u_j + \frac{1}{2}u_{j+1}, \quad (\text{B.4a})$$

$$d_1 = \frac{1}{4}, \quad u_{j+\frac{1}{2}}^{-(1)} = -\frac{1}{2}u_{j-1} + \frac{3}{2}u_j. \quad (\text{B.4b})$$

$$\beta_0 = (u_j - u_{j+1})^2, \quad (\text{B.5a})$$

$$\beta_1 = (u_{j-1} - u_j)^2. \quad (\text{B.5b})$$

For  $k = 5$  ( $r = 3$ ):

$$d_0 = \frac{5}{16}, \quad u_{j+\frac{1}{2}}^{-(0)} = \frac{3}{8}u_j + \frac{3}{4}u_{j+1} - \frac{1}{8}u_{j+2}, \quad (\text{B.6a})$$

$$d_1 = \frac{5}{8}, \quad u_{j+\frac{1}{2}}^{-(1)} = -\frac{1}{8}u_{j-1} + \frac{3}{4}u_j + \frac{3}{8}u_{j+1}, \quad (\text{B.6b})$$

$$d_2 = \frac{1}{16}, \quad u_{j+\frac{1}{2}}^{-(2)} = \frac{3}{8}u_{j-2} - \frac{5}{4}u_{j-1} + \frac{15}{8}u_j. \quad (\text{B.6c})$$

$$\beta_0 = \frac{13}{12}(u_j - 2u_{j+1} + u_{j+2})^2 + \frac{1}{4}(3u_j - 4u_{j+1} + u_{j+2})^2, \quad (\text{B.7a})$$

$$\beta_1 = \frac{13}{12}(u_{j-1} - 2u_j + u_{j+1})^2 + \frac{1}{4}(u_{j-1} - u_{j+2})^2, \quad (\text{B.7b})$$

$$\beta_2 = \frac{13}{12}(u_{j-2} - 2u_{j-1} + u_j)^2 + \frac{1}{4}(u_{j-2} - 4u_{j-1} + 3u_j)^2. \quad (\text{B.7c})$$

For  $k = 7$  ( $r = 4$ ):

$$d_0 = \frac{7}{64}, \quad u_{j+\frac{1}{2}}^{-(0)} = \frac{5}{16}u_j + \frac{15}{16}u_{j+1} - \frac{5}{16}u_{j+2} + \frac{1}{16}u_{j+3}, \quad (\text{B.8a})$$

$$d_1 = \frac{35}{64}, \quad u_{j+\frac{1}{2}}^{-(1)} = -\frac{1}{16}u_{j-1} + \frac{9}{16}u_j + \frac{9}{16}u_{j+1} - \frac{1}{16}u_{j+2}, \quad (\text{B.8b})$$

$$d_2 = \frac{21}{64}, \quad u_{j+\frac{1}{2}}^{-(2)} = \frac{1}{16}u_{j-2} - \frac{5}{16}u_{j-1} + \frac{15}{16}u_j + \frac{5}{16}u_{j+1}, \quad (\text{B.8c})$$

$$d_3 = \frac{1}{64}, \quad u_{j+\frac{1}{2}}^{-(3)} = -\frac{5}{16}u_{j-3} + \frac{21}{16}u_{j-2} - \frac{35}{16}u_{j-1} + \frac{35}{16}u_j. \quad (\text{B.8d})$$

$$\beta_0 = \frac{(-15u_j + 25u_{j+1} - 13u_{j+2} + 3u_{j+3})^2}{64} + \frac{13}{12}(2u_j - 5u_{j+1} + 4u_{j+2} - u_{j+3})^2 \quad (\text{B.9a})$$

$$+ \frac{781}{720}(-u_j + 3u_{j+1} - 3u_{j+2} + u_{j+3})^2,$$

$$\beta_1 = \frac{(-3u_{j-1} - 3u_j + 7u_{j+1} - u_{j+2})^2}{64} + \frac{13}{12}(u_{j-1} - 2u_j + u_{j+1})^2 \quad (\text{B.9b})$$

