

An Extension of the Order-Preserving Mapping to the WENO-Z-Type Schemes

Ruo Li¹ and Wei Zhong^{2,3,*}

¹ CAPT, LMAM and School of Mathematical Sciences, Peking University, Beijing 100871, China

² School of Mathematical Sciences, Peking University, Beijing 100871, China

³ Northwest Institute of Nuclear Technology, Xi'an, Shaanxi 710024, China

Received 1 February 2022; Accepted (in revised version) 19 May 2022

Abstract. In the present study, we extend the order-preserving (OP) criterion proposed in our latest studies to the WENO-Z-type schemes. Firstly, we innovatively present the concept of the generalized mapped WENO schemes by rewriting the Z-type weights in a uniform formula from the perspective of the mapping relation. Then, we naturally introduce the OP criterion to improve the WENO-Z-type schemes, and the resultant schemes are denoted as MOP-GMWENO-X, where the notation “X” is used to identify the version of the existing WENO-Z-type scheme in this paper. Finally, extensive numerical experiments have been conducted to demonstrate the benefits of these new schemes. We draw the conclusion that, the convergence properties of the proposed schemes are equivalent to the corresponding WENO-X schemes. The major benefit of the new schemes is that they have the capacity to achieve high resolutions and simultaneously remove spurious oscillations for long simulations. The new schemes have the additional benefit that they can greatly decrease the post-shock oscillations on solving 2D Euler problems with strong shock waves.

AMS subject classifications: 65M06, 65M12

Key words: WENO, Z-type weights, order-preserving generalized mapping, hyperbolic systems.

1 Introduction

Over the past several decades, the WENO methods [5–7, 22–28] have received considerable scholarly attention. The first WENO scheme that can obtain the designed convergence order of accuracy was proposed by Jiang and Shu [22], dubbed WENO-JS. By using the information of all r -point stencils of the ENO scheme [1–4], WENO-JS maintains the ENO property near the region with discontinuities or large gradients and in the

*Corresponding author.

Emails: rli@math.pku.edu.cn (R. Li), zhongwei2016@pku.edu.cn (W. Zhong)

meantime achieves the designed convergence rates of accuracy. It was pointed out by Henrick et al. in [29] that the fifth-order WENO-JS scheme can not recover the designed accuracy at critical points of order $n_{cp}=1$. For function g , one has $g'=0, g''\neq 0$ for $n_{cp}=1$. Similarly, $g'=0, g''=0, g'''\neq 0$ for $n_{cp}=2$, etc. Then, the sufficient condition for optimality of the convergence rates of accuracy was derived in [33], and this sufficient condition can be extended to higher order cases trivially [30]. In the work of Henrick et al. [29], a mapping function, namely $(g^M)_s(\omega^{JS})$, was designed and the resultant mapped WENO scheme, dubbed WENO-M, can achieve the designed convergence properties even in the presence of critical points. It is since the work of Henrick et al. [29] that the study of different mapped WENO methods has gained momentum, and a series of new mapping functions [30–32, 37, 38, 44] have been proposed by obeying the similar principles proposed by Henrick et al. [29].

Later, the work of Henrick et al. [29] inspired the development of a new family of nonlinear weights, dubbed Z-type weights. From a different perspective, Borges et al. [33] proposed another version of nonlinear weights by using available and previously unused information of the WENO-JS scheme. In other words, a global smoothness indicator (GSI) of higher order, obtained via a linear combination of the original smoothness indicators of the WENO-JS scheme, was proposed and employed to devise the new nonlinear weights. The resultant scheme was denoted as WENO-Z. Because of the success of the WENO-Z scheme that its nonlinear weights can satisfy the sufficient conditions for optimality of the convergence order without any costly mapping processes, leading to superior results with almost the same computational effort of the WENO-JS method, different researchers have developed a multitude of techniques to design their Z-type weights [8–11, 14, 19–21, 39, 45, 46] by obeying the similar principles proposed by Borges et al. [33]. In this paper, all the WENO schemes using Z-type weights is collectively called WENO-Z-type schemes. We will give a brief review of several WENO-Z-type schemes in subsection 2.3.

Despite the success mainly for short-output-time simulations, the family of mapped WENO schemes has a serious and ubiquitous problem in calculations with long output times, that is, they can hardly avoid spurious oscillations and meanwhile preserve high resolutions for long simulations. This disadvantage in long simulations of the mapped WENO methods was firstly noticed and carefully studied by Feng et al. [31] and it has attracted considerable attention over the past decade [30, 32, 34, 36, 37, 42, 43]. However, up to now, far too little attention has been paid to the long-output-time simulations of the WENO-Z-type schemes. Indeed, our extensive calculations (see Subsection 4.3 below) show that the WENO-Z-type schemes also terribly suffer from either losing high resolutions or generating numerical oscillations on long-run calculations. Nevertheless, undoubtedly, this issue is worthy of scholarly attention.

Accordingly, in this article, we would like to focus on the theme of addressing the aforementioned drawback of the WENO-Z-type schemes. First of all, we give the important observation and analysis of the implicit relationship between the nonlinear weights of the WENO-JS scheme and the Z-type weights, denoted as IMR (standing for *implicit*

mapping relation) for simplicity. It can be found that the profiles of IMRs for various WENO-Z-type schemes are very similar to the traditional mapping curve (whose mapping function was specifically designed in the mapped WENO scheme and hence we call it *designed mapping* for simplicity in this paper) that embrace evident optimal weight intervals where the nonlinear weights are replaced by the ideal weights. It is well known that this kind of replacements appears favorable for reducing the dissipation and improving the resolution at least for short-output-time simulations. However, these replacements also generate the non-order-preserving (non-OP) points where the order of the nonlinear weights is disrupted. It has been demonstrated [34, 36] that the non-OP points are extremely harmful for the WENO schemes to preserving high resolutions and meanwhile avoiding spurious oscillations for long-run calculations. In other words, the enhancements of the WENO schemes with order-preserving mappings [34, 36] benefit from the mapped optimal weight interval, in which the WENO weights are close to the ideal weights, thus reducing the numerical dissipation and improving the resolution. On the other hand, the order-preserving scheme ensures the order of the mapped nonlinear weights near the shocks, so a more dissipative weight is used to reduce the numerical oscillations. Therefore, after advancing the new concept of the generalized mapped WENO schemes by reformulating the non-normalized nonlinear Z-type weights in a uniform form from the perspective of the mapping relation, the OP property is introduced to modify the previously published WENO-Z-type schemes for the purpose of improving their performance in long simulations. Necessary theoretical analysis will be provided and extensive numerical experiments of 1D linear advection equation with various initial conditions for long output times will be conducted and carefully discussed to examine the enhancement of the new proposed schemes. In addition, we also carry out several problems modeled via 1D and 2D Euler equations to show the good performance of these schemes.

We organize the remainder of this article as follows. The preliminaries are reviewed in Section 2, where we briefly recall the procedures of several typical WENO schemes, including WENO-JS [22], WENO-Z [33] and some other WENO methods with Z-type weights. In Section 3, the analysis of the Z-type nonlinear weights from the perspective of the mapping relation is given, and the concept of the generalized mapped WENO schemes is proposed. The method to generally improve the previously published WENO-Z-type schemes by requiring the use of the OP property is devised in Section 4, and some examples of 1D linear advection equation are performed to manifest the major benefits of the proposed schemes in this Section. In Section 5, more numerical results of 1D and 2D Euler equations are provided for illustration. The conclusions are given in Section 6.

2 Brief review of the WENO-Z-type schemes

It was reported [40] that there are three versions of the odd-order WENO methods with $r \geq 3$ when applied to hyperbolic systems. They are the classical WENO schemes (e.g.,

WENO-JS [22], WENO-ZS [41], WENO-NIS [13]), the mapped WENO schemes (e.g., WENO-M [29], WENO-PMk [31], WENO-MAIMi [32], WENO-ACM [44]) and the WENO-Z-type schemes (e.g., WENO-Z [33], WENO-Z+ [10], WENO-NIP [21]). As the version of the mapped WENO schemes has been discussed carefully in our previous work [36], we mainly describe the other two versions in this section. In other words, we first introduce the classical WENO-JS scheme [22], along with the original WENO-Z scheme as designed in [33] and then several WENO schemes with improved Z-type weights.

For simplicity but without loss of generality, we only pay our attention to uniform meshes in this paper. However, the technique proposed in the present work can be naturally applied to 2D/3D unstructured meshes and this is left for future study.

For brevity's sake, we restrict our discussion to the following 1D scalar equation within the Finite Volume Method (FVM) framework

$$u_t + f(u)_x = 0. \tag{2.1}$$

We assume that the computational domain is distributed into smaller uniform cells $I_j = [x_{j-1/2}, x_{j+1/2}]$, where $\Delta x = x_{j+1/2} - x_{j-1/2}$ is the mesh width, $x_{j\pm 1/2} = x_j \pm \frac{\Delta x}{2}$ are the interfaces of I_j and $x_j = \frac{1}{2}(x_{j+1/2} + x_{j-1/2})$ are the cell centers. Let

$$\bar{u}(x_j, t) = \frac{1}{\Delta x} \int_{x_{j-1/2}}^{x_{j+1/2}} u(\eta, t) d\eta,$$

we can transform Eq. (2.1) into the semi-discretized form

$$\begin{cases} \frac{d\bar{u}_j(t)}{dt} \approx \mathcal{L}(u_j), \\ \mathcal{L}(u_j) = -\frac{1}{\Delta x} (\hat{f}_{j+1/2} - \hat{f}_{j-1/2}), \end{cases} \tag{2.2}$$

where $\bar{u}_j(t)$ is the approximation to $\bar{u}(x_j, t)$, and $\hat{f}_{j\pm 1/2} = \hat{f}(u_{j\pm 1/2}^-, u_{j\pm 1/2}^+)$ is the numerical flux used to replace $f(u)$. The WENO reconstruction techniques are used to calculate the $u_{j\pm 1/2}^\pm$. We will choose the global LF flux in this article.

Remark 2.1. In the discussion below, we only show the reconstruction of $u_{j+1/2}^-$ as that of $u_{j+1/2}^+$ is symmetric to it with respect to $x_{j+1/2}$. And for brevity, the “-” sign in the superscript will be dropped without causing any confusion.

2.1 WENO-JS

In the 5th-order WENO-JS, $u_{j+1/2}$ is approximated by

$$u_{j+1/2} = \omega_0 u_{j+1/2}^0 + \omega_1 u_{j+1/2}^1 + \omega_2 u_{j+1/2}^2, \tag{2.3}$$

with

$$u_{j+1/2}^0 = \frac{1}{6}(2\bar{u}_{j-2} - 7\bar{u}_{j-1} + 11\bar{u}_j), \quad (2.4a)$$

$$u_{j+1/2}^1 = \frac{1}{6}(-\bar{u}_{j-1} + 5\bar{u}_j + 2\bar{u}_{j+1}), \quad (2.4b)$$

$$u_{j+1/2}^2 = \frac{1}{6}(2\bar{u}_j + 5\bar{u}_{j+1} - \bar{u}_{j+2}), \quad (2.4c)$$

and (for $s=0,1,2$)

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

where ϵ is a small positive number, $d_0 = \frac{1}{10}$, $d_1 = \frac{3}{5}$, $d_2 = \frac{3}{10}$. In [22], β_s , dubbed LSI, are explicitly given as follows

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

It was indicated by Henrick et al. [29] that the convergence rates of accuracy of WENO-JS may drop to 3rd-order at critical points. In the same article, they indicated that to ensure the overall scheme retaining fifth-order accuracy, it is sufficient to require

$$\omega_s^\pm - d_s = \mathcal{O}(\Delta x^3). \quad (2.6)$$

2.2 WENO-Z

Borges et al. [33] devised the first WENO-Z-type scheme, dubbed WENO-Z. By introducing the GSI, say, $\tau_5 = |\beta_0 - \beta_2|$, they compute the nonlinear weights by

$$\omega_s^Z = \frac{\alpha_s^Z}{\sum_{l=0}^2 \alpha_l^Z}, \quad \alpha_s^Z = d_s \left(1 + \left(\frac{\tau_5}{\beta_s + \epsilon} \right)^p \right), \quad s=0,1,2, \quad (2.7)$$

where the parameters β_s and ϵ are the same as in the classical WENO-JS scheme, and p is a tunable parameter. Borges et al. demonstrated that [33], if $p=1$, the WENO-Z scheme only achieves fourth-order convergence orders in the presence of critical points; if $p=2$, it can recover the designed 5th-order convergence rates. Unless indicated otherwise, we choose $p=2$ in the present study.

2.3 Other WENO-Z-type schemes

The significant contribution of WENO-Z is the inclusion of higher-order information by introducing the GSI, τ_5 , in the definition of the nonlinear weights. This helps the WENO-Z scheme get closer to the central scheme than the WENO-JS scheme and hence generate less dissipation. The formation of the nonlinear weights of WENO-Z gives an effective way for enhancing the capacity of WENO-JS. Thus, by obeying the similar principles of WENO-Z, a series of new Z-type nonlinear weights with various kinds of GSI are proposed. In this subsection, we briefly review other WENO schemes with different Z-type weights that will be further studied in the rest of this paper. It should be noted that there are so many different types of WENO-Z-type schemes [8–11, 14, 19–21, 33, 39, 45, 46], and we only considered a limited number of them just for illustrative purposes in the present study.

WENO-Z $\eta(\tau_l)$: Fan et al. [8, 39] proposed the WENO-Z $\eta(\tau_l)$ scheme with the following Z-type nonlinear weights

$$\omega_s^{Z\eta} = \frac{\alpha_s^{Z\eta}}{\sum_{l=0}^2 \alpha_l^{Z\eta}}, \quad \alpha_s^{Z\eta} = d_s \left(1 + \left(\frac{\tau_l}{\eta_s + \epsilon} \right)^2 \right), \quad s = 0, 1, 2, \tag{2.8}$$

where the LSI is calculated using the flexible and simple formula suggested by Shen and Zha [50], and we state them explicitly here

$$\begin{cases} \eta_0 = \frac{1}{4} (\bar{u}_{j-2} - 4\bar{u}_{j-1} + 3\bar{u}_j)^2 + (\bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j)^2, \\ \eta_1 = \frac{1}{4} (\bar{u}_{j-1} - \bar{u}_{j+1})^2 + (\bar{u}_{j-1} - 2\bar{u}_j + \bar{u}_{j+1})^2, \\ \eta_2 = \frac{1}{4} (3\bar{u}_j - 4\bar{u}_{j+1} + \bar{u}_{j+2})^2 + (\bar{u}_j - 2\bar{u}_{j+1} + \bar{u}_{j+2})^2. \end{cases} \tag{2.9}$$

Fan et al. [39] proposed a fifth-, a sixth- and two eighth- order GSIs as

$$\begin{cases} \tau_5 = |\eta_0 - \eta_2|, \quad \tau_6 = \left| \eta_5 - \frac{\eta_0 + 4\eta_1 + \eta_2}{6} \right|, \\ \tau_{81} = \left| \left(|P_0^{(1)}| - |P_2^{(1)}| \right) \left(P_0^{(2)} - 2P_1^{(2)} + P_2^{(2)} \right) \right|, \\ \tau_{82} = \left(|P_0^{(1)}| - |P_2^{(1)}| \right)^2 + \left(P_0^{(2)} - 2P_1^{(2)} + P_2^{(2)} \right)^2, \end{cases}$$

where

$$\eta_5 = \frac{1}{144} \left((\bar{u}_{j-2} - 8\bar{u}_{j-1} + 8b\bar{u}_{j+1} - \bar{u}_{j+2})^2 + (\bar{u}_{j-2} - 16\bar{u}_{j-1} + 30\bar{u}_j - 16\bar{u}_{j+1} + \bar{u}_{j+2})^2 \right),$$

and

$$\left\{ \begin{array}{l} P_0^{(1)} = \frac{1}{2}\bar{u}_{j-2} - 2\bar{u}_{j-1} + \frac{3}{2}\bar{u}_j, \\ P_1^{(1)} = -\frac{1}{2}\bar{u}_{j-1} + \frac{1}{2}\bar{u}_{j+1}, \\ P_2^{(1)} = -\frac{3}{2}\bar{u}_j + 2\bar{u}_{j+1} - \frac{1}{2}\bar{u}_{j+2}, \end{array} \right. \quad \left\{ \begin{array}{l} P_0^{(2)} = \bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j, \\ P_1^{(2)} = \bar{u}_{j-1} - 2\bar{u}_j + \bar{u}_{j+1}, \\ P_2^{(2)} = \bar{u}_j - 2\bar{u}_{j+1} + \bar{u}_{j+2}. \end{array} \right.$$

WENO-Z+: Acker et al. [10] proposed the WENO-Z+ scheme with the following Z-type nonlinear weights

$$\omega_s^{Z+} = \frac{\alpha_s^{Z+}}{\sum_{l=0}^2 \alpha_l^{Z+}}, \quad \alpha_s^{Z+} = d_s \left(1 + \left(\frac{\tau_5 + \epsilon}{\beta_s + \epsilon} \right)^2 + \lambda \left(\frac{\beta_s + \epsilon}{\tau_5 + \epsilon} \right) \right), \quad s=0,1,2, \quad (2.10)$$

where λ is recommended to be $\Delta x^{2/3}$ in their work.

WENO-ZA: Shen et al. [14] proposed the WENO-ZA scheme with the following Z-type nonlinear weights

$$\omega_s^{ZA} = \frac{\alpha_s^{ZA}}{\sum_{l=0}^2 \alpha_l^{ZA}}, \quad \alpha_s^{ZA} = d_s \left(1 + \frac{A\tau_6}{\beta_s + \epsilon} \right), \quad s=0,1,2, \quad (2.11)$$

where the LSI is defined by

$$\beta_s = \gamma_1 (\bar{u}_s^{(1)})^2 + \gamma_2 (\bar{u}_s^{(2)})^2, \quad s=0,1,2, \quad (2.12)$$

with

$$\left\{ \begin{array}{l} \bar{u}_0^{(1)} = (\bar{u}_{j-2} - 4\bar{u}_{j-1} + 3\bar{u}_j) / 2, \\ \bar{u}_1^{(1)} = (-\bar{u}_{j-1} + \bar{u}_{j+1}) / 2, \\ \bar{u}_2^{(1)} = (-3\bar{u}_j + 4\bar{u}_{j+1} - \bar{u}_{j+2}) / 2, \end{array} \right. \quad \left\{ \begin{array}{l} \bar{u}_0^{(2)} = \bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j, \\ \bar{u}_1^{(2)} = \bar{u}_{j-1} - 2\bar{u}_j + \bar{u}_{j+1}, \\ \bar{u}_2^{(2)} = \bar{u}_j - 2\bar{u}_{j+1} + \bar{u}_{j+2}. \end{array} \right. \quad (2.13)$$

In Eq. (2.11), the parameter A is recommended as

$$A = \frac{\tau_6}{\beta_0 + \beta_2 - \tau_6 + \epsilon}, \quad (2.14)$$

with

$$\tau_6 = \gamma_1 (|\bar{u}_0^{(1)}| - |\bar{u}_2^{(1)}|)^2 + \gamma_2 (|\bar{u}_0^{(2)}| - |\bar{u}_2^{(2)}|)^2. \quad (2.15)$$

According to the recommendation of Shen et al. [14], $\gamma_1 = 1$, $\gamma_2 = 13/12$ are used.

