

## An Efficient Mapped WENO Scheme Using Approximate Constant Mapping

Ruo Li<sup>1,2</sup> and Wei Zhong<sup>2,3,\*</sup>

<sup>1</sup> *CAPT, LMAM, Peking University, Beijing 100871, China*

<sup>2</sup> *School of Mathematical Sciences, Peking University, Beijing 100871, China*

<sup>3</sup> *Northwest Institute of Nuclear Technology, Xi'an 710024, China*

Received 13 May 2021; Accepted (in revised version) 12 August 2021

---

**Abstract.** We present a novel mapping approach for WENO schemes through the use of an approximate constant mapping function which is constructed by employing an approximation of the classic signum function. The new approximate constant mapping function is designed to meet the overall criteria for a proper mapping function required in the design of the WENO-PM6 scheme. The WENO-PM6 scheme was proposed to overcome the potential loss of accuracy of the WENO-M scheme which was developed to recover the optimal convergence order of the WENO-JS scheme at critical points. Our new mapped WENO scheme, denoted as WENO-ACM, maintains almost all advantages of the WENO-PM6 scheme, including low dissipation and high resolution, while decreases the number of mathematical operations remarkably in every mapping process leading to a significant improvement of efficiency. The convergence rates of the WENO-ACM scheme have been shown through one-dimensional linear advection equation with various initial conditions. Numerical results of one-dimensional Euler equations for the Riemann problems, the Mach 3 shock-density wave interaction and the Woodward-Colella interacting blastwaves are improved in comparison with the results obtained by the WENO-JS, WENO-M and WENO-PM6 schemes. Numerical experiments with two-dimensional problems as the 2D Riemann problem, the shock-vortex interaction, the 2D explosion problem, the double Mach reflection and the forward-facing step problem modeled via the two dimensional Euler equations have been conducted to demonstrate the high resolution and the effectiveness of the WENO-ACM scheme. The WENO-ACM scheme provides significantly better resolution than the WENO-M scheme and slightly better resolution than the WENO-PM6 scheme, and compared to the WENO-M and WENO-PM6 schemes, the extra computational cost is reduced by more than 83% and 93%, respectively.

**AMS subject classifications:** 65M06, 65M12

---

\*Corresponding author. *Email addresses:* zhongwei2016@pku.edu.cn (W. Zhong), rli@math.pku.edu.cn (R. Li)

**Key words:** WENO, approximate constant mapping, hyperbolic problems, high resolution, low computational cost.

---

## 1. Introduction

A number of essentially non-oscillatory (ENO) schemes [16–19, 33, 34] and weighted ENO (WENO) schemes [23, 27] have been developed quite successfully to solve the hyperbolic conservation laws, which may develop discontinuities in its solutions even if the initial conditions are smooth. The goal of this paper is to propose an improved version of the fifth-order WENO scheme for the hyperbolic conservation laws taking the form

$$\mathbf{u}_t + \sum_{\alpha=1}^d \mathbf{f}_{\alpha}(\mathbf{u})_{x_{\alpha}} = 0, \quad x_{\alpha} \in \mathbb{R}, \quad t > 0.$$

Here the function  $\mathbf{u} = (u_1, u_2, \dots, u_m)^T$  is an  $m$ -dimensional vector of conserved variables, and flux  $\mathbf{f}_{\alpha}(\mathbf{u})$  is a vector-valued function of  $m$  components with  $x_{\alpha}$  and  $t$  variables.

Liu *et al.* [27] developed the first version of WENO schemes which convert an  $r$ -th order ENO scheme [16–19, 33, 34] into an  $(r + 1)$ -th order WENO scheme by using a convex combination of all candidate substencils instead of just one as in the original ENO scheme. Later, Jiang and Shu [23] proposed the classic WENO-JS schemes with an improvement that an  $r$ -th order ENO scheme can be converted into a  $(2r - 1)$ -th order WENO scheme by introducing a new definition of the smoothness indicator used to measure the smoothness of the numerical solution on a substencil. Then, the weighting method presented in [27] and the smoothness indicators designed in [23] eventually became a standard, and the WENO-JS schemes especially the fifth-order one [23] developed into one of the most popular high-order methods [25]. In recent decades, many successful works have been done to raise some issues about WENO schemes [1, 5, 6, 9, 11, 12, 20, 22, 44].

It was clearly pointed out by Henrick *et al.* [20] that, in general, the fifth-order WENO-JS scheme is only third-order or even less accurate at critical points of order  $n_{\text{cp}} = 1$  in smooth regions, where  $n_{\text{cp}}$  denotes the order of the critical point; e.g.,  $n_{\text{cp}} = 1$  corresponds to  $f' = 0, f'' \neq 0$  and  $n_{\text{cp}} = 2$  corresponds to  $f' = 0, f'' = 0, f''' \neq 0$ , etc. To overcome this problem, Henrick *et al.* [20] introduced a carefully designed mapping function leading to the first mapped WENO scheme named WENO-M. Compared to the WENO-JS scheme [23], the WENO-M scheme is able to recover the optimal convergence order near critical points in smooth regions and generate more accurate solutions. Another significant contribution of the work by Henrick *et al.* [20] is that they derived a strong sufficient condition on the weights of substencils for WENO schemes to achieve optimal convergence orders and this condition has become the primary criterion in the design of all other mapped WENO schemes [11, 12, 25, 26, 37, 38, 40]. Recently, Feng *et al.* [11] found that the mapping

function of the WENO-M scheme [20] may amplify the effect from the non-smooth substencils and thus cause a potential loss of accuracy near discontinuities. To address this issue, they proposed two additional requirements to the mapping function  $g(\omega)$ , that is,  $g'(0) = 0$  and  $g'(1) = 0$ . Then, the WENO-PM $k$  scheme has been devised [11] by employing a piecewise polynomial mapping function which satisfies the two additional requirements above and the original criteria in [20], and the WENO-PM6 scheme was recommended, which is able to generate more accurate numerical solutions [11] near the discontinuities than the classic WENO-JS scheme and the WENO-M scheme. Furthermore, the two additional requirements were considered to be very important to decrease the effect from the non-smooth substencils [40] and were used in the construction of the WENO-RM( $mn0$ ) scheme [40]. Similarly, requirements  $g'(0) = 1$  and  $g'(1) = 1$ , which are also used to decrease the effect from the non-smooth substencils, were employed when the WENO-PPM $n$  ( $n = 4, 5, 6$ ) [25], WENO-RM( $k, m, s$ ) [37], WENO-AIM( $k, m, c$ ) [38] and WENO-MAIM $i$  [26] schemes were constructed. From a different perspective, Borges *et al.* [5] proposed another version of the fifth-order WENO scheme, which is called the WENO-Z scheme. It drives the weights to the optimal values faster than the WENO-M scheme by employing a global higher order reference value for the smoothness indicators [10]. Later, the WENO-Z scheme was successfully extended to higher orders by Castro *et al.* [6]. The recommended fifth-order WENO-Z scheme with the power parameter  $p = 1$  gives less dissipation than the WENO-JS scheme but its convergence order is fourth-order at the first-order critical points [41] and will drop to second-order at higher order critical points [10]. Recently, Don *et al.* [9, 39] demonstrated that the optimal order of accuracy for the WENO-Z scheme can be obtained regardless of any order critical points by setting the parameter  $\epsilon$ , which is used to prevent the denominator becoming zero, as a function of the spacial step  $\Delta x$ . However, it was indicated [21] that adjusting  $\epsilon$  to recover accuracy is actually an implicit switch between the WENO scheme and the upstream central scheme, and its effect is dependent on the specific problem. It is worthy to note that, lately, Zhu *et al.* [46] has proposed a new type of high-order MWENO schemes with unequal-sized stencils to achieve increasingly high order of accuracy for the seventh-order and ninth-order versions. As we focus mainly on the fifth-order version here, we refer to [46] for more details.

Various existing mapped WENO schemes, e.g., WENO-M, WENO-PM6, WENO-RM260, WENO-PPM5, WENO-RM( $k, m, s$ ), WENO-AIM ( $k, m, c$ ), WENO-MAIM $i$ , etc., can improve the performances of the classic WENO-JS scheme in some ways like achieving optimal convergence orders near critical points in smooth regions, having low dissipation and generating more accurate solutions near the discontinuities. However, as some specified complicated mapping procedures must be performed, the chief drawback of these existing mapped schemes is that the computational cost increases significantly. Taking the WENO-M scheme that is one of the concerned WENO schemes in this paper as an example, as reported in [5], its extra computational cost is 20% to 30% compared to the WENO-JS scheme when calculating two dimensional Euler equations. For the WENO-PM6 scheme, another concerned WENO scheme in this paper, it was noted

that [38] the piecewise nature requires logic operations to be implemented during the mapping process, which makes it harder to vectorize the operations and increases the computational cost. Hong *et al.* indicated that [21], as the form of the mapping function designed in the WENO-PM6 scheme becomes more complicated than the WENO-M scheme, its extra computational cost will have a further increase. Actually, our extensive numerical tests show that the extra computational cost of the WENO-PM6 scheme will increase by more than 54% compared to the WENO-JS scheme, and we will give the results in Subsection 4.4 in detail.

In this paper, an approximate constant mapping function is designed at first. This new mapping function satisfies the original criteria proposed in [20] so that the new corresponding WENO scheme, abbreviated as WENO-ACM, is able to achieve the optimal convergence orders near critical points in smooth regions. And also, the new mapping function maintains  $g'(0) = 0$  and  $g'(1) = 0$  so that it can decrease the effect from the non-smooth substencils as the WENO-PM6 scheme does. Thus, the WENO-ACM scheme is able to yield low dissipation and high resolution results comparable to that of the WENO-PM6 scheme. The greatest benefit is that the new mapping function almost uses only one assignment operation to implement the mapping process, instead of evaluating the mapping functions involving multiple multiplication and division as other existing mapped WENO schemes do. Therefore, the cost of the WENO-ACM scheme is very low. Numerical experiments with various benchmark problems modeled via the two dimensional Euler equations are conducted to demonstrate that the WENO-ACM scheme generates significantly better resolution than the WENO-M scheme and slightly better resolution than the WENO-PM6 scheme, while the extra computational cost is reduced by more than 83% compared to the WENO-M scheme and reduced by more than 93% compared to the WENO-PM6 scheme.

The remainder of this paper is organized as follows. In Section 2, we give a brief description of the finite volume methodology and the procedures of the WENO-JS [23], WENO-M [20] and WENO-PM $k$  [11] schemes to clarify our major concern. In Section 3, we introduce the details on how we construct the new mapped WENO scheme with approximate constant mapping and then provide the parametric study of the new mapping function and the convergence property of the new mapped WENO scheme. In Section 4, some numerical experiments are presented to compare the performances of different WENO schemes, and the computational cost comparisons are also shown in this section. Finally, the conclusions are given in Section 5.

## 2. Review of finite volume WENO schemes

### 2.1. Finite volume methodology

Consider the following one-dimensional scalar hyperbolic conservation law:

$$\partial_t u + \partial_x f(u) = 0, \quad (2.1)$$

which is to be solved on the domain  $x \in [x_l, x_r]$  for  $t \geq 0$  with the initial condition  $u(x, 0) = u_0(x)$ . Throughout this paper, we assume that the computational domain is discretized into uniform cells  $I_j = [x_{j-1/2}, x_{j+1/2}]$ ,  $j = 1, \dots, N$  with width  $\Delta x = (x_r - x_l)/N$ . The cell center of  $I_j$  is denoted by  $x_j = x_l + (j - 1/2)\Delta x$  and its cell boundaries are denoted by  $x_{j\pm 1/2} = x_j \pm \Delta x/2$ . The cell average  $\bar{u}_j$  of  $I_j$  is defined by

$$\bar{u}_j = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(\xi, t) d\xi. \quad (2.2)$$

By integrating Eq. (2.1) over  $I_j$  and employing some simple mathematical manipulations, we can approximate Eq. (2.1) by the following finite volume conservative formulation:

$$\frac{d\bar{u}_j(t)}{dt} \approx -\frac{1}{\Delta x} \left( \hat{f}\left(u_{j+1/2}^-, u_{j+1/2}^+\right) - \hat{f}\left(u_{j-1/2}^-, u_{j-1/2}^+\right) \right). \quad (2.3)$$

In Eq. (2.3),  $\bar{u}_j(t)$  is the numerical approximation to  $\bar{u}_j$  defined in Eq. (2.2), and the numerical flux  $\hat{f}(u^-, u^+)$  where  $u^-$  and  $u^+$  refer to the left-sided and right-sided limits of  $u$  is a replacement of the physical flux function  $f(u)$ . For the possible presence of discontinuities,  $u^-$  and  $u^+$  are usually not equal. For hyperbolic laws, the numerical flux  $\hat{f}(u^-, u^+)$  is a monotone function and it is consistent with the physical flux, i.e.,  $\hat{f}(u, u) = f(u)$ . In this paper, the global Lax-Friedrichs flux  $\hat{f}(a, b) = \frac{1}{2}[f(a) + f(b) - \alpha(b - a)]$  is chosen, where  $\alpha = \max_u |f'(u)|$  is a constant and the maximum is taken over the whole range of  $u$ . In Eq. (2.3),  $u_{j\pm 1/2}^\pm$  can be computed by the technique of reconstruction, like some WENO reconstructions which are described in the following subsections. For the system of hyperbolic conservation laws a local characteristic decomposition is used in the reconstruction, and [23] is referred to for more details. Two commonly used classes of finite volume WENO schemes in two dimensional Cartesian meshes are studied in detail in [43], and the one denoted as class A is taken in this paper.

## 2.2. WENO-JS

We recall the reconstruction process of the fifth-order WENO-JS scheme [23], which has successfully been extended to higher order ones [4, 13]. We describe only the procedure of the left-biased reconstruction  $u_{j+1/2}^-$  as the right-biased one  $u_{j+1/2}^+$  can easily be obtained by mirror symmetry with respect to the location  $x_{j+1/2}$  of that for  $u_{j+1/2}^-$ . For simplicity of notation, we do not use the subscript “-” in the following content.

Explicitly, for the five-point stencil  $S^5 = \{I_{j-2}, I_{j-1}, I_j, I_{j+1}, I_{j+2}\}$ , the third-order approximations of  $u(x_{j+1/2}, t)$  associated with three left-biased substencils  $S_s =$

$\{I_{j+s-2}, I_{j+s-1}, I_{j+s}\}$ ,  $s = 0, 1, 2$  are as follows:

$$\begin{aligned} u_{j+\frac{1}{2}}^0 &= \frac{1}{6}(2\bar{u}_{j-2} - 7\bar{u}_{j-1} + 11\bar{u}_j), \\ u_{j+\frac{1}{2}}^1 &= \frac{1}{6}(-\bar{u}_{j-1} + 5\bar{u}_j + 2\bar{u}_{j+1}), \\ u_{j+\frac{1}{2}}^2 &= \frac{1}{6}(2\bar{u}_j + 5\bar{u}_{j+1} - \bar{u}_{j+2}). \end{aligned} \quad (2.4)$$

The fifth-order approximation of global stencil  $S^5$  is built via the following convex combination of the three third-order approximations in Eq. (2.4):

$$u_{j+\frac{1}{2}} = \sum_{s=0}^2 \omega_s u_{j+\frac{1}{2}}^s,$$

where  $\omega_s$  are nonlinear weights. In the classic WENO-JS scheme, the nonlinear weights are calculated by

$$\omega_s^{\text{JS}} = \frac{\alpha_s^{\text{JS}}}{\sum_{l=0}^2 \alpha_l^{\text{JS}}}, \quad \alpha_s^{\text{JS}} = \frac{d_s}{(\epsilon + \beta_s)^2}, \quad (2.5)$$

where  $\epsilon$  is a small positive number introduced to prevent the denominator being zero and it was taken to be  $10^{-6}$  in the original WENO-JS scheme, and  $d_0 = 0.1, d_1 = 0.6, d_2 = 0.3$  are ideal weights of  $\omega_s$  satisfying

$$\sum_{s=0}^2 d_s u_{j+\frac{1}{2}}^s = u(x_{j+\frac{1}{2}}, t) + \mathcal{O}(\Delta x^5)$$

in smooth regions. The parameters  $\beta_s$  named smoothness indicators are defined as follows [23]:

$$\begin{aligned} \beta_0 &= \frac{13}{12}(\bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j)^2 + \frac{1}{4}(\bar{u}_{j-2} - 4\bar{u}_{j-1} + 3\bar{u}_j)^2, \\ \beta_1 &= \frac{13}{12}(\bar{u}_{j-1} - 2\bar{u}_j + \bar{u}_{j+1})^2 + \frac{1}{4}(\bar{u}_{j-1} - \bar{u}_{j+1})^2, \\ \beta_2 &= \frac{13}{12}(\bar{u}_j - 2\bar{u}_{j+1} + \bar{u}_{j+2})^2 + \frac{1}{4}(3\bar{u}_j - 4\bar{u}_{j+1} + \bar{u}_{j+2})^2. \end{aligned}$$

In smooth regions without critical points, the classic WENO-JS scheme is able to achieve fifth-order of accuracy. However, at critical points where the first derivative vanishes but the third derivative does not simultaneously, it loses accuracy and its order of accuracy decreases to third-order or even less. More details can be found in [20].

### 2.3. WENO-M

In order to overcome the problem that the WENO-JS scheme loses accuracy at critical points, Henrick *et al.* [20] proposed the WENO-M scheme by constructing a mapping function of the nonlinear weights  $\omega$  given by

$$(g^M)_s(\omega) = \frac{\omega(d_s + d_s^2 - 3d_s\omega + \omega^2)}{d_s^2 + (1 - 2d_s)\omega}, \quad \omega \in [0, 1]. \quad (2.6)$$

Clearly,  $(g^M)_s(\omega)$  is a monotonically increasing function in  $[0, 1]$  with finite slopes, and it satisfies the following properties.