$$+ \frac{781}{720}(-u_{j-1} + 3u_j - 3u_{j+1} + u_{j+2})^2,$$

$$\beta_2 = \frac{(u_{j-2} - 7u_{j-1} + 3u_j + 3u_{j+1})^2}{64} + \frac{13}{12}(u_{j-1} - 2u_j + u_{j+1})^2 \quad (\text{B.9c})$$

$$+ \frac{781}{720}(-u_{j-2} + 3u_{j-1} - 3u_j + u_{j+1})^2,$$

$$\begin{aligned}\beta_3 = & \frac{(-3u_{j-3} + 13u_{j-2} - 25u_{j-1} + 15u_j)^2}{64} + \frac{13}{12}(-u_{j-3} + 4u_{j-2} - 5u_{j-1} + 2u_j)^2 \\ & + \frac{781}{720}(-u_{j-3} + 3u_{j-2} - 3u_{j-1} + u_j)^2.\end{aligned}\quad (\text{B.9d})$$

For  $k = 9$  ( $r = 5$ ):

$$d_0 = \frac{9}{256}, \quad u_{j+\frac{1}{2}}^{-(0)} = \frac{35}{128}u_j + \frac{35}{32}u_{j+1} - \frac{35}{64}u_{j+2} + \frac{7}{32}u_{j+3} - \frac{5}{128}u_{j+4}, \quad (\text{B.10a})$$

$$d_1 = \frac{21}{64}, \quad u_{j+\frac{1}{2}}^{-(1)} = -\frac{5}{128}u_{j-1} + \frac{15}{32}u_j + \frac{45}{64}u_{j+1} - \frac{5}{32}u_{j+2} + \frac{3}{128}u_{j+3}, \quad (\text{B.10b})$$

$$d_2 = \frac{63}{128}, \quad u_{j+\frac{1}{2}}^{-(2)} = \frac{3}{128}u_{j-2} - \frac{5}{32}u_{j-1} + \frac{45}{64}u_j + \frac{15}{32}u_{j+1} - \frac{5}{128}u_{j+2}, \quad (\text{B.10c})$$

$$d_3 = \frac{9}{64}, \quad u_{j+\frac{1}{2}}^{-(3)} = -\frac{5}{128}u_{j-3} + \frac{7}{32}u_{j-2} - \frac{35}{64}u_{j-1} + \frac{35}{32}u_j + \frac{35}{128}u_{j+1}, \quad (\text{B.10d})$$

$$d_4 = \frac{1}{256}, \quad u_{j+\frac{1}{2}}^{-(4)} = \frac{35}{128}u_{j-4} - \frac{45}{32}u_{j-3} + \frac{189}{64}u_{j-2} - \frac{105}{32}u_{j-1} + \frac{315}{128}u_j. \quad (\text{B.10e})$$

$$\begin{aligned}\beta_0 = & \frac{(-35u_j + 70u_{j+1} - 56u_{j+2} + 26u_{j+3} - 5u_{j+4})^2}{256} \\ & + \frac{(4613u_j - 13772u_{j+1} + 15198u_{j+2} - 7532u_{j+3} + 1493u_{j+4})^2}{2246400} \\ & + \frac{781}{2880}(-5u_j + 18u_{j+1} - 24u_{j+2} + 14u_{j+3} - 3u_{j+4})^2 \\ & + \frac{1421461}{1310400}(u_j - 4u_{j+1} + 6u_{j+2} - 4u_{j+3} + u_{j+4})^2,\end{aligned}\quad (\text{B.11a})$$

$$\begin{aligned}\beta_1 = & \frac{(-5u_{j-1} - 10u_j + 20u_{j+1} - 6u_{j+2} + u_{j+3})^2}{256} \\ & + \frac{(1493u_{j-1} - 2852u_j + 1158u_{j+1} + 268u_{j+2} - 67u_{j+3})^2}{2246400} \\ & + \frac{781}{2880}(-3u_{j-1} + 10u_j - 12u_{j+1} + 6u_{j+2} - u_{j+3})^2 \\ & + \frac{1421461}{1310400}(u_{j-1} - 4u_j + 6u_{j+1} - 4u_{j+2} + u_{j+3})^2,\end{aligned}\quad (\text{B.11b})$$