WENO-D/WENO-A: Wang et al. [20] proposed WENO-D and WENO-A with the following Z-type nonlinear weights

$$\left\{ \begin{array}{l} \text{WENO-D: } \omega_s^D = \frac{\alpha_s^D}{\sum_{l=0}^2 \alpha_l^D}, \quad \alpha_s^D = d_s \left(1 + \Phi \left(\frac{\tau_5}{\beta_s + \epsilon} \right)^p \right), \\ \text{WENO-A: } \omega_s^A = \frac{\alpha_s^A}{\sum_{l=0}^2 \alpha_l^A}, \quad \alpha_s^A = d_s \left(\max \left(1, \Phi \left(\frac{\tau_5}{\beta_s + \epsilon} \right)^p \right) \right), \end{array} \right. \quad s=0,1,2, \quad (2.16)$$

where $p = 2$ is used in their numerical examples, and $\tau_5 = |\beta_0 - \beta_2|$, Φ is computed by

$$\Phi = \min\{1, \phi\}, \quad \phi = \sqrt{|\beta_0 - 2\beta_1 + \beta_2|}. \tag{2.17}$$

WENO-NIP: Yuan [21] proposed the WENO-NIP scheme with the following Z-type nonlinear weights

$$\omega_s^{\text{NIP}} = \frac{\alpha_s^{\text{NIP}}}{\sum_{l=0}^2 \alpha_l^{\text{NIP}}}, \quad \alpha_s^{\text{NIP}} = d_s \left(1 + \frac{\tau}{(\epsilon + \chi_s)^2} \right), \quad s = 0, 1, 2, \tag{2.18}$$

where

$$\chi_0 = \theta |\bar{u}_{j-2} - 3\bar{u}_{j-1} + 2\bar{u}_j| + |\bar{u}_{j-2} - 2\bar{u}_{j-1} + \bar{u}_j|, \tag{2.19a}$$

$$\chi_1 = \theta |\bar{u}_{j+1} - \bar{u}_j| + |\bar{u}_{j-1} - 2\bar{u}_j + \bar{u}_{j+1}|, \tag{2.19b}$$

$$\chi_2 = \theta |\bar{u}_{j+1} - \bar{u}_j| + |\bar{u}_j - 2\bar{u}_{j+1} + \bar{u}_{j+2}|, \tag{2.19c}$$

and

$$\tau = |\bar{u}_{j-2} - 4\bar{u}_{j-1} + 6\bar{u}_j - 4\bar{u}_{j+1} + \bar{u}_{j+2}|^J.$$

As mentioned in [21], the requirement $J \geq \frac{2n_{cp} + 4}{4}$ needs to be satisfied. According to Yuan's solutions [21], $\theta = 0.1$ seems to be a better choice.

3 Study on the Z-type weights from the perspective of the mapping relation

It is well known [29,30,32,33,44] that the WENO-JS scheme can not achieve the designed convergence order at critical points, and there are usually two ways to address this issue: one is to introduce a mapping function [29–32, 37, 38, 44], the other is to introduce a GSI [10, 14, 20, 21, 33, 39]. Recently, the long-run simulations of the mapped WENO methods have been widely concerned [30–32, 34, 36, 37, 43]. The mapped WENO methods have one great drawback in common that it is very difficult to get high resolutions and remove spurious oscillations simultaneously for long-run simulations and many researches were devoted to amending it. However, any researches hardly focused on the performances of the WENO-Z-type schemes for this topic. In this study, this will be studied and discussed carefully. Actually, our numerical calculations indicate that the drawback of long-run simulations mentioned above also exist for the WENO-Z-type schemes. From Figs. 4-7 below, we can see that most of the existing WENO-Z-type schemes suffer from either losing high resolutions or generating spurious oscillations. This issue will be addressed in this paper.

In practical calculations, for any output time t , we can calculate uniquely the ω_s^{JS} of each location x_j according to Eq. (2.5). And also, for any WENO-X scheme, we can uniquely get its nonlinear weights, say ω_s^{X} , according to its nonlinear weights formula,

like the formulae of the various Z-type weights as shown in Subsection 2.3. Trivially, although no mapping function formula of $\omega_s^X(\omega_s^{JS})$ for WENO-Z-type schemes was explicitly given in previously published literature, the relation of $\omega_s^{JS} \sim \omega_s^X$ is uniquely determined. Thus, we call it *Implicit Mapping Relation (IMR)* of the WENO-X scheme. Indeed, one also can treat ω_s^X as a function of ω_s^{JS} as stated in Eq. (3.3) below.

Example 3.1 (SLP). The following Linear Problem proposed by Shu et al. in [22], denoted as SLP, is widely used to examine the performance of high resolution schemes. It is defined by

$$\begin{cases} u_t + u_x = 0, & -1 < x < 1, \\ u(x, 0) = u_0(x). \end{cases} \quad (3.1)$$

Let

$$G(x, \beta, z) = \exp(-\beta(x-z)^2), \quad F(x, \alpha, a) = \left(\max(1 - \alpha^2(x-a)^2, 0) \right)^{1/2},$$

and the initial condition $u_0(x)$ is given as

$$u_0(x) = \begin{cases} \frac{1}{6} [G(x, \beta, z - \hat{\delta}) + 4G(x, \beta, z) + G(x, \beta, z + \hat{\delta})], & -0.8 \leq x \leq -0.6, \\ 1, & -0.4 \leq x \leq -0.2, \\ 1 - |10(x - 0.1)|, & 0.0 \leq x \leq 0.2, \\ \frac{1}{6} [F(x, \alpha, a - \hat{\delta}) + 4F(x, \alpha, a) + F(x, \alpha, a + \hat{\delta})], & 0.4 \leq x \leq 0.6, \\ 0, & \text{otherwise,} \end{cases} \quad (3.2)$$

with $a = \frac{1}{2}$, $\alpha = 10$, $\hat{\delta} = \frac{1}{200}$, $z = -\frac{7}{10}$ and $\beta = \log \frac{2}{36\hat{\delta}^2}$.

Taking SLP as an example, by choosing $t = 2000$ and $N = 800$, we plot the IMRs of the WENO-Z, WENO-Z $\eta(\tau_5)$, WENO-Z $\eta(\tau_{81})$, WENO-Z+, WENO-ZA, WENO-D, WENO-A and WENO-NIP schemes in Fig. 1. For comparison purpose, we also plot the designed mapping of WENO-JS, or say identity mapping, and that of WENO-M. It is very interesting that the IMRs of all the considered WENO-Z-type schemes are similar to the designed mapping of the WENO-M scheme in general. In other words, it is quite obvious that there are “optimal weight intervals” where the mappings, as well as the Z-type weight formulae, replace the nonlinear weights using the associated ideal weights. We refer to Subsection 3.4.2 of [32] for more details about the “optimal weight intervals”. As far as we know, these features are not accounted for in previously published articles. However, it is indicated that [30, 32, 43] the *optimal weight interval* plays a crucial role in recovering the designed convergence orders in the presence of critical points and improving the resolution of the corresponding WENO scheme. Therefore, based on this meaningful observation, we innovatively build the concept of Generalized Mapped WENO schemes.

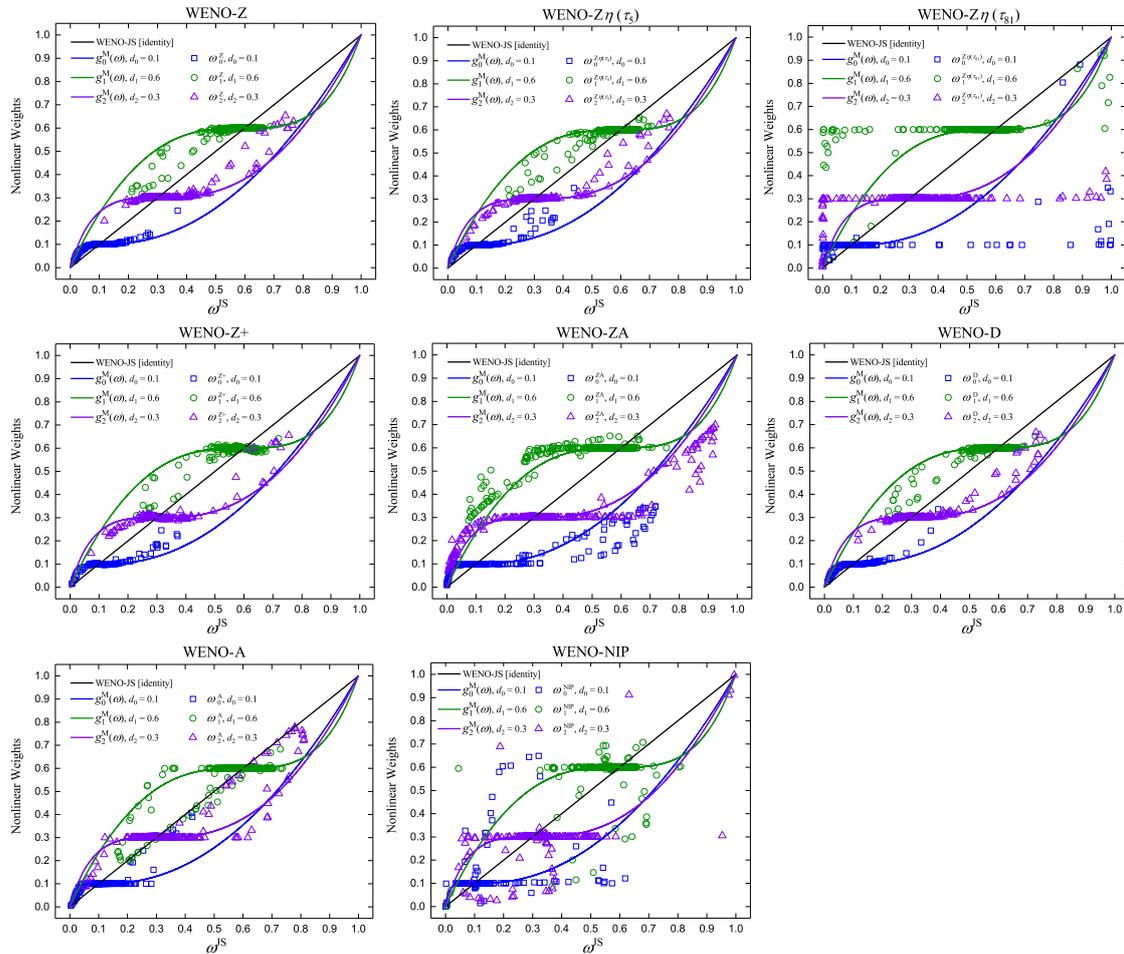


Figure 1: The IMRs for various WENO-Z-type schemes.

Definition 3.1 (Generalized Mapped WENO scheme). *If the IMR of WENO-X exhibits the key characteristic of the traditional mapped WENO scheme, that is, existing apparent optimal weight intervals, we say WENO-X is a Generalized Mapped WENO scheme.*

In fact, the various Z-type weights can be written in the following generalized form

$$\begin{cases} \omega_s^X = \frac{\alpha_s^X}{\sum_{l=0}^2 \alpha_l^X}, & s=0,1,2, \\ \alpha_s^X = \psi_{s,1}^X(d_s) + \frac{d_s / (\beta_s^{JS} + \epsilon)^2}{\sum_{l=0}^2 \alpha_l^{JS}} \cdot \psi_{s,2}^X(d_s) + \psi_{s,3}^X(d_s) \\ = \psi_{s,1}^X(d_s) + \omega_s^{JS} \cdot \psi_{s,2}^X(d_s) + \psi_{s,3}^X(d_s), \end{cases} \quad (3.3)$$

where the $\psi_{s,1}^X(d_s)$, $\psi_{s,2}^X(d_s)$, and $\psi_{s,3}^X(d_s)$ can be obtained trivially by some simple mathe-

Table 1: $\psi_{s,1}^X(d_s)$, $\psi_{s,2}^X(d_s)$, and $\psi_{s,3}^X(d_s)$ of various Z-type weights.

Scheme, WENO-X	$\psi_{s,1}^X(d_s)$	$\psi_{s,2}^X(d_s)$	$\psi_{s,3}^X(d_s)$
WENO-Z	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot \tau_5^2$	0
WENO-Z $\eta(\tau_5)$	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot \frac{(\beta_s^{JS} + \epsilon)^2}{(\eta_s + \epsilon)^2} \cdot \tau_5^2$	0
WENO-Z $\eta(\tau_{81})$	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot \frac{(\beta_s^{JS} + \epsilon)^2}{(\eta_s + \epsilon)^2} \cdot \tau_{81}^2$	0
WENO-Z+	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot (\tau_5 + \epsilon)^2$	$d_s \cdot \lambda \left(\frac{\beta_s + \epsilon}{\tau_5 + \epsilon} \right)$
WENO-ZA	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot \frac{(\beta_s^{JS} + \epsilon)^2}{(\beta_s^{ZA} + \epsilon)} \cdot A \tau_6$	0
WENO-D	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot \Phi \tau_5^p \cdot (\beta_s^{JS} + \epsilon)^{2-p}$	0
WENO-A	$d_s \cdot B$	$\sum_{l=0}^2 \alpha_l^{JS} \cdot \Phi \tau_5^p \cdot (\beta_s^{JS} + \epsilon)^{2-p} (1-B)$	0
WENO-NIP	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot \tau \cdot \frac{(\beta_s^{JS} + \epsilon)^2}{(\chi_s + \epsilon)^2}$	0
(1) $B = \text{BOOL} \left(\max \left(1, \Phi \left(\frac{\tau_5}{\beta_s + \epsilon} \right)^p \right) = 1 \right)$, Φ is computed by Eq. (2.17);			
(2) the other parameters are the same as in Subsections 2.2 and 2.3.			

mathematical manipulations. As examples, we provide the $\psi_{s,1}^X(d_s)$, $\psi_{s,2}^X(d_s)$, and $\psi_{s,3}^X(d_s)$ of the WENO-Z, WENO-Z $\eta(\tau_5)$, WENO-Z $\eta(\tau_{81})$, WENO-Z+, WENO-ZA, WENO-D, WENO-A and WENO-NIP schemes in Table 1.

Furthermore, we give a uniform formula of the nonlinear weights of all three versions of the odd-order WENO schemes, taking the form

$$\begin{cases} \omega_s^X = \frac{\alpha_s^X}{\sum_{l=0}^2 \alpha_l^X}, & s = 0, 1, 2, \\ \alpha_s^X = \left(g^{\text{GMX}} \right)_s (\omega_s^Y) = \psi_{s,1}^X(d_s) + \mathcal{H}^X(\omega_s^Y, d_s) \cdot \psi_{s,2}^X(d_s) + \psi_{s,3}^X(d_s), \end{cases} \quad (3.4)$$

where ω_s^Y is the nonlinear weights of some classical WENO scheme, and $\omega_s^Y = \omega_s^{JS}$ is used usually. Also, $\psi_{s,i}^X(d_s)$, $i = 1, 2, 3$ and $\mathcal{H}^X(\omega_s^Y)$ can easily be obtained. As examples, we provide the $\psi_{s,i}^X(d_s)$, $i = 1, 2, 3$ and $\mathcal{H}^X(\omega_s^Y)$ of the WENO-JS, WENO-M, WENO-Z and WENO-Z+ schemes in Table 2.

Definition 3.2 (Generalized Mapping Function). We call $(g^{\text{GMX}})_s(\omega_s^Y)$ in Eq. (3.4) the Generalized Mapping Function of WENO-X, and for simplicity, $(g^{\text{GMX}})_s(\omega)$ is used without causing any confusion.

Remark 3.1. It can be observed that the $\psi_{s,2}^X(d_s)$ is still complex. However, the primary purpose of rewriting the Z-type weights in the form of Eq. (3.3) or (3.4) is just to explain the fact that one can treat ω_s^X as a function of ω_s^{JS} and then to introduce the definition

Table 2: $\psi_{s,i}^X(d_s), i=1,2,3$ and $\mathcal{H}^X(\omega_s^Y)$ for WENO-JS, WENO-M, WENO-Z and WENO-Z+.

Scheme, WENO-X	ω_s^Y	$\psi_{s,1}^X(d_s)$	$\psi_{s,2}^X(d_s)$	$\psi_{s,3}^X(d_s)$	$\mathcal{H}^X(\omega_s^Y, d_s)$
WENO-JS	ω_s^{JS}	0	1	0	ω_s^Y
WENO-M	ω_s^{JS}	d_s	1	0	$(g^M)_s(\omega_s^Y)$, see [29]
WENO-Z	ω_s^{JS}	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot \tau_5^2$	0	ω_s^Y
WENO-Z+	ω_s^{JS}	d_s	$\sum_{l=0}^2 \alpha_l^{JS} \cdot (\tau_5 + \epsilon)^2$	$d_s \cdot \lambda \left(\frac{\beta_s + \epsilon}{\tau_5 + \epsilon} \right)$	ω_s^Y

3.2 leading to a concise description of Algorithm 1 below. It should be noted that one can use a much simpler form of Eq. (3.3) or (3.4) without performing $\psi_{s,2}^X(d_s)$ in practical programming.

Corollary 3.1. *If WENO-X is a traditional mapped WENO scheme, then its generalized mapping function is exactly its designed mapping function; if WENO-X is a WENO-Z-type WENO scheme, then its generalized mapping function is the non-normalized form of the associated IMR.*

The proof of Corollary 3.1 is very trivial and we omit it here just for the sake of brevity.

4 Design and properties of WENO-Z-type schemes with order-preserving generalized mappings

4.1 The new WENO-Z-type schemes

In order to provide convenience to the reader and clarify our major concern, we firstly restate the following notations and preliminaries, originally proposed in [36], that will be needed for the present study.