**Lemma 2.1.** *The mapping function defined by Eq. (2.6) satisfies:*

- C1.  $0 \leq (g^M)_s(\omega) \leq 1$ ,  $(g^M)_s(0) = 0$ ,  $(g^M)_s(d_s) = d_s$ ,  $(g^M)_s(1) = 1$ ;
- C2.  $(g^M)'_s(d_s) = (g^M)''_s(d_s) = 0$ .

By employing Eqs. (2.5)-(2.6), one can obtain the mapped weights as follows:

$$\omega_s^M = \frac{\alpha_s^M}{\sum_{l=0}^2 \alpha_l^M}, \quad \alpha_s^M = (g^M)_s(\omega_s^{JS}).$$

It has been analyzed and proved in detail in [20] that the WENO-M scheme is able to achieve the optimal order of accuracy in smooth regions even near the first-order critical point.

### 2.4. WENO-PM6

Feng *et al.* [11] found that the mapping operation of the WENO-M scheme will cause the potential loss of accuracy near the discontinuities or the parts with sharp gradients. To overcome this drawback, they add two requirements, that is,  $g'_s(0) = 0$  and  $g'_s(1) = 0$ , to the original criteria (see Lemma 2.1) by Henrick *et al.* [20]. And then a new mapping function is defined by the following piecewise polynomial function:

$$(g^{PM})_s(\omega) = C_1(\omega - d_s)^{k+1}(\omega + C_2) + d_s, \quad k \geq 2, \quad (2.7)$$

where  $C_1, C_2$  are constants with specified parameters  $k$  and  $d_s$ , and they are calculated by

$$\begin{cases} C_1 = (-1)^k \frac{k+1}{d_s^{k+1}}, & C_2 = \frac{d_s}{k+1}, & \text{if } 0 \leq \omega \leq d_s, \\ C_1 = -\frac{k+1}{(1-d_s)^{k+1}}, & C_2 = \frac{d_s - (k+2)}{k+1}, & \text{if } d_s < \omega \leq 1. \end{cases}$$

**Lemma 2.2.** *The mapping function  $(g^{PM})_s(\omega)$  defined by Eq. (2.7) satisfies:*

- C1.  $(g^{\text{PM}})'_s(\omega) \geq 0, \omega \in [0, 1];$   
 C2.  $(g^{\text{PM}})_s(0) = 0, (g^{\text{PM}})_s(d_s) = d_s, (g^{\text{PM}})_s(1) = 1;$   
 C3.  $(g^{\text{PM}})'_s(d_s) = \dots = (g^{\text{PM}})_s^{(k)}(d_s) = 0;$   
 C4.  $(g^{\text{PM}})'_s(0) = (g^{\text{PM}})'_s(1) = 0.$

Similarly, by employing Eq. (2.5) and Eq. (2.7) where the parameter  $k$  is taken to be 6 as recommended in [11], one can obtain the mapped weights of the WENO-PM6 scheme as follows:

$$\omega_s^{\text{PM6}} = \frac{\alpha_s^{\text{PM6}}}{\sum_{l=0}^2 \alpha_l^{\text{PM6}}}, \quad \alpha_s^{\text{PM6}} = (g^{\text{PM6}})_s(\omega_s^{\text{JS}}).$$

It has been verified that the WENO-PM6 scheme is able to achieve the optimal order of accuracy as the WENO-M scheme does at critical points. In addition, the resolution of the WENO-PM6 scheme is significantly higher than the WENO-JS scheme and the WENO-M scheme, especially for a long output time. One can see [11] for more details.

## 2.5. Time discretization

Following the method of lines (MOL) approach, the Partial Differential Equation (PDE) Eq. (2.1) can be turned into an Ordinary Differential Equation (ODE) system of the form

$$\frac{d\bar{u}_j(t)}{dt} = \mathcal{L}(u_j), \quad (2.8)$$

where

$$\mathcal{L}(u_j) := -\frac{1}{\Delta x} \left( \hat{f} \left( u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+ \right) - \hat{f} \left( u_{j-\frac{1}{2}}^-, u_{j-\frac{1}{2}}^+ \right) \right).$$

Then, the WENO schemes can be applied to obtain  $\mathcal{L}(u_j)$ .

In all the numerical experiments in this paper, the ODE system Eq. (2.8) is solved using the following explicit, third-order, Strong Stability Preserving (SSP) Runge-Kutta method [14, 15, 33]:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t \mathcal{L}(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t \mathcal{L}(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t \mathcal{L}(u^{(2)}), \end{aligned}$$

where  $u^{(1)}$  and  $u^{(2)}$  are the intermediate stages,  $u^n$  is the value of  $u$  at time level  $t^n = n\Delta t$ , and  $\Delta t$  is the time step satisfying some proper CFL condition.

### 3. The new mapped WENO scheme

#### 3.1. Design and properties of the approximate constant mapping function

##### 3.1.1. The new mapping function

Let  $\text{sgm}(x, \delta, A)$  denote the signum-like function, taking the form

$$\text{sgm}(x, \delta, A) = \begin{cases} \frac{x}{|x|}, & |x| \geq \delta, \\ \frac{x}{(A(\delta^2 - x^2))^{k+3} + |x|}, & |x| < \delta, \end{cases}$$

where  $k \in \mathbb{N}^+$ ,  $\delta > 0$  and  $\delta \rightarrow 0$ . The positive parameter  $A$  is a scale transformation factor introduced to adjust the shape of the mapping function below. As mentioned in [26], we can easily verify that  $\text{sgm}(x, \delta, A)$  is monotone increasing.

Then we can construct a global monotonically increasing mapping function, denoted as  $(g^{\text{ACM}})_s(\omega)$ , by directly splicing two signum-like functions as

$$(g^{\text{ACM}})_s(\omega) = \begin{cases} \frac{d_s}{2} \text{sgm}(\omega - \text{CFS}_s, \delta_s, A) + \frac{d_s}{2}, & \omega \leq d_s, \\ \frac{1 - d_s}{2} \text{sgm}(\omega - \overline{\text{CFS}}_s, \delta_s, A) + \frac{1 + d_s}{2}, & \omega > d_s, \end{cases} \quad (3.1)$$

where the Control Factor of Smoothness  $\text{CFS}_s$  is the same as that in [26] satisfying  $\text{CFS}_s \in (0, d_s)$ , and  $\overline{\text{CFS}}_s = 1 - (1 - d_s)/d_s \times \text{CFS}_s$  with  $\overline{\text{CFS}}_s \in (d_s, 1)$ . In addition, the splicing condition  $\text{CFS}_s + \delta_s < d_s < \overline{\text{CFS}}_s - \delta_s$  and the requirements  $\text{CFS}_s - \delta_s > 0$  and  $\overline{\text{CFS}}_s + \delta_s < 1$  need to be satisfied. Therefore, the value of the parameter  $\delta_s$  is limited by

$$\delta_s < \min \left\{ \text{CFS}_s, d_s - \text{CFS}_s, (1 - d_s) \left( 1 - \frac{\text{CFS}_s}{d_s} \right), \frac{1 - d_s}{d_s} \text{CFS}_s \right\}.$$

The effects of parameters  $\text{CFS}_s, k, A$  and  $\delta_s$  on the mapping function  $(g^{\text{ACM}})_s(\omega)$  will be discussed in the following subsection.

**Remark 3.1.** The splicing condition  $\text{CFS}_s + \delta_s < d_s < \overline{\text{CFS}}_s - \delta_s$  is used to guarantee

$$(g^{\text{ACM}})_s(d_s) = d_s \quad \text{and} \quad (g^{\text{ACM}})'_s(d_s) = (g^{\text{ACM}})''_s(d_s) = \dots = 0.$$

Similarly, the requirements  $\text{CFS}_s - \delta_s > 0$  and  $\overline{\text{CFS}}_s + \delta_s < 1$  are imposed to ensure that  $(g^{\text{ACM}})_s(\omega)$  satisfies the properties at the boundaries  $\omega = 0$  and  $\omega = 1$ , that is,

$$(g^{\text{ACM}})_s(0) = (g^{\text{ACM}})'_s(0^+) = (g^{\text{ACM}})'_s(1^-) = 0 \quad \text{and} \quad (g^{\text{ACM}})_s(1) = 1.$$

As a summary, we state the trivial theorem without proof in the following.

**Theorem 3.1.** *With appropriate parameters  $CFS_s$ ,  $A$  and  $\delta_s$ , the mapping function  $(g^{\text{ACM}})_s(\omega)$  defined by Eq. (3.1) satisfies the following properties:*

- C1.  $(g^{\text{ACM}})_s(0) = 0$ ,  $(g^{\text{ACM}})_s(d_s) = d_s$ ,  $(g^{\text{ACM}})_s(1) = 1$ ;
- C2.  $(g^{\text{ACM}})'_s(\omega) \geq 0$ ,  $\omega \in (0, 1)$ ;
- C3.  $(g^{\text{ACM}})'_s(d_s) = (g^{\text{ACM}})''_s(d_s) = \dots = 0$ ;
- C4.  $(g^{\text{ACM}})'_s(0^+) = (g^{\text{ACM}})'_s(1^-) = 0$ .

We can observe the properties in Theorem 3.1 intuitively from Fig. 1. Now, we give the approximate-constant-mapped WENO scheme, denoted as WENO-ACM, with the mapped weights

$$\omega_s^{\text{ACM}} = \frac{\alpha_s^{\text{ACM}}}{\sum_{l=0}^k \alpha_l^{\text{ACM}}}, \quad \alpha_s^{\text{ACM}} = (g^{\text{ACM}})_s(\omega_s^{\text{JS}}). \quad (3.2)$$

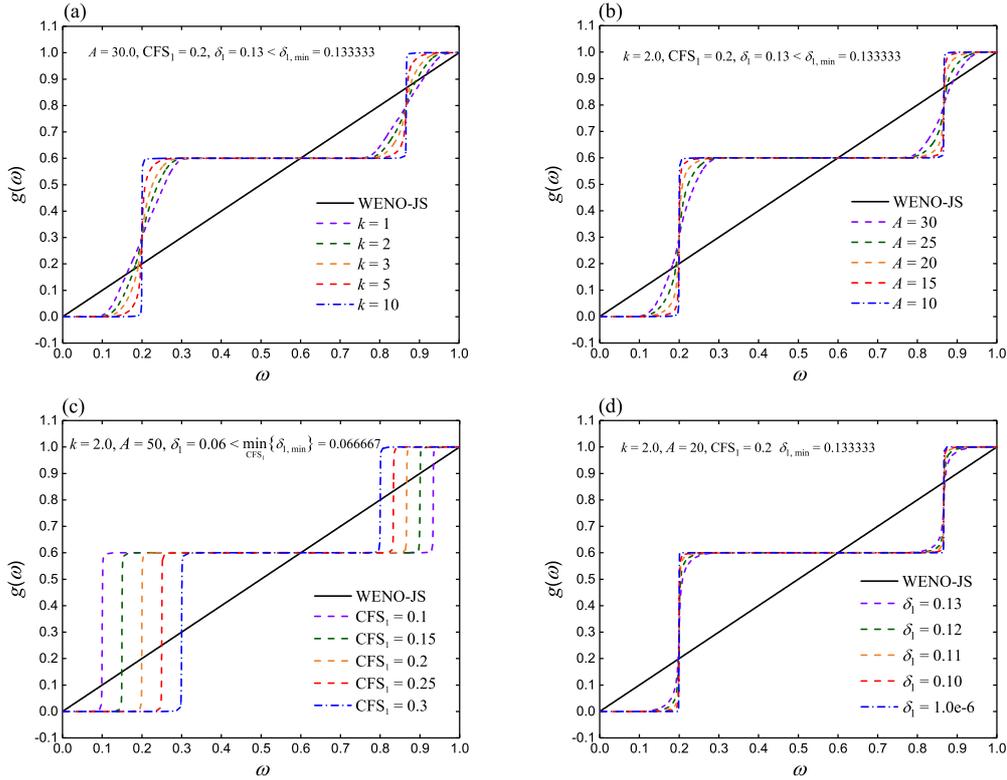


Figure 1: The new mapping function  $(g^{\text{ACM}})_1(\omega)$ , and effects of varying parameters  $k$ ,  $A$ ,  $CFS_1$  and  $\delta_1$  for  $d_1 = 0.6$ .

**Remark 3.2.** By submitting the nonlinear weights  $\omega_s^{\text{JS}}$  of the  $(2r - 1)$ -th order WENO-JS scheme (in [4] and [13] the results of  $r = 2, \dots, 9$  were given) into Eq. (3.2), the above fifth-order WENO-ACM scheme can be easily extended to  $(2r - 1)$ -th order ones.

### 3.1.2. Parametric study

The curves of  $(g^{\text{ACM}})_s(\omega)$  varying with  $\omega$  and the effects of the parameters  $k$ ,  $A$ ,  $\text{CFS}_s$  and  $\delta_s$  are shown in Fig. 1, taking the case of  $d_1 = 0.6$  as an example.

In Fig. 1, we can see the following properties:

- (1) for given  $A$ ,  $\text{CFS}_1$  and  $\delta_1$ , increasing  $k$  will widen the optimal weight interval (standing for the interval about  $\omega = d_s$  over which the mapping process attempts to use the corresponding optimal weight, see [26]), but narrow the transition intervals (standing for the intervals about  $\omega = \text{CFS}_s$  or  $\omega = \overline{\text{CFS}}_s$  over which the mapping results satisfying  $0 < g(\omega) < d_s$  or  $d_s < g(\omega) < 1$  respectively);
- (2) for given  $k$ ,  $\text{CFS}_1$  and  $\delta_1$ , decreasing  $A$  will widen the optimal weight interval and narrow the transition intervals;
- (3) for given  $k$ ,  $A$  and  $\delta_1$ , increasing  $\text{CFS}_1$  will narrow the optimal weight interval and maintain the width of the transition intervals unchanged;
- (4) for given  $k$ ,  $A$  and  $\text{CFS}_1$ , decreasing  $\delta_1$  will widen the optimal weight interval and narrow the transition intervals.

For smooth problems, a wider optimal weight interval will bring the scheme closer to the corresponding linear upwind scheme leading to lower dissipation and higher resolution, and it is apparent from the success of fifth-order WENO-IM(2, 0.1) scheme [12] that widening the optimal weight interval results in better performance. However, an excessive optimal weight interval may lead to possible over-amplification of the contribution from a non-smooth stencil that creates a serious problem when a shock or nonlinear interaction between two shocks, like in the blastwave problem [42], appears in the solution. It would generate numerical oscillations, even produce negative density and pressure, due to the mapping. This would be likely to happen in various mapped WENO schemes as their mappings push the nonlinear weights over the optimal weight intervals to be the ideal weights. Taking the WENO-ACM scheme as an example, its mapping functions compress all the large and small weights  $\omega_s \in (\text{CFS}_s + \delta_s, \overline{\text{CFS}}_s - \delta_s)$  closer together toward the ideal weights with less numerical dissipation but higher risk of generating numerical oscillations. It is easy to verify from Eq. (3.1) that the maximum optimal weight interval of the mapping function  $(g^{\text{ACM}})_s(\omega)$  is determined by  $\text{CFS}_s$  and we can also find this intuitively from Fig. 1. Therefore, in the WENO-ACM scheme, an optimal  $\text{CFS}_s$  should be desired that helps to obtain solutions with less numerical dissipation leading to higher resolution and prevent the scheme from generating numerical oscillations in the meantime. In other words, we can treat  $\text{CFS}_s$  as a tunable parameter: a larger  $\text{CFS}_s$  makes the performance of the WENO-ACM

scheme get closer to that of the WENO-M or even WENO-JS scheme, and a smaller  $CFS_s$  makes the performance of the WENO-ACM scheme get closer to that of the linear upwind scheme. It is difficult and unsolved here yet to determine the optimal  $CFS_s$  theoretically, but after extensive numerical tests, we find that  $CFS_s = d_s/10$  should be a good choice. We will give a more detailed discussion about this through the blastwave problem as shown in Example 4.8 in Subsection 4.2.

Clearly, the new mapping method uses only one assignment operation when  $\omega$  is out of the transition intervals. However, when  $\omega$  is on the transition intervals, the new mapping method uses multiple multiplication and division as the existing mapped WENO schemes (like the WENO-M and WENO-PM6 schemes) do in their mapping processes on the whole interval of  $\omega \in [0, 1]$ . Thus, narrower transition intervals will introduce fewer mathematical operations during the mapping process of the WENO-ACM scheme so that the CPU time will decrease significantly. Furthermore, our tests have shown that narrower transition intervals will not bring any adverse effects on the resolution and convergence rate of accuracy of the WENO-ACM scheme. Actually, the WENO-ACM scheme still performs very well even if the transition intervals decrease to near zero, and this will be verified in the calculation results of the numerical experiments in Section 4.

### 3.2. The rate of convergence

Before giving Theorem 3.2 to show the convergence property of the WENO-ACM scheme, we state the following two necessary lemmas explicitly that can be found in the statement in [12, pp. 456-457] and [20, p. 565].