$$\begin{aligned}\beta_2 = & \frac{(u_{j-2} - 10u_{j-1} + 10u_{j+1} - u_{j+2})^2}{256} \\ & + \frac{(-67u_{j-2} + 1828u_{j-1} - 3522u_j + 1828u_{j+1} - 67u_{j+2})^2}{2246400} \\ & + \frac{781}{2880}(-u_{j-2} + 2u_{j-1} - 2u_{j+1} + u_{j+2})^2 \\ & + \frac{1421461}{1310400}(u_{j-2} - 4u_{j-1} + 6u_j - 4u_{j+1} + u_{j+2})^2,\end{aligned}\quad (\text{B.11c})$$

$$\begin{aligned}
\beta_3 = & \frac{(-u_{j-3} + 6u_{j-2} - 20u_{j-1} + 10u_j + 5u_{j+1})^2}{256} \\
& + \frac{(-67u_{j-3} + 268u_{j-2} + 1158u_{j-1} - 2852u_j + 1493u_{j+1})^2}{2246400} \\
& + \frac{781}{2880} (u_{j-3} - 6u_{j-2} + 12u_{j-1} - 10u_j + 3u_{j+1})^2 \\
& + \frac{1421461}{1310400} (u_{j-3} - 4u_{j-2} + 6u_{j-1} - 4u_j + u_{j+1})^2,
\end{aligned} \tag{B.11d}$$

$$\begin{aligned}
\beta_4 = & \frac{(5u_{j-4} - 26u_{j-3} + 56u_{j-2} - 70u_{j-1} + 35u_j)^2}{256} \\
& + \frac{(1493u_{j-4} - 7532u_{j-3} + 15198u_{j-2} - 13772u_{j-1} + 4613u_j)^2}{2246400} \\
& + \frac{781}{2880} (3u_{j-4} - 14u_{j-3} + 24u_{j-2} - 18u_{j-1} + 5u_j)^2 \\
& + \frac{1421461}{1310400} (u_{j-4} - 4u_{j-3} + 6u_{j-2} - 4u_{j-1} + u_j)^2.
\end{aligned} \tag{B.11e}$$

We remark here that to derive the classical smoothness indicator up to  $r = 5$ , we employ the following sum-of-squares (SOS) decomposition that is similar to the one given in [4, Appendix B]. For any  $p(x) \in \mathcal{P}^4([- \frac{h}{2}, \frac{h}{2}])$ ,

$$\begin{aligned}
& \int_{-\frac{h}{2}}^{\frac{h}{2}} \sum_{l=1}^{\infty} h^{2l-1} \left( p^{(l)}(x) \right)^2 dx \\
& = \left( hp^{(1)}(0) + \frac{1}{24} h^3 p^{(3)}(0) \right)^2 + \frac{(520h^2 p^{(2)}(0) + 21h^4 p^{(4)}(0))^2}{249600} + \frac{781}{720} \left( h^3 p^{(3)}(0) \right)^2 + \frac{1421461}{1310400} \left( h^4 p^{(4)}(0) \right)^2.
\end{aligned} \tag{B.12}$$

Then, to compute the smoothness indicator  $\beta$ 's in any cell  $[x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ , it only remains to provide the interpolation-based derivative values at the cell center  $x_j$ .

## Appendix C High-order flux correction terms

We give the  $2r = (k+1)$ th-order central difference implementation here. All the coefficients are computed by `Mathematica`, and are verified to be consistent with those in [4] and [1, Appendix A], to ensure that

$$\frac{1}{\Delta x} \left( \left( \mathbf{f}_{j+\frac{1}{2}} + \mathbf{f}_{j+\frac{1}{2}}^{\text{cor},r} \right) - \left( \mathbf{f}_{j-\frac{1}{2}} + \mathbf{f}_{j-\frac{1}{2}}^{\text{cor},r} \right) \right) = \frac{d}{dx} \mathbf{f}|_j + \mathcal{O}(\Delta x^{2r}). \tag{C.1}$$