Definition 4.1. *Construct the Min.DIST function taking the form*

$$\begin{cases} \text{Min.DIST}(v_0, \dots, v_{r-1}; \bar{\omega}_0, \dots, \bar{\omega}_{r-1}; \omega) = v_{l^*}, \\ l^* = \min \left(\text{INDEX} \left(\min \left\{ |\omega - \bar{\omega}_0|, \dots, |\omega - \bar{\omega}_{r-1}| \right\} \right) \right), \end{cases} \quad (4.1)$$

where $\bar{\omega}_s$ stands for the ideal weight and

$$\text{INDEX} \left(\min \left\{ |\omega - \bar{\omega}_0|, \dots, |\omega - \bar{\omega}_{r-1}| \right\} \right) = \{q_1, \dots, q_M\}, \quad (4.2)$$

if

$$\min \left\{ |\omega - \bar{\omega}_0|, \dots, |\omega - \bar{\omega}_{r-1}| \right\} = |\omega - \bar{\omega}_{q_1}| = \dots = |\omega - \bar{\omega}_{q_M}|.$$

Definition 4.2. Denote $\mathcal{S} = \{d_0, \dots, d_{r-1}\}$ and $\widehat{\mathcal{S}} = \{\widehat{d}_0, \dots, \widehat{d}_{r-1}\}$, and they satisfy:

- (1) the elements of \mathcal{S} and $\widehat{\mathcal{S}}$ are only different in turn;
- (2) $0 < \widehat{d}_0 < \dots < \widehat{d}_{r-1} < 1$.

Let

$$\mathcal{G} = \left\{ (g^{\text{GMX}})_0(\omega), \dots, (g^{\text{GMX}})_{r-1}(\omega) \right\},$$

we define

$$\widehat{\mathcal{G}} = \left\{ (\widehat{g^{\text{GMX}}})_0(\omega), \dots, (\widehat{g^{\text{GMX}}})_{r-1}(\omega) \right\},$$

where $(\widehat{g^{\text{GMX}}})_s(\omega)$ is computed by substituting \widehat{d}_s into the corresponding $(g^{\text{GMX}})_s(\omega)$.

Lemma 4.1. Assume $\widehat{d}_{-1} = 0, \widehat{d}_r = 1$, and denote $\ddot{d}_{-1} = \widehat{d}_{-1}, \ddot{d}_0 = (\widehat{d}_0 + \widehat{d}_1)/2, \dots, \ddot{d}_{r-2} = (\widehat{d}_{r-2} + \widehat{d}_{r-1})/2, \ddot{d}_{r-1} = \widehat{d}_r$, then we have

$$\min \left(\text{INDEX} \left(\min \left\{ |\omega - \widehat{d}_0|, \dots, |\omega - \widehat{d}_{r-1}| \right\} \right) \right) = j \quad \text{for } \forall \omega \in (\ddot{d}_{j-1}, \ddot{d}_j].$$

Proof. We refer to the proof of the Lemma 3 in [36]. □

Lemma 4.2. Let

$$\Omega_i = \left\{ \omega \mid \mathbf{Min_DIST}(\widehat{d}_0, \dots, \widehat{d}_{r-1}; \widehat{d}_0, \dots, \widehat{d}_{r-1}; \omega) = \widehat{d}_i \right\}$$

and $a, b \in \{0, 1, \dots, r-1\}$, if $a > b$, then for $\forall \omega_\alpha \in \Omega_a$ and $\forall \omega_\beta \in \Omega_b$, one can get

- C1. $\Omega_i = (\ddot{d}_{i-1}, \ddot{d}_i]$;
- C2. $\omega_\alpha > \omega_\beta, \widehat{d}_\alpha > \widehat{d}_\beta$;
- C3. $(\widehat{g^{\text{GMX}}})_a(\omega_\alpha) > (\widehat{g^{\text{GMX}}})_b(\omega_\beta)$.

Proof. According to Definition 4.1 and Lemma 4.1, we can prove C1 trivially. As $a > b$, then Ω_b is on the left of Ω_a . It is already given that $\omega_\alpha \in \Omega_a, \omega_\beta \in \Omega_b$, so

$$\omega_\alpha > \omega_\beta, \quad \widehat{d}_\alpha > \widehat{d}_\beta. \tag{4.3}$$

Thus, the C2 is proved.

The proof of C3 is case by case. For simplicity but without loss of generality, we present the proof for the case of "WENO-X = WENO-Z" as an example. According to Eq. (3.4), we can get

$$(\widehat{g^{\text{GMZ}}})_a(\omega_\alpha) = \psi_1^Z(\widehat{d}_\alpha) + \mathcal{H}^Z(\omega_\alpha, \widehat{d}_\alpha) \cdot \psi_2^Z(\widehat{d}_\alpha) + \psi_3^Z(\widehat{d}_\alpha), \tag{4.4a}$$

$$(\widehat{g^{\text{GMZ}}})_b(\omega_\beta) = \psi_1^Z(\widehat{d}_\beta) + \mathcal{H}^Z(\omega_\beta, \widehat{d}_\beta) \cdot \psi_2^Z(\widehat{d}_\beta) + \psi_3^Z(\widehat{d}_\beta). \tag{4.4b}$$

From Table 2, we can obtain

$$\mathcal{H}^Z(\omega_\alpha, \widehat{d}_\alpha) = \omega_\alpha, \quad \mathcal{H}^Z(\omega_\beta, \widehat{d}_\beta) = \omega_\beta, \tag{4.5a}$$

$$\psi_1^Z(\widehat{d}_\alpha) = \widehat{d}_\alpha, \quad \psi_1^Z(\widehat{d}_\beta) = \widehat{d}_\beta, \tag{4.5b}$$

$$\psi_2^Z(\widehat{d}_\alpha) = \psi_2^Z(\widehat{d}_\beta) = \sum_{l=0}^2 \alpha_l^{\text{JS}} \cdot \tau_5^2 > 0, \tag{4.5c}$$

$$\psi_3^Z(\widehat{d}_\alpha) = \psi_3^Z(\widehat{d}_\beta) = 0. \tag{4.5d}$$

Then, from Eqs. (4.3), (4.4) and (4.5), we get

$$\widehat{(g^{\text{GMZ}})}_a(\omega_\alpha) > \widehat{(g^{\text{GMZ}})}_b(\omega_\beta).$$

Thus, C3 for the case of “WENO-X = WENO-Z” is proved. The other cases can also be proved analogously and they are omitted here to save space. Actually, from Fig. 2 below, it can be intuitively observed from the IMR curve of MOP-GMWENO-X, where MOP-GMWENO-X stands for the new WENO-Z-type scheme derived from the WENO-X scheme taking the OP property and we will present it in the following. \square

It has been reported [34, 36] that the OP property of the mapping plays an essential role in preserving high resolutions and meanwhile avoiding spurious oscillations for long-run calculations. Now, we extend the concept of the order-preserving mapping (see Definition 2 of [34] or Definition 1 of [36]) to the generalized mapping function as follows.

Definition 4.3. For $\forall m > n$, and $\omega_{l_m} \in \Omega_m, \omega_{l_n} \in \Omega_n$, if

$$(g^{\text{GMX}})_{l_m}(\omega_{l_m}) > (g^{\text{GMX}})_{l_n}(\omega_{l_n}), \tag{4.6}$$

then the set of generalized mappings $\{(g^{\text{GMX}})_s(\omega), s=0, \dots, r-1\}$ is called to be order-preserving (OP) mapping. Otherwise, it is called to be non-order-preserving (non-OP) mapping.

In Algorithm 1, we introduce the OP property into the previously published WENO-Z-type schemes whose generalized mappings are non-OP. Before giving Algorithm 1, we rewrite the generalized mapping $(g^{\text{GMX}})_s(\omega)$ defined in Eq. (3.4) in a more meaningful form

$$(g^{\text{GMX}})_s(\omega_s^{\text{JS}}) = g^{\text{GMX}}(\omega_s^{\text{JS}}; \psi_{s,1}, \psi_{s,2}, \psi_{s,3}). \tag{4.7}$$

Theorem 4.1. $\{(g^{\text{MOP-GMX}})_s(\omega_s^{\text{JS}}), s=0, 1, \dots, r-1\}$ computed by Algorithm 1 is OP.

Proof. Without loss of generality, assume $\omega_m^{\text{JS}}, \omega_n^{\text{JS}} \in [0, 1]$ and $\omega_m^{\text{JS}} \in \Omega_{l_m^*}, \omega_n^{\text{JS}} \in \Omega_{l_n^*}$, then

$$\begin{cases} (g^{\text{MOP-GMX}})_m(\omega_m^{\text{JS}}) = g^{\text{GMX}}(\omega_m^{\text{JS}}; \psi_{l_m^*,1}, \psi_{l_m^*,2}, \psi_{l_m^*,3}), \\ (g^{\text{MOP-GMX}})_n(\omega_n^{\text{JS}}) = g^{\text{GMX}}(\omega_n^{\text{JS}}; \psi_{l_n^*,1}, \psi_{l_n^*,2}, \psi_{l_n^*,3}). \end{cases}$$

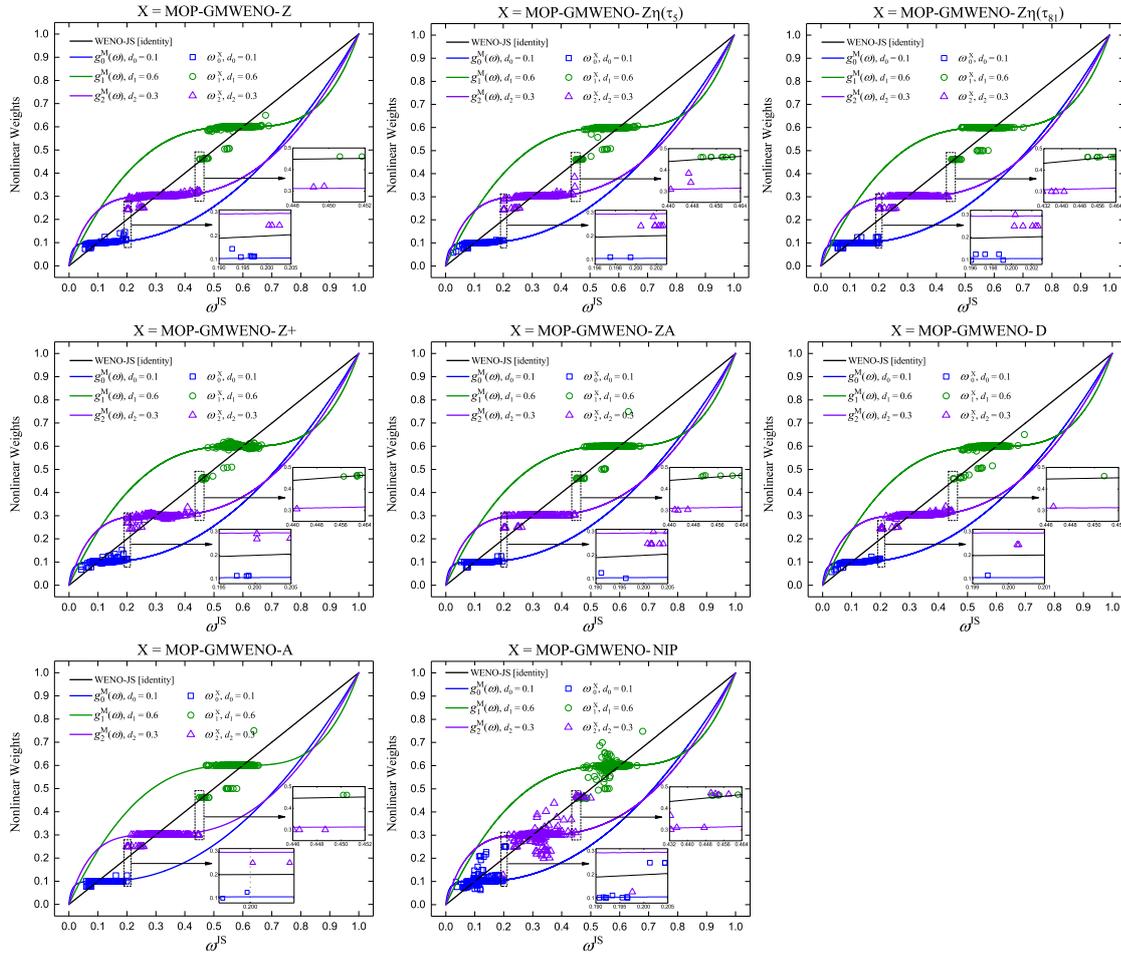


Figure 2: The IMRs for various MOP-GMWENO-X schemes.

Trivially, we have

$$\begin{cases} g^{\text{GMX}}(\omega_m^{\text{JS}}; \psi_{l^*,1}, \psi_{l^*,2}, \psi_{l^*,3}) = (\widehat{g^{\text{GMX}}})_{l^*}(\omega_m^{\text{JS}}), \\ g^{\text{GMX}}(\omega_n^{\text{JS}}; \psi_{l^*,1}, \psi_{l^*,2}, \psi_{l^*,3}) = (\widehat{g^{\text{GMX}}})_{l^*}(\omega_n^{\text{JS}}). \end{cases}$$

Up to now, the proof can be finished easily as Lemma 4.2 holds true. □

Now, we give the improved Z-type weights satisfying the *OP* property by

$$\omega_s^{\text{MOP-GMX}} = \frac{\alpha_s^{\text{MOP-GMX}}}{\sum_{l=0}^{r-1} \alpha_l^{\text{MOP-GMX}}}, \quad \alpha_s^{\text{MOP-GMX}} = (g^{\text{MOP-GMX}})_s(\omega_s^{\text{JS}}), \quad s=0, \dots, r-1, \quad (4.8)$$

Algorithm 1: The method to get new Z-type weights satisfying *OP* property.

input : $s, d_s, \omega_s^{\text{JS}}, \psi_{s,m}$ with $m = 1, 2, 3$
output: the new *OP* generalized mappings, namely
 $\left\{ (g^{\text{MOP-GMX}})_s(\omega_s^{\text{JS}}), s = 0, 1, \dots, r-1 \right\}$

- 1 $(g^{\text{GMX}})_s(\omega)$ is a generalized mapping function, $\left\{ (g^{\text{GMX}})_s(\omega), s = 0, 1, \dots, r-1 \right\}$ is non-*OP*;
- 2 set $s = 0$;
- 3 **while** $s \leq r-1$ **do**
- 4 compute ω_s^{JS} ;
- 5 set $j = 1, d_{\text{MIN}} = |\omega_s^{\text{JS}} - d_0|, l_s^* = 0$;
- 6 **while** $j \leq r-1$ **do**
- 7 **while** $-d_{\text{MIN}} < \omega_s^{\text{JS}} - d_j < d_{\text{MIN}}$ **do**
- 8 $d_{\text{MIN}} = |\omega_s^{\text{JS}} - d_j|$;
- 9 $l_s^* = j$;
- 10 **end**
- 11 $j++$;
- 12 **end**
- 13 set $m = 1$;
- 14 **while** $m \leq 3$ **do**
- 15 $\bar{\psi}_{s,m} = \psi_{l_s^*, m}$;
- 16 $m++$;
- 17 **end**
- 18 $s++$;
- 19 **end**
- 20 set $s = 0$;
- 21 **while** $s \leq r-1$ **do**
- 22 $(g^{\text{MOP-GMX}})_s(\omega_s^{\text{JS}}) = g^{\text{GMX}}(\omega_s^{\text{JS}}; \bar{\psi}_{s,1}, \bar{\psi}_{s,2}, \bar{\psi}_{s,3})$;
- 23 $s++$;
- 24 **end**

where $(g^{\text{MOP-GMX}})_s(\omega_s^{\text{JS}})$ is determined by Algorithm 1 and the resultant scheme is named MOP-GMWENO- χ .

In Fig. 2, we plot the profiles of the generalized mappings. It can be seen that, for the improved WENO-Z-type schemes:

- (1) the *OP* property is obtained in general;
- (2) the widths of the “optimal weight intervals” are decreased compared to the WENO-

X schemes.

Next, we conduct numerical examples to show the convergence properties of the MOP-GMWENO-X schemes.

4.2 Convergence at critical points

Since Henrick et al. [29] pointed out that the WENO-JS scheme achieves only third-order convergence rate of accuracy at critical points of smooth solutions, it has become a focus of discussion [30–34, 36–38, 44].

We compute the function $f(x) = x^3 + \cos(x)$ to test the convergence property of the MOP-GMWENO-X schemes at critical points. It is trivial to know that $f'_{x=0} = 0$ and $f'''_{x=0} \neq 0$. For comparison purpose, the WENO-X schemes, as well as WENO-JS and WENO5 using ideal linear weights (denoted as WENO5-ILW in this paper for simplicity), are also conducted.

Table 3 shows the L_∞ convergence behaviors for the considered schemes at the critical point $x = 0$. As expected, the WENO-JS scheme only gets third-order accuracy. It should be noted that, in this test, the WENO-Z+ scheme can also only obtain third-order accuracy (in consistency with the results reported in [10]), leading to the fact that the MOP-GMWENO-Z+ scheme only gets third-order accuracy. Moreover, the WENO-JS, WENO-Z+ and MOP-GMWENO-Z+ schemes obtain the errors of at least 4 orders of magnitude larger than those of the WENO-ILW scheme. The other WENO-X schemes, say, WENO-Z, WENO-Z $\eta(\tau_5)$, WENO-Z $\eta(\tau_{81})$, WENO-ZA, WENO-D, WENO-A, WENO-NIP, and their MOP-GMWENO-X schemes can get fifth-order accuracy, but only the WENO-Z $\eta(\tau_{81})$, WENO-ZA, WENO-A, WENO-NIP schemes, and their MOP-GMWENO-X schemes, can obtain the errors with the same order of magnitude as those of the WENO-ILW scheme. The errors generated by the WENO-Z, WENO-Z $\eta(\tau_5)$, WENO-D schemes, and their MOP-GMWENO-X schemes, are about 3 orders of magnitude larger than those of the WENO-ILW scheme. It is worthy to note that all the new WENO schemes perform similarly as the WENO-X schemes for this test.

4.3 Long-run simulations of 1D linear advection equation for comparison

4.3.1 With high-order critical points

For the purpose of demonstrating the improvement of the new WENO schemes that they can preserve high resolutions for the problem with high-order critical points at long output times, we perform the following test.

Example 4.1. We compute

$$u_t + u_x = 0, \quad x \in (7.5, 10.5),$$

with the following initial condition

$$u(x, 0) = \exp\left(- (x - 9.0)^{10}\right) \cos^9\left(\pi(x - 9.0)\right). \quad (4.9)$$

Table 3: Convergence rate of accuracy at the critical point.