**Lemma 3.1.** *The sufficient condition for the  $(2r - 1)$ -th order WENO scheme to achieve the optimal order of accuracy is*

$$\omega_s - d_s = \mathcal{O}((\Delta x)^r), \quad r = 2, \dots, 9, \quad s = 0, \dots, r - 1.$$

**Lemma 3.2.** *For  $n_{cp} \leq r - 1$ , the weights  $\omega_s^{JS}$  in the  $(2r - 1)$ -th order WENO-JS scheme satisfy*

$$\omega_s^{JS} - d_s = \mathcal{O}((\Delta x)^{r-1-n_{cp}}), \quad r = 2, 3, \dots, 9,$$

*then, the convergence order is*

$$r_c = \begin{cases} 2r - 1, & \text{if } n_{cp} = 0, \\ 2r - 2 - n_{cp}, & \text{if } n_{cp} = 1, 2, \dots, r - 1. \end{cases}$$

**Theorem 3.2.** *For  $n_{cp} < r - 1$ , the  $(2r - 1)$ -th order WENO-ACM scheme can achieve the optimal convergence rate of accuracy if the new mapping function  $(g^{ACM})_s(\omega)$  is applied to weights of the  $(2r - 1)$ -th order WENO-JS scheme.*

One can easily prove Theorem 3.2 by employing the Taylor series analysis and using Theorem 3.1, Lemmas 3.1 and 3.2, and the detailed proof process is almost identical to the one in [20].

## 4. Numerical experiments

In this section, we compare the calculation results of the WENO-ACM scheme with those of several typical WENO schemes, i.e., the classic WENO-JS scheme, the original WENO-M scheme, and the WENO-PM6 scheme that generates less dissipation and better resolution. The value of  $\epsilon$  is chosen to be  $10^{-40}$  in all considered schemes. Several problems with different initial and boundary conditions are used for the comparison, such as the one-dimensional linear advection equation, one- and two-dimensional Euler equations of compressible gas dynamics. The one-dimensional linear advection equation is computed for the accuracy test, and the numerical experiments of the Euler equations are conducted to demonstrate the performance of the WENO-ACM scheme in solving hyperbolic systems. Meanwhile, the computational costs of the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes in simulating the benchmark problems of two-dimensional Euler equations are compared by the CPU timing per Runge-Kutta step.

In all the numerical experiments below, the global Lax-Friedrichs numerical flux is employed, and unless otherwise noted, the parameters in the WENO-ACM scheme are chosen to be  $k = 2$ ,  $A = 20$ ,  $\delta_s = 1.0e-6$ ,  $CFS_s = 0.1d_s$ .

### 4.1. One-dimensional linear advection equation

**Example 4.1.** We solve the one-dimensional linear advection equation  $u_t + u_x = 0$  with the periodic boundary conditions and the following initial condition [12]:

$$u(x, 0) = \sin(\pi x). \quad (4.1)$$

The computational domain is  $x \in (-1; 1)$  and it is the same for the rest examples in this subsection. It is noted that although the initial condition in Eq. (4.1) has two first-order critical points, their first and third derivatives vanish simultaneously. To ensure that the error for the overall scheme is a measure of the spatial convergence only, and note that we consider only the fifth-order methods here, we set the CFL number to be  $(\Delta x)^{2/3}$ . The  $L_1, L_2, L_\infty$  norms of the error are calculated by comparing the numerical solution  $(u_h)_j$  with the exact solution  $u_j^{\text{exact}}$  according to

$$\begin{aligned} L_1 &= h \cdot \sum_j |u_j^{\text{exact}} - (u_h)_j|, \\ L_2 &= \sqrt{h \cdot \sum_j (u_j^{\text{exact}} - (u_h)_j)^2}, \\ L_\infty &= \max_j |u_j^{\text{exact}} - (u_h)_j|, \end{aligned}$$

where  $h = \Delta x$  is the uniform spatial step size.

Table 1 shows the  $L_1, L_2, L_\infty$  errors and convergence orders of various considered WENO schemes for Example 4.1 at output time  $t = 2.0$ . All the schemes can achieve

Table 1: Convergence properties of various schemes as applied to Example 4.1 at output time  $t = 2$  with  $CFL = (\Delta x)^{2/3}$ .

Scheme	$h = \Delta x$	$L_1$ error	$L_1$ order	$L_2$ error	$L_2$ order	$L_\infty$ error	$L_\infty$ order
WENO-JS	0.2	6.18328e-02	-	4.72306e-02	-	4.87580e-02	-
	0.1	2.96529e-03	4.3821	2.42673e-03	4.2826	2.57899e-03	4.2408
	0.05	9.27609e-05	4.9985	7.64332e-05	4.9887	9.05453e-05	4.8320
	0.025	2.89265e-06	5.0031	2.33581e-06	5.0322	2.90709e-06	4.9610
	0.0125	9.03392e-08	5.0009	7.19259e-08	5.0213	8.85753e-08	5.0365
	0.00625	2.82330e-09	4.9999	2.23105e-09	5.0107	2.72458e-09	5.0228
WENO-M	0.2	2.01781e-02	-	1.55809e-02	-	1.47767e-02	-
	0.1	5.18291e-04	5.2829	4.06148e-04	5.2616	3.94913e-04	5.2256
	0.05	1.59422e-05	5.0228	1.25236e-05	5.0193	1.24993e-05	4.9816
	0.025	4.98914e-07	4.9979	3.91875e-07	4.9981	3.91808e-07	4.9956
	0.0125	1.56021e-08	4.9990	1.22541e-08	4.9991	1.22538e-08	4.9988
	0.00625	4.88356e-10	4.9977	3.83568e-10	4.9976	3.83541e-10	4.9977
WENO-PM6	0.2	1.74869e-02	-	1.35606e-02	-	1.27577e-02	-
	0.1	5.02923e-04	5.1198	3.95215e-04	5.1006	3.94515e-04	5.0151
	0.05	1.59130e-05	4.9821	1.25010e-05	4.9825	1.24960e-05	4.9805
	0.025	4.98858e-07	4.9954	3.91831e-07	4.9957	3.91795e-07	4.9952
	0.0125	1.56020e-08	4.9988	1.22541e-08	4.9989	1.22538e-08	4.9988
	0.00625	4.88355e-10	4.9977	3.83568e-10	4.9976	3.83543e-10	4.9977
WENO-ACM	0.2	1.52184e-02	-	1.19442e-02	-	1.17569e-02	-
	0.1	5.02844e-04	4.9196	3.95138e-04	4.9178	3.94406e-04	4.8977
	0.05	1.59130e-05	4.9818	1.25010e-05	4.9822	1.24960e-05	4.9801
	0.025	4.98858e-07	4.9954	3.91831e-07	4.9957	3.91795e-07	4.9952
	0.0125	1.56020e-08	4.9988	1.22541e-08	4.9989	1.22538e-08	4.9988
	0.00625	4.88355e-10	4.9977	3.83568e-10	4.9976	3.83543e-10	4.9977

the optimal convergence orders. In terms of accuracy, the WENO-M, WENO-PM6 and WENO-ACM schemes provide more accurate numerical solutions than the results of the WENO-JS scheme in general. Besides, it is noted that, in terms of the  $L_1$ -norm error, the WENO-ACM scheme gives almost equally accurate numerical solutions as those of the WENO-PM6 scheme which are more accurate than the numerical solutions of the WENO-M scheme.

We use this example to test the efficiency of various WENO schemes. Here we also consider the efficiency of the MWENO5 scheme [46], which is a fifth-order mapped WENO scheme constructed by obeying the similar principles proposed by Henrick *et al.* [20] and employing the ‘Type (i)’ mapping function proposed by Zhu *et al.* [46] very recently. In Fig. 2, we draw the graphs for the CPU time versus the computing errors (we only present the results of the  $L_1$ - and  $L_2$ - norm errors here just for the sake of brevity in the presentation, hereinafter the same). Clearly, the scheme with the

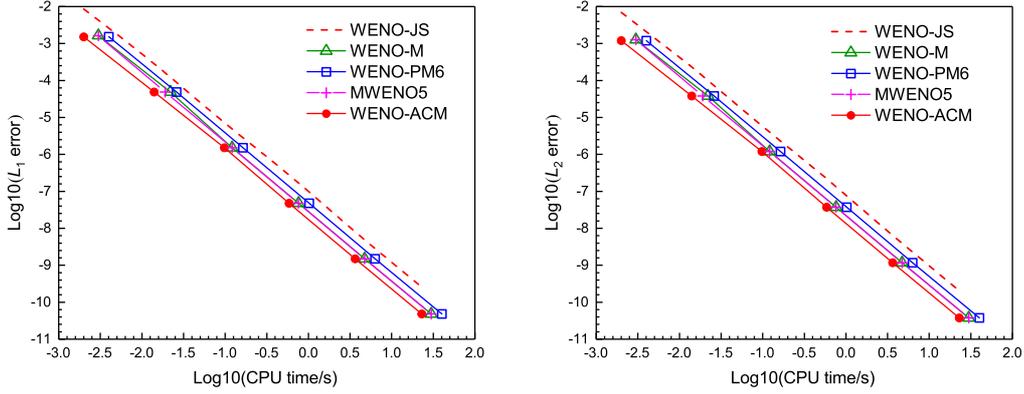


Figure 2: Comparison of various WENO schemes for Example 4.1 in CPU time and computing errors.

least CPU time to achieve the same errors would have the best efficiency. Therefore, the lower line has higher efficiency. Then, from Fig. 2, we can easily observe that the WENO-ACM scheme has the best efficiency. We also find that the MWENO5 scheme performs almost identical to the WENO-M scheme, and they have the second best efficiency, followed by the WENO-PM6 scheme. Unsurprisingly, the WENO-JS scheme has the lowest efficiency.

**Example 4.2.** We solve the one-dimensional linear advection equation  $u_t + u_x = 0$  with the periodic boundary conditions and the following initial condition [20]:

$$u(x, 0) = \sin\left(\pi x - \frac{\sin(\pi x)}{\pi}\right). \quad (4.2)$$

As mentioned earlier, the CFL number is set to be  $(\Delta x)^{2/3}$ . It is easy to verify that the particular initial condition Eq. (4.2) has two first-order critical points, which both have a non-vanishing third derivative.

The  $L_1, L_2, L_\infty$  errors and convergence orders of various considered WENO schemes at output time  $t = 2.0$  are shown in Table 2. From Table 2, we can observe that the WENO-M, WENO-PM6 and WENO-ACM schemes can retain the optimal orders even in the presence of critical points. Moreover, in terms of accuracy, the WENO-ACM scheme provides the equally accurate results as those of the WENO-M and WENO-PM6 schemes, which are much more accurate than the solutions of the WENO-JS scheme whose  $L_\infty$  convergence rate of accuracy drops by almost 2 orders leading to an overall accuracy loss shown with  $L_1$  and  $L_2$  convergence orders.

We also use this example to test the efficiency of various WENO schemes and the MWENO5 scheme [46] is taken into account once again. Fig. 3 shows that the WENO-ACM scheme costs the least CPU time among all considered schemes, so it has the best efficiency. Again, the MWENO5 scheme performs almost identical to the WENO-M scheme, and they have the second best efficiency, followed by the WENO-PM6 scheme. As expected, the WENO-JS scheme has the lowest efficiency.

Table 2: Convergence properties of various schemes as applied to Example 4.2 at output time  $t = 2$  with  $CFL = (\Delta x)^{2/3}$ .

Scheme	$h = \Delta x$	$L_1$ error	$L_1$ order	$L_2$ error	$L_2$ order	$L_\infty$ error	$L_\infty$ order
WENO-JS	0.2	1.24488e-01	-	1.09463e-01	-	1.24471e-01	-
	0.1	1.01260e-02	3.6199	8.72198e-03	3.6496	1.43499e-02	3.1167
	0.05	7.22169e-04	3.8096	6.76133e-04	3.6893	1.09663e-03	3.7099
	0.025	3.42286e-05	4.3991	3.63761e-05	4.2162	9.02485e-05	3.6030
	0.0125	1.58510e-06	4.4326	2.29598e-06	3.9858	8.24022e-06	3.4531
	0.00625	7.95517e-08	4.3165	1.68304e-07	3.7700	8.31702e-07	3.3085
WENO-M	0.2	7.53259e-02	-	6.39017e-02	-	7.49250e-02	-
	0.1	3.70838e-03	4.3443	3.36224e-03	4.2484	5.43666e-03	3.7847
	0.05	1.45082e-04	4.6758	1.39007e-04	4.5962	2.18799e-04	4.6350
	0.025	4.80253e-06	4.9169	4.52646e-06	4.9406	6.81451e-06	5.0049
	0.0125	1.52120e-07	4.9805	1.42463e-07	4.9897	2.14545e-07	4.9893
	0.00625	4.77083e-09	4.9948	4.45822e-09	4.9980	6.71080e-09	4.9987
WENO-PM6	0.2	9.51313e-02	-	7.83600e-02	-	9.32356e-02	-
	0.1	4.82173e-03	4.3023	4.29510e-03	4.1894	5.91037e-03	3.9796
	0.05	1.55428e-04	4.9552	1.43841e-04	4.9001	2.09540e-04	4.8180
	0.025	4.87327e-06	4.9952	4.54036e-06	4.9855	6.83270e-06	4.9386
	0.0125	1.52750e-07	4.9956	1.42488e-07	4.9939	2.14532e-07	4.9932
	0.00625	4.77729e-09	4.9988	4.45807e-09	4.9983	6.71079e-09	4.9986
WENO-ACM	0.2	8.75629e-02	-	6.98131e-02	-	7.91292e-02	-
	0.1	4.39527e-03	4.3163	4.02909e-03	4.1150	5.89045e-03	3.7478
	0.05	1.52219e-04	4.8517	1.42172e-04	4.8247	2.09893e-04	4.8107
	0.025	4.86436e-06	4.9678	4.53770e-06	4.9695	6.83017e-06	4.9416
	0.0125	1.52735e-07	4.9931	1.42486e-07	4.9931	2.14533e-07	4.9926
	0.00625	4.77728e-09	4.9987	4.45807e-09	4.9983	6.71079e-09	4.9986

**Example 4.3.** We solve the one-dimensional linear advection equation  $u_t + u_x = 0$  with the following initial condition:

$$u(x, 0) = \begin{cases} \frac{1}{6} [G(x, \beta, z - \hat{\delta}) + 4G(x, \beta, z) + G(x, \beta, z + \hat{\delta})], & x \in [-0.8, -0.6], \\ 1, & x \in [-0.4, -0.2], \\ 1 - |10(x - 0.1)|, & x \in [0.0, 0.2], \\ \frac{1}{6} [F(x, \alpha, a - \hat{\delta}) + 4F(x, \alpha, a) + F(x, \alpha, a + \hat{\delta})], & x \in [0.4, 0.6], \\ 0, & \text{otherwise,} \end{cases} \quad (4.3)$$

where

$$G(x, \beta, z) = e^{-\beta(x-z)^2}, \quad F(x, \alpha, a) = \sqrt{\max(1 - \alpha^2(x-a)^2, 0)},$$

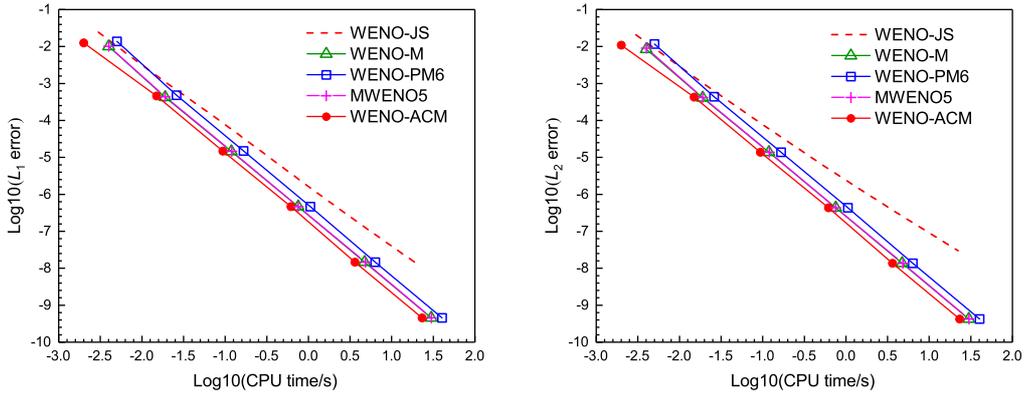


Figure 3: Comparison of various WENO schemes for Example 4.2 in CPU time and computing errors.

and the constants are  $z = -0.7, \hat{\delta} = 0.005, \beta = \log 2 / (36\hat{\delta}^2), a = 0.5$  and  $\alpha = 10$ . The periodic boundary conditions are used in the two directions and the CFL number is set to be 0.1. This example is obtained from [23] and it consists of a Gaussian, a square wave, a sharp triangle and a semi-ellipse.

Tables 3 and 4 show the  $L_1, L_2, L_\infty$  errors and convergence orders of various considered WENO schemes at output times  $t = 2$  and  $t = 2000$ . At output time  $t = 2$ , we find: (1) for all schemes, the  $L_1$  and  $L_2$  orders are approximately 1.0 and 0.4 to 0.5, respectively, and the  $L_\infty$  orders are all negative; (2) in terms of accuracy, the WENO-ACM scheme provides the most accurate results closely followed by the WENO-PM6 and WENO-M schemes, which are more accurate than that of the WENO-JS scheme. At output time  $t = 2000$ , we find: (1) for the WENO-JS and WENO-M schemes, the  $L_1, L_2$  orders decrease to very small values and even become negative; (2) however, for

Table 3: Convergence properties of various schemes as applied to Example 4.3 at output time  $t = 2$  with CFL = 0.1.