For  $k = 1$  ( $r = 1$ ):

$$\mathbf{f}_{j+\frac{1}{2}}^{\text{cor},1} = 0. \tag{C.2}$$

For  $k = 3$  ( $r = 2$ ):

$$\mathbf{f}_{j+\frac{1}{2}}^{\text{cor},2} = -\frac{1}{48} (\mathbf{f}_{j-1} + \mathbf{f}_{j+2}) + \frac{1}{48} (\mathbf{f}_j + \mathbf{f}_{j+1}). \tag{C.3}$$

For  $k = 5$  ( $r = 3$ ):

$$\mathbf{f}_{j+\frac{1}{2}}^{\text{cor},3} = \frac{19}{3840} (\mathbf{f}_{j-2} + \mathbf{f}_{j+3}) - \frac{137}{3840} (\mathbf{f}_{j-1} + \mathbf{f}_{j+2}) + \frac{59}{1920} (\mathbf{f}_j + \mathbf{f}_{j+1}). \tag{C.4}$$

For  $k = 7$  ( $r = 4$ ):

$$\mathbf{f}_{j+\frac{1}{2}}^{\text{cor},4} = -\frac{81}{71680}(\mathbf{f}_{j-3} + \mathbf{f}_{j+4}) + \frac{2279}{215040}(\mathbf{f}_{j-2} + \mathbf{f}_{j+3}) - \frac{9859}{215040}(\mathbf{f}_{j-1} + \mathbf{f}_{j+2}) + \frac{7823}{215040}(\mathbf{f}_j + \mathbf{f}_{j+1}). \quad (\text{C.5})$$

For  $k = 9$  ( $r = 5$ ):

$$\begin{aligned} \mathbf{f}_{j+\frac{1}{2}}^{\text{cor},5} = & \frac{5359}{20643840}(\mathbf{f}_{j-4} + \mathbf{f}_{j+5}) - \frac{60841}{20643840}(\mathbf{f}_{j-3} + \mathbf{f}_{j+4}) + \frac{81491}{5160960}(\mathbf{f}_{j-2} + \mathbf{f}_{j+3}) \\ & - \frac{274129}{5160960}(\mathbf{f}_{j-1} + \mathbf{f}_{j+2}) + \frac{413017}{10321920}(\mathbf{f}_j + \mathbf{f}_{j+1}). \end{aligned} \quad (\text{C.6})$$

## References

- [1] Dinshaw S. Balsara, Deepak Bhoriya, Chi-Wang Shu, and Harish Kumar, *Efficient alternative finite difference WENO schemes for hyperbolic conservation laws*, Communications on Applied Mathematics and Computation (2024).
- [2] Stephen F. Davis, *Simplified second-order Godunov-type methods*, SIAM Journal on Scientific and Statistical Computing **9** (1988), no. 3, 445–473.
- [3] Bernd Einfeldt, Claus-Dieter Munz, Philip L. Roe, and Björn Sjögreen, *On Godunov-type methods near low densities*, Journal of Computational Physics **92** (1991), no. 2, 273–295.
- [4] Zhen Gao, Li-Li Fang, Bao-Shan Wang, Yinghua Wang, and Wai Sun Don, *Seventh and ninth orders characteristic-wise alternative WENO finite difference schemes for hyperbolic conservation laws*, Computers and Fluids **202** (2020), no. 104519.
- [5] Zhen Gao, Shuang Guo, Bao-Shan Wang, and Yaguang Gu, *High order bound- and positivity-preserving finite difference affine-invariant AWENO scheme for the five-equation model of two-medium flows*, Communications in Computational Physics **36** (2024), no. 3, 781–820.
- [6] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor, *Strong stability-preserving high-order time discretization methods*, SIAM Review **43** (2001), no. 1, 89–112.
- [7] Yaguang Gu, Zhen Gao, Guanghui Hu, Peng Li, and Lifeng Wang, *High order finite difference alternative WENO scheme for multi-component flows*, Journal of Scientific Computing **89** (2021), no. 3, 52.
- [8] Jean-Luc Guermond and Bojan Popov, *Fast estimation from above of the maximum wave speed in the Riemann problem for the Euler equations*, Journal of Computational Physics **321** (2016), 908–926.
- [9] Amiram Harten, Björn Engquist, Stanley Osher, and Sukumar R. Chakravarthy, *Uniformly high order accurate essentially non-oscillatory schemes, iii*, Journal of Computational Physics **71** (1987), no. 2, 231–303.
- [10] Amiram Harten, Peter David Lax, and Bram van Leer, *On upstream differencing and Godunov-type schemes for hyperbolic conservation laws*, SIAM Review **25** (1983), no. 1, 35–61.