Δx	WENO5-ILW		WENO-JS		WENO-Z		MOP-GMWENO-Z	
	Error	Order	Error	Order	Error	Order	Error	Order
0.0125	2.72851e-11	-	3.57689e-06	-	5.36240e-08	-	5.36245e-08	-
0.00625	1.11260e-12	4.6161	3.64804e-07	3.2935	1.14381e-09	5.5510	1.14381e-09	5.5510
0.003125	3.47692e-14	5.0000	3.81177e-08	3.2586	2.50723e-11	5.5116	2.50723e-11	5.5116
0.0015625	1.08654e-15	5.0000	3.99276e-09	3.2550	5.58613e-13	5.4881	5.58613e-13	5.4881
Δx	WENO-Z $\eta(\tau_5)$		MOP-GMWENO-Z $\eta(\tau_5)$		WENO-Z $\eta(\tau_{81})$		MOP-GMWENO-Z $\eta(\tau_{81})$	
	Error	Order	Error	Order	Error	Order	Error	Order
0.0125	5.23932e-08	-	5.23937e-08	-	3.50808e-11	-	3.65801e-11	-
0.00625	1.11430e-09	5.5552	1.11430e-09	5.5552	1.11260e-12	4.9787	1.11260e-12	5.0391
0.003125	2.43877e-11	5.5138	2.43877e-11	5.5138	3.47692e-14	5.0000	3.47692e-14	5.0000
0.0015625	5.42433e-13	5.4906	5.42433e-13	5.4906	1.08654e-15	5.0000	1.08654e-15	5.0000
Δx	WENO-Z+		MOP-GMWENO-Z+		WENO-ZA		MOP-GMWENO-ZA	
	Error	Order	Error	Order	Error	Order	Error	Order
0.0125	6.57316e-07	-	6.57315e-07	-	1.07615e-10	-	1.08743e-10	-
0.00625	8.77288e-08	2.9055	8.77288e-08	2.9055	1.50193e-12	6.1629	1.50193e-12	6.1780
0.003125	1.08696e-08	3.0128	1.08696e-08	3.0128	3.69247e-14	5.3461	3.69247e-14	5.3461
0.0015625	1.29563e-09	3.0686	1.29563e-09	3.0686	1.09865e-15	5.0708	1.09865e-15	5.0708
Δx	WENO-D		MOP-GMWENO-D		WENO-A		MOP-GMWENO-A	
	Error	Order	Error	Order	Error	Order	Error	Order
0.0125	5.36240e-08	-	5.36245e-08	-	3.62665e-11	-	3.67745e-11	-
0.00625	1.14328e-09	5.5516	1.14328e-09	5.5516	1.11260e-12	5.0266	1.11260e-12	5.0467
0.003125	2.50723e-11	5.5109	2.50723e-11	5.5109	3.47692e-14	5.0000	3.47692e-14	5.0000
0.0015625	5.58613e-13	5.4881	5.58613e-13	5.4881	1.08654e-15	5.0000	1.08654e-15	5.0000
Δx	WENO-NIP		MOP-GMWENO-NIP					
	Error	Order	Error	Order				
0.0125	3.56050e-11	-	3.56511e-11	-				
0.00625	1.11264e-12	5.0000	1.11264e-12	5.0019				
0.003125	3.47695e-14	5.0000	3.47695e-14	5.0000				
0.0015625	1.08654e-15	5.0000	1.08654e-15	5.0000				

Here, we set $CFL = (\Delta x)^{2/3}$.

To compare the dissipations, the following L_1 and L_∞ errors are computed

$$L_1 = h \cdot \sum_{i=1}^N |u_i^{\text{exact}} - (u_h)_i|, \tag{4.10a}$$

$$L_\infty = \max_{1 \leq i \leq N} |u_i^{\text{exact}} - (u_h)_i|, \tag{4.10b}$$

where N is the number of the cells, h the mesh size, $(u_h)_i$ the computing result and u_i^{exact} the exact solution that can be easily computed by

$$u(x, t) = \exp\left(-((x-t)-9.0)^{10}\right) \cos^9\left(\pi((x-t)-9.0)\right).$$

For the scheme ‘‘Y’’, its increased errors compared to WENO5-ILW are also been computed by

$$\chi_1 = \frac{L_1^Y(t) - L_1^{\text{ILW}}(t)}{L_1^{\text{ILW}}(t)} \times 100\%, \quad \chi_\infty = \frac{L_\infty^Y(t) - L_\infty^{\text{ILW}}(t)}{L_\infty^{\text{ILW}}(t)} \times 100\%,$$

where $L_1^{\text{ILW}}(t)$ and $L_\infty^{\text{ILW}}(t)$ stand for the L_1 and L_∞ errors of WENO5-ILW respectively.

In Table 4, we show the results computed by considered schemes with $N = 300$ and $t = 300, 600, 900, 1200$. It can be seen that: (1) WENO-JS produces the largest L_1 and L_∞ errors, leading to the largest increased errors, among all considered schemes for all output times; (2) the WENO-Z, WENO-Z $\eta(\tau_5)$, WENO-Z+, WENO-D and WENO-A schemes also generate very large L_1 and L_∞ errors, although slightly smaller than the WENO-JS scheme, and their associated increased errors are extremely large naturally; (3) however, the MOP-GMWENO-Z, MOP-GMWENO-Z $\eta(\tau_5)$, MOP-GMWENO-Z+, MOP-GMWENO-D and MOP-GMWENO-A schemes can significantly decrease the increased errors to a tolerable level, as they can achieve much smaller L_1 and L_∞ errors that actually get very close to that of WENO-ILW; (4) moreover, the MOP-GMWENO-Z $\eta(\tau_{81})$, MOP-GMWENO-ZA and MOP-GMWENO-NIP schemes can also ensure that their L_1 and L_∞ errors get close to that of WENO-ILW, and thus their increased errors are also at a tolerable level. It seems that the WENO-Z $\eta(\tau_{81})$, WENO-ZA and WENO-NIP schemes can get solutions almost as accurate as, or even more accurate than, that of WENO-ILW, and this is good for this test. However, it should be pointed out that these schemes may suffer from lack of robustness as their dissipations are too small for long-run calculations. Indeed, we will demonstrate this in detail by numerical examples below.

Fig. 3 shows the solutions at $t = 1200$. For comparison purpose, we also plot the results computed by the WENO-JS [22] and WENO-M [29] schemes. From Fig. 3, it can be observed that: (1) WENO-JS shows the lowest resolution, followed by WENO-M; (3) the WENO-Z, WENO-Z $\eta(\tau_5)$, WENO-Z+, WENO-D and WENO-A schemes also show very low resolutions but the improved MOP-GMWENO-X schemes can significantly improve the resolutions; (4) the resolutions of MOP-GMWENO-Z $\eta(\tau_{81})$, MOP-GMWENO-ZA and MOP-GMWENO-NIP are slightly smaller than that of the WENO-Z $\eta(\tau_{81})$, WENO-ZA and WENO-NIP schemes but they are still far better than the WENO-JS, WENO-M, and the other WENO-Z-type schemes.

4.3.2 With discontinuities

To examine the major benefit of the new schemes that they are able to achieve high resolutions and meanwhile avoid spurious oscillations for long-output-time simulations, we solve

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial u}{\partial x} = 0, & -1 < x < 1, \\ u(x, 0) = u_0(x), \end{cases}$$

with two different $u_0(x)$.

Example 4.2 (Case 1). The initial condition $u_0(x)$ is computed by

$$u_0(x) = \begin{cases} 1, & -1 < x \leq 0, \\ 0, & 0 < x < 1. \end{cases} \quad (4.11)$$

This problem simply consists of two constant states separated by sharp discontinuities at $x = 0.0, \pm 1.0$.

Table 4: L_1 , L_∞ errors and the increased errors (in percentage) on solving Example 4.1.

Time, t	WENO5-ILW				WENO-JS			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	5.39974E-03	-	8.81363E-03	-	7.93589E-02	1370%	1.34321E-01	1424%
600	9.94133E-03	-	1.50917E-02	-	2.10016E-01	2013%	3.04860E-01	1920%
900	1.38061E-02	-	1.96281E-02	-	2.84632E-01	1962%	4.20080E-01	2040%
1200	1.74067E-02	-	2.39652E-02	-	3.26687E-01	1777%	5.14072E-01	2045%
Time, t	WENO-Z				MOP-GMWENO-Z			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	3.56420e-02	560%	1.23424e-01	1300%	1.18556e-02	120%	2.21312e-02	151%
600	8.58998e-02	777%	2.36754e-01	1530%	1.56335e-02	60%	2.43793e-02	68%
900	1.13762e-01	726%	2.77206e-01	1324%	2.57924e-02	87%	3.33097e-02	71%
1200	1.27002e-01	631%	2.78133e-01	1070%	2.51627e-02	45%	4.06869e-02	71%
Time, t	WENO-Z η (τ_5)				MOP-GMWENO-Z η (τ_5)			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	4.52391e-02	738%	1.14337e-01	1197%	1.16734e-02	116%	2.42074e-02	175%
600	9.44265e-02	864%	2.36673e-01	1530%	1.56751e-02	60%	2.51991e-02	74%
900	1.09605e-01	695%	2.12852e-01	993%	2.52547e-02	83%	3.20434e-02	65%
1200	1.38434e-01	697%	3.05553e-01	1185%	2.15191e-02	24%	3.08777e-02	30%
Time, t	WENO-Z η (τ_{81})				MOP-GMWENO-Z η (τ_{81})			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	5.57811e-03	3%	9.60994e-03	9%	9.38301e-03	74%	1.47213e-02	67%
600	1.04455e-02	7%	1.50933e-02	4%	1.70600e-02	74%	2.32755e-02	60%
900	1.47709e-02	7%	2.00720e-02	3%	2.09101e-02	52%	3.42640e-02	76%
1200	1.83129e-02	5%	2.40373e-02	1%	2.66533e-02	54%	3.22354e-02	36%
Time, t	WENO-Z+				MOP-GMWENO-Z+			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	5.09235e-02	843%	1.59585e-01	1710%	9.45604e-03	75%	2.50726e-02	184%
600	8.20317e-02	737%	1.84579e-01	1171%	1.50895e-02	54%	3.40525e-02	135%
900	1.03748e-01	653%	1.76101e-01	804%	2.18063e-02	58%	6.96188e-02	258%
1200	1.15534e-01	565%	2.22997e-01	838%	3.00894e-02	73%	5.66315e-02	138%
Time, t	WENO-ZA				MOP-GMWENO-ZA			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	5.10578e-03	-5%	8.86265e-03	1%	8.87687e-03	64%	1.54090e-02	75%
600	9.05646e-03	-8%	1.48176e-02	2%	1.82371e-02	86%	2.52104e-02	74%
900	1.41249e-02	3%	2.36763e-02	22%	1.99799e-02	45%	3.82169e-02	96%
1200	2.32315e-02	34%	6.80578e-02	186%	2.81372e-02	62%	3.76059e-02	58%
Time, t	WENO-D				MOP-GMWENO-D			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	3.56420e-02	560%	1.23424e-01	1300%	1.18556e-02	120%	2.21312e-02	151%
600	8.58998e-02	777%	2.36754e-01	1530%	1.56335e-02	60%	2.43793e-02	68%
900	1.13762e-01	726%	2.77206e-01	1324%	2.57924e-02	87%	3.33097e-02	71%
1200	1.27002e-01	631%	2.78133e-01	1070%	2.51627e-02	45%	4.06869e-02	71%
Time, t	WENO-A				MOP-GMWENO-A			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	1.18154e-01	2088%	2.14177e-01	2330%	9.18713e-03	70%	1.47167e-02	67%
600	1.33707e-01	1265%	2.38106e-01	1540%	1.68488e-02	72%	2.48995e-02	71%
900	1.50686e-01	993%	2.33834e-01	1101%	2.07575e-02	51%	3.50088e-02	80%
1200	1.56130e-01	799%	2.35669e-01	891%	2.41207e-02	39%	3.23873e-02	36%
Time, t	WENO-NIP				MOP-GMWENO-NIP			
	L_1 error	χ_1	L_∞ error	χ_∞	L_1 error	χ_1	L_∞ error	χ_∞
300	5.06954e-03	-6%	8.83927e-03	0%	9.18661e-03	70%	1.14320e-02	30%
600	8.65059e-03	-12%	1.48766e-02	2%	1.51265e-02	54%	2.28329e-02	57%
900	1.13469e-02	-18%	1.91798e-02	-2%	1.72611e-02	25%	2.19778e-02	13%
1200	1.36442e-02	-21%	2.29721e-02	-3%	2.13328e-02	23%	2.68344e-02	13%

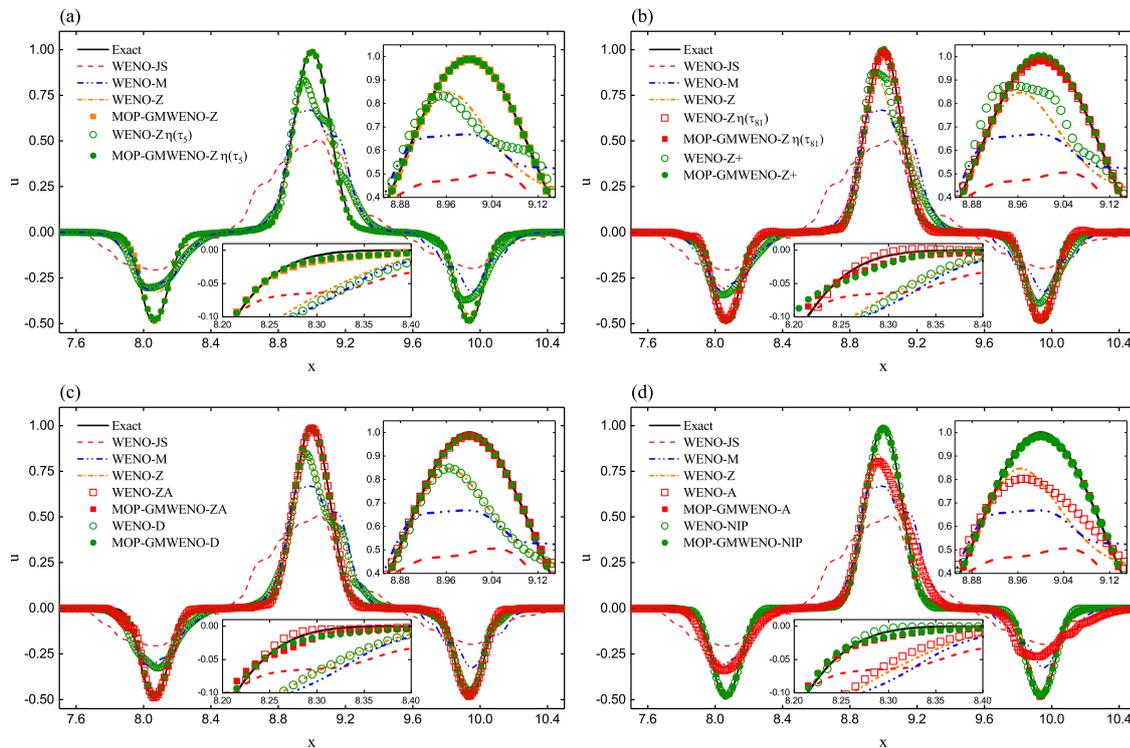


Figure 3: The solutions on solving Example 4.1.

Example 4.3 (Case 2). The SLP defined by Eq. (3.2).

In order to test the convergence properties, we compute both Case 1 and Case 2 to the final time $t=2000$ with the CFL=0.1. For the purpose of comparison, we also solve Case 1 and Case 2 by WENO-JS and WENO5-ILW.

In Tables 5, 6, the L_1 , L_∞ errors and the convergence orders are presented. It can be observed that: (1) because of its highest dissipation, the WENO-JS scheme generates significantly larger numerical errors than all other schemes, leading to the smallest L_1 convergence orders; (2) the L_1 convergence orders of the MOP-GMWENO-X schemes are distinctly higher than the WENO-X schemes, and the L_1 errors produced by the MOP-GMWENO-Z, MOP-GMWENO-Z $\eta(\tau_5)$, MOP-GMWENO-Z+, MOP-GMWENO-ZA, MOP-GMWENO-D and MOP-GMWENO-A schemes are slightly smaller than the WENO-X schemes in general; (3) the L_1 errors produced by the MOP-GMWENO-NIP scheme on solving Case 1 are also slightly smaller than the WENO-NIP scheme, while this holds true only for the computing case of $N = 800$ on solving Case 2 as the MOP-GMWENO-NIP scheme generates slightly and evidently larger L_1 errors for the computing cases of $N = 400$ and $N = 200$ respectively; (4) the L_1 errors produced by the MOP-GMWENO-Z $\eta(\tau_{81})$ scheme are slightly larger on solving both Case 1 and Case 2; (5) the L_∞ errors of the MOP-GMWENO-X schemes on solving Case 1 are smaller than

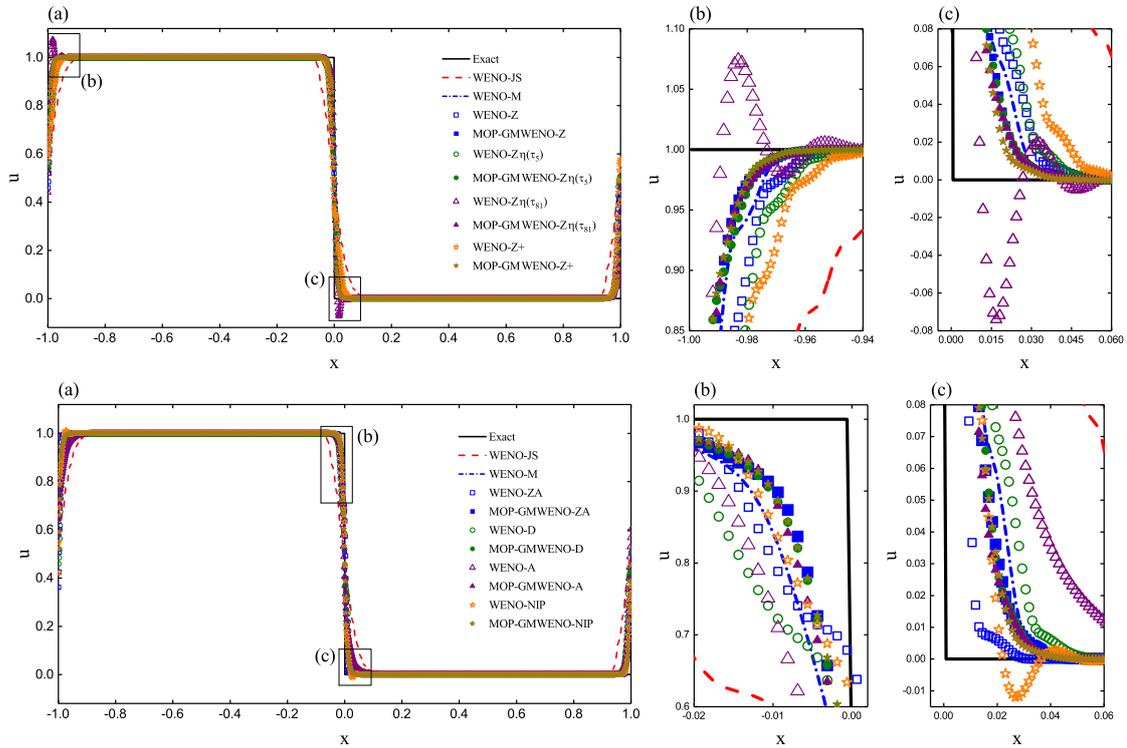


Figure 4: Results of different schemes on solving Example 4.2 (Case 1) with $t=200$, $N=1600$.

the WENO-X schemes; (6) for Case 2, the L_∞ errors of the MOP-GMWENO-X schemes are very close to, or even smaller for many cases than, the WENO-X schemes. In addition, by taking a view of the $x-u$ profiles, we find that the resolutions of the WENO-Z, WENO- $Z\eta(\tau_5)$, WENO-Z+, WENO-ZA, WENO-D and WENO-A schemes are significantly lower than the MOP-GMWENO-Z, MOP-GMWENO- $Z\eta(\tau_5)$, MOP-GMWENO-Z+, MOP-GMWENO-ZA, MOP-GMWENO-D and MOP-GMWENO-A schemes, and the WENO- $Z\eta(\tau_{81})$ and WENO-NIP schemes produce spurious oscillations but the MOP-GMWENO- $Z\eta(\tau_{81})$ and MOP-GMWENO-NIP schemes, as well as all other MOP-GMWENO-X schemes, can remove these oscillations properly. To demonstrate this, we next perform detailed tests and show the solutions carefully.