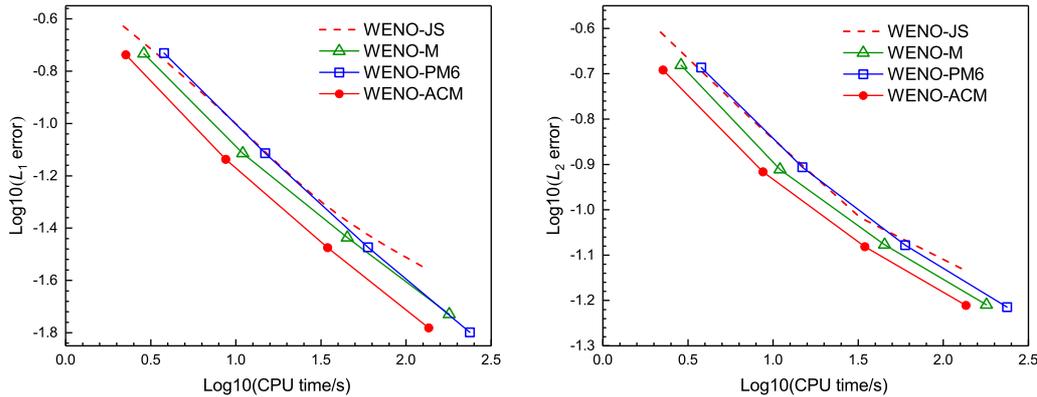
Scheme	$h = \Delta x$	$L_1$ error	$L_1$ order	$L_2$ error	$L_2$ order	$L_\infty$ error	$L_\infty$ order
WENO-JS	0.01	6.30497e-02	-	1.08621e-01	-	4.09733e-01	-
	0.005	2.81654e-02	1.2103	7.71111e-02	0.4943	4.19594e-01	-0.0343
	0.0025	1.41364e-02	0.9945	5.69922e-02	0.4362	4.28463e-01	-0.0302
WENO-M	0.01	4.77201e-02	-	9.53073e-02	-	3.94243e-01	-
	0.005	2.23407e-02	1.0949	6.91333e-02	0.4632	4.05856e-01	-0.0419
	0.0025	1.11758e-02	0.9993	5.09232e-02	0.4411	4.16937e-01	-0.0389
WENO-PM6	0.01	4.66681e-02	-	9.45566e-02	-	3.96866e-01	-
	0.005	2.13883e-02	1.1256	6.82948e-02	0.4694	4.06118e-01	-0.0332
	0.0025	1.06477e-02	1.0063	5.03724e-02	0.4391	4.15277e-01	-0.0322
WENO-ACM	0.01	4.45059e-02	-	9.24356e-02	-	3.92505e-01	-
	0.005	2.03633e-02	1.1280	6.69718e-02	0.4649	4.03456e-01	-0.0397
	0.0025	1.02139e-02	0.9954	4.95672e-02	0.4342	4.13217e-01	-0.0345

Table 4: Convergence properties of various schemes as applied to Example 4.3 at output time  $t = 2000$  with CFL = 0.1.

Scheme	$h = \Delta x$	$L_1$ error	$L_1$ order	$L_2$ error	$L_2$ order	$L_\infty$ error	$L_\infty$ order
WENO-JS	0.01	6.12899e-01	-	5.08726e-01	-	7.99265e-01	-
	0.005	5.99215e-01	0.0326	5.01160e-01	0.0216	8.20493e-01	-0.0378
	0.0025	5.50158e-01	0.1232	4.67585e-01	0.1000	8.14650e-01	0.0103
WENO-M	0.01	3.81597e-01	-	3.59205e-01	-	6.89414e-01	-
	0.005	3.25323e-01	0.2302	3.12970e-01	0.1988	6.75473e-01	0.0295
	0.0025	3.48528e-01	-0.0994	3.24373e-01	-0.0516	6.25645e-01	0.1106
WENO-PM6	0.01	2.17323e-01	-	2.28655e-01	-	5.63042e-01	-
	0.005	1.05197e-01	1.0467	1.47518e-01	0.6323	5.04977e-01	0.1570
	0.0025	4.47030e-02	1.2347	9.34250e-02	0.6590	4.71368e-01	0.0994
WENO-ACM	0.01	2.21313e-01	-	2.28433e-01	-	5.36234e-01	-
	0.005	1.06583e-01	1.0541	1.46401e-01	0.6418	5.03925e-01	0.0897
	0.0025	4.76305e-02	1.1620	9.40930e-02	0.6378	5.15924e-01	-0.0339

the WENO-ACM and WENO-PM6 schemes, their  $L_1$  orders are maintained at approximately 1.0, and their  $L_2$  orders increase to approximately 0.6 to 0.7; (3) for all schemes, the  $L_\infty$  orders are very small and even become negative. Overall, for both short and long output times, the WENO-ACM scheme performs as well as the WENO-PM6 scheme in calculating this problem that includes various discontinuities.

We consider the cases of  $t = 20$  and  $t = 200$  respectively to test the efficiency of various considered WENO schemes for short and long output time calculations. Figs. 4 and 5 show that the WENO-ACM scheme uses less CPU time than all the other considered WENO schemes to get the same quantities of computing errors for both short and long output times, so it is the most efficient one of all considered WENO schemes on solving this one-dimensional linear advection problem with the initial condition including discontinuities.

Figure 4: Comparison of various WENO schemes for Example 4.3 with output time  $t = 20$  in CPU time and computing errors.

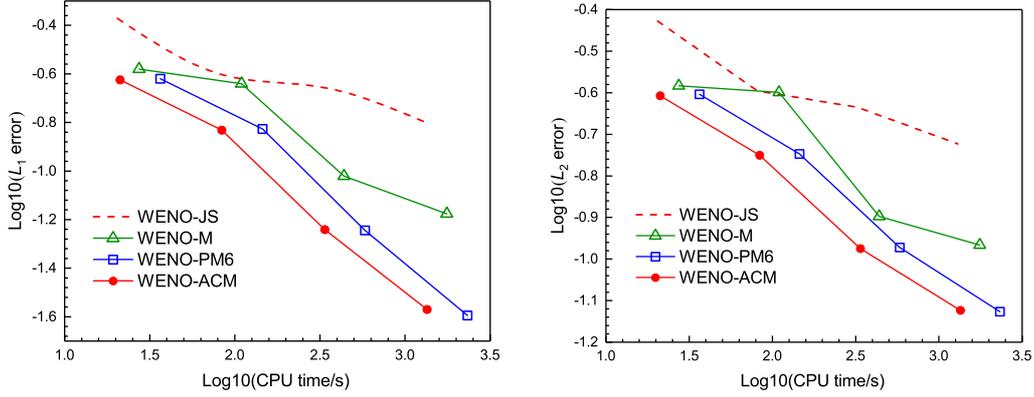


Figure 5: Comparison of various WENO schemes for Example 4.3 with output time  $t = 200$  in CPU time and computing errors.

**Example 4.4.** We solve the one-dimensional linear advection equation  $u_t + u_x = 0$  with the periodic boundary conditions and the following initial condition [11]:

$$u(x, 0) = \sin^9(\pi x). \quad (4.4)$$

Again, the CFL number is set to be  $(\Delta x)^{2/3}$ . It is easy to verify that the initial condition in Eq. (4.4) has high-order critical points.

Table 5 shows the  $L_1, L_2, L_\infty$  errors of various considered WENO schemes at several output times with a uniform mesh size of  $\Delta x = 1/100$ . Clearly, at short output times, the WENO-ACM scheme achieves similar results as those of the WENO-M and WENO-PM6 schemes. However, at long output times, the numerical solutions computed by the WENO-M scheme are far less accurate than that of the WENO-PM6 scheme, while the solutions of the WENO-ACM scheme are still as accurate as results of the WENO-PM6 scheme. Another observation is that the WENO-M, WENO-PM6 and WENO-ACM schemes all provide more accurate numerical solutions than the WENO-JS scheme.

Fig. 6 shows the performance of the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 1000$  with a uniform mesh size of  $\Delta x = 1/100$ . Clearly the WENO-ACM and WENO-PM6 schemes give the highest resolution followed by the WENO-M scheme whose resolution decreases significantly, and the WENO-JS scheme shows the lowest resolution.

We consider the cases of  $t = 10$  and  $t = 100$  respectively to test the efficiency of various considered WENO schemes for short and long output time calculations. Figs. 7 and 8 show that the WENO-ACM scheme consumes the least CPU time among all considered WENO schemes to obtain the same quantities of computing errors for both short and long output times, so it is the most efficient one of all considered WENO schemes on solving this one-dimensional linear advection problem with the initial condition including high-order critical points.

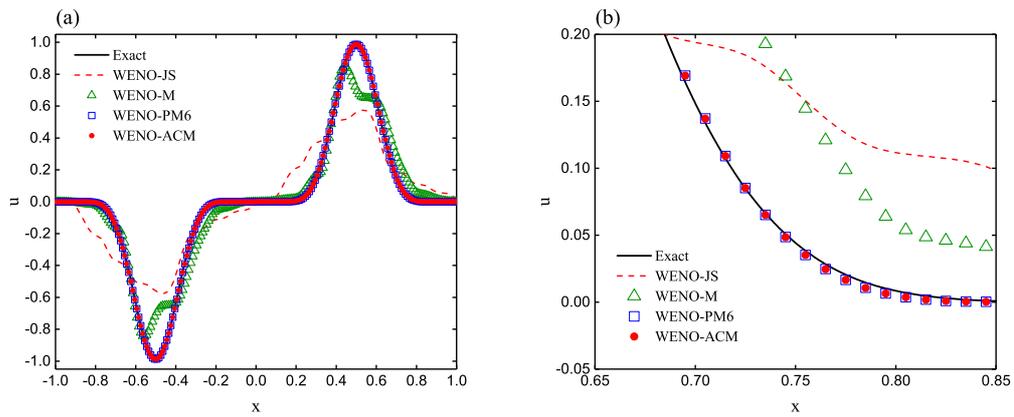


Figure 6: Performance of the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes for Example 4.4 at output time  $t = 1000$  with a uniform mesh size of  $\Delta x = 1/100$ .

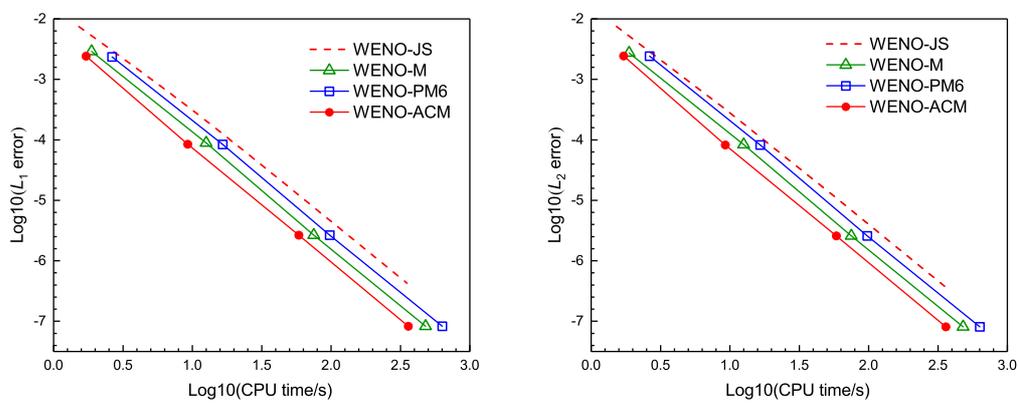


Figure 7: Comparison of various WENO schemes for Example 4.4 with output time  $t = 10$  in CPU time and computing errors.

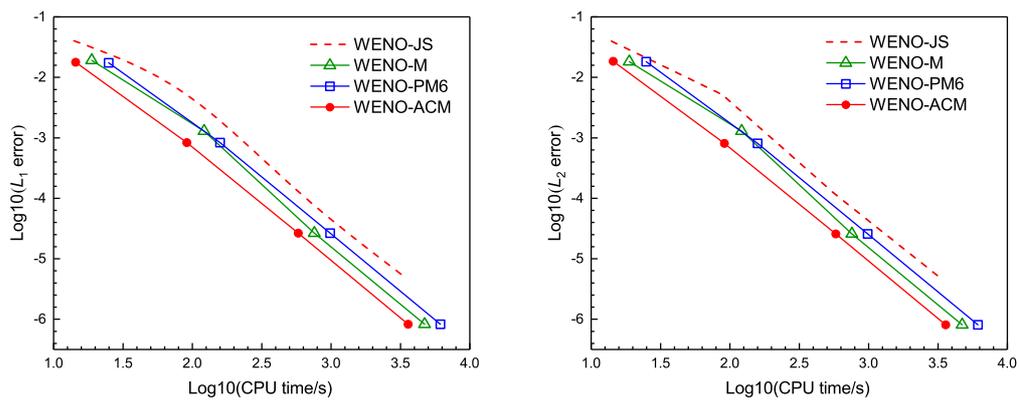


Figure 8: Comparison of various WENO schemes for Example 4.4 with output time  $t = 100$  in CPU time and computing errors.

Table 5: The  $L_1$ ,  $L_2$ ,  $L_\infty$  errors of various schemes as applied to Example 4.4 at several output times with  $\Delta x = 1/100$ ,  $\text{CFL} = (\Delta x)^{2/3}$ .

Time	WENO-JS			WENO-M		
	$L_1$ error	$L_2$ error	$L_\infty$ error	$L_1$ error	$L_2$ error	$L_\infty$ error
$t = 1$	3.87826e-05	3.62689e-05	6.69118e-05	8.84565e-06	8.31248e-06	1.38461e-05
$t = 10$	3.86931e-04	3.52611e-04	5.36940e-04	8.90890e-05	8.32089e-05	1.38348e-04
$t = 30$	1.17988e-03	1.06511e-03	1.58134e-03	2.73430e-04	2.51737e-04	4.13887e-04
$t = 50$	2.05488e-03	1.84782e-03	2.69500e-03	4.81901e-04	4.39983e-04	6.87879e-04
$t = 100$	5.42288e-03	5.17716e-03	1.20056e-02	1.29154e-03	1.28740e-03	3.32665e-03
$t = 200$	2.35657e-02	2.68753e-02	6.47820e-02	5.74021e-03	7.66721e-03	2.37125e-02
$t = 500$	1.55650e-01	1.46859e-01	2.57663e-01	4.89290e-02	6.23842e-02	1.78294e-01
$t = 1000$	2.91359e-01	2.66692e-01	4.44664e-01	1.34933e-01	1.46524e-01	3.17199e-01
Time	WENO-PM6			WENO-ACM		
	$L_1$ error	$L_2$ error	$L_\infty$ error	$L_1$ error	$L_2$ error	$L_\infty$ error
$t = 1$	8.52448e-06	8.22944e-06	1.38389e-05	8.43356e-06	8.20366e-06	1.38389e-05
$t = 10$	8.40259e-05	8.19676e-05	1.38205e-04	8.42873e-05	8.19107e-05	1.38205e-04
$t = 30$	2.51117e-04	2.45084e-04	4.13397e-04	2.52378e-04	2.45090e-04	4.13398e-04
$t = 50$	4.17588e-04	4.07311e-04	6.86969e-04	4.19825e-04	4.07429e-04	6.86983e-04
$t = 100$	8.30374e-04	8.09152e-04	1.36410e-03	8.35747e-04	8.09679e-04	1.36404e-03
$t = 200$	1.63963e-03	1.59697e-03	2.68938e-03	1.65557e-03	1.59929e-03	2.68955e-03
$t = 500$	3.88864e-03	3.83159e-03	6.45650e-03	3.95849e-03	3.84802e-03	6.45564e-03
$t = 1000$	7.17606e-03	7.19008e-03	1.21637e-02	7.24723e-03	7.21626e-03	1.21593e-02

## 4.2. One-dimensional Euler system

In this subsection, we calculate the one-dimensional Euler system of gas dynamics with different initial and boundary conditions. The one-dimensional Euler system is given by the following strong conservation form of mass, momentum and energy:

$$\begin{aligned}
\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} &= 0, \\
\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} &= 0, \\
\frac{\partial E}{\partial t} + \frac{\partial(uE + up)}{\partial x} &= 0,
\end{aligned} \tag{4.5}$$

where  $\rho, u, p$  and  $E$  are the density, velocity, pressure and total energy, respectively. The Euler system Eq. (4.5) is closed by the equation of state for an ideal polytropic gas, which is given by

$$p = (\gamma - 1) \left( E - \frac{1}{2} \rho u^2 \right),$$

where  $\gamma$  is the ratio of specific heat, and we use  $\gamma = 1.4$  in this paper. The finite volume version of the characteristic-wise one-dimensional WENO procedure is employed, and we refer to [43] for details. In all examples of this subsection, the CFL number is set to be 0.5.

**Example 4.5** (Sod's shock tube problem). We consider the Sod's shock tube problem [35], specified by the following initial condition:

$$(\rho, u, p)(x, 0) = \begin{cases} (1.0, 0.0, 1.0), & x \in [0.0, 0.5], \\ (0.125, 0.0, 0.1), & x \in [0.5, 1.0]. \end{cases}$$

The transmissive boundary conditions are used in two directions.

Fig. 9 presents the density profiles computed by the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 0.25$  with a uniform mesh size of  $N = 200$ . We observe that the WENO-M, WENO-PM6 and WENO-ACM schemes capture sharper discontinuity compared to the WENO-JS scheme, and the WENO-ACM scheme gives slightly better resolution than the WENO-PM6 and WENO-M schemes.

**Example 4.6** (Riemann problem of Lax). The second considered problem modeled by the one-dimensional Euler system Eq. (4.5) is the Lax's problem [24], specified by the following initial condition:

$$(\rho, u, p)(x, 0) = \begin{cases} (0.445, 0.698, 3.528), & x \in [-5, 0], \\ (0.500, 0.000, 0.571), & x \in [0, 5]. \end{cases}$$

The transmissive boundary conditions are used at  $x = \pm 5$ , and the uniform cell number is chosen to be  $N = 200$ .