- [11] Xiangyu Hu, Nikolaus A. Adams, and Chi-Wang Shu, *Positivity-preserving method for high-order conservative schemes solving compressible Euler equations*, Journal of Computational Physics **242** (2013), 169–180.
- [12] Guang-Shan Jiang and Chi-Wang Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics **126** (1996), no. 1, 202–228.
- [13] Yan Jiang, Chi-Wang Shu, and Mengping Zhang, *An alternative formulation of finite difference weighted ENO schemes with Lax–Wendroff time discretization for conservation laws*, SIAM Journal on Scientific Computing **35** (2013), no. 2, A1137–A1160.
- [14] Xu-Dong Liu, Stanley Osher, and Tony Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics **115** (1994), no. 1, 200–212.
- [15] Xu-Dong Liu and Eitan Tadmor, *Third order nonoscillatory central scheme for hyperbolic conservation laws*, Numerische Mathematik **79** (1998), no. 3, 397–425.
- [16] Haim Nessyahu and Eitan Tadmor, *Non-oscillatory central differencing for hyperbolic conservation laws*, Journal of Computational Physics **87** (1990), no. 2, 408–463.
- [17] Jianxian Qiu and Chi-Wang Shu, *On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes*, Journal of Computational Physics **183** (2002), no. 1, 187–209.
- [18] Omer San and Kursat Kara, *Evaluation of Riemann flux solvers for WENO reconstruction schemes: Kelvin–Helmholtz instability*, Computers and Fluids **117** (2015), 24–41.
- [19] Jing Shi, Yong-Tao Zhang, and Chi-Wang Shu, *Resolution of high order WENO schemes for complicated flow structures*, Journal of Computational Physics **186** (2003), no. 2, 690–696.
- [20] Chi-Wang Shu and Stanley Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics **77** (1988), no. 2, 439–471.
- [21] ———, *Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii*, Journal of Computational Physics **83** (1989), no. 1, 32–78.
- [22] Chi-Wang Shu, Thomas A. Zang, Gordon Erlebacher, David Whitaker, and Stanley Osher, *High-order ENO schemes applied to two- and three-dimensional compressible flow*, Applied Numerical Mathematics **9** (1992), no. 1, 45–71.
- [23] Joel Alan Smoller, *Shock waves and reaction–diffusion equations*, 2 ed., Grundlehren der mathematischen Wissenschaften, vol. 258, Springer New York, New York, 10 1994.
- [24] Eleuterio Francisco Toro, *Riemann solvers and numerical methods for fluid dynamics*, 3 ed., Springer Berlin Heidelberg, Berlin, Heidelberg, 04 2009.
- [25] François Vilar, Pierre-Henri Maire, and Rémi Abgrall, *Cell-centered discontinuous Galerkin discretizations for two-dimensional scalar conservation laws on unstructured grids and for one-dimensional Lagrangian hydrodynamics*, Computers and Fluids **46** (2011), no. 1, 498–504.

- [26] Bao-Shan Wang and Wai Sun Don, *Affine-invariant WENO weights and operator*, Applied Numerical Mathematics **181** (2022), 630–646.
- [27] Paul Woodward and Phillip Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, Journal of Computational Physics **54** (1984), no. 1, 115–173.
- [28] Ziyao Xu and Chi-Wang Shu, *Local characteristic decomposition-free high-order finite difference WENO schemes for hyperbolic systems endowed with a coordinate system of Riemann invariants*, SIAM Journal on Scientific Computing **46** (2024), no. 2, A1352–A1372.
- [29] Xiangxiong Zhang and Chi-Wang Shu, *On positivity-preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, Journal of Computational Physics **229** (2010), no. 23, 8918–8934.