

A simple weighted essentially non-oscillatory limiter for the correction procedure via reconstruction (CPR) framework

Jie Du¹, Chi-Wang Shu² and Mengping Zhang³

Abstract

In this paper, we adapt a simple weighted essentially non-oscillatory (WENO) limiter, originally designed for discontinuous Galerkin (DG) schemes [45], to the correction procedure via reconstruction (CPR) framework for solving conservation laws. The objective of this simple WENO limiter is to simultaneously maintain uniform high order accuracy of the CPR framework in smooth regions and control spurious numerical oscillations near discontinuities. The WENO limiter we adopt in this paper is particularly simple to implement and will not harm the conservativeness of the CPR framework. Also, it uses information only from the target cell and its immediate neighbors, thus can maintain the compactness of the CPR framework. Since the CPR framework with the WENO limiter does not in general preserve positivity of the solution, we also extend the positivity-preserving limiters in [43, 44, 36] to the CPR framework. Numerical results in one and two dimensions are provided to illustrate the good behavior of this procedure.

Key Words: CPR framework, WENO limiter, positivity-preserving.

¹School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, P.R. China. E-mail: dujie@mail.ustc.edu.cn

²Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: shu@dam.brown.edu. Research supported by NASA grant NNX12AJ62A and NSF grant DMS-1112700.

³School of Mathematical Sciences, University of Science and Technology of China, Hefei, Anhui 230026, P.R. China. E-mail: mpzhang@ustc.edu.cn. Research supported by NSFC grants 11071234, 91130016 and 91024025.

1 Introduction

In this paper, we consider the following hyperbolic conservation law

$$\begin{aligned}u_t + f(u)_x &= 0, \\u(x, 0) &= u_0(x),\end{aligned}\tag{1}$$

and its two-dimensional version

$$\begin{aligned}u_t + f(u)_x + g(u)_y &= 0, \\u(x, y, 0) &= u_0(x, y),\end{aligned}\tag{2}$$

where u , $f(u)$ and $g(u)$ can be either scalars or vectors. There has been a surge of recent research activities on high-order methods capable of solving conservation laws, such as the discontinuous Galerkin (DG) method [30, 8, 7, 6, 5, 9], the spectral and spectral-element type methods [11, 4, 13, 18], the staggered-grid (SG) multi-domain spectral method [21, 19, 20], the spectral volume (SV) [37, 39, 40, 42] and spectral difference (SD) methods [24, 25]. The SD method can be viewed as the extension (or generalization) of the SG method to triangular meshes. More recently, a new correction procedure via reconstruction (CPR) framework [14, 15, 38, 10, 12] was developed. This method was originally developed in [14] to solve conservation laws on structured meshes, under the name of flux reconstruction. In [38, 10], the CPR framework was extended to 2D triangular and mixed grids under lifting collocation penalty. In [12], CPR was further extended to 3D hybrid meshes. The CPR formulation has some nice properties. By choosing certain correction functions, the CPR framework can unify several well-known methods such as the DG, SG and the SV/SD methods and leads to simplified versions of these methods, at least for linear equations. The CPR framework is based on a nodal differential form, with an element-wise discontinuous piecewise polynomial solution space. It can be considered as the DG method with a suitable numerical quadrature for the integration of the nonlinear terms. The advantage is that it solves the conservation laws in a differential form, similar to a finite difference scheme. In the SG or SD method, two groups of grid points are needed, i.e., the solution points and the flux points. However,

the CPR framework involves only one group of grid points, namely, the solution points. Hence, the CPR framework is easier to understand and more efficient to implement. The CPR method is compact because only immediate face neighbors play a role in updating the solutions in the current cell. Therefore, the complexity of implementation can be reduced, especially for multi-dimensional meshes including unstructured meshes.

The main difficulty in solving conservation laws is that solutions may contain discontinuities even if the initial conditions are smooth. However, the CPR method is only a high-order linear scheme, hence it will generate spurious oscillations for problems containing strong discontinuities. Therefore, we need to apply nonlinear limiters to control these oscillations. There are many types of limiters, such as the minmod type total variation bounded (TVB) limiter [8, 7, 6, 5, 9], the moment-based limiter [2] and the improved moment limiter [3]. In [24, 41, 25], TVB and TVD (total variation diminishing) limiters are used in the SD method to control spurious oscillations. Although these limiters can control spurious numerical oscillations near discontinuities, they tend to degrade accuracy when mistakenly used in smooth regions of the solution. To achieve the full potential of high order accuracy and efficiency of high order methods, we would like to find a robust and high order limiting procedure to simultaneously maintain uniform high order accuracy in smooth regions and control spurious numerical oscillations near discontinuities. Limiters based on the WENO methodology [23, 16, 31] would serve such a purpose. Qiu and Shu [28] and Zhu et al. [46] designed limiters using the usual WENO reconstruction for the DG methods. They need to reconstruct the values of the solutions at certain Gaussian quadrature points in the target cell using the cell averages of neighboring cells. This kind of limiters uses information from not only the immediate neighboring cells but also neighbors' neighbors. If we apply this limiter to the CPR framework, the compactness of the original CPR framework will be destroyed. To reduce the spread of reconstruction stencils, the effort in [26, 27] attempts to construct Hermite type WENO approximations. However the information of neighbors' neighbors

is still needed for higher order methods. Very recently, a new and simple WENO limiter [45] was designed for the Runge-Kutta discontinuous Galerkin (RKDG) methods. This simple WENO limiter attempts to reconstruct the entire polynomial in one shot, using the information only from the target cell and its immediate neighbors. Extension to 2D triangulations is studied in [47].

Since the CPR framework is relatively new, discussion about high order limiting procedure in this framework is sparse. In this paper, we attempt to adapt the WENO limiter in [45] to the CPR framework on structured meshes to make it more robust for shocked flows while maintaining high order accuracy. Comparing with the usual WENO reconstruction type limiters and the limiters based on Hermite type WENO approximations, this new WENO limiter uses the information only from the target cell and its immediate neighbors, thus it can maintain the compactness of the original CPR method.

An important property of the entropy solution of scalar conservation laws is that it satisfies a strict maximum principle. For example, considering the one-dimensional scalar conservation law (1), if

$$M = \max_x u_0(x), \quad m = \min_x u_0(x), \quad (3)$$

then $u(x, t) \in [m, M]$ for any x and t . In particular, the solution will not be negative if $u(x, 0) \geq 0$. In [43], a genuinely high order accurate maximum-principle-satisfying scheme for scalar conservation laws was developed. For hyperbolic conservation law systems, the entropy solutions generally do not satisfy the maximum principle. However, for the Euler equations, the density and pressure should both be positive physically. A uniformly high order accurate DG scheme which preserve positivity of density and pressure for Euler equations of compressible gas dynamics was constructed in [44], which can be considered as a generalization of the maximum-principle-satisfying scheme in [43]. [36] further discussed an extension to design arbitrarily high order positivity-preserving DG schemes for reactive Euler equations and presented a simpler and more robust im-

plementation of the positivity-preserving limiter than the one in [44].

In this paper, the CPR framework with the WENO limiter does not in general satisfy the positivity property. Hence, we extend the positivity-preserving limiters in [43], [44] and [36] to the CPR framework. Note that the CPR framework is conservative, and it can be written in a form similar to a scheme satisfied by the cell averages of a DG method. Also, the WENO limiter will not harm the conservativeness of the CPR framework. Hence, it is straightforward to extend the positivity-preserving limiter to the CPR framework with the WENO limiter. However, there are still some differences. For the two-dimensional conservation law, as we will see in this paper, there are two new solution polynomials after we apply the WENO limiter. Hence, we need to apply the positivity-preserving limiter to both of these two solution polynomials.

This paper is organized as follows. We first review the CPR formulation in Section 2. In Section 3, we review the details of the WENO limiting procedure and describe the way to introduce it to the CPR framework. In Section 4, we describe the detailed procedure to perform the positivity-preserving limiter. In Section 5, numerical experiments are provided to verify the accuracy and stability of these limiters. Finally, concluding remarks are provided in Section 6.

2 Formulation of the CPR framework

In this section, we give an overview of the formulation of the CPR method for solving hyperbolic conservation laws (1) and (2) and give some comments on the order of accuracy of the CPR framework.

2.1 Review of the CPR framework in one dimension

Consider the one-dimensional conservation law denoted by Equation (1), which can be either a scalar equation or a system of equations. Assuming the calculation domain is $x \in [a, b]$, we first divide it into N small cells, with the j -th cell defined by

$I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, $j = 1, \dots, N$. Instead of dealing with each element cell, it is more convenient to transform them into a standard element $I = [-1, 1]$. With ξ varying on I and x varying on I_j , the mapping from I onto I_j is

$$x(\xi) = x_j + \xi h_j/2, \quad (4)$$

where $x_j = (x_{j-\frac{1}{2}} + x_{j+\frac{1}{2}})/2$ is the cell center of I_j and $h_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}$ is the width of I_j . We further denote $h = \max_{j=1, \dots, N} h_j$. For convenience, we mainly deal with I for the following discussions.