We re-run both Case 1 and Case 2 by considered WENO schemes. Just for the purpose of providing better illustrations but without loss of generality, we set the output time to be $t=200$ and use the uniform meshes of $N=1600$ and $N=3200$ this time.

Figs. 4 and 6 give the solutions of different schemes with $t=200$, $N=1600$ for Case 1 and Case 2, respectively. It can be observed that: (1) the MOP-GMWENO-Z, MOP-GMWENO- $Z\eta(\tau_5)$, MOP-GMWENO-Z+, MOP-GMWENO-ZA, MOP-GMWENO-D and MOP-GMWENO-A schemes generate no any spurious oscillations and obtain numerical results with significantly better resolutions than the WENO-Z, WENO- $Z\eta(\tau_5)$,

Table 5: Numerical errors and convergence orders of accuracy on Example 4.2 at $t=2000$.

N	WENO5-ILW				WENO-JS			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	1.03240E-01	-	4.67252E-01	-	4.48148E-01	-	5.55748E-01	-
400	5.79848E-02	0.8323	4.70837E-01	-0.0110	3.37220E-01	0.4103	5.77105E-01	-0.0544
800	3.25843E-02	0.8315	4.74042E-01	-0.0098	2.93752E-01	0.1991	5.17829E-01	0.1564
N	WENO-Z				MOP-GMWENO-Z			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.08722e-01	-	4.98364e-01	-	1.18245e-01	-	4.88876e-01	-
400	1.25878e-01	0.7296	5.94047e-01	-0.2534	6.67048e-02	0.8259	5.32787e-01	-0.1241
800	8.56252e-02	0.5559	6.02088e-01	-0.0194	3.79314e-02	0.8144	5.30365e-01	0.0066
N	WENO-Z $\eta(\tau_5)$				MOP-GMWENO-Z $\eta(\tau_5)$			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.35848e-01	-	5.46963e-01	-	1.18115e-01	-	4.82687e-01	-
400	1.34410e-01	0.8112	5.77563e-01	-0.0785	6.57450e-02	0.8452	4.99738e-01	-0.0501
800	8.88830e-02	0.5967	6.10969e-01	-0.0811	3.74280e-02	0.8128	5.14095e-01	-0.0409
N	WENO-Z $\eta(\tau_{81})$				MOP-GMWENO-Z $\eta(\tau_{81})$			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	1.06673e-01	-	5.57758e-01	-	1.16619e-01	-	4.77852e-01	-
400	6.02857e-02	0.8233	5.67245e-01	-0.0243	6.51575e-02	0.8398	4.95518e-01	-0.0524
800	3.50816e-02	0.7811	5.87196e-01	-0.0499	3.70925e-02	0.8128	5.15105e-01	-0.0559
N	WENO-Z+				MOP-GMWENO-Z+			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.35835e-01	-	5.53876e-01	-	9.90921e-02	-	4.99725e-01	-
400	1.46265e-01	0.6892	6.02883e-01	-0.1223	5.76023e-02	0.7826	4.90011e-01	0.0283
800	6.51259e-02	1.1673	5.83954e-01	0.0460	3.24177e-02	0.8293	4.81302e-01	0.0259
N	WENO-ZA				MOP-GMWENO-ZA			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	1.31936e-01	-	5.50979e-01	-	1.17961e-01	-	5.10189e-01	-
400	1.13400e-01	0.2184	6.01658e-01	-0.1269	6.61903e-02	0.8336	5.02192e-01	0.0228
800	6.70946e-02	0.7572	5.18541e-01	0.2145	3.65564e-02	0.8565	4.80328e-01	0.0642
N	WENO-D				MOP-GMWENO-D			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.08722e-01	-	4.98364e-01	-	1.18950e-01	-	4.90750e-01	-
400	1.25884e-01	0.7295	5.94701e-01	-0.2550	6.56664e-02	0.8571	4.81270e-01	0.0281
800	8.52679e-02	0.5620	5.98136e-01	-0.0083	3.80765e-02	0.7863	4.84771e-01	-0.0105
N	WENO-A				MOP-GMWENO-A			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	3.29771e-01	-	5.51867e-01	-	1.16473e-01	-	4.80577e-01	-
400	2.02749e-01	0.7018	5.35326e-01	0.0439	6.65634e-02	0.8072	4.97352e-01	-0.0495
800	1.05379e-01	0.9441	5.54605e-01	-0.0510	3.62328e-02	0.8774	4.87916e-01	0.0276
N	WENO-NIP				MOP-GMWENO-NIP			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	1.19293e-01	-	5.83513e-01	-	1.14420e-01	-	4.85124e-01	-
400	7.46960e-02	0.6754	5.61235e-01	0.0562	6.10704e-02	0.9058	4.82764e-01	0.0070
800	4.56612e-02	0.7101	5.18618e-01	0.1139	3.35626e-02	0.8636	4.75671e-01	0.0214

WENO-Z+, WENO-ZA, WENO-D, WENO-A schemes, and the WENO-JS and WENO-M schemes; (2) the WENO-Z $\eta(\tau_{81})$ and WENO-NIP schemes inevitably generate severe spurious oscillations while the MOP-GMWENO-Z $\eta(\tau_{81})$ and MOP-GMWENO-NIP schemes can avoid these spurious oscillations successfully and at the same time they can achieve considerably high resolutions, as accurate as the other MOP-GMWENO-X schemes.

In Figs. 5 and 7, we give the solutions of different schemes respectively for the case

Table 6: Numerical errors and convergence orders of accuracy on Example 4.3 at $t = 2000$.

N	WENO5-ILW				WENO-JS			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.27171E-01	-	5.14236E-01	-	6.12899E-01	-	7.99265E-01	-
400	1.15918E-01	0.9707	4.77803E-01	0.1060	5.99215E-01	0.0326	8.20493E-01	-0.0378
800	5.35871E-02	1.1131	4.74317E-01	0.0106	5.50158E-01	0.1232	8.14650E-01	0.0103
N	WENO-Z				MOP-GMWENO-Z			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	3.86995e-01	-	6.85835e-01	-	4.51514e-01	-	7.68680e-01	-
400	2.02287e-01	0.9359	5.18993e-01	0.4021	1.76765e-01	1.3529	5.01169e-01	0.6171
800	1.66703e-01	0.2791	5.04564e-01	0.0407	6.44772e-02	1.4550	5.04415e-01	-0.0093
N	WENO-Z $\eta(\tau_5)$				MOP-GMWENO-Z $\eta(\tau_5)$			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	3.24546e-01	-	6.51925e-01	-	4.48811e-01	-	7.65748e-01	-
400	2.31374e-01	0.4882	5.42432e-01	0.2653	1.82206e-01	1.3005	5.47587e-01	0.4838
800	1.55489e-01	0.5734	5.12270e-01	0.0825	6.41444e-02	1.5062	4.98985e-01	0.1341
N	WENO-Z $\eta(\tau_{81})$				MOP-GMWENO-Z $\eta(\tau_{81})$			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.42963e-01	-	6.39818e-01	-	3.77388e-01	-	7.39311e-01	-
400	1.33752e-01	0.8612	6.01344e-01	0.0895	1.61629e-01	1.2234	4.91776e-01	0.5882
800	5.89144e-02	1.1829	5.73819e-01	0.0676	6.41956e-02	1.3321	4.95672e-01	-0.0114
N	WENO-Z+				MOP-GMWENO-Z+			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.99492e-01	-	5.45598e-01	-	2.66825e-01	-	7.08732e-01	-
400	2.42059e-01	0.3072	4.96267e-01	0.1367	1.57975e-01	0.7562	5.38619e-01	0.3960
800	1.48193e-01	0.7079	5.49331e-01	-0.1466	5.99206e-02	1.3986	4.94470e-01	0.1234
N	WENO-ZA				MOP-GMWENO-ZA			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.27174e-01	-	6.35754e-01	-	3.82186e-01	-	7.40853e-01	-
400	2.03221e-01	0.1607	5.99664e-01	0.0843	1.81782e-01	1.0721	4.90704e-01	0.5943
800	1.63892e-01	0.3103	5.32228e-01	0.1721	6.35776e-02	1.5156	4.94734e-01	-0.0118
N	WENO-D				MOP-GMWENO-D			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	3.86995e-01	-	6.85835e-01	-	4.52027e-01	-	7.69911e-01	-
400	2.02287e-01	0.9359	5.18995e-01	0.4021	1.79447e-01	1.3329	5.13353e-01	0.5947
800	1.66552e-01	0.2804	5.04564e-01	0.0407	6.28595e-02	1.5134	4.94367e-01	0.0544
N	WENO-A				MOP-GMWENO-A			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	5.31200e-01	-	7.70910e-01	-	4.53035e-01	-	7.77065e-01	-
400	4.08352e-01	0.3794	6.93282e-01	0.1531	1.73916e-01	1.3812	5.29427e-01	0.5536
800	2.95123e-01	0.4685	5.90637e-01	0.2312	6.35906e-02	1.4515	4.87979e-01	0.1176
N	WENO-NIP				MOP-GMWENO-NIP			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
200	2.40800e-01	-	5.64249e-01	-	4.37570e-01	-	7.64352e-01	-
400	1.34966e-01	0.8352	5.68093e-01	-0.0098	1.58619e-01	1.4639	5.19438e-01	0.5573
800	6.62377e-02	1.0269	5.18541e-01	0.1317	5.80606e-02	1.4499	4.84905e-01	0.0992

of $t = 200$, $N = 3200$. We can see that: (1) with the increase of grid number, the spurious oscillations produced by WENO-Z $\eta(\tau_{81})$ and WENO-NIP appear to be closer to the discontinuities, and the amplitudes of these spurious oscillations seem to become larger, however, MOP-GMWENO-Z $\eta(\tau_{81})$ and MOP-GMWENO-NIP are still able to prevent the spurious oscillations while provide greatly improved resolutions; (2) all the other MOP-GMWENO-X schemes still generate no spurious oscillations and evidently provide much better resolutions than the WENO-X schemes and the WENO-JS and WENO-M schemes.

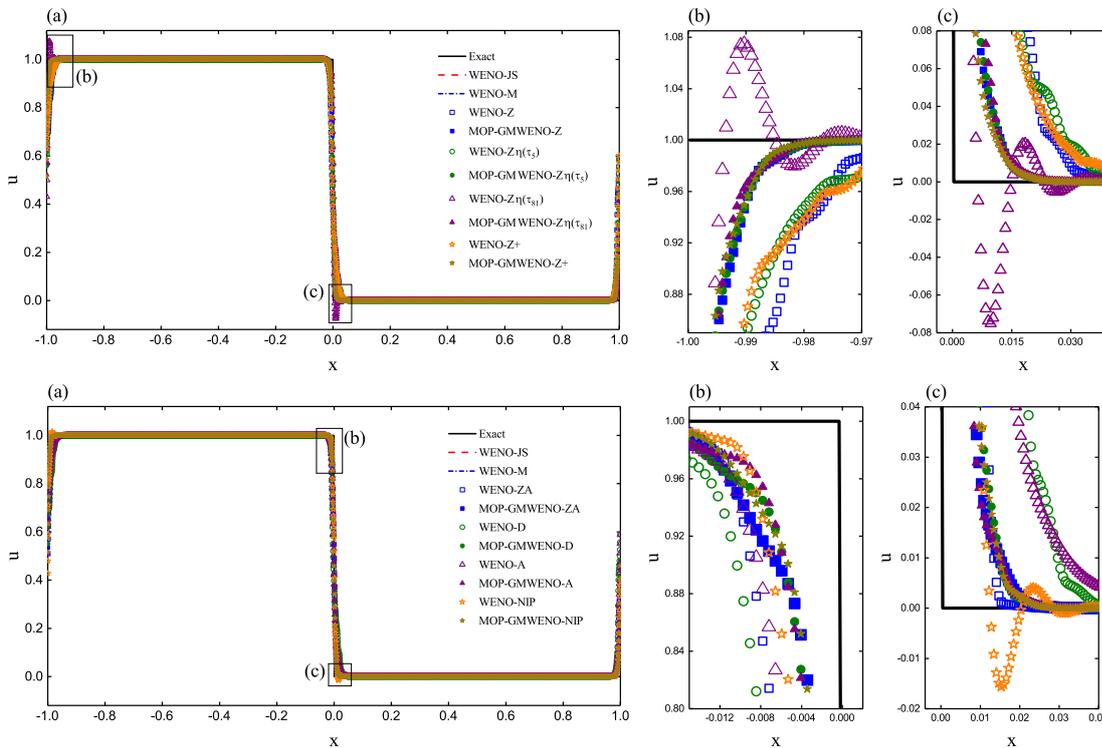


Figure 5: Results of different schemes on solving Example 4.2 (Case 1) with $t = 200$, $N = 3200$.

To summarize, we may conclude that the order-preserving property introduced in the present study, can help the WENO-Z-type schemes to get high resolutions and remove spurious oscillations at the same time for long-run simulations. Although we have only shown the results of 1D linear advection equations so far, it is trivial to get the following conclusions.

Remark 4.1. The technique and algorithm of introducing the order-preserving property proposed previously in this paper can be directly extended to higher dimensions using dimension by dimension approach and to system of equations using the characteristics-wise approach [22].

Remark 4.2. In the same way that many previously published articles did [30–32, 34–37, 42, 43], the long-output-time simulations are only considered for the 1D linear advection equation while not for the nonlinear hyperbolic conservation laws in the present work. As mentioned in [31], the reason is that, for most of the nonlinear hyperbolic conservation laws, only solutions under short output time are available and with output time getting larger the waves will move out of the boundary. However, even if under short output time, introducing the order-preserving property for WENO-Z-type schemes also brings great improvement on simulating the nonlinear hyperbolic conservation laws. In

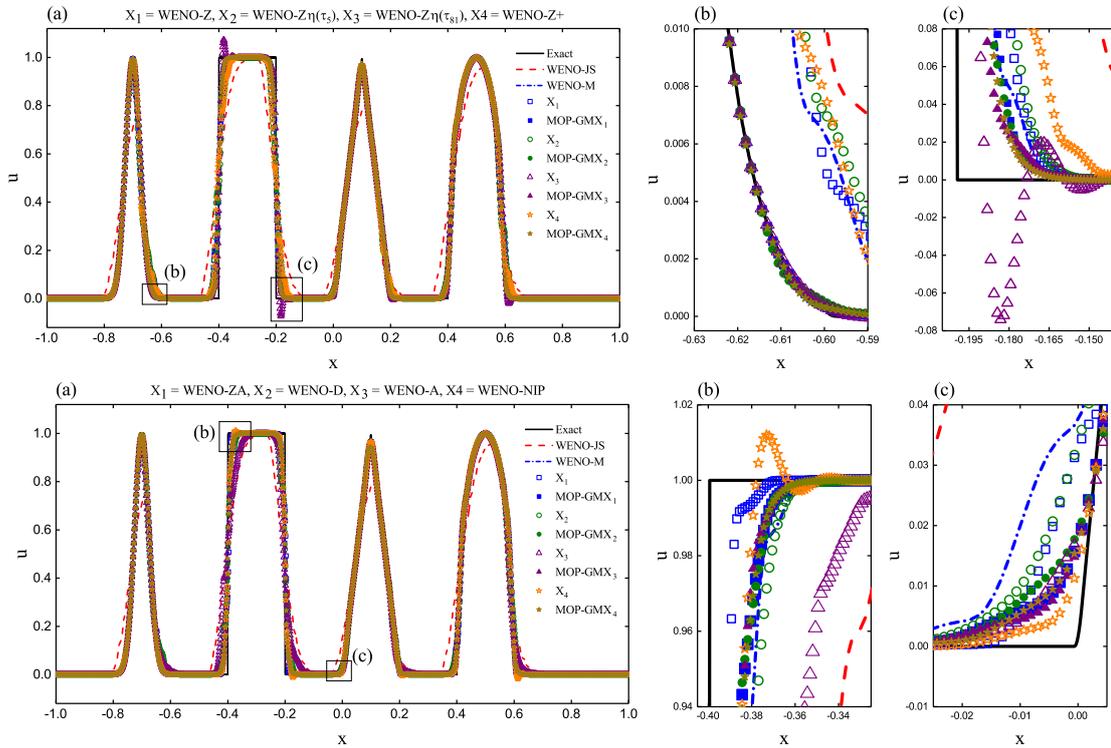


Figure 6: Results of different schemes on solving Example 4.2 (Case 2) with $t=200$, $N=1600$.

Subsection 5.2, we will demonstrate this numerically in detail.