Fig. 10 presents the density profiles computed by the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 1.3$ . It is observed that the WENO-M, WENO-PM6 and WENO-ACM schemes have higher resolution than the WENO-JS

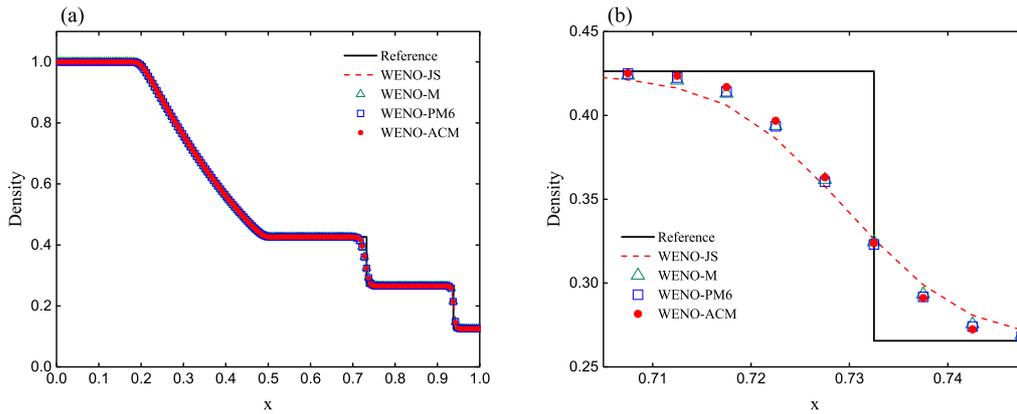


Figure 9: The density profiles of the Sod's shock tube problem.

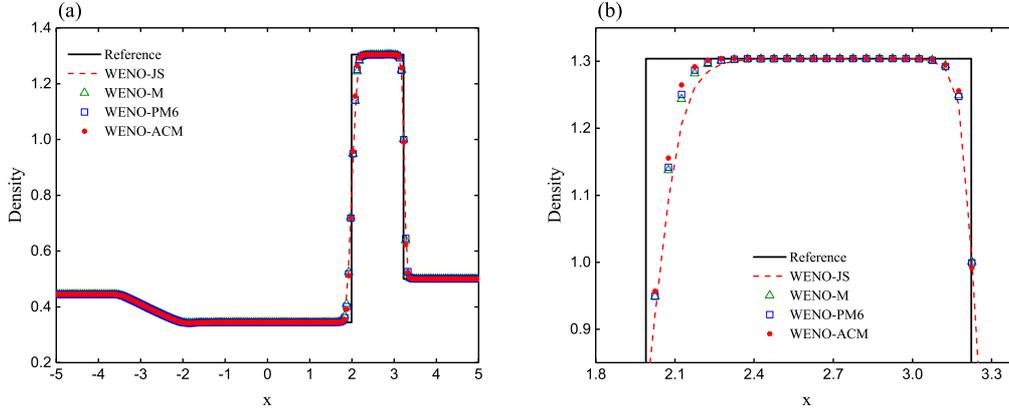


Figure 10: The density profiles of the Riemann problem of Lax.

scheme near the discontinuity. Also, if we take a closer look at  $x \in (1.8, 3.3)$ , it demonstrates that the WENO-ACM scheme performs slightly better than the WENO-PM6 and WENO-M schemes.

**Example 4.7** (Mach 3 shock-density wave interaction). We solve the Mach 3 shock-density wave interaction [34], whose solution consists of several shocklets and fine-scale structures that are located behind a main right-going shock [5]. Its boundaries at  $x = \pm 5$  are specified by zero-gradient boundary condition, and its initial condition is given by

$$(\rho, u, p)(x, 0) = \begin{cases} (3.857143, 2.629369, 10.333333), & x \in [-5.0, -4.0], \\ (1.0 + 0.2 \sin(5x), 0, 1), & x \in [-4.0, 5.0]. \end{cases}$$

Fig. 11 gives the comparison on density between the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 1.8$  with the uniform cell num-

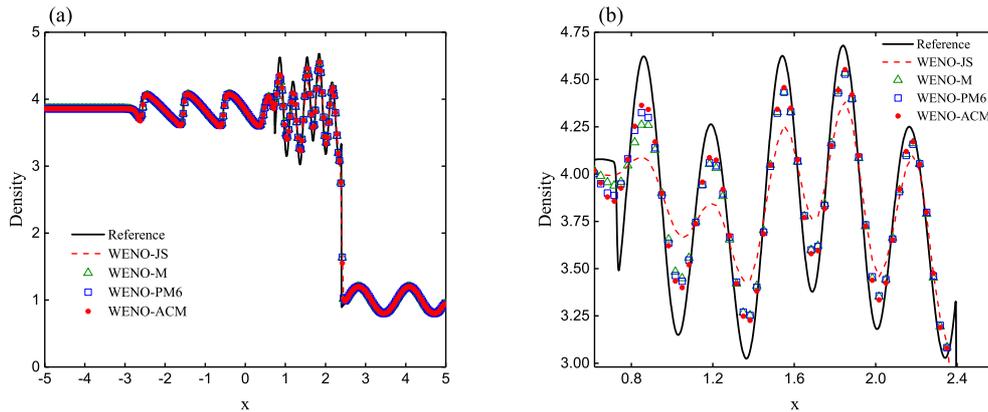


Figure 11: The density profiles of the Mach 3 shock-density wave interaction.

ber  $N = 300$ . The solid line is the reference solution using the WENO-JS scheme with  $N = 10000$ . The WENO-M, WENO-PM6 and WENO-ACM schemes capture much more fine-scale structures of the solution than the WENO-JS scheme. Furthermore, the WENO-ACM scheme shows the best description near shocklets and high-frequency waves behind the main shock.

**Example 4.8** (Woodward-Colella interacting blastwaves). We solve the standard test problem first used by Woodward and Colella [42]. The reflective boundary conditions are used at  $x = 0, 1$ , and the initial condition is set to be

$$(\rho, u, p)(x, 0) = \begin{cases} (1, 0, 1000), & x \in [0, 0.1), \\ (1, 0, 0.01), & x \in [0.1, 0.9), \\ (1, 0, 100), & x \in [0.9, 1.0]. \end{cases}$$

In order to test the robustness of the WENO-ACM scheme as discussed in Subsection 3.1.2, we test this blastwave problem by using the WENO-ACM scheme with more different  $CFS_s$ , that is,  $CFS_s = 0.001d_s, 0.01d_s, 0.095d_s, 0.099d_s, 0.0999d_s, 0.1d_s, 0.3d_s, 0.5d_s, 0.7d_s, 0.9d_s$ , as well as the considered WENO schemes used in previous examples. Fig. 12 shows the density profiles at output time  $t = 0.038$  with the uniform cell number  $N = 400$ . The reference solution is calculated by using the WENO-JS scheme with  $N = 10000$ .

As expected, when  $CFS_s$  gets too small, like  $CFS_s < 0.1d_s$ , the solutions have blown-up. In other words, the effect of the parameter  $CFS_s$  in the WENO-ACM scheme for solving time-dependent PDEs, especially for the robustness of the WENO-ACM scheme, is nonnegligible. In addition, for the computing cases when  $CFS_s \geq 0.1d_s$ , the solutions have not blown-up. From Fig. 12, we can see these phenomena very clear that: (1) when  $CFS_s$  gets larger and  $CFS_s \geq 0.1d_s$ , the WENO-ACM scheme gives solution with more numerical dissipation and lower resolution; (2) when  $CFS_s = 0.3d_s$ , the WENO-ACM scheme gives solution close to that of the WENO-PM6 scheme whose solution is comparable with that of the WENO-M scheme; (3) when  $CFS_s = 0.5d_s$ , the WENO-ACM scheme gives solution close to that of the WENO-JS scheme; (4) when  $CFS_s = 0.7d_s, 0.9d_s$ , the WENO-ACM scheme gives solutions with lower resolution than that of the WENO-JS scheme whose solution shows lowest resolution among those of the WENO-JS, WENO-M and WENO-PM6 schemes; (5) when  $CFS_s = 0.1d_s$ , the WENO-ACM scheme gives solution with the highest resolution among those of all considered schemes, while the solutions blow up when  $CFS_s < 0.1d_s$ .

### 4.3. Two-dimensional Euler system

In this subsection, we calculate the two-dimensional Euler system of gas dynamics with different initial and boundary conditions. The two-dimensional Euler system is given by the following strong conservation form of mass, momentum and energy:

$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u)}{\partial x} + \frac{\partial(\rho v)}{\partial y} = 0, \quad (4.6a)$$

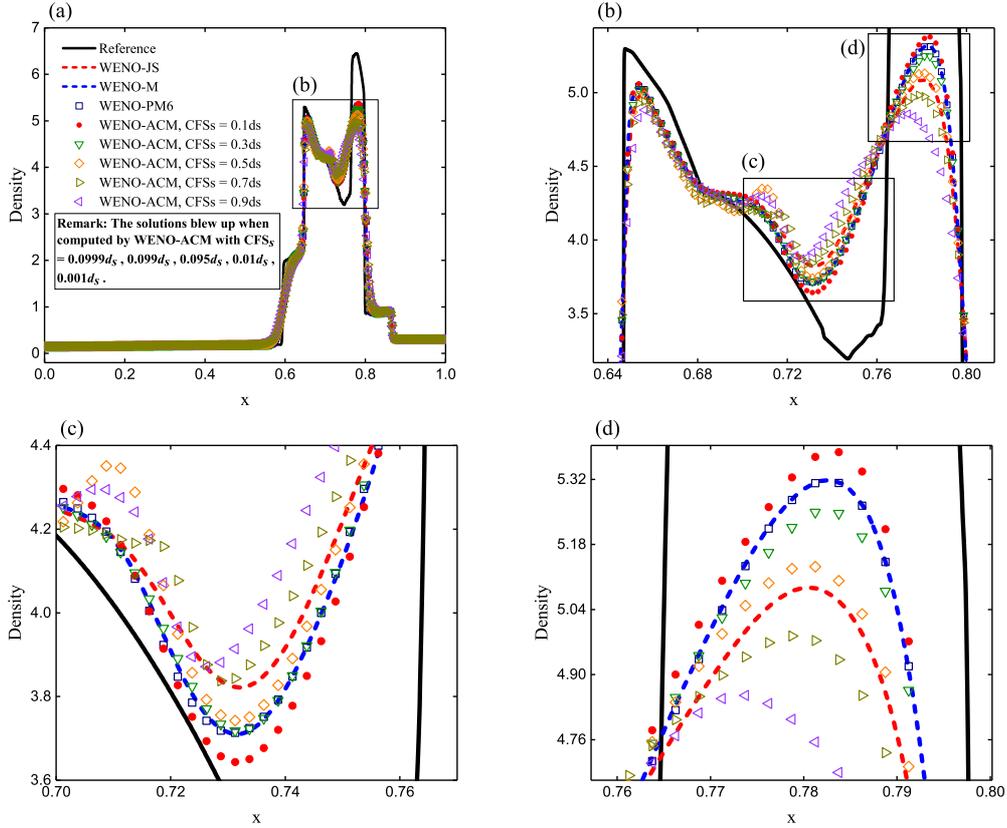


Figure 12: The density profiles of the Woodward-Colella interacting blastwaves.

$$\frac{\partial(\rho u)}{\partial t} + \frac{\partial(\rho u^2 + p)}{\partial x} + \frac{\partial(\rho uv)}{\partial y} = 0, \quad (4.6b)$$

$$\frac{\partial(\rho v)}{\partial t} + \frac{\partial(\rho vu)}{\partial x} + \frac{\partial(\rho v^2 + p)}{\partial y} = 0, \quad (4.6c)$$

$$\frac{\partial E}{\partial t} + \frac{\partial(uE + up)}{\partial x} + \frac{\partial(vE + vp)}{\partial y} = 0, \quad (4.6d)$$

where  $v$  represents the velocity component in the  $y$ -direction, and the other variables are the same as in Eq. (4.5). The equation of state for an ideal polytropic gas used to close the two-dimensional Euler system Eq. (4.6) is given by

$$p = (\gamma - 1) \left( E - \frac{1}{2} \rho (u^2 + v^2) \right).$$

In all numerical examples of this subsection, the CFL number is set to be 0.5.

**Example 4.9** (Shock-vortex interaction). In the shock-vortex interaction problem [7, 28, 30], a left-moving shock wave interacts with a right-moving vortex. The initial

condition is set over the computational domain  $[0, 1] \times [0, 1]$  by

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (\rho_L, u_L, v_L, p_L), & x < 0.5, \\ (\rho_R, u_R, v_R, p_R), & x \geq 0.5, \end{cases}$$

where the left state is taken as  $(\rho_L, u_L, v_L, p_L) = (1, \sqrt{\gamma}, 0, 1)$ , and the right state is given as

$$\begin{aligned} p_R &= 1.3, \quad \rho_R = \rho_L \left( \frac{\gamma - 1 + (\gamma + 1)p_R}{\gamma + 1 + (\gamma - 1)p_R} \right), \\ u_R &= u_L \left( \frac{1 - p_R}{\sqrt{\gamma - 1 + p_R(\gamma + 1)}} \right), \quad v_R = 0. \end{aligned}$$

A vortex given by perturbations  $(\delta\rho, \delta u, \delta v, \delta p)$  is superimposed onto the state when  $x < 0.5$ , and the perturbations are defined by

$$\begin{aligned} \delta\rho &= \frac{\rho_L^2}{(\gamma - 1)p_L} \delta T, & \delta u &= \epsilon \frac{y - y_c}{r_c} e^{\alpha(1-r^2)}, \\ \delta v &= -\epsilon \frac{x - x_c}{r_c} e^{\alpha(1-r^2)}, & \delta p &= \frac{\gamma \rho_L^2}{(\gamma - 1)\rho_L} \delta T, \end{aligned}$$

where  $\epsilon = 0.3, r_c = 0.05, \alpha = 0.204, x_c = 0.25, y_c = 0.5$  and

$$r = \sqrt{\frac{1}{r_c^2}((x - x_c)^2 + (y - y_c)^2)}, \quad \delta T = -\frac{\gamma - 1}{4\alpha\gamma} \epsilon^2 e^{2\alpha(1-r^2)}.$$

We discretize the computational domain with a uniform mesh size of  $400 \times 400$ . The transmissive boundary conditions are used and the output time is taken as  $t = 0.35$ . The initial and final positions of the shock and vortex in density profile, computed using the WENO-ACM scheme, have been shown in Fig. 13. We can easily observe that the WENO-ACM scheme performs very well in capturing the complex structure of the shock and vortex after the interaction. In Fig. 14, we have presented the cross-sectional slices of density profile along the plane  $y = 0.5$ , computed by the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 0.35$ . The reference solution is obtained using the WENO-JS scheme with a uniform mesh size of  $1000 \times 1000$ . As shown in the zoomed-in plots around the shock, the WENO-ACM scheme resolves the shock in a non-oscillatory manner and provides a better resolution than the other considered schemes. The WENO-PM6 and WENO-M schemes perform better than the WENO-JS scheme, while the WENO-PM6 scheme performs slightly better than the WENO-M scheme.

**Example 4.10** (Explosion problem). We solve the explosion problem [36] modeled by the two-dimensional Euler equations (4.6) on the square domain  $[-1, 1] \times [-1, 1]$  in the

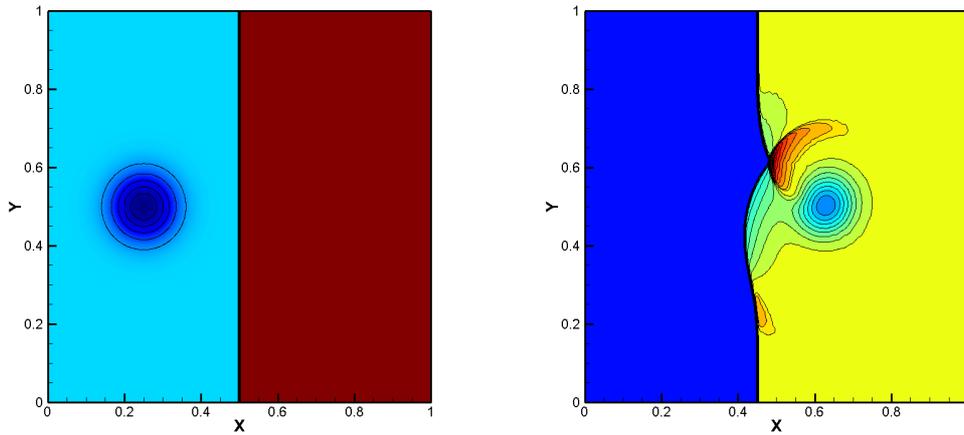


Figure 13: Density plots for the shock-vortex interaction problem using 30 contour lines with range from 0.9 to 1.4, computed using the WENO-ACM scheme at output time  $t = 0.35$  with a uniform mesh size of  $400 \times 400$ . (left: initial density, right: final density).