On each cell I_j , the solution is approximated by K pieces of data $u_{j,k}(t)$, $k = 1, \dots, K$, which are the numerical approximations to the exact solution at the K solution points $x_{j,k}$, $k = 1, \dots, K$. In [14], Fourier (Von Neumann) analysis is provided to get information on both stability and accuracy of the CPR scheme for one-dimensional linear scalar equation. It is shown that Fourier stability and accuracy analysis are independent of the solution points chosen. In other words, the results from Fourier analysis will be the same by using another set of solution points. Here, we assume that the same type of solution points is employed for all cells. Hence, by denoting the solution points on I as ξ_k , $k = 1, \dots, K$, the solution points on I_j can be computed by

$$x_{j,k} = x_j + \xi_k h_j/2, \quad k = 1, \dots, K. \quad (5)$$

Note that each numerical solution $u_{j,k}(t)$ depends on t , and that $u_{j,k}(t^n)$ is the numerical solution at time level n . For simplicity, $u_{j,k}(t^n)$ is abbreviated to $u_{j,k}$ when there is no confusion. For each time level n and cell I_j , supposing all the numerical solutions $u_{j,k}$, $k = 1, \dots, K$ are known, we would need to compute $\frac{du_{j,k}}{dt}$, $k = 1, \dots, K$ at $t = t^n$, and then march in time, for example by a Runge-Kutta method to get $u_{j,k}(t^{n+1})$, $k = 1, \dots, K$.

For each cell I_j , let us first approximate u by the following solution polynomial, which is a polynomial of degree $K - 1$ interpolating $u_{j,k}$, $k = 1, \dots, K$:

$$u_j(\xi) = \sum_{k=1}^K u_{j,k} \phi_k(\xi), \quad \xi \in [-1, 1], \quad (6)$$

where

$$\phi_k(\xi) = \prod_{l=1, l \neq k}^K \frac{\xi - \xi_l}{\xi_k - \xi_l}, \quad k = 1, \dots, K \quad (7)$$

are the Lagrange polynomials. Denote the values of $u_j(\xi)$ at the two cell boundaries as

$$u_{j-\frac{1}{2}}^+ = u_j(-1), \quad u_{j+\frac{1}{2}}^- = u_j(1). \quad (8)$$

Note that if the solution points contain the two cell boundaries, namely, $\xi_1 = -1$ and $\xi_K = 1$, then immediately we have

$$u_{j-\frac{1}{2}}^+ = u_{j,1}, \quad u_{j+\frac{1}{2}}^- = u_{j,K}. \quad (9)$$

Next, compute the fluxes at the K solution points using

$$f_{j,k} = f(u_{j,k}), \quad k = 1, \dots, K, \quad (10)$$

and interpolate them by

$$f_j(\xi) = \sum_{k=1}^K f_{j,k} \phi_k(\xi), \quad \xi \in [-1, 1]. \quad (11)$$

Obviously, $f_j(\xi)$ is a polynomial of degree $K - 1$. At each cell boundary $x_{j+\frac{1}{2}}$, we generally have $f_j(1) \neq f_{j+1}(-1)$. Hence, we refer to $f_j(\xi)$ as the discontinuous flux function. To account for the data interaction between adjacent cells, a continuous flux function denoted by $F_j(\xi)$ is designed, which must satisfy the following conditions:

- The continuous flux function $F_j(\xi)$ is required to take the Riemann fluxes at both ends of the cell I_j , namely,

$$F_j(-1) = \hat{f}_{j-\frac{1}{2}}, \quad F_j(1) = \hat{f}_{j+\frac{1}{2}}, \quad (12)$$

where $\hat{f}_{j+\frac{1}{2}} = \hat{f}(u_{j+\frac{1}{2}}^-, u_{j+\frac{1}{2}}^+)$ is any Riemann flux given the two discontinuous solutions at the left and right of the interface.

- In each cell I_j , the degree of $F_j(\xi)$ should be K , i.e., one degree higher than the solution polynomial $u_j(\xi)$. This requirement is explained in [14].

- $F_j(\xi)$ is required to approximate the discontinuous flux function $f_j(\xi)$ in some sense. Hence, $F_j(\xi) - f_j(\xi)$ approximates the function 0.

Considering the above three conditions, the continuous flux polynomial is defined by

$$F_j(\xi) = f_j(\xi) + [\hat{f}_{j-\frac{1}{2}} - f_j(-1)]g_{LB}(\xi) + [\hat{f}_{j+\frac{1}{2}} - f_j(1)]g_{RB}(\xi), \quad \xi \in [-1, 1]. \quad (13)$$

Here, g_{LB} is the correction function at the left boundary satisfying

$$g_{LB}(-1) = 1, \quad g_{LB}(1) = 0; \quad (14)$$

g_{RB} is the correction function at the right boundary satisfying

$$g_{RB}(-1) = 0, \quad g_{RB}(1) = 1. \quad (15)$$

Both g_{LB} and g_{RB} are polynomials of degree K which approximate the function 0 in some sense. The derivative of $F_j(\xi)$ at the solution points ξ_k , $k = 1, \dots, K$ can then be computed by

$$(F_j)_\xi(\xi_k) = (f_j)_\xi(\xi_k) + [\hat{f}_{j-\frac{1}{2}} - f_j(-1)]g'_{LB}(\xi_k) + [\hat{f}_{j+\frac{1}{2}} - f_j(1)]g'_{RB}(\xi_k). \quad (16)$$

Using the mapping defined by Equation (4), we know that the derivative of $F_j(x)$ at the solution point $x_{j,k}$ is

$$(F_j)_x(x_{j,k}) = \frac{2}{h_j}(F_j)_\xi(\xi_k) \quad (17)$$

Finally, we update the numerical solutions using the following ODE

$$\frac{du_{j,k}}{dt} + \frac{2}{h_j}(F_j)_\xi(\xi_k) = 0, \quad (18)$$

which can be solved by various time discretizations such as a Runge-Kutta method. In [14], it is proved that Equation (18) is a conservative scheme in the sense of the following equation:

$$\frac{\partial}{\partial t} \int_{I_j} u_j(x, t^n) dx = \hat{f}_{j-\frac{1}{2}} - \hat{f}_{j+\frac{1}{2}}. \quad (19)$$

The continuity of $F_j(\xi)$ between adjacent cells is crucial in the proof.

Note that the correction functions $g_{LB}(\xi)$ and $g_{RB}(\xi)$ have not been uniquely defined. Because of symmetry, we only need to consider the correction function g_{LB} . For simplicity of notation, we denote g_{LB} as g . We already know that g is a polynomial of degree K , and that $g(-1) = 1$ and $g(1) = 0$. Hence, $K - 1$ additional conditions are required to define g . By choosing different specific correction functions, the CPR framework successfully recovers the DG, SG or SD/SV methods, at least for linear equations [14]. In fact, the SV and SD methods are identical for one-dimensional conservation laws if the flux points in the SD method coincide with the partition boundaries in the SV method [35]. We will discuss the relationship between CPR and SG or SD method in Section 2.3 of this paper. For more details about the choice of the correction functions and the relationship between CPR and other methods including DG, we refer to [14].

2.2 Review of the CPR framework in two dimension

Consider the two-dimensional conservation law denoted by Equation (2), which can be either a scalar equation or a system of equations. It is straight-forward to extend the one-dimensional operations to two-dimensional rectangular cells. First, let the domain of calculation be divided into uniform rectangular cells. We consider uniform cells for the simplicity of presentation, the method itself works also for non-uniform meshes. The (i, j) -th cell is denoted by $I_{i,j} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$, with the length h_x , height h_y , and center (x_i, y_j) . As in the one-dimensional case, we transform $I_{i,j}$ onto the standard cell $[-1, 1] \times [-1, 1]$, with (x, y) varying on $I_{i,j}$ and (ξ, η) varying on $[-1, 1] \times [-1, 1]$. Let the solution points on $\xi \in [-1, 1]$ be $\xi_k, k = 1, \dots, K$. For simplicity, the same solution points in the η direction are used: $\eta_l = \xi_l, l = 1, \dots, K$. Then the solution points on $[-1, 1] \times [-1, 1]$ are given by the tensor product of these one-dimensional solution points, i.e., $(\xi_k, \eta_l), k = 1, \dots, K, l = 1, \dots, K$. Hence, the solution points on $I_{i,j}$ can

be computed by

$$(x_{i,k}, y_{j,l}) = (x_i + \xi_k h_x/2, y_j + \eta_l h_y/2), \quad k = 1, \dots, K, l = 1, \dots, K. \quad (20)$$

On $I_{i,j}$, the numerical solutions are $K \times K$ pieces of data $u_{i,j;k,l}$, $k = 1, \dots, K, l = 1, \dots, K$ at the solution points. The solution polynomial is then constructed by

$$u_{i,j}(\xi, \eta) = \sum_{k,l=1}^K u_{i,j;k,l} \phi_k(\xi) \phi_l(\eta), \quad (\xi, \eta) \in [-1, 1] \times [-1, 1], \quad (21)$$

where

$$\phi_k(\xi) = \prod_{m=1, m \neq k}^K \frac{\xi - \xi_m}{\xi_k - \xi_m}, \quad \phi_l(\eta) = \prod_{m=1, m \neq l}^K \frac{\eta - \eta_m}{\eta_l - \eta_m}. \quad (22)$$