5 Numerical results

5.1 Accuracy test

To test the accuracy of different WENO schemes, we consider the smooth density perturbation advection problem [12]. It is governed by the one-dimensional Euler equation

$$\begin{cases} \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} = 0, \\ \mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ E \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ u(E+p) \end{pmatrix}, \end{cases} \quad (5.1)$$

where ρ is the density, u the velocity in the x direction, p the pressure, E the total energy, and $p = (\gamma - 1)(E - \rho u^2 / 2)$, $\gamma = 1.4$. In all calculations of this subsection, the CFL number

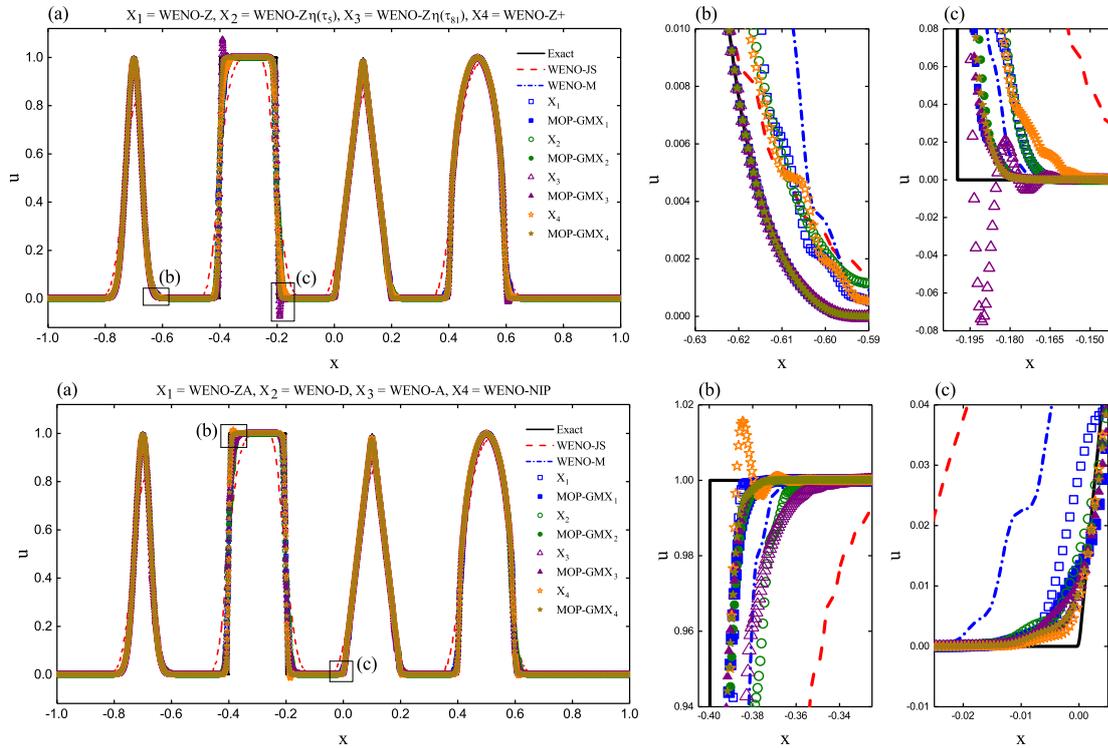


Figure 7: Results of different schemes on solving Example 4.2 (Case 2) with $t = 200, N = 3200$.

is set to be $\Delta x^{2/3}$ to rule out the effect of the time advancement on the convergence order of accuracy. The L_1, L_∞ error of the density are computed by

$$L_1 = h \cdot \sum_j |\rho_j^{\text{exact}} - (\rho_h)_j|, \quad L_\infty = \max_j |\rho_j^{\text{exact}} - (\rho_h)_j|,$$

where h is the mesh size and $h = \Delta x$. $(\rho_h)_j$ is the computing results and ρ_j^{exact} is the exact solution.

The initial conditions of this problem in our tests are given in Example 5.1 and Example 5.2 below.

Example 5.1. The first accuracy test is defined by

$$\rho(x,0) = 1 + 0.2\sin(\pi x), \quad u(x,0) = 1, \quad p(x,0) = 1. \tag{5.2}$$

The exact solution is computed by

$$\rho(x,t) = 1 + 0.2\sin(\pi(x - ut)), \quad u(x,t) = 1, \quad p(x,t) = 1.$$

The computational domain is $[0,2]$ and the output time is taken to be $t = 2.0$.

The numerical errors of the density and the convergence orders of different WENO schemes are presented in Table 7. For comparison purpose, we also present the results computed by the WENO5-ILW scheme. As expected, all the considered WENO schemes can achieve the designed convergence orders. In addition, the L_1 and L_∞ errors of all considered schemes are very close to those of the WENO-ILW scheme for this smooth problem. Here, we point out that the MOP-GMWENO-X scheme and the WENO-X scheme are essentially the same method for smooth solution in the cases whose grid numbers are not too small (e.g., $N \geq 20$ here). Actually, the minor accuracy loss of the MOP-GMWENO-X schemes that only occurs with very small grid numbers and will disappear immediately when the grid number increases very slightly is negligible and this has been discussed carefully and detailedly in [34]. Furthermore, the MOP-GMWENO-X schemes have solutions with better accuracy than WENO-JS in general.

Example 5.2. The second accuracy test is defined by

$$\rho(x,0) = 1 + 0.2 \sin\left(\pi x - \frac{\sin(\pi x)}{\pi}\right), \quad u(x,0) = 1, \quad p(x,0) = 1. \quad (5.3)$$

The exact solution is computed by

$$\rho(x,t) = 1 + 0.2 \sin\left(\pi(x-ut) - \frac{\sin(\pi(x-ut))}{\pi}\right), \quad u(x,t) = 1, \quad p(x,t) = 1.$$

Also, the computational domain is $[0,2]$ and the output time is taken to be $t = 2.0$.

Table 8 shows the numerical errors of the density and the convergence orders of different WENO schemes. Again, we give the results computed by the WENO5-ILW scheme. Firstly, it can be observed that WENO-JS provides the lowest accurate results among all considered schemes. Its L_∞ convergence rate of accuracy drops by nearly 2 orders that leads to the noticeable accuracy loss shown with the L_1 convergence orders. However, it can be seen that the MOP-GMWENO-X and WENO-X schemes can recover the designed convergence orders even in the presence of critical points.

5.2 Two-dimensional Euler equations

To examine the enhancement of the MOP-GMWENO-X schemes, we solve the two-dimensional Euler equations in this subsection. Its strong conservative form is given as

$$\begin{cases} \frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{F}(\mathbf{U})}{\partial x} + \frac{\partial \mathbf{G}(\mathbf{U})}{\partial y} = 0, \\ \mathbf{U} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \mathbf{F}(\mathbf{U}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad \mathbf{G}(\mathbf{U}) = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}, \end{cases} \quad (5.4)$$

Table 7: The L_1 , L_∞ errors of the density and the convergence rates of accuracy of different WENO schemes for Example 5.1.

N	WENO5-ILW				WENO-JS			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	5.57925e-03	-	4.31021e-03	-	2.46709e-02	-	1.76565e-02	-
20	1.87285e-04	4.8968	1.46004e-04	4.8837	1.30211e-03	4.2439	1.00423e-03	4.1360
40	5.94483e-06	4.9775	4.66039e-06	4.9694	4.15463e-05	4.9700	3.74857e-05	4.7436
80	1.86489e-07	4.9945	1.46400e-07	4.9925	1.29861e-06	4.9997	1.20927e-06	4.9541
160	5.83322e-09	4.9987	4.58129e-09	4.9980	4.05543e-08	5.0010	3.81502e-08	4.9863
320	1.81943e-10	5.0027	1.44265e-10	4.9890	1.26670e-09	5.0007	1.17287e-09	5.0236
N	WENO-Z				MOP-GMWENO-Z			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	6.08762e-03	-	4.50504e-03	-	1.35332e-02	-	1.00260e-02	-
20	1.88487e-04	5.0133	1.47175e-04	4.9359	1.88487e-04	6.1659	1.47175e-04	6.0901
40	5.94480e-06	4.9867	4.66387e-06	4.9799	5.94480e-06	4.9867	4.66387e-06	4.9799
80	1.86489e-07	4.9945	1.46410e-07	4.9934	1.86489e-07	4.9945	1.46410e-07	4.9934
160	5.83323e-09	4.9987	4.58132e-09	4.9981	5.83323e-09	4.9987	4.58133e-09	4.9981
320	1.81941e-10	5.0028	1.44251e-10	4.9891	1.81941e-10	5.0028	1.44251e-10	4.9891
N	WENO-Z $\eta(\tau_5)$				MOP-GMWENO-Z $\eta(\tau_5)$			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	6.02580e-03	-	4.51422e-03	-	1.34684e-02	-	9.98726e-03	-
20	1.88232e-04	5.0006	1.47368e-04	4.9370	1.88232e-04	6.1609	1.47368e-04	6.0826
40	5.94481e-06	4.9847	4.66425e-06	4.9816	5.94481e-06	4.9847	4.66425e-06	4.9816
80	1.86489e-07	4.9945	1.46410e-07	4.9936	1.86489e-07	4.9945	1.46410e-07	4.9936
160	5.83323e-09	4.9987	4.58133e-09	4.9981	5.83323e-09	4.9987	4.58133e-09	4.9981
320	1.81938e-10	5.0028	1.44269e-10	4.9889	1.81946e-10	5.0027	1.44260e-10	4.9890
N	WENO-Z $\eta(\tau_{81})$				MOP-GMWENO-Z $\eta(\tau_{81})$			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	5.57331e-03	-	4.29858e-03	-	1.23782e-02	-	9.20236e-03	-
20	1.87285e-04	4.8952	1.46003e-04	4.8798	1.87285e-04	6.0464	1.46003e-04	5.9779
40	5.94483e-06	4.9775	4.66039e-06	4.9694	5.94483e-06	4.9775	4.66039e-06	4.9694
80	1.86489e-07	4.9945	1.46400e-07	4.9925	1.86489e-07	4.9945	1.46400e-07	4.9925
160	5.83323e-09	4.9987	4.58130e-09	4.9980	5.83323e-09	4.9987	4.58130e-09	4.9980
320	1.81942e-10	5.0027	1.44275e-10	4.9889	1.81944e-10	5.0027	1.44261e-10	4.9890
N	WENO-Z+				MOP-GMWENO-Z+			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	5.91748e-03	-	3.89100e-03	-	4.61518e-03	-	3.04605e-03	-
20	4.31540e-04	3.7774	3.83185e-04	3.3440	4.31540e-04	3.4188	3.83185e-04	2.9908
40	1.28245e-05	5.0725	1.21272e-05	4.9817	1.28245e-05	5.0725	1.21272e-05	4.9817
80	3.78662e-07	5.0818	3.90521e-07	4.9567	3.78662e-07	5.0818	3.90521e-07	4.9567
160	1.15394e-08	5.0363	1.21818e-08	5.0026	1.15394e-08	5.0363	1.21818e-08	5.0026
320	3.60494e-10	5.0004	3.72739e-10	5.0304	3.60491e-10	5.0005	3.72757e-10	5.0303
N	WENO-ZA				MOP-GMWENO-ZA			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	5.78359e-03	-	4.46507e-03	-	1.26046e-02	-	9.33088e-03	-
20	1.87474e-04	4.9472	1.46141e-04	4.9332	1.87474e-04	6.0711	1.46141e-04	5.9966
40	5.94486e-06	4.9789	4.66039e-06	4.9708	5.94486e-06	4.9789	4.66039e-06	4.9708
80	1.86489e-07	4.9945	1.46400e-07	4.9925	1.86489e-07	4.9945	1.46400e-07	4.9925
160	5.83323e-09	4.9987	4.58130e-09	4.9980	5.83322e-09	4.9987	4.58128e-09	4.9980
320	1.81941e-10	5.0028	1.44262e-10	4.9890	1.81942e-10	5.0027	1.44257e-10	4.9890

N	WENO-D				MOP-GMWENO-D			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	6.08762e-03	-	4.50504e-03	-	1.35332e-02	-	1.00260e-02	-
20	1.88487e-04	5.0133	1.47175e-04	4.9359	1.88487e-04	6.1659	1.47175e-04	6.0901
40	5.94480e-06	4.9867	4.66387e-06	4.9799	5.94480e-06	4.9867	4.66387e-06	4.9799
80	1.86489e-07	4.9945	1.46410e-07	4.9934	1.86489e-07	4.9945	1.46410e-07	4.9934
160	5.83323e-09	4.9987	4.58132e-09	4.9981	5.83323e-09	4.9987	4.58133e-09	4.9981
320	1.81941e-10	5.0028	1.44251e-10	4.9891	1.81941e-10	5.0028	1.44251e-10	4.9891
N	WENO-A				MOP-GMWENO-A			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	5.57925e-03	-	4.31021e-03	-	1.23660e-02	-	9.21706e-03	-
20	1.87285e-04	4.8968	1.46004e-04	4.8837	1.87285e-04	6.0450	1.46004e-04	5.9802
40	5.94483e-06	4.9775	4.66039e-06	4.9694	5.94483e-06	4.9775	4.66039e-06	4.9694
80	1.86489e-07	4.9945	1.46400e-07	4.9925	1.86489e-07	4.9945	1.46400e-07	4.9925
160	5.83323e-09	4.9987	4.58129e-09	4.9980	5.83323e-09	4.9987	4.58130e-09	4.9980
320	1.81943e-10	5.0027	1.44279e-10	4.9888	1.81941e-10	5.0028	1.44249e-10	4.9891
N	WENO-NIP				MOP-GMWENO-NIP			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	1.59200e-03	-	1.88378e-03	-	1.04077e-02	-	7.92518e-03	-
20	1.81848e-04	3.1300	1.34479e-04	3.8082	1.81848e-04	5.8388	1.34479e-04	5.8810
40	5.94395e-06	4.9352	4.64675e-06	4.8550	5.94395e-06	4.9352	4.64675e-06	4.8550
80	1.86489e-07	4.9943	1.46376e-07	4.9885	1.86489e-07	4.9943	1.46376e-07	4.9885
160	5.83322e-09	4.9987	4.58125e-09	4.9978	5.83323e-09	4.9987	4.58125e-09	4.9978
320	1.81943e-10	5.0027	1.44258e-10	4.9890	1.81943e-10	5.0027	1.44241e-10	4.9892

where ρ, u, p and E are the same as in Eq. (5.1) and v is velocity in the y direction. Similarly, we use the following equations of state for ideal gases to close Eq. (5.4)

$$p = (\gamma - 1) \left(E - \frac{1}{2} \rho (u^2 + v^2) \right), \quad \gamma = 1.4.$$

For brevity in the discussion, we only focus on two standard examples. The first one is the 2D Riemann problem [15–17], and the second one is the shock-vortex interaction problem [47–49]. We choose the CFL number to be 0.5 in all calculations of this subsection.

5.2.1 2D Riemann problem

Example 5.3. We consider the 2D Riemann problem. Since the first test 2D Riemann problem was introduced by [15], it has become a widely-used two-dimensional test case for the high-resolution numerical methods [17, 18, 38]. In our present example, the Configuration 9 of [17] is taken. It is defined on the rectangular computational domain $[0, 1] \times [0, 1]$ that is divided into four quadrants by lines $x = 0.5$ and $y = 0.5$, and the following initial constant states in each quadrant are specified

$$\mathbf{U}(x, y, 0) = \begin{cases} (1.0, 0.0, 0.3, 1.0)^T, & 0.5 \leq x \leq 1.0, \quad 0.5 \leq y \leq 1.0, \\ (2.0, 0.0, -0.3, 1.0)^T, & 0.0 \leq x \leq 0.5, \quad 0.5 \leq y \leq 1.0, \\ (1.039, 0.0, -0.8133, 0.4)^T, & 0.0 \leq x \leq 0.5, \quad 0.0 \leq y \leq 0.5, \\ (0.5197, 0.0, -0.4259, 0.4)^T, & 0.5 \leq x \leq 1.0, \quad 0.0 \leq y \leq 0.5. \end{cases}$$

Table 8: The L_1 , L_∞ errors of the density and the convergence rates of accuracy of different WENO schemes for Example 5.2.

N	WENO5-ILW				WENO-JS			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	2.65885e-02	-	2.42757e-02	-	3.96420e-02	-	4.33175e-02	-
20	1.85214e-03	3.8435	2.25478e-03	3.4285	4.12024e-03	3.2662	4.83249e-03	3.1641
40	6.72815e-05	3.7828	9.26702e-05	4.6047	2.73458e-04	3.9133	3.72118e-04	3.6989
80	2.15464e-06	4.9647	3.01580e-06	4.9415	1.42262e-05	4.2647	3.21426e-05	3.5332
160	6.77495e-08	4.9911	9.51825e-08	4.9857	6.77367e-07	4.3925	3.02134e-06	3.4112
320	2.12023e-09	4.9979	2.98236e-09	4.9962	3.38590e-08	4.3223	3.05139e-07	3.3077
N	WENO-Z				MOP-GMWENO-Z			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	2.04632e-02	-	2.06144e-02	-	3.04124e-02	-	3.38563e-02	-
20	1.18306e-03	4.1124	1.13020e-03	4.1890	2.04082e-03	3.8974	3.07733e-03	3.4597
40	5.62173e-05	4.3954	9.47028e-05	3.5770	7.54893e-05	4.7567	1.06191e-04	4.8569
80	2.04617e-06	4.7800	3.01102e-06	4.9751	2.04617e-06	5.2053	3.01102e-06	5.1403
160	6.68022e-08	4.9369	9.51811e-08	4.9834	6.68022e-08	4.9369	9.51810e-08	4.9834
320	2.11297e-09	4.9826	2.98203e-09	4.9963	2.11297e-09	4.9826	2.98203e-09	4.9963
N	WENO-Z $\eta(\tau_5)$				MOP-GMWENO-Z $\eta(\tau_5)$			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	2.09002e-02	-	2.09558e-02	-	3.04489e-02	-	3.39484e-02	-
20	1.19359e-03	4.1301	1.17806e-03	4.1529	2.01481e-03	3.9177	3.04597e-03	3.4784
40	5.64972e-05	4.4010	9.45530e-05	3.6391	7.53576e-05	4.7407	1.05810e-04	4.8474
80	2.04941e-06	4.7849	3.01125e-06	4.9727	2.04941e-06	5.2005	3.01125e-06	5.1350
160	6.68307e-08	4.9386	9.51809e-08	4.9835	6.68307e-08	4.9386	9.51809e-08	4.9835
320	2.11313e-09	4.9831	2.98254e-09	4.9961	2.11313e-09	4.9831	2.98253e-09	4.9961
N	WENO-Z $\eta(\tau_{81})$				MOP-GMWENO-Z $\eta(\tau_{81})$			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	2.62185e-02	-	2.41549e-02	-	2.86051e-02	-	3.25261e-02	-
20	1.85177e-03	3.8236	2.25439e-03	3.4215	2.27118e-03	3.6548	3.22579e-03	3.3339
40	6.72815e-05	4.7826	9.26702e-05	4.6045	7.70601e-05	4.8813	1.04861e-04	4.9431
80	2.15464e-06	4.9647	3.01580e-06	4.9415	2.15464e-06	5.1605	3.01580e-06	5.1198
160	6.77495e-08	4.9911	9.51825e-08	4.9857	6.77495e-08	4.9911	9.51825e-08	4.9857
320	2.12024e-09	4.9979	2.98237e-09	4.9962	2.12023e-09	4.9979	2.98236e-09	4.9962
N	WENO-Z+				MOP-GMWENO-Z+			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	7.73904e-03	-	7.48121e-03	-	1.27380e-02	-	1.14846e-02	-
20	4.95302e-04	3.9658	5.45534e-04	3.7775	1.92014e-03	2.7299	2.50856e-03	2.1948
40	5.00701e-05	3.3063	8.02140e-05	2.7657	5.50151e-05	5.1252	1.16909e-04	4.4234
80	3.04854e-06	4.0378	8.56431e-06	3.2274	3.04854e-06	4.1736	8.56431e-06	3.7709
160	1.77633e-07	4.1011	9.41566e-07	3.1852	1.77633e-07	4.1011	9.41566e-07	3.1852
320	1.10103e-08	4.0120	1.06615e-07	3.1427	1.10103e-08	4.0120	1.06615e-07	3.1427
N	WENO-ZA				MOP-GMWENO-ZA			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	2.62478e-02	-	2.43179e-02	-	2.89498e-02	-	3.24003e-02	-
20	1.84035e-03	3.8341	2.24260e-03	3.4388	2.10030e-03	3.7849	2.99185e-03	3.4369
40	6.72478e-05	4.7743	9.26639e-05	4.5970	7.70265e-05	4.7691	1.04867e-04	4.8344
80	2.15457e-06	4.9640	3.01580e-06	4.9414	2.15457e-06	5.1599	3.01580e-06	5.1199
160	6.77493e-08	4.9911	9.51825e-08	4.9857	6.77493e-08	4.9911	9.51825e-08	4.9857
320	2.12024e-09	4.9979	2.98237e-09	4.9962	2.12024e-09	4.9979	2.98237e-09	4.9962