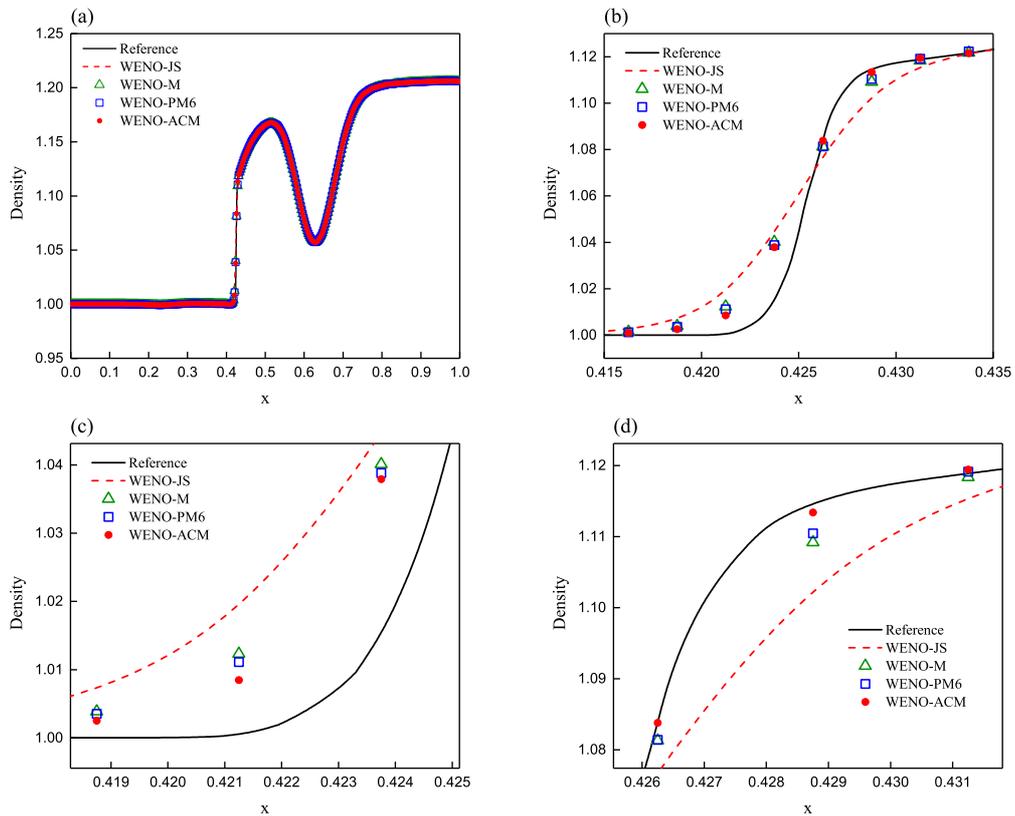


Figure 14: The cross-sectional slices of density plot along the plane  $y = 0.5$ , computed using the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 0.35$  with a uniform mesh size of  $400 \times 400$ .

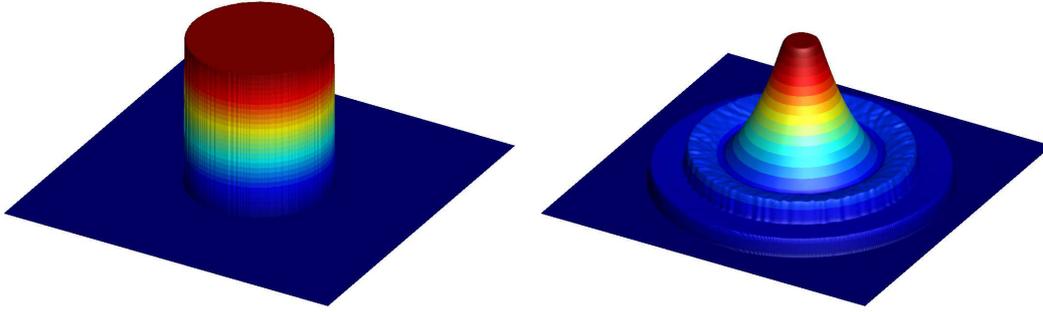


Figure 15: Density plots for the explosion problem with range from 0.125 to 1.0, computed using the WENO-ACM scheme at output time  $t = 0.25$  with a uniform mesh size of  $400 \times 400$ . (left: initial density, right: final density).

$x - y$  plane. It involves two constant states of flow variables separated with a circle of radius  $R = 0.4$  centered at  $(0, 0)$ . The initial condition is given as

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (1, 0, 0, 1), & \text{if } \sqrt{x^2 + y^2} < 0.4, \\ (0.125, 0, 0, 0.1), & \text{otherwise.} \end{cases}$$

Transmissive boundary conditions are used on all boundaries. The computational domain is discretized with a uniform mesh size of  $400 \times 400$  and the final time is chosen to be  $t = 0.25$ .

The density contours of the initial and final states, computed using the WENO-ACM scheme, have been shown in Fig. 15. The WENO-ACM scheme is able to capture the structure of the explosion problem. In Fig. 16, we have presented the cross-sectional slices of density profile along the plane  $y = 0.0$ , calculated by the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes. The reference solution is obtained using the WENO-JS scheme with a uniform mesh size of  $1000 \times 1000$ . We can observe that the WENO-M, WENO-PM6 and WENO-ACM schemes provide significantly better resolutions than the WENO-JS scheme. In Fig. 16(b) and (d), at  $x \in (-0.15, 0.15)$  and  $x \in (0.8, 0.82)$ , the WENO-ACM scheme presents a higher resolution than the WENO-PM6 and WENO-M schemes whose resolutions are comparable. In Fig. 16(c), the WENO-ACM scheme gives better resolution at  $x \in (0.38, 0.62)$  than the WENO-PM6 scheme which performs slightly better than the WENO-M scheme.

**Example 4.11** (2D Riemann problem). The calculation of the 2D Riemann problem [31, 32] is done over a unit square domain  $[0, 1] \times [0, 1]$ , initially involving the constant states of flow variables over each quadrant which is got by dividing the computational domain using lines  $x = x_0$  and  $y = y_0$ . There are many different configurations for the 2D Riemann problem [31]. In this example, the configuration is taken from [25, 31]

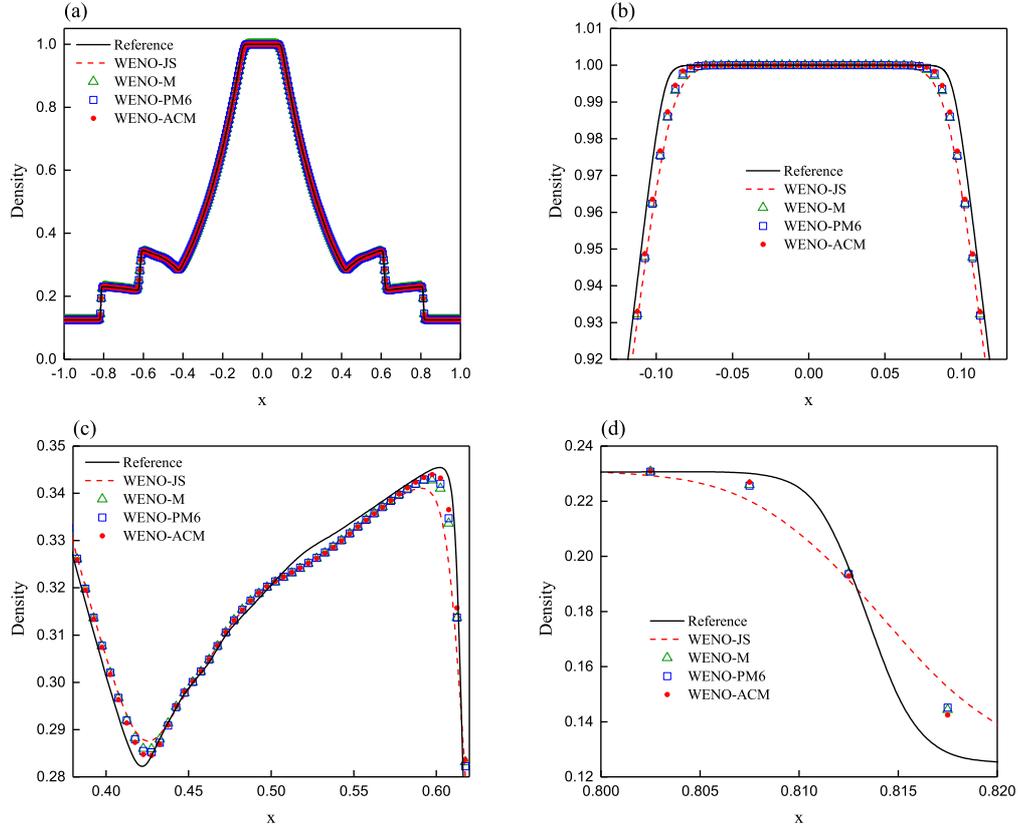


Figure 16: The cross-sectional slices of density plot along the plane  $y = 0.0$ , computed using the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 0.25$  with a uniform mesh size of  $400 \times 400$ .

with the following initial data

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} (1.0, 0.0, -0.3, 1.0), & 0.5 \leq x \leq 1.0, & 0.5 \leq y \leq 1.0, \\ (2.0, 0.0, 0.3, 1.0), & 0.0 \leq x \leq 0.5, & 0.5 \leq y \leq 1.0, \\ (1.0625, 0.0, 0.8145, 0.4), & 0.0 \leq x \leq 0.5, & 0.0 \leq y \leq 0.5, \\ (0.5313, 0.0, 0.4276, 0.4), & 0.5 \leq x \leq 1.0, & 0.0 \leq y \leq 0.5. \end{cases}$$

The transmission boundary conditions are used on all boundaries, and the numerical solutions are calculated using considered WENO schemes at output time  $t = 0.3$  with a mesh size of  $1200 \times 1200$ .

In Fig. 17, we have shown the numerical results of density obtained using the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes. All considered schemes can capture the main structure of the solution. However, this example is commonly focused on the description of instability of the slip line [25, 29], and we have displayed the close-up view of this instability in Fig. 18. We can observe that the WENO-JS and WENO-M schemes failed to resolve the instability of the slip line under current spa-

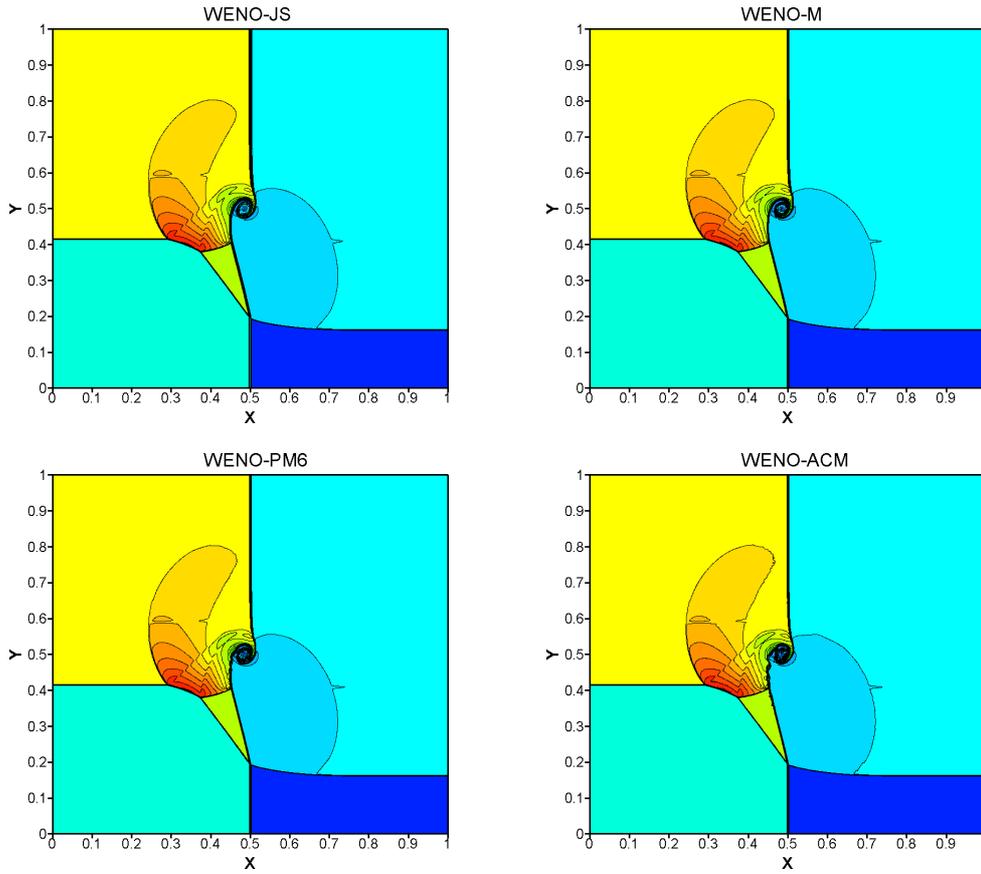


Figure 17: Density plots for the 2D Riemann problem using 30 contour lines with range from 0.5 to 2.4, computed using the WENO-JS, WENO-M, WENO-PM6, WENO-ACM schemes at output time  $t = 0.3$  with a uniform mesh size of  $1200 \times 1200$ .

tial resolution, while both the WENO-PM6 and WENO-ACM schemes have evidently resolved this instability. The unstable physical structures in the solution of the WENO-ACM scheme appear to have larger length scale and bigger wave numbers when compared with WENO-PM6.

**Example 4.12** (Double Mach reflection, DMR). Now we apply the considered WENO schemes to the two-dimensional double Mach reflection problem [42]. This problem is an important test where a vertical shock wave moves horizontally into a wedge that is inclined by some angle. The computational domain of this problem is  $[0, 4] \times [0, 1]$  and the initial condition is given by

$$(\rho, u, v, p)(x, y, 0) = \begin{cases} \left(8.0, 8.25 \cos \frac{\pi}{6}, -8.25 \sin \frac{\pi}{6}, 116.5\right), & x < x_0 + \frac{y}{\sqrt{3}}, \\ (1.4, 0.0, 0.0, 1.0), & x \geq x_0 + \frac{y}{\sqrt{3}}, \end{cases}$$

where  $x_0 = 1/6$ . The left boundary condition is inflow, with the post-shock values

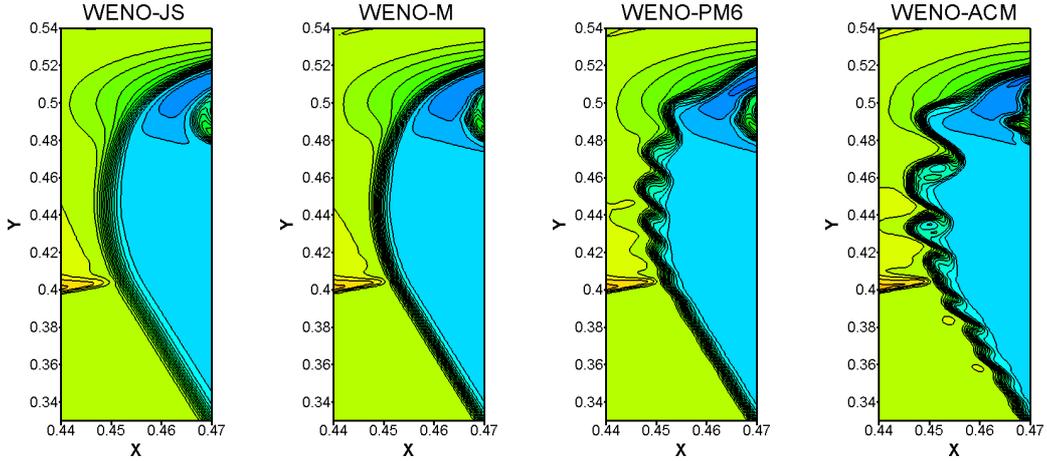


Figure 18: The zoomed-in density plots for the 2D Riemann problem, computed using the WENO-JS, WENO-M, WENO-PM6, WENO-ACM schemes at output time  $t = 0.3$  with a uniform mesh size of  $1200 \times 1200$ .

as stated above, and the outflow boundary condition is used on the right boundary. On the bottom boundary, the reflective boundary condition is applied to the interval  $[x_0, 4]$ , while at  $(0, x_0)$ , the post-shock values are imposed. The boundary condition on the upper boundary is implemented as follows:

$$(\rho, u, v, p)(x, 1, t) = \begin{cases} \left(8.0, 8.25 \cos \frac{\pi}{6}, -8.25 \sin \frac{\pi}{6}, 116.5\right), & x \in [0, s(t)), \\ (1.4, 0.0, 0.0, 1.0), & x \in [s(t), 4], \end{cases}$$

where  $s(t)$  is the position of the shock wave at time  $t$  on the upper boundary and given by  $s(t) = x_0 + (1 + 20t)/\sqrt{3}$ . The computational domain  $[0, 3] \times [0, 1]$  is discretized with a uniform mesh size of  $2000 \times 500$  and the output time is chosen to be  $t = 0.2$ .