At each solution point, we can evaluate the corresponding fluxes:

$$f_{i,j;k,l} = f(u_{i,j;k,l}), \quad g_{i,j;k,l} = g(u_{i,j;k,l}). \quad (23)$$

Then the discontinuous flux polynomials are given by

$$f_{i,j}(\xi, \eta) = \sum_{k,l=1}^K f_{i,j;k,l} \phi_k(\xi) \phi_l(\eta), \quad (\xi, \eta) \in [-1, 1] \times [-1, 1], \quad (24)$$

and

$$g_{i,j}(\xi, \eta) = \sum_{k,l=1}^K g_{i,j;k,l} \phi_k(\xi) \phi_l(\eta), \quad (\xi, \eta) \in [-1, 1] \times [-1, 1]. \quad (25)$$

To approximate $f(u)_x$ in Equation (2), we let the solution point η_l be fixed, say, $l = 2$. The solution polynomial $u_{i,j}(\xi, \eta_2)$ and the discontinuous polynomial $f_{i,j}(\xi, \eta_2)$ now become polynomials of degree $K - 1$ in ξ :

$$u_{i,j}(\xi, \eta_2) = \sum_{k=1}^K u_{i,j,k,2} \phi_k(\xi), \quad \xi \in [-1, 1], \quad (26)$$

$$f_{i,j}(\xi, \eta_2) = \sum_{k=1}^K f_{i,j,k,2} \phi_k(\xi). \quad \xi \in [-1, 1]. \quad (27)$$

Now, the continuous flux polynomial along $\eta = \eta_2$ can be constructed in the same way as in one dimension:

$$F_{i,j}(\xi, \eta_2) = f_{i,j}(\xi, \eta_2) + [\hat{f}_{i-\frac{1}{2}}(\eta_2) - f_{i,j}(-1, \eta_2)]g_{LB}(\xi) + [\hat{f}_{i+\frac{1}{2}}(\eta_2) - f_{i,j}(1, \eta_2)]g_{RB}(\xi), \quad (28)$$

where $\hat{f}_{i+\frac{1}{2}}(\eta_2) = \hat{f}(u_{i,j}(1, \eta_2), u_{i+1,j}(-1, \eta_2))$ is any Riemann flux. Then, $f(u)_x$ at the point (ξ_k, η_2) is approximated by $\frac{2}{h_x}(F_{i,j})_\xi(\xi_k, \eta_2)$. The procedure to compute the continuous flux function $G_{i,j}$ along the η direction is similar, we just fix the solution point ξ_k and use the same algorithm along the η direction. We omit the details.

Finally, the numerical solutions in cell $I_{i,j}$ are updated by the following ODE:

$$\frac{d}{dt}u_{i,j;k,l}(t) + \frac{2}{h_x}(F_{i,j})_\xi(\xi_k, \eta_l) + \frac{2}{h_y}(G_{i,j})_\eta(\xi_k, \eta_l) = 0. \quad (29)$$

2.3 Comments on the order of accuracy of CPR

Recall the CPR procedure in one dimension. We can see that $f_j(\xi)$ is a Lagrange interpolation polynomial of degree $K - 1$, and that the derivative values of it, i.e., $(f_j)_\xi(\xi_k)$, $k = 1, \dots, K$ are used in Equation (16). We denote this approach as the LP (Lagrange polynomial) approach. Note that $(f_j)_\xi(\xi)$ is a polynomial of degree $K - 2$, namely, one degree lower than $u_j(\xi)$. Wang and Gao [38] have pointed out that this LP approach will lead to a slight accuracy loss (half an order to one order). Instead of using $(f_j)_\xi(\xi_k)$ in Equation (16), they replace it by the following expression,

$$(f_u)(u_{j,k}) \cdot (u_j)_\xi(\xi_k), \quad (30)$$

which is derived using the chain-rule (the CR approach). Accuracy studies in [38] have indicated that this CR approach is significantly more accurate than the traditional LP approach. However, we generally have $(f_u)(u_{j,k}) \cdot (u_j)_\xi(\xi_k) \neq (f_j)_\xi(\xi_k)$ for a nonlinear function $f(u)$. In this case, $(F_j)_\xi$ computed by Eq. (16) using the CR approach can be considered as derivative values of another function which is no longer continuous between adjacent cells. Hence, the CR approach will harm the conservativeness of the

original CPR framework. In this section, we attempt to show that, by requiring the $K - 1$ solution points to coincide with the inner roots of the correction functions, the SG/SD type CPR using the original LP approach will not lose accuracy and the scheme is still conservative.

Let us first recall the algorithm of the SG/SD method in one dimension. Here, we consider the standard cell $I = [-1, 1]$. Except for the set of solution points $\{\xi_k, k = 1, \dots, K\}$, an additional set of points is also needed, namely, the flux points:

$$\zeta_l, l = 1, \dots, K + 1, \quad (31)$$

which must contain the two end points of the cell, i.e., $\zeta_1 = -1$ and $\zeta_{K+1} = 1$. The way to construct the solution polynomial $u_j(\xi)$ is the same as shown in Equation (6). The main difference between the SG/SD method and the CPR framework is the way to construct the continuous flux function. In the SG/SD method, the values of the solution polynomial at the flux points are computed:

$$u_j(\zeta_l), l = 1, \dots, K + 1. \quad (32)$$

Then, the K degree continuous flux polynomial $\tilde{F}_j(\xi)$ is an interpolation polynomial required to satisfy:

$$\tilde{F}_j(\zeta_l) = f(u_j(\zeta_l)), l = 2, \dots, K, \quad \tilde{F}_j(-1) = \hat{f}_{j-\frac{1}{2}}, \quad \tilde{F}_j(1) = \hat{f}_{j+\frac{1}{2}}. \quad (33)$$

Finally, the numerical solutions are updated using

$$\frac{du_{j,k}}{dt} + \frac{2}{h_j}(\tilde{F}_j)_\xi(\xi_k) = 0. \quad (34)$$

We now consider the CPR framework. Note that we have only given the values of the correction functions at the two end points of the cell, and that $K - 1$ additional conditions are needed to determine the correction functions. If we require

$$g_{LB}(\zeta_l) = g_{RB}(\zeta_l) = 0, \quad l = 2, \dots, K, \quad (35)$$

then the two correction functions can be determined. We call this kind of CPR as SG/SD type CPR. From Equation (13), we can easily see that

$$F_j(\zeta_l) = f_j(\zeta_l), \quad l = 2, \dots, K, \quad F_j(-1) = \hat{f}_{j-\frac{1}{2}}, \quad F_j(1) = \hat{f}_{j+\frac{1}{2}}. \quad (36)$$

By comparing Equation (33) and Equation (36), we can see that the SG/SD method and the SG/SD type CPR are almost the same. However, there are still some differences. One can see that for a nonlinear function $f(u)$, we generally have

$$f_j(\zeta_l) \neq f(u_j(\zeta_l)), \quad l = 2, \dots, K. \quad (37)$$

If we let $K-1$ solution points coincide with the inner roots of the correction functions, for example, we can let $\xi_k = \zeta_k, k = 2, \dots, K$, then we have

$$f_j(\zeta_l) = f_j(\xi_l) = f_{j,l} = f(u_{j,l}) = f(u_j(\xi_l)) = f(u_j(\zeta_l)), \quad l = 2, \dots, K. \quad (38)$$

Now, the SG/SD type CPR is exactly the same as the SG/SD method. Numerical results in Section 5 show that the accuracy will not be lost using the original LP approach.

3 The WENO limiter

In this section, we attempt to adapt the simple WENO limiter in [45] to the CPR framework to control the oscillations for shocked flows as well as maintain the original high order accuracy. To maintain the conservativeness of the CPR framework, we only use the WENO limiter to reconstruct the solution polynomials (not the flux polynomials) on each cell. Since the WENO limiter uses information only from immediate neighbors, it can maintain the compactness of the CPR framework.

3.1 The WENO limiting procedure for the one-dimensional scalar case

In the one-dimensional scalar CPR framework, the solution $u(x, t)$ on cell I_j is approximated by the solution polynomial $u_j(x), x \in I_j$. As in [45], we first use the TVB

minmod limiter [7] to check whether the cell I_j is a troubled cell, namely, whether $u_j(x)$, $x \in I_j$ contains possible shocks and needs the limiting procedure.