N	WENO-D				MOP-GMWENO-D			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	2.04632e-02	-	2.06144e-02	-	3.04124e-02	-	3.38563e-02	-
20	1.18306e-03	4.1124	1.13020e-03	4.1890	2.04082e-03	3.8974	3.07733e-03	3.4597
40	5.62173e-05	4.3954	9.47028e-05	3.5770	7.54893e-05	4.7567	1.06191e-04	4.8567
80	2.04617e-06	4.7800	3.01102e-06	4.9751	2.04617e-06	5.2053	3.01102e-06	5.1403
160	6.68022e-08	4.9369	9.51811e-08	4.9834	6.68022e-08	4.9369	9.51810e-08	4.9834
320	2.11297e-09	4.9826	2.98203e-09	4.9963	2.11297e-09	4.9826	2.98203e-09	4.9963
N	WENO-A				MOP-GMWENO-A			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	1.83037e-02	-	1.90477e-02	-	2.99262e-02	-	3.35183e-02	-
20	1.25386e-03	3.8677	1.77100e-03	3.4270	1.92453e-03	3.9588	2.85230e-03	3.5548
40	6.72815e-05	4.2200	9.26702e-05	4.2563	7.70601e-05	4.6424	1.04861e-04	4.7656
80	2.15464e-06	4.9647	3.01580e-06	4.9415	2.15464e-06	5.1605	3.01580e-06	5.1198
160	6.77495e-08	4.9911	9.51825e-08	4.9857	6.77495e-08	4.9911	9.51825e-08	4.9857
320	2.12024e-09	4.9979	2.98238e-09	4.9962	2.12023e-09	4.9979	2.98237e-09	4.9962
N	WENO-NIP				MOP-GMWENO-NIP			
	L_1 error	L_1 order	L_∞ error	L_∞ order	L_1 error	L_1 order	L_∞ error	L_∞ order
10	2.33776e-02	-	2.39846e-02	-	2.01935e-02	-	2.86901e-02	-
20	1.64174e-03	3.8318	2.00222e-03	3.5824	2.19785e-03	3.1997	3.03570e-03	3.2405
40	6.70209e-05	4.6145	9.18945e-05	4.4455	7.68682e-05	4.8376	1.04304e-04	4.8632
80	2.15445e-06	4.9592	3.01546e-06	4.9295	2.15445e-06	5.1570	3.01546e-06	5.1123
160	6.77491e-08	4.9910	9.51826e-08	4.9855	6.77491e-08	4.9910	9.51826e-08	4.9855
320	2.12024e-09	4.9979	2.98236e-09	4.9962	2.12023e-09	4.9979	2.98236e-09	4.9962

The outflow condition is employed on all edges. We discretize the computational domain into 800×800 cells and set $t = 0.3$.

The solutions are shown in Fig. 8. In the first two columns, we give the density profiles of the 2D Riemann problem simulated by the WENO-X schemes and MOP-GMWENO-X schemes respectively. In the last two columns, we display the close-up view of the zone where the post-shock oscillations are generated. We can observe that: (1) the main structure of the 2D Riemann problem was captured properly by all the considered schemes; (2) in the solutions of the WENO-X schemes, evident post-shock oscillations can be observed, however, in almost all the solutions of the MOP-GMWENO-X schemes except MOP-GMWENO-NIP, the post-shock oscillations are dramatically reduced; (3) moreover, from the close-up views in the last two columns, it is easy to see that the amplitudes of the post-shock oscillations produced by the WENO-X schemes (except WENO-NIP) are much larger than the MOP-GMWENO-X schemes; (4) although both the WENO-NIP and MOP-GMWENO-NIP schemes produce evident post-shock oscillations, it appears that the MOP-GMWENO-NIP scheme performs slightly better than the WENO-NIP scheme, and we will discuss this further in the next example. In summary, we claim that this should be an additional benefit of the new proposed WENO schemes that satisfy the *OP* property.

5.2.2 Shock-vortex interaction

Example 5.4. This is a very favorable two-dimensional test case for high-resolution methods [47–49]. The computational domain is $[0,1] \times [0,1]$, and the initial condition is given

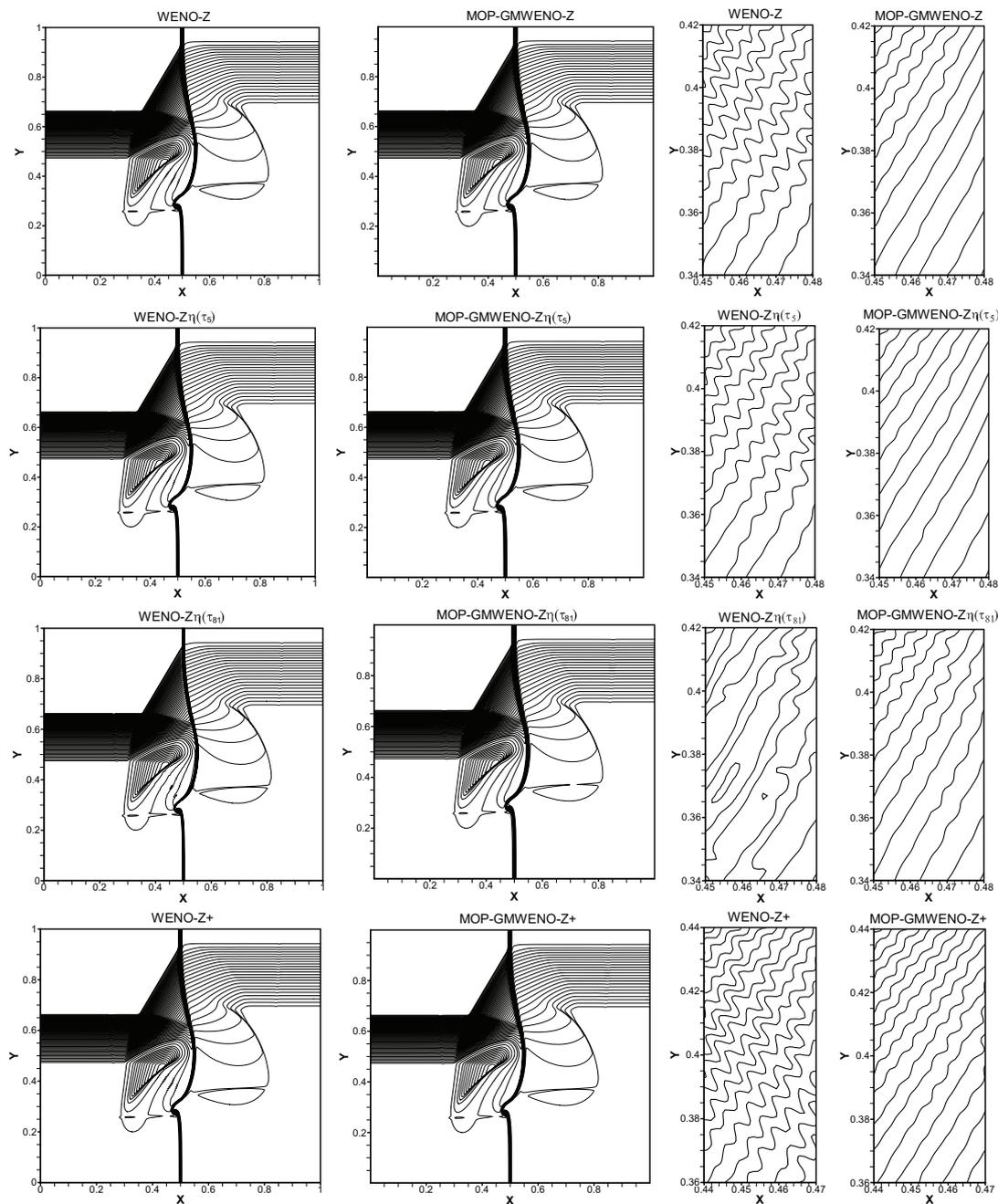


Figure 8: Density contours of Example 5.3.

by

$$\mathbf{U}(x,y,0) = \begin{cases} \mathbf{U}_L = (\rho_L + \delta\rho, u_L + \delta u, v_L + \delta v, p_L + \delta p)^T, & x < 0.5, \\ \mathbf{U}_R = (\rho_R, u_R, v_R, p_R)^T, & x \geq 0.5, \end{cases}$$

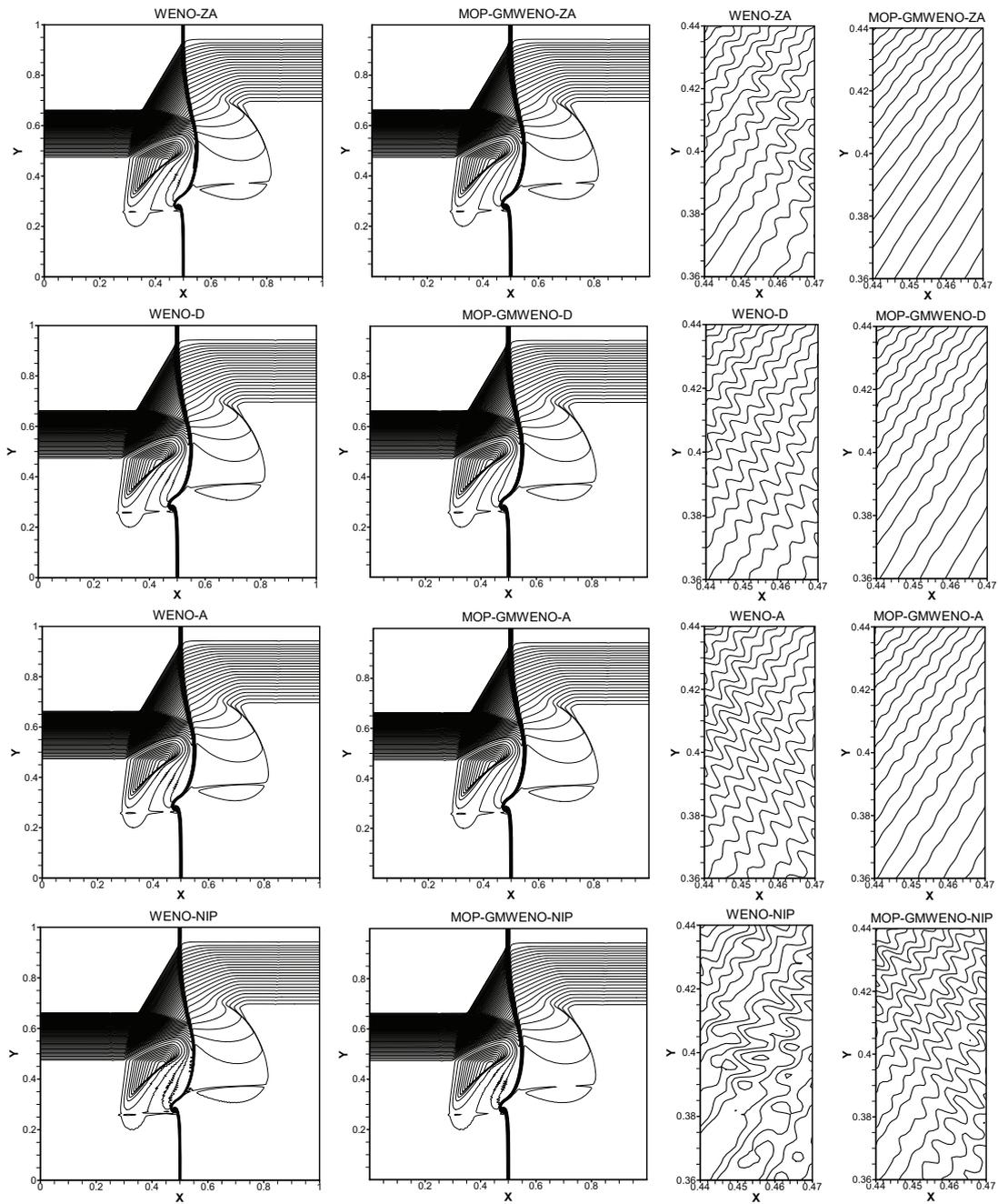


Figure 9: Continued.

with

$$\begin{aligned}
 \rho_L &= 1, & u_L &= \sqrt{\gamma}, & v_L &= 0, \\
 p_L &= 1, & \rho_R &= \rho_L \left(\frac{\gamma - 1 + (\gamma + 1)p_R}{\gamma + 1 + (\gamma - 1)p_R} \right), & u_R &= \frac{u_L(1 - p_R)}{(\gamma - 1 + p_R(\gamma + 1))^{1/2}},
 \end{aligned}$$

$$\begin{aligned}
v_R &= 0, & p_R &= 1.3, & \delta\rho &= \frac{\rho_L^2}{p_L} \frac{\delta T}{\gamma - 1}, \\
\delta u &= \frac{\epsilon}{r_c} (y - y_c) e^{\alpha(1-r^2)}, & \delta v &= \frac{-\epsilon}{r_c} (x - x_c) e^{\alpha(1-r^2)}, & \delta p &= \frac{\gamma \rho_L^2}{\rho_L} \frac{\delta T}{\gamma - 1},
\end{aligned}$$

where

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

and ϵ is 0.3, r_c is 0.05, α is 0.204, x_c is 0.25, y_c is 0.5. The outflow condition is used on all edges. We discretize the computational domain into 800×800 cells and set $t = 0.35$.

We show the results computed by the MOP-GMWENO-X schemes and the WENO-X schemes in Fig. 10. In the first two columns, the solutions in the density profile of the WENO-X schemes and the MOP-GMWENO-X schemes are given, respectively. In order to show the enhancement of the MOP-GMWENO-X schemes more clearly, in the last column, we show the density cross-sectional slices at $y = 0.65$. We can observe that: (1) the main structure of this complicated flow are captured properly by all considered schemes; (2) the WENO-X schemes generate evident numerical oscillations, however, the MOP-GMWENO-X schemes, except MOP-GMWENO-NIP, can considerably decrease these oscillations; (3) it is interesting that the oscillation produced by WENO-NIP is extremely violent, leading to the fact that the MOP-GMWENO-NIP scheme produces oscillations with larger amplitudes than that of the other MOP-GMWENO-X schemes, even larger than the other WENO-X schemes; (4) in spite of this, we can easily find that the MOP-GMWENO-NIP scheme significantly reduces the oscillations comparing with the WENO-NIP scheme; (5) also, from the last column, we can easily find that the amplitudes of the oscillations produced by the WENO-X schemes are much greater than the MOP-GMWENO-X schemes. As mentioned before, this should be a benefit of the WENO-Z-type schemes with *OP* generalized mappings.

5.3 Computational cost comparison for 2D problems

In this subsection, we present the computational cost comparison for 2D problems. In [33], the CPU time comparison of two-dimensional problems for the WENO-JS, WENO-M and WENO-Z schemes were performed. It was indicated that WENO-Z was the most efficient scheme among these three schemes and about 25% faster than the WENO-M scheme. Thus, for illustrative purpose and to save space in the present study, we take the MOP-GMWENO-Z scheme which was an improvement of the classical WENO-Z scheme as an example to examine the CPU time-consuming for 2D system of Euler equations with test cases taken in the previous subsection. For comparison purpose, we also present the results of the WENO-JS, WENO-M and WENO-Z schemes.

We test Example 5.3 and Example 5.4 by using the same computing conditions as used in Subsection 5.2.1. Moreover, a different mesh size is considered for each test respectively. An in-house code developed in C++ is employed, running in serial mode

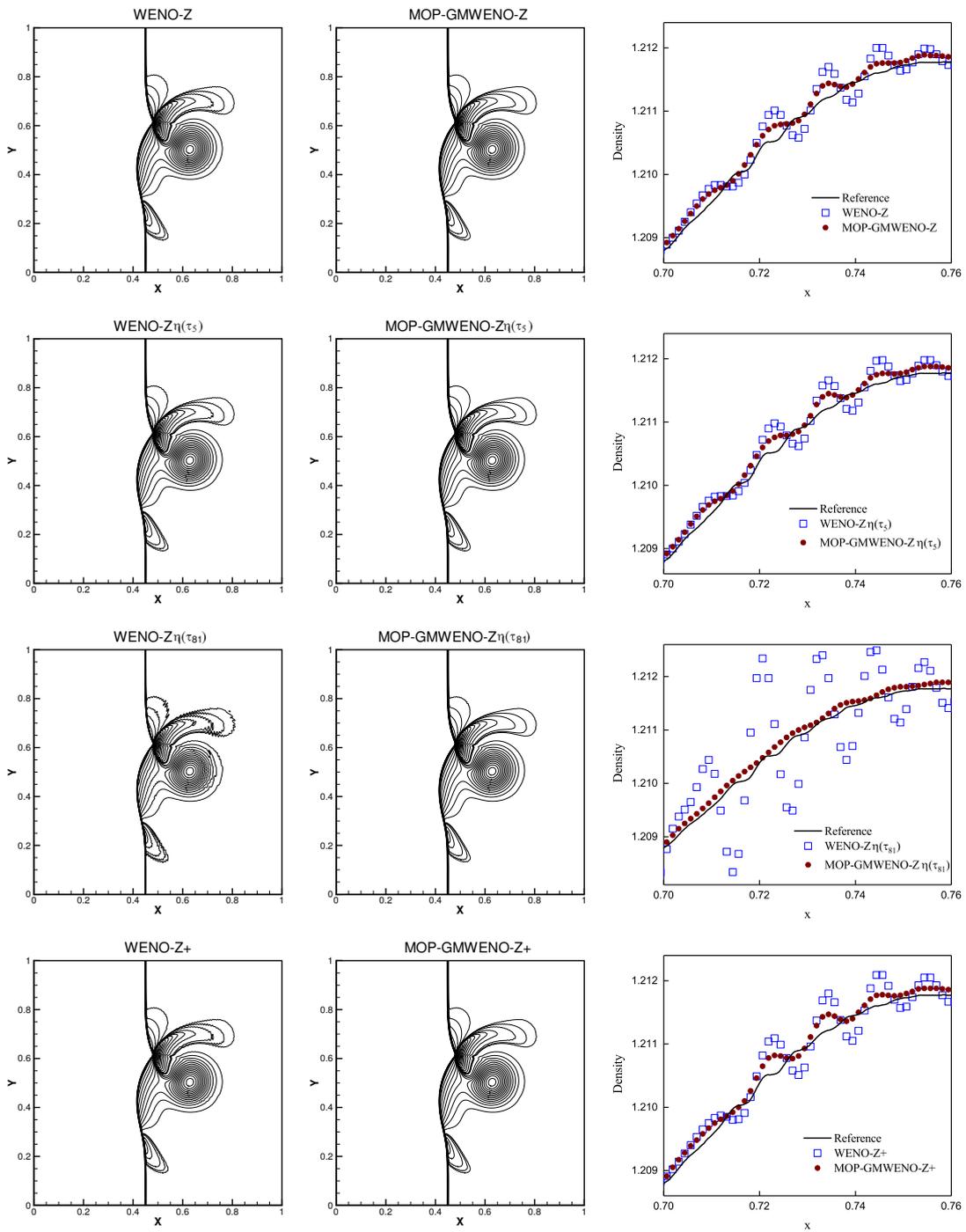


Figure 10: Density contours of Example 5.4 and the density cross-sectional slices at $y = 0.65$.