In Fig. 19, we have shown the numerical results of density obtained using the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes. Further, in Fig. 20, we have displayed the close-up view of the region around the double Mach stems to observe more clearly the numerical solutions of all considered WENO schemes. In general, the global structure of the solution is very similar for different schemes, and all schemes are able to capture the companion structures behind the lower half of the right-moving reflection shock. However, the dissipation of the various schemes can be distinguished by the number and size of the small vortices generated along the slip lines. We can clearly see that the WENO-ACM and WENO-PM6 schemes capture more in number and bigger in size of the small vortices along the slip lines than the WENO-M and WENO-JS schemes, and it indicates that the resolving ability of the WENO-ACM and WENO-PM6 schemes is better than the other ones. In the solution of the WENO-JS scheme, the vortex rolled up near the slip lines is significantly dissipated due to the excessive numerical dissipation. It should be noted that all considered schemes suffered from the post shock oscillations in this test. The post shock oscillation seems slight in

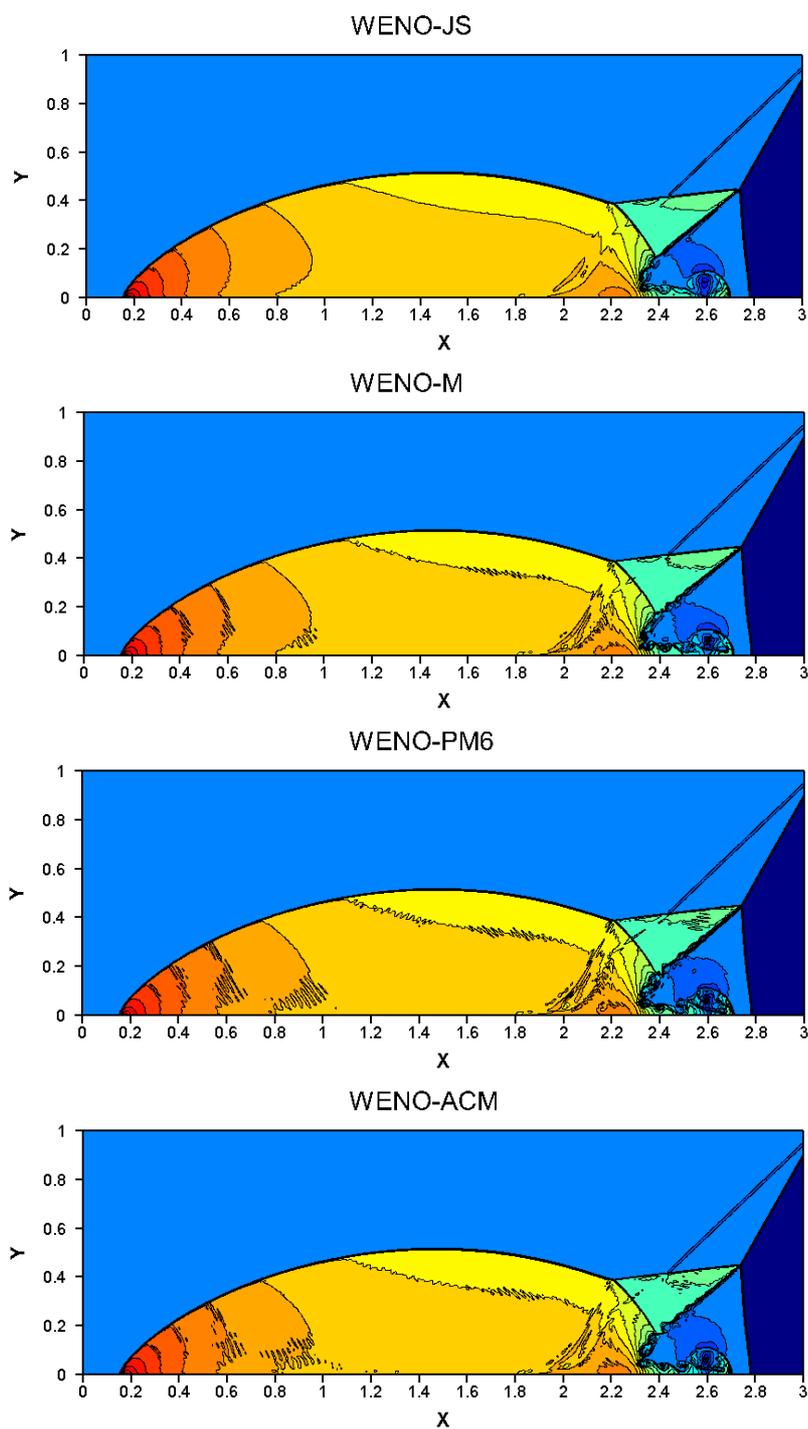


Figure 19: Density plots for the DMR problem using 30 contour lines with range from 1.4 to 25, computed using the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 0.2$  with a uniform mesh size of  $2000 \times 500$ .

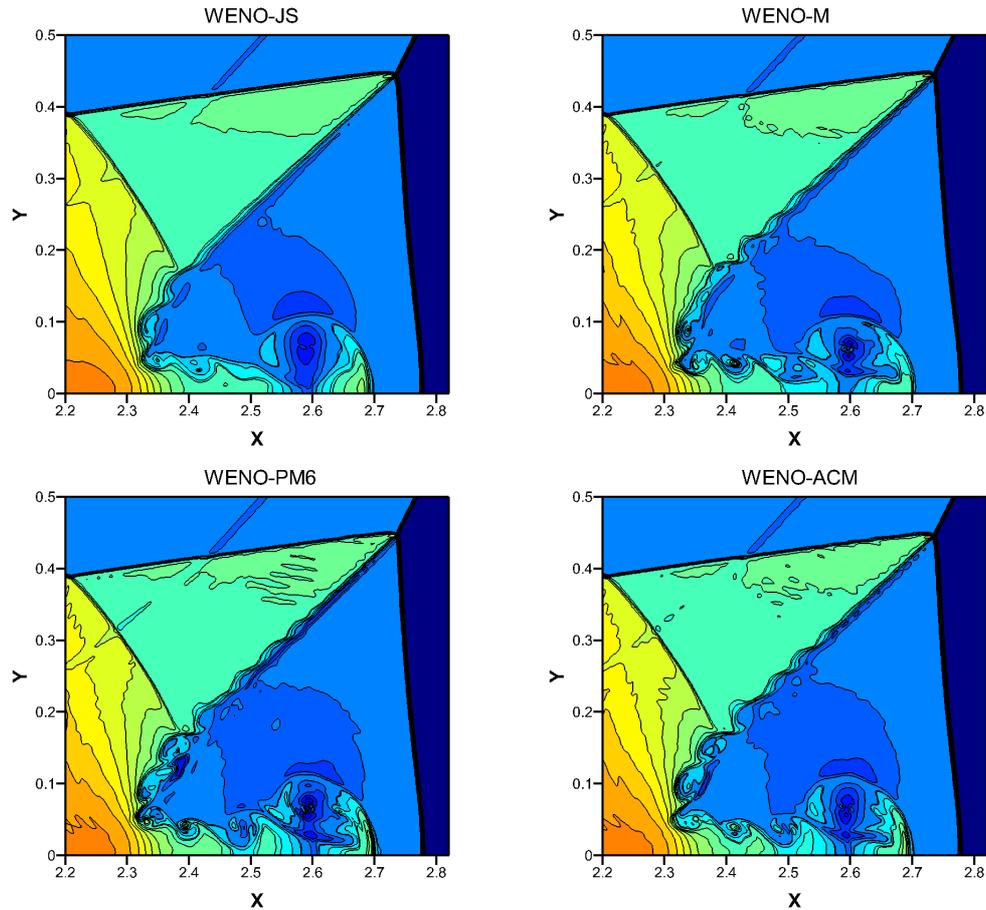


Figure 20: The zoomed-in density plots for the DMR problem, computed using the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 0.2$  with a uniform mesh size of  $2000 \times 500$ .

the solution of the WENO-JS schemes and it becomes more serious in the solution of the WENO-M scheme. Furthermore, it gets much more serious in the solutions of both the WENO-PM6 and WENO-ACM schemes. As mentioned in [44], these oscillations do not affect the “ENO (essentially non-oscillatory)” property of WENO schemes, but they are indeed responsible for the numerical residue to hang at the truncation error level of the scheme. As it is not the key point we are concerned about here, we refer to [44] for more details, in which Zhang *et al.* proposed a systematic analysis and detailed discussion about this issue.

**Example 4.13** (Forward facing step problem, FFS). This model problem was first presented by Woodward and Colella [42]. Recently, important details such as the physical instability and roll-up of the vortex sheet that emanates from the Mach stem have been successfully captured by various high order schemes [2, 3, 8, 10, 45]. Our purpose is to prove that the WENO-ACM scheme is also able to successfully capture the roll-up of the vortex sheet and perform robustly on this severely stringent problem.

The setup of the problem is as follows. There is a step with a height of 0.2 length units located 0.6 length units from the left-hand end of a wind tunnel, which is 1 length unit wide and 3 length units long. The computational domain of this problem is  $\Omega = [0, 0.6] \times [0, 1] \cup [0.6, 3] \times [0.2, 1]$  and the flow is initialized by

$$(\rho, u, v, p)(x, y, 0) = (1.4, 3.0, 0.0, 1.0), \quad (x, y) \in \Omega.$$

Reflective boundary conditions are used along the walls of the wind tunnel and the step, and inflow and outflow conditions are used at the entrance and the exit of the wind tunnel respectively. In order to handle the singularity at the left top corner of the step, the same technique used in [42] is employed. The computational domain is discretized with uniform mesh sizes of  $900 \times 300$  and  $1200 \times 400$ , and the output time is chosen to be  $t = 4$ .

The density contours obtained by all considered schemes are shown in Figs. 21 and 22 for different computing mesh cells. We can observe that these considered schemes capture all the shocks properly with sharp profiles. From Fig. 21, we find that on the uniform mesh size of  $900 \times 300$ , the roll-up of the vortex sheet is clearly visible when the WENO-ACM scheme is used, while not observed in solutions of the other three considered WENO schemes. From Fig. 22, we can see that with an increase of the mesh size from  $900 \times 300$  to  $1200 \times 400$ , the roll-up of the vortex sheet becomes clearly visible when the WENO-PM6 scheme is used, while it is not particularly clear for the WENO-M scheme and still invisible for the WENO-JS scheme. Moreover, the roll-up of the vortex sheet is observed more evidently and maintains the most intensely when the WENO-ACM scheme is used. These demonstrate the advantage of the WENO-ACM scheme that has less dissipation and better resolution in capturing details of the complicated flow structures. Again, we can see the post shock oscillations in solutions of all considered schemes as mentioned in the DMR example above. And very similar phenomenon is observed.

#### 4.4. Computational cost comparison for 2D Euler problems

In this subsection, we compare the computational costs of the WENO-ACM scheme with the WENO-JS, WENO-M and WENO-PM6 schemes. The numerical experiments of the two-dimensional Euler system simulated earlier, that is, Example 4.9 to Example 4.13 in Subsection 4.3, are taken as the test objects. These examples are tested with all conditions the same as in Subsection 4.3. In addition, we also test each example with a different mesh size respectively. An in-house code developed in C++ is employed, running in serial mode under Windows system, and the CPU is Intel(R) Core(TM) i9-9880H. In order to rule out the effect of other operations such as the boundary treatment, initialization, etc., as applied in reference [5], the CPU timing per Runge-Kutta step is considered here. And to mitigate the influence of randomness, each test is repeated three times under the same condition.

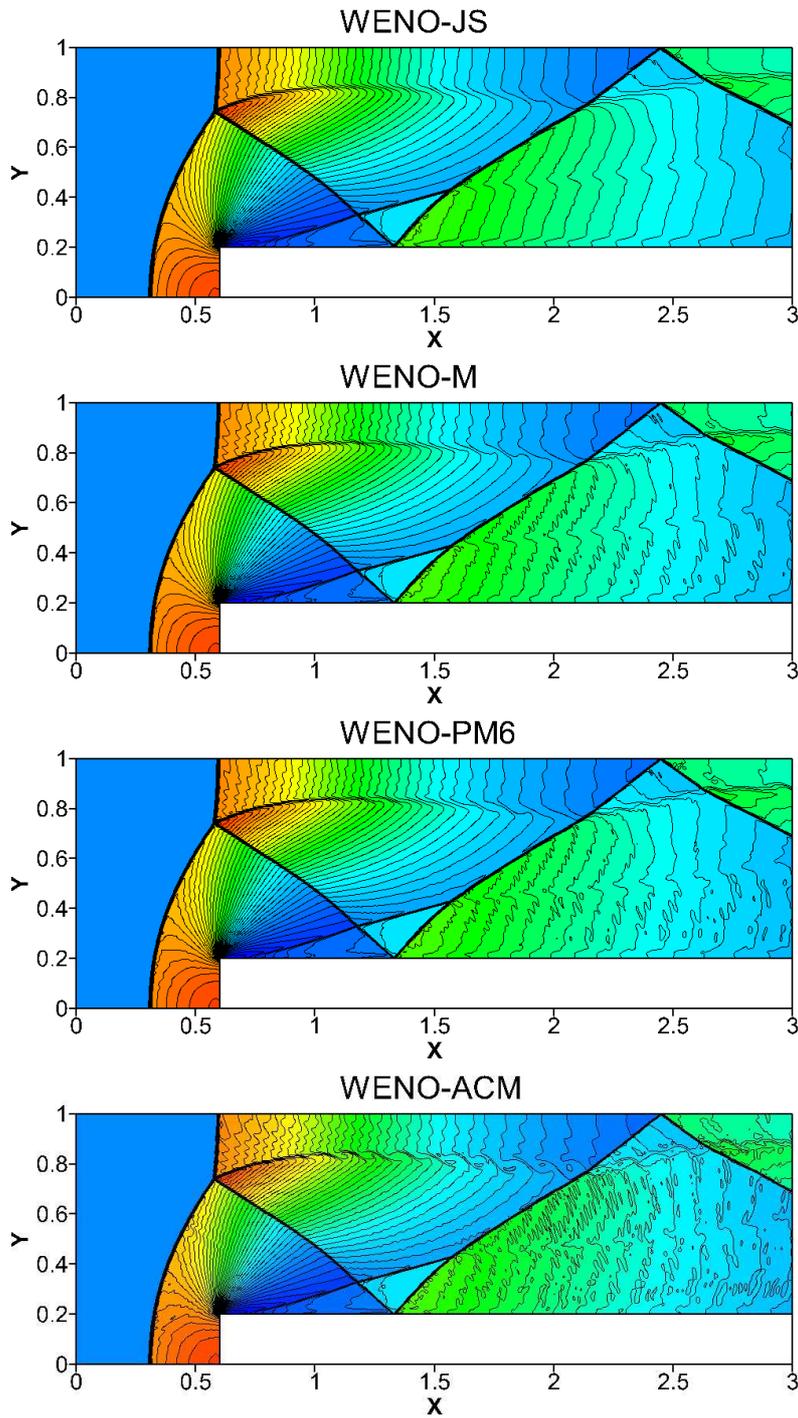


Figure 21: Density plots for the forward facing step problem using 50 density contour lines with range from 0.32 to 6.5, computed using the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 4.0$  with a uniform mesh size of  $900 \times 300$ .

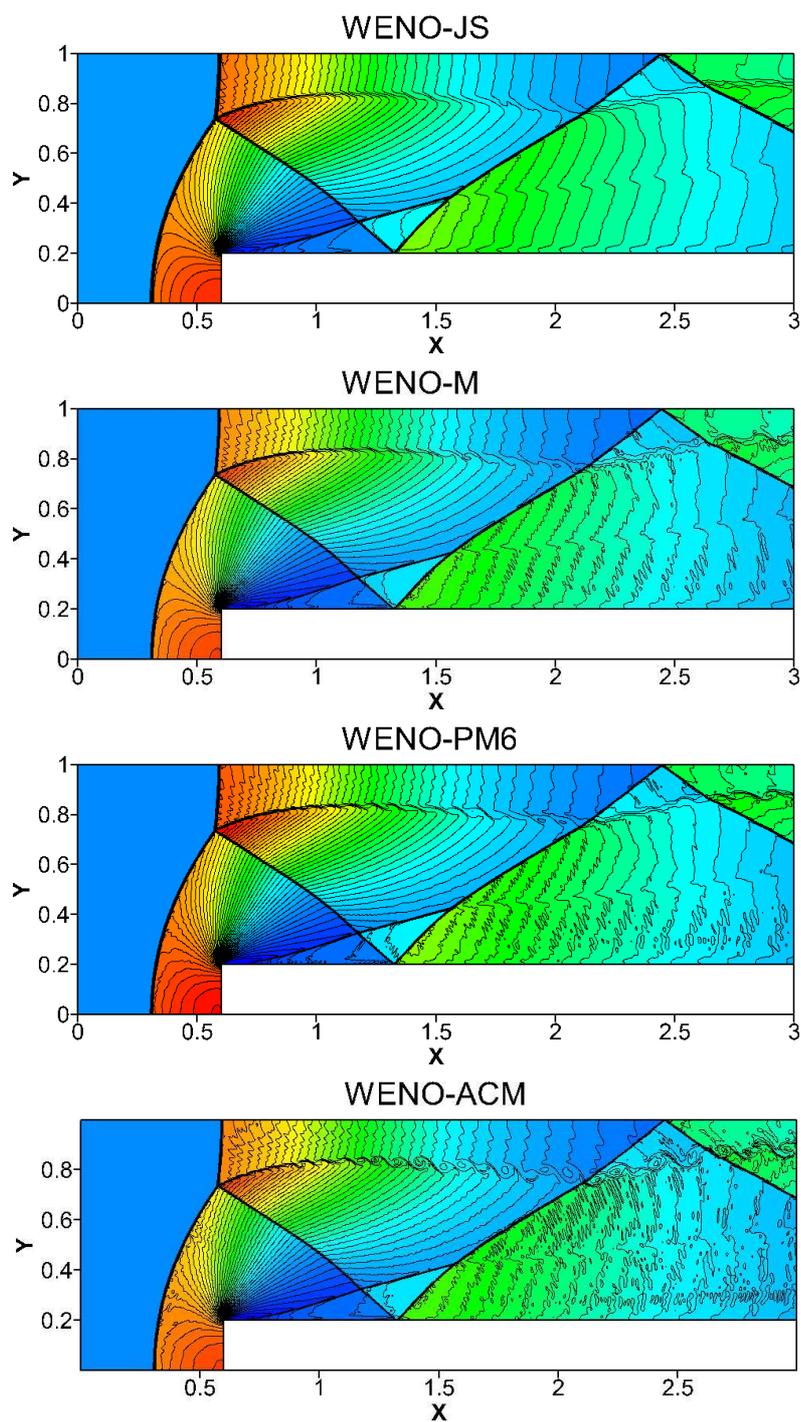


Figure 22: Density plots for the forward facing step problem using 50 density contour lines with range from 0.32 to 6.5, computed using the WENO-JS, WENO-M, WENO-PM6 and WENO-ACM schemes at output time  $t = 4.0$  with a uniform mesh size of  $1200 \times 400$ .