For the solution polynomial $u_j(x)$, we denote

$$\bar{u}_j = \frac{1}{h_j} \int_{I_j} u_j(x) dx, \quad (39)$$

and

$$\delta_+ u_j = u_{j+\frac{1}{2}}^- - \bar{u}_j, \quad \delta_- u_j = \bar{u}_j - u_{j-\frac{1}{2}}^+. \quad (40)$$

We further denote

$$\delta_+ u_j^{(mod)} = \tilde{m}(\delta_+ u_j, \Delta_+ \bar{u}_j, \Delta_- \bar{u}_j), \quad \delta_- u_j^{(mod)} = \tilde{m}(\delta_- u_j, \Delta_+ \bar{u}_j, \Delta_- \bar{u}_j), \quad (41)$$

where $\Delta_+ \bar{u}_j = \bar{u}_{j+1} - \bar{u}_j$ and $\Delta_- \bar{u}_j = \bar{u}_j - \bar{u}_{j-1}$, with the TVB modified minmod function

$$\tilde{m}(a_1, \dots, a_l) = \begin{cases} a_1, & \text{if } |a_1| \leq Mh^2, \\ m(a_1, \dots, a_l), & \text{otherwise,} \end{cases} \quad (42)$$

where the minmod function m is defined by

$$m(a_1, \dots, a_l) = \begin{cases} s \min_{1 \leq j \leq l} |a_j|, & \text{if } s = \text{sign}(a_1) = \dots = \text{sign}(a_l), \\ 0, & \text{otherwise.} \end{cases} \quad (43)$$

Here the TVB parameter M has to be chosen adequately depending on the solution of the problem. In smooth regions away from extrema, Taylor expansions reveal that the minmod function will return the first argument. In the region near smooth extrema, the condition $|a_1| \leq Mh^2$ will be satisfied with a suitable choice of M (which depends on the second derivative of the function at the extrema) and hence the modified minmod function will also return the first argument. A detailed analysis is given in [7] to show that, if M is chosen suitably depending on the second derivative of the solution, the modified minmod function will always return the first argument (that is, the limiter will not take any effect) for smooth functions or for general functions in smooth regions. Whenever one of the minmod functions in Equation (41) gets enacted (returns other than the first argument), the cell I_j is marked as a troubled cell. We remark that the TVB

troubled cell indicator is just one of the many possibilities for troubled cell indicators and may not be the best one, see [29] for a detailed discussion. We use it here for its simplicity, as our main focus of this paper is not on troubled cell indicators.

Assuming that the cell I_j is a troubled cell, we now introduce the WENO limiting procedure in [45] to reconstruct the solution polynomial on it. We denote the solution polynomials on the cells I_{j-1}, I_j, I_{j+1} as $p_{-1}(x), p_0(x), p_1(x)$, respectively, i.e.,

$$p_l(x) = u_{j-l}(x), \quad l = -1, 0, 1, \quad (44)$$

and use them to reconstruct a new solution polynomial $u_j^{new}(x)$ on I_j . In order to make sure that $u_j^{new}(x)$ maintains the original cell average of $u_j(x)$, which is essential to keep the conservativeness of the original CPR framework, we make the following modifications:

$$\tilde{p}_{-1}(x) = p_{-1}(x) - \bar{p}_{-1} + \bar{p}_0, \quad \tilde{p}_1(x) = p_1(x) - \bar{p}_1 + \bar{p}_0, \quad (45)$$

where

$$\bar{p}_{-1} = \frac{1}{h_j} \int_{I_j} p_{-1}(x) dx, \quad \bar{p}_0 = \frac{1}{h_j} \int_{I_j} p_0(x) dx, \quad \bar{p}_1 = \frac{1}{h_j} \int_{I_j} p_1(x) dx. \quad (46)$$

In this way the new polynomials $\tilde{p}_{-1}(x)$ and $\tilde{p}_1(x)$ will have the same cell average as that of $u_j(x)$ on cell I_j . The final nonlinear WENO reconstruction polynomial $u_j^{new}(x)$ is now defined by a convex combination of these modified polynomials:

$$u_j^{new}(x) = \omega_{-1} \tilde{p}_{-1}(x) + \omega_0 p_0(x) + \omega_1 \tilde{p}_1(x). \quad (47)$$

Following [16, 1], the normalized nonlinear weights are defined as

$$\omega_l = \frac{\tilde{\omega}_l}{\sum_s \tilde{\omega}_s}, \quad (48)$$

where the non-normalized nonlinear weights $\tilde{\omega}_l$ are functions of the linear weights γ_l and the so-called smoothness indicators β_l defined as follows:

$$\tilde{\omega}_l = \frac{\gamma_l}{(\varepsilon + \beta_l)^2}. \quad (49)$$

Here $\varepsilon > 0$ is introduced to avoid the denominator to become 0. We take $\varepsilon = 10^{-6}$ in all our numerical tests. As in [16, 1], the smoothness indicator is defined by

$$\beta_l = \sum_{s=1}^K \int_{I_j} \Delta x_j^{2s-1} \left(\frac{\partial^s}{\partial x^s} p_l(x) \right)^2 dx. \quad (50)$$

We can easily see that $\omega_{-1} + \omega_0 + \omega_1 = 1$. From Equation (47), we know that $u_j^{new}(x)$ has the same cell average and order of accuracy as $u_j(x)$. As discussed in [45], because we have used the complete information of the three polynomials $p_{-1}(x)$, $p_0(x)$ and $p_1(x)$, there is no extra requirements on the linear weights in order to maintain the original high order accuracy. Since for smooth solutions the central cell is usually the best one, we put a larger linear weight on the central cell than on the neighboring cells. In our numerical tests, we take

$$\gamma_{-1} = 0.000001, \quad \gamma_0 = 0.999998, \quad \gamma_1 = 0.000001. \quad (51)$$

Starting from $u_{j,k}(t^n)$, $j = 1, \dots, N$, $k = 1, \dots, K$ at time level n , we summarize the procedure to obtain $u_{j,k}(t^{n+1})$, $j = 1, \dots, N$, $k = 1, \dots, K$ as follows. For simplicity, we consider the forward Euler time discretization of the semi-discrete scheme (18).

1. As shown in Section 2.1, we first get the solution polynomials $u_j(x)$, $j = 1, \dots, N$ by interpolating the numerical solutions at the solution points in each cell.
2. Using the TVB limiting procedure described above to find out those troubled cells.
3. For each cell I_j , $j = 1, \dots, N$, if it is a troubled cell, we replace the entire solution polynomial $u_j(x)$, $x \in I_j$ with the new polynomial $u_j^{new}(x)$, $x \in I_j$, which is a convex combination of polynomials on this cell and its immediate neighboring cells. If the cell I_j is not a troubled cell, we just let $u_j^{new}(x) = u_j(x)$, $x \in I_j$.
4. For each cell I_j , $j = 1, \dots, N$, compute $u_{j,k}^{new} = u_j^{new}(x_{j,k})$, $k = 1, \dots, K$. We now use the original CPR procedure (see Section 2.1) to construct the continuous flux polynomial by using the new numerical solutions $u_{j,k}^{new}$, and denote it as $F_j^{new}(x)$.

5. We march in time and get the numerical solutions at time level $n + 1$ by

$$\frac{u_{j,k}(t^{n+1}) - u_{j,k}^{new}}{\Delta t} + (F_j^{new})_x(x_{j,k}) = 0. \quad (52)$$

For simplicity, we have only considered the forward Euler time discretization in the above illustration, while higher order methods, such as the TVD Runge-Kutta time discretization [33] which is just a convex combination of such Euler forward steps, could be applied.

Note that we have only reconstructed the solution polynomials in those troubled cells, and the remaining procedures are exactly the same as the original CPR. Hence, the new polynomials $F_j^{new}(x)$ are also continuous between adjacent cells. Using the same proof as in [14], we can prove that

$$\frac{\partial}{\partial t} \int_{I_j} u_j^{new}(x, t^n) dx = \hat{f}_{j-\frac{1}{2}}^{new} - \hat{f}_{j+\frac{1}{2}}^{new}, \quad (53)$$

where $\hat{f}_{j-\frac{1}{2}}^{new}$ and $\hat{f}_{j+\frac{1}{2}}^{new}$ are the new common numerical fluxes at the cell boundaries. Hence, the WENO limiting procedure will not harm the conservation property of the original CPR.

3.2 WENO limiting procedure for the one-dimensional system case

Consider the one-dimensional conservation law system as shown in Equation (1), where u and f are vectors with m components. Denote the Jacobian matrix by $A_j = \frac{df}{du}|_{\bar{u}_j}$, and the left and right eigenvectors of A_j by $l_j^{(p)}, r_j^{(p)}, p = 1, \dots, m$, normalized so that $l_j^{(p)} \cdot r_j^{(q)} = \delta_{pq}$. Let $R(\bar{u}_j)$ be the $m \times m$ matrix with the right eigenvectors as columns, i.e.,

$$R(\bar{u}_j) = (r_j^{(1)}, r_j^{(2)}, \dots, r_j^{(m)}). \quad (54)$$

Clearly, $R^{-1}(\bar{u}_j)$ is a $m \times m$ matrix with the left eigenvectors as rows, i.e.,

$$R^{-1}(\bar{u}_j) = (l_j^{(1)}, l_j^{(2)}, \dots, l_j^{(m)})^T. \quad (55)$$

As in [45], we first identify the troubled cells using the following characteristic-wise TVB minmod limiter. For cell I_j , we denote

$$\Delta_+ \bar{v}_j = R^{-1}(\bar{u}_{j+1} - \bar{u}_j), \quad \Delta_- \bar{v}_j = R^{-1}(\bar{u}_j - \bar{u}_{j-1}), \quad (56)$$

and

$$\delta_+ v_j = R^{-1}(u_{j+\frac{1}{2}}^- - \bar{u}_j), \quad \delta_- v_j = R^{-1}(\bar{u}_j - u_{j-\frac{1}{2}}^+). \quad (57)$$

We compute

$$\delta_+ v_j^{(mod)} = \tilde{m}(\delta_+ v_j, \Delta_+ \bar{v}_j, \Delta_- \bar{v}_j), \quad \delta_- v_j^{(mod)} = \tilde{m}(\delta_- v_j, \Delta_+ \bar{v}_j, \Delta_- \bar{v}_j), \quad (58)$$

with the modified minmod function \tilde{m} defined in Equation (42) for each component of the vectors. If

$$\delta_+ v_j^{(mod)} = \delta_+ v_j \quad \& \quad \delta_- v_j^{(mod)} = \delta_- v_j, \quad (59)$$

then I_j is not a troubled cell and we simply let $u_j^{new}(x) = u_j(x)$.