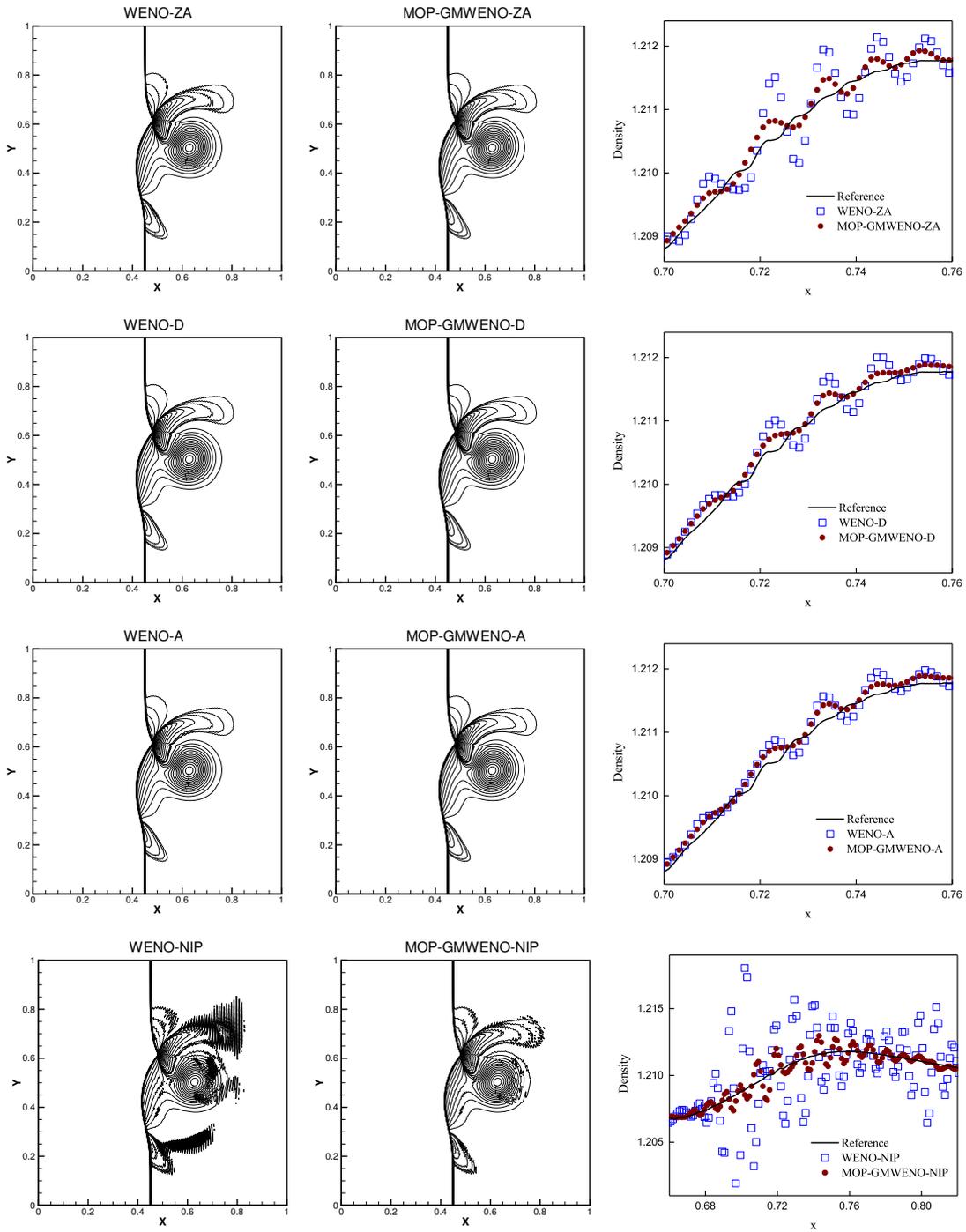


Figure 11: Continued.

Table 9: CPU time-consuming (in seconds) and the extra cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 5.3 as computed by the WENO-JS, WENO-M, WENO-Z and MOP-GMWENO-Z schemes with 400×400 meshes.

Scheme	Test 1	Test 2	Tetst 3	Aver
WENO-JS	0.889(-)	0.858(-)	0.827(-)	0.858(-)
WENO-M	1.108(24.63%)	1.092(27.27%)	1.045(26.36%)	1.082(26.07%)
WENO-Z	0.874(-1.69%)	0.842(-1.86%)	0.811(-1.93%)	0.842(-1.83%)
MOP-GMWENO-Z	0.967(8.77%)	0.951(10.84%)	0.889(7.50%)	0.936(9.05%)

Table 10: CPU time-consuming (in seconds) and the extra cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 5.3 as computed by the WENO-JS, WENO-M, WENO-Z and MOP-GMWENO-Z schemes with 800×800 meshes.

Scheme	Test 1	Test 2	Tetst 3	Aver
WENO-JS	3.401(-)	3.463(-)	3.416(-)	3.427(-)
WENO-M	4.290(26.14%)	4.415(27.49%)	4.368(27.87%)	4.358(27.17%)
WENO-Z	3.325(-2.23%)	3.416(-1.36%)	3.354(-1.81%)	3.365(-1.80%)
MOP-GMWENO-Z	3.698(8.73%)	3.753(8.37%)	3.726(9.07%)	3.725(8.73%)

Table 11: CPU time-consuming (in seconds) and the extra cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 5.4 as computed by the WENO-JS, WENO-M, WENO-Z and MOP-GMWENO-Z schemes with 400×400 meshes.

Scheme	Test 1	Test 2	Tetst 3	Aver
WENO-JS	0.936(-)	0.967(-)	0.952(-)	0.951(-)
WENO-M	1.217(30.02%)	1.279(32.26%)	1.232(29.41%)	1.243(30.58%)
WENO-Z	0.904(-3.42%)	0.951(-1.65%)	0.920(-3.36%)	0.925(-2.80%)
MOP-GMWENO-Z	1.030(10.04%)	1.061(9.72%)	1.046(9.87%)	1.046(9.87%)

Table 12: CPU time-consuming (in seconds) and the extra cost compared to the WENO-JS scheme (in percentage) per Runge-Kutta step of Example 5.4 as computed by the WENO-JS, WENO-M, WENO-Z and MOP-GMWENO-Z schemes with 800×800 meshes.

Scheme	Test 1	Test 2	Tetst 3	Aver
WENO-JS	4.165(-)	4.118(-)	4.134(-)	4.139(-)
WENO-M	5.538(32.97%)	5.289(28.44%)	5.320(28.69%)	5.382(30.04%)
WENO-Z	4.132(-0.79%)	3.978(-3.40%)	4.056(-1.89%)	4.055(-2.02%)
MOP-GMWENO-Z	4.524(8.62%)	4.493(9.11%)	4.509(9.07%)	4.508(8.93%)

under Windows system, and the CPU is Intel(R) Core(TM) i7-6700 3.40GHz. To eliminate the influences of other operations such as the initialization, boundary treatment, etc., as applied in reference [33], 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.

Let T^X denote the CPU time-consuming (in seconds) of a Runge-Kutta step of the WENO-X scheme and ζ^X denote the extra cost (in percentage) of the WENO-X scheme compared to the WENO-JS scheme, taking the form $\zeta^X = (T^X - T^{JS}) / T^{JS} \times 100\%$.

In Tables 9-12, the CPU time-consuming of the WENO-JS, WENO-M, WENO-Z and

MOP-GMWENO-Z schemes and the extra cost (in brackets) of the WENO-M, WENO-Z and MOP-GMWENO-Z schemes are presented. From Tables 9-12, we can find that: (1) for each example with different mesh sizes, the CPU time-consuming in the three tests has a certain degree of fluctuation; (2) the most efficient scheme is WENO-Z and the most inefficient one is WENO-M whose extra cost compared to WENO-JS is about 25% to 30%, and these results are consistent with those obtained in [33,44]; (3) as expected, the computational cost of MOP-GMWENO-Z has increased due to the additional operations for the order-preserving implementation as shown in Algorithm 1; (4) however, the extra costs of MOP-GMWENO-Z for all tests are no more than 10% on average, and they are far less than those obtained by the WENO-M scheme.

6 Conclusions

In this paper we improved the family of the WENO-Z-type schemes. We have extended the order-preserving (OP) criterion to the WENO-Z-type schemes. As reported in the literature, the OP property plays an important role in preserving high resolutions and meanwhile avoiding spurious oscillations for the traditional mapped WENO schemes on long simulations.

As the fact that the real-time one-to-one relationships between the nonlinear weights of the WENO-JS scheme and those of the WENO-Z-type schemes are very similar to the designed mapping relationships of the traditional mapped WENO schemes, we generated the idea of establishing the concept of generalized mapped WENO scheme. Accordingly, we devised a uniform formula for the Z-type weights. And then, we designed a general algorithm to implement our idea. Therefore, the improved WENO-Z-type schemes, dubbed MOP-GMWENO-X, have been proposed by naturally introducing the OP criterion. Compared to the traditional WENO-Z-type schemes, the improved WENO-Z-type schemes have the following enhancements: (1) they can amend the drawback of the traditional WENO-Z-type schemes of suffering from either generating spurious oscillations or losing high resolutions in long simulations of hyperbolic problems; (2) they can significantly decrease the post-shock oscillations in the simulations of the 2D Euler problems with strong shock waves.

Acknowledgements

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

References

- [1] A. HARTEN, B. ENGQUIST, S. OSHER AND S. R. CHAKRAVARTHY, *Uniformly high order accurate essentially non-oscillatory schemes III*, J. Comput. Phys., 71 (1987), pp. 231–303.

- [2] A. HARTEN AND S. OSHER, *Uniformly high order accurate essentially non-oscillatory schemes I*, SIAM J. Numer. Anal., 24 (1987), pp. 279–309.
- [3] 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), pp. 347–377.
- [4] A. HARTEN, *ENO schemes with subcell resolution*, J. Comput. Phys., 83 (1989), pp. 148–184.
- [5] C. W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.
- [6] C. W. SHU AND S. OSHER, *Efficient implementation of essentially non-oscillatory shock-capturing schemes II*, J. Comput. Phys., 83 (1989), pp. 32–78.
- [7] X. D. LIU, S. OSHER AND T. CHAN, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.
- [8] P. FAN, *High order weighted essentially non-oscillatory WENO- η schemes for hyperbolic conservation laws*, J. Comput. Phys., 269 (2014), pp. 355–385.
- [9] C. H. KIM, Y. HA AND J. YOON, *Modified non-linear weights for fifth-order weighted essentially non-oscillatory schemes*, J. Sci. Comput., 67 (2016), pp. 299–323.
- [10] F. ACKER, R. B. DE R. BORGES AND B. COSTA, *An improved WENO-Z scheme*, J. Comput. Phys., 313 (2016), pp. 726–753.
- [11] X. LUO AND S. WU, *Improvement of the WENO-Z+ scheme*, Comput. Fluids, 218 (2021), 104855.
- [12] J. PENG, C. ZHAI, G. NI, H. YONG AND Y. SHEN, *An adaptive characteristic-wise reconstruction WENO-Z scheme for gas dynamic euler equations*, Comput. Fluids, 179 (2019), pp. 34–51.
- [13] F. HU, *The weighted ENO scheme based on the modified smoothness indicator*, Comput. Fluids, 150 (2017), pp. 1–7.
- [14] S. LIU, Y. SHEN, F. ZENG AND M. YU, *A new weighting method for improving the WENO-Z scheme*, Int. J. Numer. Meth. Fluids, 87 (2018), pp. 271–291.
- [15] C. W. SCHULZ-RINNE, JAMES P. COLLINS, AND HARLAND M. GLAZ, *Numerical solution of the Riemann problem for two-dimensional gas dynamics*, SIAM J. Sci. Comput., 14 (1993), pp. 1394–1414.
- [16] C. W. SCHULZ-RINNE, *Classification of the Riemann problem for two-dimensional gas dynamics*, SIAM J. Math. Anal., 24 (1993), pp. 76–88.
- [17] P. D. LAX AND X. D. LIU, X. D., *Solution of two-dimensional Riemann problems of gas dynamics by positive schemes*, SIAM J. Sci. Comput., 19 (1998), pp. 319–340.
- [18] S. PIROZZOLI, *Numerical methods for high-speed flows*, Annu. Rev. Fluid Mech., 43 (2011), pp. 163–194.
- [19] F. ZENG, Y. SHEN AND S. LIU, *A perturbational weighted essentially non-oscillatory scheme*, Comput. Fluids, 172 (2018), pp. 196–208.
- [20] Y. WANG, B. S. WANG AND W. S. DON, *Generalized sensitivity parameter free fifth order WENO finite difference scheme with Z-type weights*, J. Sci. Comput., 81 (2019), pp. 1329–1358.
- [21] M. YUAN, *A new weighted essentially non-oscillatory WENO-NIP scheme for hyperbolic conservation laws*, Comput. Fluids, 197 (2020), 104168.
- [22] G. S. JIANG AND C. W. SHU, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.
- [23] C. W. SHU, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, Lecture Notes in Mathematics, Springer, 1998.
- [24] J. LI, G. LI, S. QIAN AND J. GAO, *High-order well-balanced finite volume WENO schemes with conservative variables decomposition for shallow water equations*, Adv. Appl. Math. Mech., 13

- (2021), pp. 827–849.
- [25] Z. WANG, J. ZHU, Y. YANG, L. TIAN AND N. ZHAO, *A class of robust low dissipation nested multi-resolution WENO schemes for solving hyperbolic conservation laws*, Adv. Appl. Math. Mech., 13 (2021), pp. 1064–1095.
- [26] R. LI AND W. ZHONG, *Improvement of the WENO-NIP scheme for hyperbolic conservation laws*, Axioms, 11 (2022), 190.
- [27] Q. LI, D. SUN AND F. XU, *WENO interpolation-based and upwind-biased free-stream preserving nonlinear schemes*, Adv. Appl. Math. Mech., 13 (2021), pp. 1441–1484.
- [28] U. S. RAJPUT AND K. M. SINGH, *A fifth order alternative mapped WENO scheme for nonlinear hyperbolic conservation laws*, Adv. Appl. Math. Mech., 14 (2022), pp. 275–298.
- [29] 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), pp. 542–567.
- [30] H. FENG, C. HUANG AND R. WANG, *An improved mapped weighted essentially non-oscillatory scheme*, Appl. Math. Comput., 232 (2014), pp. 453–468.
- [31] H. FENG, F. HU AND R. WANG, *A new mapped weighted essentially non-oscillatory scheme*, J. Sci. Comput., 51 (2012), pp. 449–473.
- [32] R. LI AND W. ZHONG, *A Modified adaptive improved mapped WENO method*, Commun. Comput. Phys., 30 (2021), pp. 1545–1588.
- [33] 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), pp. 3101–3211.
- [34] R. LI AND W. ZHONG, *A new mapped WENO scheme using order-preserving mapping*, Commun. Comput. Phys., 31 (2022), pp. 548–592.
- [35] R. LI AND W. ZHONG, *A general improvement in the WENO-Z-type schemes*, Commun. Comput. Phys., 31 (2022), pp. 1362–1401.
- [36] R. LI AND W. ZHONG, *Towards building the OP-mapped WENO schemes: a general methodology*, Math. Comput. Appl., 26 (2021), 67.
- [37] R. WANG, H. FENG AND C. HUANG, *A new mapped weighted essentially non-oscillatory method using rational function*, J. Sci. Comput., 67 (2016), pp. 540–580.
- [38] Q. LI, P. LIU AND H. ZHANG, *Piecewise polynomial mapping method and corresponding WENO scheme with improved resolution*, Commun. Comput. Phys., 18 (2015), pp. 1417–1444.
- [39] 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), pp. 329–354.
- [40] M. CASTRO, B. COSTA, AND W. S. DON, *High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws*, J. Comput. Phys., 230 (2011), pp. 1766–1792.
- [41] S. ZHANG, C. W. SHU, J. ZHU, AND J. QIU, *A new smoothness indicator for the WENO schemes and its effect on the convergence to steady state solutions*, J. Sci. Comput., 31 (2007), pp. 273–305.
- [42] U. S. VEVEK, B. ZANG, AND T. H. NEW, *A New Mapped WENO Method for Hyperbolic Problems*, Tenth International Conference on Computational Fluid Dynamics, Barcelona, Spain, 2018.
- [43] U. S. VEVEK, B. ZANG, AND T. H. NEW, *Adaptive mapping for high order WENO methods*, J. Comput. Phys., 381 (2019), pp. 162–188.
- [44] R. LI, AND W. ZHONG, *An efficient mapped WENO scheme using approximate constant mapping*, Numer. Math. Theor. Meth. Appl., 15 (2022), pp. 1–41.
- [45] Y. HA, C. H. KIM, Y. J. LEE AND J. YOON, *An improved weighted essentially non-oscillatory scheme with a new smoothness indicator*, J. Comput. Phys., 232 (2013), pp. 68–86.
- [46] RATHAN SAMALA, AND G. NAGA RAJU, *A modified fifth-order WENO scheme for hyperbolic*

- conservation laws*, Comput. Math. Appl., 75 (2018), pp. 1531–1549.
- [47] A. CHATTERJEE, *Shock wave deformation in shock-vortex interactions*, Shock Waves, 9 (1999), pp. 95–105.
- [48] S. P. PAO, M. D. SALAS, *A numerical study of two-dimensional shock-vortex interaction*, AIAA 14th Fluid and Plasma Dynamics Conference, California, Palo Alto, 1981.
- [49] 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), pp. 365–386.
- [50] Y. SHEN, AND G. ZHA, *A robust seventh-order WENO scheme and its application*, 46th AIAA Aerospace Sciences Meeting and Exhibit, Reno, Nevada, 2008.