Let  $T(X)$  denote the *cost* (the CPU time that a Runge-Kutta step consumes) of the WENO-X scheme and  $P(X)$  denote the *extra cost* brought by the mapping process of the WENO-X scheme compared to the WENO-JS scheme, that is,

$$P(X) = \frac{T(X) - T(\text{JS})}{T(\text{JS})} \times 100\%.$$

In Table 6 to Table 10, we have given  $T(X)$  of all considered schemes and  $P(X)$  (in brackets) of the WENO-M, WENO-PM6 and WENO-ACM schemes for Example 4.9 to Example 4.13, respectively. In order to measure the *reduced cost* of the mapping process of the WENO-ACM scheme compared to those of the WENO-M and WENO-PM6 schemes, in the last two columns of each table, we have shown these *reduced costs* in percentages computed by

$$\chi_M = \frac{P(M) - P(\text{ACM})}{P(M)} = \frac{T(M) - T(\text{ACM})}{T(M) - T(\text{JS})},$$

Table 6: CPU time (in seconds) and the extra computational cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 4.9 as computed by considered WENO schemes.

Grid size	Test	WENO-JS	WENO-M	WENO-PM6	WENO-ACM	Saved	
						$\chi_M$	$\chi_{\text{PM6}}$
200 × 200	Test 1	0.219	0.281(28.31%)	0.359(63.93%)	0.229(4.57%)	83.87%	92.86%
	Test 2	0.226	0.282(24.78%)	0.361(59.73%)	0.234(3.54%)	85.71%	94.07%
	Test 3	0.218	0.274(25.69%)	0.345(58.26%)	0.228(4.59%)	82.14%	92.13%
	Average	0.221	0.279(26.24%)	0.355(60.63%)	0.230(4.22%)	83.91%	93.03%
400 × 400	Test 1	0.890	1.118(25.62%)	1.415(58.99%)	0.921(3.48%)	86.40%	94.10%
	Test 2	0.875	1.124(28.46%)	1.399(59.89%)	0.906(3.54%)	87.55%	94.08%
	Test 3	0.885	1.115(25.99%)	1.410(59.32%)	0.924(4.41%)	83.04%	92.57%
	Average	0.883	1.119(26.68%)	1.408(59.40%)	0.917(3.81%)	85.71%	93.58%

Table 7: CPU time (in seconds) and the extra computational cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 4.10 as computed by considered WENO schemes.

Grid size	Test	WENO-JS	WENO-M	WENO-PM6	WENO-ACM	Saved	
						$\chi_M$	$\chi_{\text{PM6}}$
200 × 200	Test 1	0.207	0.258(24.64%)	0.321(55.07%)	0.213(2.90%)	88.24%	94.74%
	Test 2	0.204	0.263(28.92%)	0.328(60.78%)	0.215(5.39%)	81.36%	91.13%
	Test 3	0.210	0.268(27.62%)	0.343(63.33%)	0.218(3.81%)	86.21%	93.98%
	Average	0.207	0.263(27.05%)	0.331(59.74%)	0.215(4.03%)	85.12%	93.26%
400 × 400	Test 1	0.794	0.987(24.31%)	1.219(53.53%)	0.807(1.64%)	93.26%	96.94%
	Test 2	0.782	0.982(25.58%)	1.232(57.54%)	0.811(3.71%)	85.50%	93.56%
	Test 3	0.788	0.998(26.65%)	1.211(53.68%)	0.814(3.30%)	87.62%	93.85%
	Average	0.788	0.898(25.51%)	1.221(54.91%)	0.811(2.88%)	88.72%	94.76%

Table 8: CPU time (in seconds) and the extra computational cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 4.11 as computed by considered WENO schemes.

Grid size	Test	WENO-JS	WENO-M	WENO-PM6	WENO-ACM	Saved	
						$\chi_M$	$\chi_{PM6}$
600 × 600	Test 1	1.852	2.414(30.35%)	2.917(57.51%)	1.884(1.73%)	94.31%	97.00%
	Test 2	1.843	2.319(25.83%)	2.905(57.62%)	1.900(3.09%)	88.03%	94.63%
	Test 3	1.861	2.404(29.18%)	2.935(57.71%)	1.918(3.06%)	89.50%	94.69%
	Average	1.852	2.379(28.46%)	2.919(57.61%)	1.900(2.63%)	90.77%	95.44%
1200 × 1200	Test 1	6.678	8.699(30.26%)	10.920(63.52%)	6.978(4.49%)	85.16%	92.93%
	Test 2	6.749	8.694(28.82%)	10.943(62.14%)	6.882(1.97%)	93.16%	96.83%
	Test 3	6.740	8.743(29.72%)	11.000(63.20%)	7.001(3.87%)	86.97%	93.87%
	Average	6.722	8.712(29.60%)	10.954(62.95%)	6.954(3.44%)	88.37%	94.53%

and

$$\chi_{PM6} = \frac{P(\text{PM6}) - P(\text{ACM})}{P(\text{PM6})} = \frac{T(\text{PM6}) - T(\text{ACM})}{T(\text{PM6}) - T(\text{JS})},$$

respectively.

From Table 6 to Table 10, we can easily observe that:

- (1) for each example with different mesh sizes, the *cost* in the three tests has a certain degree of fluctuation;
- (2) on average, the *extra costs* compared to the WENO-JS scheme are higher than 24% for the WENO-M scheme, higher than 54% for the WENO-PM6 scheme, while lower than 5% for the WENO-ACM scheme;
- (3) the *reduced costs* of the WENO-ACM scheme are more than 83% and 93% compared to the WENO-M scheme and the WENO-PM6 scheme, respectively.

Table 9: CPU time (in seconds) and the extra computational cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 4.12 as computed by considered WENO schemes.

Grid size	Test	WENO-JS	WENO-M	WENO-PM6	WENO-ACM	Saved	
						$\chi_M$	$\chi_{PM6}$
1000 × 250	Test 1	1.132	1.462(29.15%)	1.794(58.48%)	1.178(4.06%)	86.06%	93.05%
	Test 2	1.160	1.478(27.41%)	1.757(51.47%)	1.191(2.67%)	90.25%	94.81%
	Test 3	1.156	1.469(27.08%)	1.765(52.68%)	1.186(2.60%)	90.42%	95.07%
	Average	1.149	1.469(27.87%)	1.772(54.18%)	1.185(3.10%)	88.87%	94.27%
2000 × 500	Test 1	4.299	5.610(30.50%)	6.749(56.99%)	4.462(3.79%)	87.57%	93.35%
	Test 2	4.278	5.618(31.32%)	6.739(57.53%)	4.435(3.67%)	88.28%	93.62%
	Test 3	4.307	5.648(31.14%)	6.766(57.09%)	4.502(4.53%)	85.46%	92.07%
	Average	4.295	5.625(30.98%)	6.751(57.20%)	4.466(4.00%)	87.10%	93.01%

Table 10: CPU time (in seconds) and the extra computational cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 4.13 as computed by considered WENO schemes.

Grid size	Test	WENO-JS	WENO-M	WENO-PM6	WENO-ACM	Saved	
						$\chi_M$	$\chi_{PM6}$
$900 \times 300$	Test 1	1.873	2.497(33.32%)	3.142(67.75%)	1.945(3.84%)	88.46%	94.33%
	Test 2	1.880	2.502(33.09%)	3.124(66.17%)	1.926(2.45%)	92.60%	96.30%
	Test 3	1.868	2.465(31.96%)	3.147(68.47%)	1.928(3.21%)	89.95%	95.31%
	Average	1.874	2.488(32.79%)	3.138(67.46%)	1.933(3.17%)	90.34%	95.31%
$1200 \times 400$	Test 1	3.221	4.308(33.75%)	5.372(66.78%)	3.360(4.32%)	87.21%	93.54%
	Test 2	3.238	4.359(34.62%)	5.498(69.80%)	3.393(4.79%)	86.17%	93.14%
	Test 3	3.230	4.288(32.76%)	5.423(67.89%)	3.381(4.67%)	85.73%	93.11%
	Average	3.229	4.318(33.71%)	5.431(68.16%)	3.378(4.59%)	86.37%	93.26%

## 5. Conclusions

In order to reduce the computational cost introduced by mapping processes of the mapped WENO-M [20] and WENO-PM6 [11] schemes on the premise of retaining their advantages, we have devised a new mapped WENO scheme named WENO-ACM for hyperbolic conservation laws by introducing an approximate constant mapping function. It is theoretically and numerically demonstrated that the WENO-ACM scheme achieves the optimal convergence orders at critical points as the original mapped WENO-M scheme does. The new approximate constant mapping function satisfies the two additional properties first proposed in the WENO-PM6 scheme, and these properties ensure that the WENO-ACM scheme is able to generate comparable or better numerical solutions compared with the WENO-PM6 scheme, which has lower dissipation and provides higher resolution results than the classic WENO-JS and WENO-M schemes. Extensive numerical tests with two dimensional Euler equations show that the WENO-ACM scheme can reduce the cost of the mapping process by more than 83% compared to the WENO-M scheme and by more than 93% compared to the WENO-PM6 scheme, making the extra computational cost reduced from more than 24% for the WENO-M scheme and more than 54% for the WENO-PM6 scheme to a more acceptable value of no more than 5%.

## Acknowledgments

The authors would like to thank the referees for the helpful suggestions.

## References

- [1] F. ARÀNDIGA, A. BAEZA, A. M. BELDA AND P. MULET, *Analysis of WENO schemes for full and global accuracy*, SIAM J. Numer. Anal. 49 (2011), 893–915.

- [2] D. S. BALSARA, C. MEYER, M. DUMBSER, H. DU AND Z. XU, *Efficient implementation of ADER schemes for Euler and magnetohydrodynamical flows on structured meshes-Speed comparisons with Runge-Kutta methods*, J. Comput. Phys. 235 (2013), 934–969.
- [3] D. S. BALSARA, T. RUMPF, M. DUMBSER AND C. D. MUNZ, *Efficient, high accuracy ADER-WENO schemes for hydrodynamics and divergence-free magnetohydrodynamics*, J. Comput. Phys. 228 (2009), 2480–2516.
- [4] D. S. BALSARA AND C. W. SHU, *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, J. Comput. Phys. 160 (2000), 405–452.
- [5] R. BORGES, M. CARMONA, B. COSTA AND W. S. DON, *An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws*, J. Comput. Phys. 227 (2008), 3101–3211.
- [6] M. CASTRO, B. COSTA, W. S. DON, *High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws*, J. Comput. Phys. 230 (2011), 1766–1792.
- [7] A. CHATTERJEE, *Shock wave deformation in shock-vortex interactions*, Shock Waves 9 (1999), 95–105.
- [8] B. COCKBURN AND C. W. SHU, *The Runge-Kutta discontinuous Galerkin method for Conservation Laws V: multidimensional systems*, J. Comput. Phys. 141 (1998), 199–224.
- [9] W. S. DON AND R. BORGES, *Accuracy of the weighted essentially non-oscillatory conservative finite difference schemes*, J. Comput. Phys. 250 (2013), 347–372.
- [10] P. FAN, Y. SHEN, B. TIAN AND C. YANG, *A new smoothness indicator for improving the weighted essentially non-oscillatory scheme*, J. Comput. Phys. 269 (2014), 329–354.
- [11] H. FENG, F. HU AND R. WANG, *A new mapped weighted essentially non-oscillatory scheme*, J. Sci. Comput. 51 (2012), 449–473.
- [12] H. FENG, C. HUANG AND R. WANG, *An improved mapped weighted essentially non-oscillatory scheme*, Appl. Math. Comput. 232 (2014), 453–468.
- [13] G. A. GEROLYMOS, D. SÉNÉCHAL AND I. VALLET, *Very-high-order WENO schemes*, J. Comput. Phys. 228 (2009), 8481–8524.
- [14] S. GOTTLIEB AND C. W. SHU, *Total variation diminishing Runge-Kutta schemes*, Math. Comput. 67 (1998), 73–85.
- [15] S. GOTTLIEB, C. W. SHU AND E. TADMOR, *Strong stability-preserving high-order time discretization methods*, SIAM Rev. 43 (2001), 89–112.
- [16] A. HARTEN, *ENO schemes with subcell resolution*, J. Comput. Phys. 83 (1989), 148–184.
- [17] A. HARTEN, B. ENGQUIST, S. OSHER AND S. R. CHAKRAVARTHY, *Uniformly high order accurate essentially non-oscillatory schemes III*, J. Comput. Phys. 71 (1987), 231–303.
- [18] A. HARTEN AND S. OSHER, *Uniformly high order accurate essentially non-oscillatory schemes I*, SIAM J. Numer. Anal. 24 (1987), 279–309.
- [19] A. HARTEN, S. OSHER, B. ENGQUIST AND S. R. CHAKRAVARTHY, *Some results on uniformly high order accurate essentially non-oscillatory schemes*, Appl. Numer. Math. 2 (1986), 347–377.
- [20] A. K. HENRICK, T. D. ASLAM AND J. M. POWERS, *Mapped weighted essentially non-oscillatory schemes: Achieving optimal order near critical points*, J. Comput. Phys. 207 (2005), 542–567.
- [21] Z. HONG, Z. YE AND X. MENG, *A mapping-function-free WENO-M scheme with low computational cost*, J. Comput. Phys. 405 (2020), 109145.
- [22] F. HU, R. WANG AND X. CHEN, *A modified fifth-order WENOZ method hyperbolic conservation laws*, J. Comput. Appl. Math. 303 (2016), 56–68.
- [23] G. S. JIANG AND C. W. SHU, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys. 126 (1996), 202–228.

- [24] P. D. LAX, *Weak solutions of nonlinear hyperbolic equations and their numerical computation*, Commun. Pure Appl. Math. 7 (1954), 159–193.
- [25] Q. LI, P. LIU AND H. ZHANG, *Piecewise polynomial mapping method and corresponding WENO scheme with improved resolution*, Commun. Comput. Phys. 18 (2015), 1417–1444.
- [26] R. LI AND W. ZHONG, *A modified adaptive improved mapped WENO method*, Commun. Comput. Phys., (to appear).
- [27] X. D. LIU, S. OSHER AND T. CHAN, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys. 115 (1994), 200–212.
- [28] S. P. PAO AND M. D. SALAS, *A numerical study of two-dimensional shock-vortex interaction*, In: AIAA 14th Fluid and Plasma Dynamics Conference, California, Palo Alto, 1981.
- [29] S. PIROZZOLI, *Numerical methods for high-speed flows*, Annu. Rev. Fluid Mech. 43 (2011), 163–194.
- [30] Y. X. REN, M. LIU AND H. ZHANG, *A characteristic-wise hybrid compact-WENO scheme for solving hyperbolic conservation laws*, J. Comput. Phys. 192 (2003), 365–386.
- [31] C. W. SCHULZ-RINNE, *Classification of the Riemann problem for two-dimensional gas dynamics*, SIAM J. Math. Anal. 24 (1993), 76–88.
- [32] C. W. SCHULZ-RINNE, J. P. COLLINS AND H. M. GLAZ, *Numerical solution of the Riemann problem for two-dimensional gas dynamics*, SIAM J. Sci. Comput. 14 (1993), 1394–1414.
- [33] C. W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys. 77 (1988), 439–471.
- [34] C. W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes II*, J. Comput. Phys. 83 (1989), 32–78.
- [35] G. A. SOD, *A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws*, J. Comput. Phys. 27 (1978), 1–31.
- [36] E. F. TORO, *Riemann Solvers and Numerical Methods for Fluid Dynamics*, Springer-Verlag, 1999.
- [37] U. S. VEVEK, B. ZANG AND T. H. NEW, *A New Mapped WENO Method for Hyperbolic Problems*. In: Tenth International Conference on Computational Fluid Dynamics (ICCFD10), Barcelona, Spain, 2018.
- [38] U. S. VEVEK, B. ZANG AND T. H. NEW, *Adaptive mapping for high order WENO methods*, J. Comput. Phys. 381 (2019), 162–188.
- [39] B. S. WANG, P. LI, Z. GAO AND W. S. DON, *An improved fifth order alternative WENO-Z finite difference scheme for hyperbolic conservation laws*, J. Comput. Phys. 374 (2018), 469–477.
- [40] R. WANG, H. FENG AND C. HUANG, *A new mapped weighted essentially non-oscillatory method using rational function*, J. Sci. Comput. 67 (2016), 540–580.
- [41] Y. WANG, Y. DU, K. ZHAO AND L. YUAN, *Modified stencil approximations for fifth-order weighted essentially non-oscillatory schemes*, J. Sci. Comput. 81 (2019), 898–922.
- [42] P. WOODWARD AND P. COLELLA, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys. 54 (1984), 115–173.
- [43] R. ZHANG, M. ZHANG, C. W. SHU, *On the order of accuracy and numerical performance of two classes of finite volume WENO schemes*, Commun. Comput. Phys. 9 (2011), 807–827.
- [44] S. ZHANG AND C. W. SHU, *A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions*, J. Sci. Comput. 31 (2007), 273–305.
- [45] J. ZHU AND J. QIU, *A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws*, J. Comput. Phys. 318 (2016), 110–121.
- [46] J. ZHU AND J. QIU, *New Finite Difference Mapped WENO Schemes with Increasingly High Order of Accuracy*, Commun. Appl. Math. Comput. (2021).