Assuming the cell I_j is a troubled cell, we project the solution polynomials $u_{j-1}(x)$, $u_j(x)$ and $u_{j+1}(x)$ into the characteristic fields as

$$v_{j-1}(x) = R^{-1}u_{j-1}(x), \quad v_j(x) = R^{-1}u_j(x), \quad v_{j+1}(x) = R^{-1}u_{j+1}(x). \quad (60)$$

Then, we perform the WENO limiting procedure as in the one-dimensional scalar case for each component of $v_j(x)$ which is a troubled component, namely the corresponding component makes the condition (59) not satisfied, and obtain the updated vector $v_j^{new}(x)$. Finally, the reconstructed solution polynomials on cell I_j is computed by

$$u_j^{new}(x) = Rv_j^{new}(x). \quad (61)$$

Starting from $u_{j,k}(t^n), j = 1, \dots, N, k = 1, \dots, K$ at time level n , we also use Equation (52) to get the numerical solutions at time level $n + 1$. We omit the details since it is similar to the one-dimensional scalar case.

3.3 WENO limiting procedure for the two-dimensional scalar and system cases

Consider the conservation laws as shown in Equation (2), where u , f and g can be either scalars or vectors. As explained in Section 2.2, the procedure in the two-dimensional CPR framework on rectangular meshes is as simple as in the one-dimensional case. In fact, we just fix one of the variables, and deal with another variable in the same way as in the one-dimensional case. Hence, we only need to use the one-dimensional WENO limiting procedure to reconstruct the solution polynomials along different directions, namely, x and y , respectively. We show the details below.

Starting from the numerical solutions on the cell $I_{i,j}$ at time level n , i.e., $u_{i,j;k,l}(t^n)$, $k = 1, \dots, K$, $l = 1, \dots, K$, we now show the procedure to obtain $u_{i,j;k,l}(t^{n+1})$, $k = 1, \dots, K$, $l = 1, \dots, K$.

We first perform the procedure in the x -direction. For each fixed l , say $l = 2$, the numerical solutions along the x -direction become $u_{i,j;k,2}(t^n)$, $k = 1, \dots, K$. We now attempt to compute a set of new numerical solutions $u_{i,j;k,2}^{x,new}$, $k = 1, \dots, K$ as follows:

1. We first interpolate $u_{i,j;k,2}(t^n)$, $k = 1, \dots, K$ to get the solution polynomial $u_{i,j}(x, y_{j,2})$, which is a polynomial of degree $K - 1$ in x .
2. For the scalar case, we use the TVB minmod limiter introduced in Section 3.1 to identify whether the solution function $u_{i,j}(x, y_{j,2})$ contains shocks and needs the limiting procedure. For the system case, we use the characteristic-wise TVB limiter described in Section 3.2. Be careful that the Jacobian matrix we use now is $\frac{\partial f}{\partial u}$.
3. If the solution function $u_{i,j}(x, y_{j,2})$ does not need the limiting procedure, we just let $u_{i,j}^{x,new}(x, y_{j,2}) = u_{i,j}(x, y_{j,2})$. Otherwise, we use the polynomials $u_{i-1,j}(x, y_{j,2})$, $u_{i,j}(x, y_{j,2})$ and $u_{i+1,j}(x, y_{j,2})$ to reconstruct a new polynomial $u_{i,j}^{x,new}(x, y_{j,2})$ and replace the entire solution polynomial $u_{i,j}(x, y_{j,2})$ with it. The WENO limiting procedure we use here is exactly the same as in the one-dimensional case.

4. Compute $u_{i,j;k,2}^{x,new} = u_{i,j}^{x,new}(x_{i,k}, y_{j,2})$, $k = 1, \dots, K$.

After doing all these procedures for each fixed l ($l = 1, \dots, K$), we get a new set of numerical solutions $u_{i,j;k,l}^{x,new}$, $k = 1, \dots, K$, $l = 1, \dots, K$.

Next, we perform a similar procedure along the y -direction to get $u_{i,j;k,l}^{y,new}$, $k = 1, \dots, K$, $l = 1, \dots, K$. We omit the details, and only caution that, for the system case, the Jacobian matrix we use now is $\frac{\partial g}{\partial u}$.

Finally, we compute $u_{i,j;k,l}(t^{n+1})$, $k = 1, \dots, K$, $l = 1, \dots, K$ at time level $n + 1$ by

$$\frac{u_{i,j;k,l}(t^{n+1}) - u_{i,j;k,l}^{new}}{\Delta t} + (F_{i,j}^{x,new})_x(x_{i,k}, y_{j,l}) + (G_{i,j}^{y,new})_y(x_{i,k}, y_{j,l}) = 0, \quad (62)$$

where $u_{i,j;k,l}^{new} = \frac{1}{2}(u_{i,j;k,l}^{x,new} + u_{i,j;k,l}^{y,new})$, the continuous flux function $F_{i,j}^{x,new}$ is computed by using $u_{i,j;k,l}^{x,new}$, $k = 1, \dots, K$, $l = 1, \dots, K$ and $G_{i,j}^{y,new}$ is computed by using $u_{i,j;k,l}^{y,new}$, $k = 1, \dots, K$, $l = 1, \dots, K$. For simplicity, we have only considered the forward Euler time discretization in the above procedure, while higher order methods, such as Runge-Kutta time discretization could be applied.

4 The positivity-preserving limiter

In this section, we extend the positivity-preserving limiters in [43, 44, 36] to the CPR framework. At each time level, we first reconstruct the solution polynomials using the WENO limiter described in Section 3 to control oscillations, and then use the positivity-preserving limiter to further modify the solution polynomials in each cell. The other procedures will remain the same as in the original CPR framework. As shown in [43, 44, 36], these positivity-preserving limiters can enforce positivity without compromising the original high order accuracy.

4.1 Positivity-preserving limiter for the one-dimensional scalar case

Consider the M -point Legendre Gauss-Lobatto quadrature rule on the interval $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$, which is exact for the integral of polynomials of degree up to $2M - 3$. We

denote these quadrature points on I_j as S_j . Here we choose M to be the smallest integer satisfying $2M - 3 \geq K - 1$.

At each time level, we can obtain a new solution polynomial $u_j^{new}(x)$ in I_j during the WENO limiting procedure. We now use it to construct a modified solution polynomial $\tilde{u}_j(x)$ such that $\tilde{u}_j(x) \geq 0$ for all $x \in S_j$:

$$\tilde{u}_j(x) = \theta(u_j^{new}(x) - \bar{u}_j^n) + \bar{u}_j^n, \quad \theta = \min \left\{ \left| \frac{\bar{u}_j^n}{\bar{u}_j^n - m_j} \right|, 1 \right\}, \quad (63)$$

with $m_j = \min_{x \in S_j} u_j^{new}(x)$. Here \bar{u}_j^n is the cell average of the solution polynomial as defined in Equation (39) with $\bar{u}_j^n \geq 0$. We add the superscript n to denote the time level. In the case of $m_j \geq 0$, we have $\theta = 1$ and $\tilde{u}_j(x) = u_j^{new}(x) \geq m_j \geq 0, \forall x \in S_j$. In the case of $m_j < 0$, we have $\theta = \frac{\bar{u}_j^n}{\bar{u}_j^n - m_j}$ and $\tilde{u}_j(x) \geq \theta(m_j - \bar{u}_j^n) + \bar{u}_j^n = 0, \forall x \in S_j$. Then we replace $u_j^{new}(x)$ by the modified polynomial $\tilde{u}_j(x)$ for all j . The other procedure remains unchanged as in the original CPR framework with the WENO limiter.

We have described the bound-preserving limiter for the simple case in which the only bound to preserve is the lower bound zero (positivity-preserving). The procedure for other lower bound and/or upper bound is similar. We refer to [43, 44, 36] for more details.

4.2 Positivity-preserving limiter for the one-dimensional Euler system

Consider the one-dimensional Euler system of compressible gas dynamics

$$\mathbf{u}_t + \mathbf{f}(\mathbf{u})_x = 0, \quad (64)$$

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

Here ρ is the density, v is the velocity, $m = \rho v$ is the momentum, E is the total energy, and p is the pressure, with

$$p(\mathbf{u}) = (\gamma - 1)(E - \frac{1}{2}\rho v^2); \quad (65)$$

$\gamma = 1.4$ for the air.

At time level n , assume that the solution polynomial reconstructed by the WENO limiter in cell I_j is $\mathbf{u}_j^{new}(x) = (\rho_j(x), m_j(x), E_j(x))^T$, and that the cell average of $\mathbf{u}_j^{new}(x)$ is $\bar{\mathbf{u}}_j^n = (\bar{\rho}_j^n, \bar{m}_j^n, \bar{E}_j^n)^T$. We use the following algorithm to get a new modified solution polynomial $\tilde{\mathbf{u}}_j(x)$ and use it instead of $\mathbf{u}_j^{new}(x)$.

1. In each cell, we modify the density first. Assume there exists a small number $\varepsilon > 0$ such that $\bar{\rho}_j^n \geq \varepsilon$ for all j . In practice, we can choose $\varepsilon = 10^{-13}$. Replace $\rho_j(x)$ by

$$\hat{\rho}_j(x) = \theta_1(\rho_j(x) - \bar{\rho}_j^n) + \bar{\rho}_j^n, \quad \theta_1 = \min\left\{\frac{\bar{\rho}_j^n - \varepsilon}{\bar{\rho}_j^n - \rho_{min}}, 1\right\}, \quad (66)$$

with $\rho_{min} = \min_{x \in S_j} \rho_j(x)$. Now we have $\hat{\rho}_j(x) \geq \varepsilon > 0, \forall x \in S_j$. This fact is proved in [22].

2. We then enforce the positivity of the pressure. Define $\hat{\mathbf{u}}_j(x) = (\hat{\rho}_j(x), m_j(x), E_j(x))^T$. For each $x \in S_j$, if $p(\hat{\mathbf{u}}_j(x)) \geq 0$, set $\theta_x = 1$; otherwise, set

$$\theta_x = \frac{p(\bar{\mathbf{u}}_j^n)}{p(\bar{\mathbf{u}}_j^n) - p(\hat{\mathbf{u}}_j(x))}. \quad (67)$$

Then we obtain the limited polynomial

$$\tilde{\mathbf{u}}_j(x) = \theta_2(\hat{\mathbf{u}}_j(x) - \bar{\mathbf{u}}_j^n) + \bar{\mathbf{u}}_j^n, \quad \theta_2 = \min_{x \in S_j} \theta_x. \quad (68)$$

The proof of $p(\tilde{\mathbf{u}}_j(x)) \geq 0, \forall x \in S_j$ can be found in [36]. It can be shown that this limiter does not destroy accuracy for smooth solutions [36, 44].

4.3 Positivity-preserving limiter for the two-dimensional scalar case

In this section, we need to use the Gauss quadrature with L points, which is exact for single variable polynomials of degree $K - 1$. We assume S_i^x denote the Gauss quadrature points on $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, and S_j^y denote the Gauss quadrature points on $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. We still need to use the Gauss-Lobatto quadrature rule. We assume \hat{S}_i^x denote the Gauss-Lobatto quadrature points on $[x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, and \hat{S}_j^y denote the Gauss-Lobatto quadrature

points on $[y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. We use \times to denote the Cartesian product, for instance, $S_i^x \times S_j^y = \{(a, b) | a \in S_i^x, b \in S_j^y\}$. Define the sets $S_{i,j}^x$ and $S_{i,j}^y$ as

$$S_{i,j}^x = \hat{S}_i^x \times S_j^y, \quad S_{i,j}^y = S_i^x \times \hat{S}_j^y. \quad (69)$$

During the WENO limiting procedure, we get two solution polynomials $u_{i,j}^{x,new}(x, y)$ and $u_{i,j}^{y,new}(x, y)$. Note that the WENO limiter does not change the cell average of the original solution polynomial. Hence, the cell averages of both of these two solution polynomials are $\bar{u}_{i,j}^n$. To preserve the positivity of the cell averages of the solution polynomials at the next time level, we need $u_{i,j}^{x,new}(x, y) \geq 0, \forall (x, y) \in S_{i,j}^x$ and $u_{i,j}^{y,new}(x, y) \geq 0, \forall (x, y) \in S_{i,j}^y$ for all i and j . The proof is similar to that in [44] and is omitted.

We now construct the modified solution polynomial $\tilde{u}_{i,j}^x(x, y)$ as

$$\tilde{u}_{i,j}^x(x, y) = \theta(u_{i,j}^{x,new}(x, y) - \bar{u}_{i,j}^n) + \bar{u}_{i,j}^n, \quad \theta = \min \left\{ \left| \frac{\bar{u}_{i,j}^n}{\bar{u}_{i,j}^n - m_{i,j}} \right|, 1 \right\}, \quad (70)$$

with $m_{i,j} = \min_{(x,y) \in S_{i,j}^x} u_{i,j}^{x,new}(x, y)$, and construct the modified solution polynomial $\tilde{u}_{i,j}^y(x, y)$ as

$$\tilde{u}_{i,j}^y(x, y) = \theta(u_{i,j}^{y,new}(x, y) - \bar{u}_{i,j}^n) + \bar{u}_{i,j}^n, \quad \theta = \min \left\{ \left| \frac{\bar{u}_{i,j}^n}{\bar{u}_{i,j}^n - m_{i,j}} \right|, 1 \right\}, \quad (71)$$

with $m_{i,j} = \min_{(x,y) \in S_{i,j}^y} u_{i,j}^{y,new}(x, y)$. In each cell $I_{i,j}$, we replace $u_{i,j}^{x,new}(x, y)$ and $u_{i,j}^{y,new}(x, y)$ by the two modified polynomials $\tilde{u}_{i,j}^x(x, y)$ and $\tilde{u}_{i,j}^y(x, y)$, respectively. The other procedure remain unchanged as in the CPR framework with the WENO limiter.

4.4 Positivity-preserving limiter for two-dimensional Euler system

Let us consider the two-dimensional Euler system which is given by

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

$$\mathbf{u} = \begin{pmatrix} \rho \\ m \\ n \\ E \end{pmatrix}, \mathbf{f}(\mathbf{u}) = \begin{pmatrix} m \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{pmatrix}, \mathbf{g}(\mathbf{u}) = \begin{pmatrix} n \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{pmatrix}. \quad (72)$$

Here, ρ is the density, (u, v) is the velocity, $m = \rho u$ and $n = \rho v$ are the momenta. E is the total energy, and p is the pressure, with

$$p(\mathbf{u}) = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2)\right). \quad (73)$$

At time level n , assume that the two solution polynomials reconstructed by the WENO limiter in cell $I_{i,j}$ are $\mathbf{u}_{i,j}^{x,new}(x, y) = (\rho_{i,j}^x(x, y), m_{i,j}^x(x, y), n_{i,j}^x(x, y), E_{i,j}^x(x, y))^T$ and $\mathbf{u}_{i,j}^{y,new}(x, y) = (\rho_{i,j}^y(x, y), m_{i,j}^y(x, y), n_{i,j}^y(x, y), E_{i,j}^y(x, y))^T$, and the cell averages of them are both $\bar{\mathbf{u}}_{i,j}^n = (\bar{\rho}_{i,j}^n, \bar{m}_{i,j}^n, \bar{n}_{i,j}^n, \bar{E}_{i,j}^n)^T$. We first use the following algorithm to get a new modified solution polynomial $\tilde{\mathbf{u}}_{i,j}^x(x, y)$.

1. In each cell, we modify the density first. Set up a small number ε such that $\bar{\rho}_{i,j}^n \geq \varepsilon$ for all cell $I_{i,j}$. In practice, we can choose $\varepsilon = 10^{-13}$. Replace $\rho_{i,j}^x(x, y)$ by

$$\hat{\rho}_{i,j}^x(x, y) = \theta_1(\rho_{i,j}^x(x, y) - \bar{\rho}_{i,j}^n) + \bar{\rho}_{i,j}^n, \quad \theta_1 = \min\left\{\frac{\bar{\rho}_{i,j}^n - \varepsilon}{\bar{\rho}_{i,j}^n - \rho_{min}^x}, 1\right\}, \quad (74)$$

with $\rho_{min}^x = \min_{(x,y) \in S_{i,j}^x} \rho_{i,j}^x(x, y)$.

2. Define $\hat{\mathbf{u}}_{i,j}^x(x, y) = (\hat{\rho}_{i,j}^x(x, y), m_{i,j}^x(x, y), n_{i,j}^x(x, y), E_{i,j}^x(x, y))^T$. For each $(x, y) \in S_{i,j}^x$, if $p(\hat{\mathbf{u}}_{i,j}^x(x, y)) \geq 0$, set $\theta_{x,y} = 1$; otherwise, set

$$\theta_{x,y} = \frac{p(\bar{\mathbf{u}}_{i,j}^n)}{p(\bar{\mathbf{u}}_{i,j}^n) - p(\hat{\mathbf{u}}_{i,j}^x(x, y))}. \quad (75)$$

Then get the limited polynomial

$$\tilde{\mathbf{u}}_{i,j}^x(x, y) = \theta_2(\hat{\mathbf{u}}_{i,j}^x(x, y) - \bar{\mathbf{u}}_{i,j}^n) + \bar{\mathbf{u}}_{i,j}^n, \quad \theta_2 = \min_{(x,y) \in S_{i,j}^x} \theta_{x,y}, \quad (76)$$

with $p(\tilde{\mathbf{u}}_{i,j}^x(x, y)) \geq 0, \forall (x, y) \in S_{i,j}^x$. See [36] for the proof of the positivity of the pressure.

Next, we compute the modified solution polynomial $\tilde{\mathbf{u}}_{i,j}^y(x, y)$.

1. In each cell, we modify the density first. Replace $\rho_{i,j}^y(x, y)$ by

$$\hat{\rho}_{i,j}^y(x, y) = \theta_1(\rho_{i,j}^y(x, y) - \bar{\rho}_{i,j}^n) + \bar{\rho}_{i,j}^n, \quad \theta_1 = \min\left\{\frac{\bar{\rho}_{i,j}^n - \varepsilon}{\bar{\rho}_{i,j}^n - \rho_{min}^y}, 1\right\}, \quad (77)$$

with $\rho_{min}^y = \min_{(x,y) \in S_{i,j}^y} \rho_{i,j}^y(x, y)$.

2. Define $\hat{\mathbf{u}}_{i,j}^y(x, y) = (\hat{\rho}_{i,j}^y(x, y), m_{i,j}^y(x, y), n_{i,j}^y(x, y), E_{i,j}^y(x, y))^T$. For each $(x, y) \in S_{i,j}^y$, if $p(\hat{\mathbf{u}}_{i,j}^y(x, y)) \geq 0$, set $\theta_{x,y} = 1$; otherwise ,set

$$\theta_{x,y} = \frac{p(\bar{\mathbf{u}}_{i,j}^n)}{p(\bar{\mathbf{u}}_{i,j}^n) - p(\hat{\mathbf{u}}_{i,j}^y(x, y))}. \quad (78)$$

Then get the limited polynomial

$$\tilde{\mathbf{u}}_{i,j}^y(x, y) = \theta_2(\hat{\mathbf{u}}_{i,j}^y(x, y) - \bar{\mathbf{u}}_{i,j}^n) + \bar{\mathbf{u}}_{i,j}^n, \quad \theta_2 = \min_{(x,y) \in S_{i,j}^y} \theta_{x,y}, \quad (79)$$

with $p(\tilde{\mathbf{u}}_{i,j}^y(x, y)) \geq 0, \forall (x, y) \in S_{i,j}^y$. See [36] for the proof.

In each cell $I_{i,j}$, we replace $\mathbf{u}_{i,j}^{x,new}(x, y)$ and $\mathbf{u}_{i,j}^{y,new}(x, y)$ by the two modified polynomials $\tilde{\mathbf{u}}_{i,j}^x(x, y)$ and $\tilde{\mathbf{u}}_{i,j}^y(x, y)$, respectively.

5 Numerical results

In this section, we provide numerical experiments to demonstrate the performance of the WENO limiter for the CPR framework. We use the third order TVD Runge-Kutta method [33] for the time discretization. Third ($K = 3$), fourth ($K = 4$) and fifth ($K = 5$) order CPR schemes with the WENO limiter are tested. Since the time discretization is only third order accurate, we take $\Delta t \sim \Delta x^{K/3}$ to obtain the high order accurate results for the accuracy test examples.

For all the numerical experiments, uniform meshes are tested. In several of them, we also show the results of non-uniform meshes. The non-uniform meshes are obtained from a 40% random perturbation of each node of the uniform mesh. Here, 40% denotes the total range of possible variation of the perturbation of the point position, i.e., 20% to the left and 20% to the right.

As we know, there are several different kinds of correction functions in the CPR framework. In our computation, the numerical results for different correction functions are similar, so we just give the results of one specific correction function. To demonstrate the correctness of our comments in Section 2.3, we provide the results with the SG/SD

type correction function in this section. Here, we let the solution points coincide with the inner roots of the correction functions and use the original LP approach. We can see that the order of accuracy will not be lost.

In general, there is no universal way to automatically choose the TVB constant M . A lemma for the choice of M in the one-dimensional scalar case is given in [7]. For all the accuracy tests, in order to see the effect of the WENO limiter on the accuracy of the CPR method, we use the TVB minmod limiter with a small TVB constant $M = 0.01$ to identify troubled cells, resulting in many good cells being identified as troubled cells. We list in each table the percentage of troubled cells among all the cells. For the figures, the solid lines are for the exact solutions or grid converged reference solutions, and the symbols are for the numerical solutions (just one point per cell is plotted).

Example 1. We first test the performance of the WENO limiter on the one-dimensional linear scalar problem

$$u_t + u_x = 0, \quad 0 \leq x \leq 2\pi, \quad (80)$$

with the initial condition $u(x, 0) = \sin(x)$ and a 2π -periodic boundary condition. The exact solution is $u(x, t) = \sin(x - t)$. The errors at $t = 2\pi$ are listed in Tables 1 and 2. We list in the last column of each table the percentage of troubled cells among all the cells. We can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the errors of the original CPR method.

Example 2. We consider the Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x = 0, \quad 0 \leq x \leq 2\pi, \quad (81)$$

with the initial condition $u(x, 0) = 0.5 + \sin(x)$ and periodic boundary conditions. The errors at $t = 0.5$ when the solution is smooth are listed in Tables 3 and 4. We can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the errors of the original CPR method. In Figure 1, we show the numerical results

Table 1: 1D linear equation with initial condition $u(x, 0) = \sin(x)$ at $t = 2\pi$. Uniform meshes.

N	CPR without limiter				CPR with WENO limiter (M=0.01)					
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage	
3rd order										
20	9.17E-05	–	2.76E-04	–	9.17E-05	–	2.76E-04	–	20.00%	
40	9.87E-06	3.22	3.15E-05	3.13	9.87E-06	3.22	3.15E-05	3.13	10.00%	
80	1.16E-06	3.09	3.73E-06	3.08	1.16E-06	3.09	3.73E-06	3.08	5.00%	
160	1.41E-07	3.04	4.53E-07	3.04	1.41E-07	3.04	4.53E-07	3.04	2.50%	
320	1.74E-08	3.02	5.57E-08	3.02	1.74E-08	3.02	5.57E-08	3.02	1.25%	
4th order										
20	1.37E-05	–	4.03E-05	–	1.37E-05	–	4.03E-05	–	20.00%	
40	8.56E-07	4.00	2.54E-06	3.99	8.56E-07	4.00	2.54E-06	3.99	10.00%	
80	5.36E-08	4.00	1.59E-07	4.00	5.36E-08	4.00	1.59E-07	3.99	5.00%	
160	3.35E-09	4.00	9.93E-09	4.00	3.35E-09	4.00	1.00E-08	3.99	2.50%	
320	2.09E-10	4.00	6.21E-10	4.00	2.09E-10	4.00	6.34E-10	3.98	1.25%	
5th order										
20	4.07E-06	–	6.46E-06	–	4.07E-06	–	6.47E-06	–	20.00%	
40	1.28E-07	5.00	2.01E-07	5.00	1.28E-07	5.00	2.02E-07	5.00	10.00%	
80	3.99E-09	5.00	6.28E-09	5.00	3.99E-09	5.00	6.29E-09	5.00	5.00%	
160	1.25E-10	5.00	1.96E-10	5.00	1.25E-10	5.00	1.97E-10	5.00	2.50%	
320	3.91E-12	4.99	6.22E-12	4.98	3.91E-12	4.99	6.21E-12	4.99	1.25%	

using 80 cells at $t = 1.5$, when a shock has already appeared in the solution. We can see that the schemes of all orders perform well in capturing this discontinuity without oscillations.

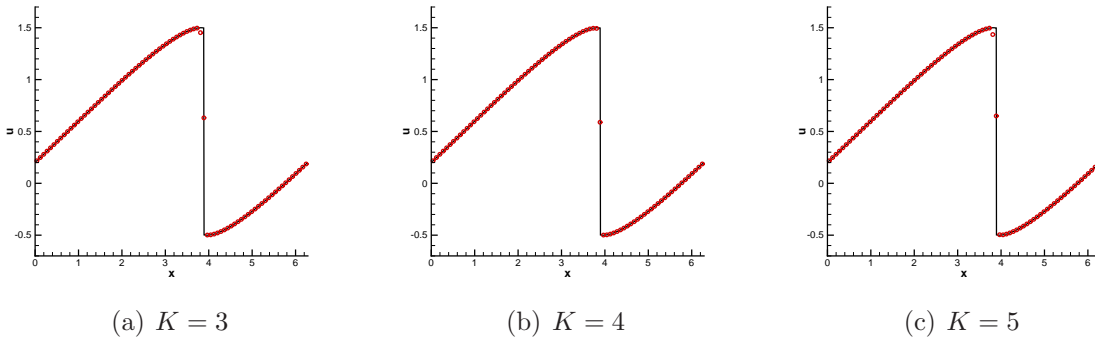


Figure 1: 1D Burgers equation at $t = 1.5$ with $N = 80$ cells

Example 3. Consider the one-dimensional Euler system described in Equation (64). The initial condition is set to be $\rho(x, 0) = 1 + 0.2 \sin(x)$, $v(x, 0) = 1$, $p(x, 0) = 1$, with

Table 2: 1D linear equation with initial condition $u(x, 0) = \sin(x)$ at $t = 2\pi$. Non-uniform meshes.

N	CPR without limiter				CPR with WENO limiter (M=0.01)					
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage	
3rd order										
20	1.27E-04	–	4.50E-04	–	1.27E-04	–	4.50E-04	–	20.00%	
40	1.64E-05	3.33	6.61E-05	3.13	1.64E-05	3.33	6.61E-05	3.13	12.50%	
80	2.18E-06	3.15	9.43E-06	3.04	2.18E-06	3.15	9.43E-06	3.04	6.25%	
160	2.61E-07	2.97	1.02E-06	3.12	2.61E-07	2.97	1.02E-06	3.12	3.12%	
320	3.29E-08	3.03	1.26E-07	3.06	3.29E-08	3.03	1.26E-07	3.06	1.56%	
4th order										
20	2.95E-05	–	8.51E-05	–	2.95E-05	–	8.51E-05	–	20.00%	
40	2.53E-06	4.01	6.88E-06	4.11	2.53E-06	4.01	6.88E-06	4.11	10.50%	
80	1.94E-07	4.01	5.20E-07	4.02	1.94E-07	4.01	5.20E-07	4.02	5.00%	
160	1.12E-08	4.00	3.01E-08	4.00	1.12E-08	4.00	3.01E-08	4.00	2.50%	
320	7.26E-10	4.00	2.06E-09	3.92	7.26E-10	4.00	2.06E-09	3.92	1.25%	
5th order										
20	1.10E-05	–	1.74E-05	–	1.10E-05	–	1.74E-05	–	20.00%	
40	5.14E-07	5.00	8.11E-07	5.00	5.14E-07	5.00	8.11E-07	5.00	10.00%	
80	2.08E-08	5.00	3.27E-08	5.00	2.08E-08	5.00	3.27E-08	5.00	5.00%	
160	5.89E-10	5.00	9.26E-10	5.00	5.89E-10	5.00	9.27E-10	5.00	2.50%	
320	1.92E-11	5.00	3.02E-11	5.00	1.92E-11	5.00	3.03E-11	5.00	1.25%	

a 2π -periodic boundary condition. The exact solution is $\rho(x, t) = 1 + 0.2 \sin(x - t)$, $v(x, t) = 1$ and $p(x, t) = 1$. The errors for the density at $t = 2\pi$ are listed in Tables 5 and 6. We can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the errors of the original CPR method.

Example 4. We consider two well-known problems of the Euler equations (64) with the following Riemann type initial conditions

$$\mathbf{u}(x, 0) = \begin{cases} \mathbf{u}_L, & -5 \leq x < 0, \\ \mathbf{u}_R, & 0 \leq x \leq 5. \end{cases} \quad (82)$$

The first one is the Sod problem in which the initial condition is

$$(\rho_L, v_L, p_L) = (1, 0, 1); \quad (\rho_R, v_R, p_R) = (0.125, 0, 0.1). \quad (83)$$

The second one is the Lax problem in which the initial condition is

$$(\rho_L, v_L, p_L) = (0.445, 0.698, 3.528); \quad (\rho_R, v_R, p_R) = (0.5, 0, 0.571). \quad (84)$$

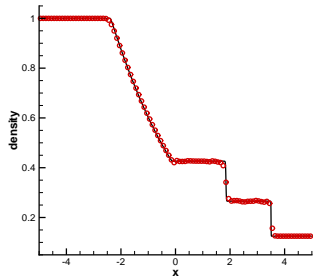
Table 3: 1D Burgers equation with initial condition $u(x, 0) = 0.5 + \sin(x)$ at $t = 0.5$. Uniform meshes.

N	CPR without limiter				CPR with WENO limiter (M=0.01)				
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage
3rd order									
20	1.93E-04	–	1.90E-03	–	1.93E-04	–	1.90E-03	–	35.00%
40	2.17E-05	3.15	3.09E-04	2.62	2.17E-05	3.15	3.09E-04	2.62	20.00%
80	2.39E-06	3.18	4.75E-05	2.70	2.39E-06	3.18	4.75E-05	2.70	10.00%
160	2.66E-07	3.17	6.36E-06	2.90	2.66E-07	3.17	6.36E-06	2.90	5.63%
320	3.06E-08	3.12	8.21E-07	2.95	3.06E-08	3.12	8.21E-07	2.95	2.81%
4th order									
20	1.74E-05	–	4.87E-04	–	1.74E-05	–	4.87E-04	–	30.00%
40	9.17E-07	4.24	6.40E-05	2.93	9.17E-07	4.24	6.40E-05	2.93	17.50%
80	5.03E-08	4.19	4.57E-06	3.81	5.03E-08	4.19	4.57E-06	3.81	8.75%
160	2.68E-09	4.23	3.11E-07	3.88	2.68E-09	4.23	3.11E-07	3.88	4.38%
320	1.44E-10	4.22	2.03E-08	3.93	1.44E-10	4.22	2.03E-08	3.93	2.19%
5th order									
20	1.49E-06	–	8.32E-05	–	1.49E-06	–	8.32E-05	–	30.00%
40	4.36E-08	5.10	5.09E-06	4.03	4.36E-08	5.10	5.09E-06	4.03	15.00%
80	1.17E-09	5.21	1.83E-07	4.79	1.17E-09	5.21	1.83E-07	4.79	7.50%
160	2.99E-11	5.30	6.89E-09	4.73	2.99E-11	5.30	6.89E-09	4.73	3.75%
320	7.97E-13	5.23	2.30E-10	4.90	7.97E-13	5.23	2.30E-10	4.90	1.88%

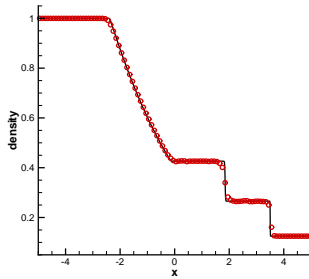
The numerical results of the Sod problem at $t = 2$ and the results of the Lax problem at $t = 1.3$ are shown in Figures 2 and 3. To save space, we only show the plots for density. For the TVB constant $M = 0.01$, we can see that there is no oscillation near the discontinuities, however we also observe rather severe smearing, due to the strong WENO limiter. The results become more accurate by relaxing the limiting through adjusting M . For the Sod problem, we relax the limiting by taking $M = 5, 5, 10$ for $K = 3, 4, 5$ respectively. For the Lax problem, we relax the limiting by taking $M = 10, 15, 30$ for $K = 3, 4, 5$ respectively.

Table 4: 1D Burgers equation with initial condition $u(x, 0) = 0.5 + \sin(x)$ at $t = 0.5$. Non-uniform meshes.

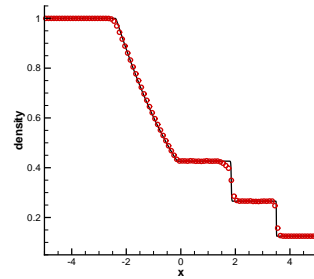
N	CPR without limiter				CPR with WENO limiter (M=0.01)					
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage	
3rd order										
20	2.37E-04	–	1.20E-03	–	2.37E-04	–	1.20E-03	–	35.00%	
40	2.50E-05	3.39	2.49E-04	2.38	2.50E-05	3.39	2.49E-04	2.38	20.00%	
80	2.58E-06	3.01	5.99E-05	1.89	2.58E-06	3.01	5.99E-05	1.89	11.25%	
160	3.18E-07	3.15	1.18E-05	2.44	3.18E-07	3.15	1.18E-05	2.44	5.63%	
320	3.84E-08	3.10	1.87E-06	2.71	3.84E-08	3.10	1.87E-06	2.71	2.81%	
4th order										
20	1.98E-05	–	1.20E-03	–	1.98E-05	–	1.20E-03	–	30.00%	
40	1.28E-06	4.14	7.81E-05	4.12	1.28E-06	4.14	7.81E-05	4.12	17.50%	
80	5.79E-08	4.10	5.74E-06	3.46	5.79E-08	4.10	5.74E-06	3.46	8.75%	
160	3.63E-09	4.16	6.86E-07	3.19	3.63E-09	4.16	6.86E-07	3.19	4.38%	
320	1.98E-10	4.27	3.66E-08	4.30	1.98E-10	4.27	3.66E-08	4.30	2.19%	
5th order										
20	2.68E-06	–	1.91E-04	–	2.68E-06	–	1.91E-04	–	30.00%	
40	5.30E-08	5.93	3.22E-06	6.16	5.30E-08	5.93	3.22E-06	6.16	17.50%	
80	1.73E-09	4.54	2.69E-07	3.29	1.73E-09	4.54	2.69E-07	3.29	7.50%	
160	5.26E-11	5.25	1.77E-08	4.09	5.26E-11	5.25	1.77E-08	4.09	4.38%	
320	1.62E-12	5.11	7.97E-10	4.55	1.62E-12	5.11	7.97E-10	4.55	1.88%	



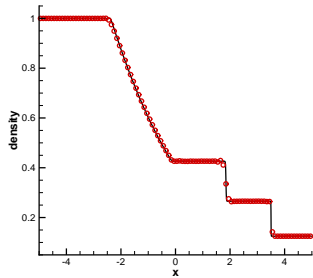
(a) $K = 3, M = 0.01$



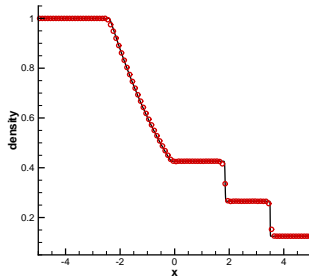
(b) $K = 4, M = 0.01$



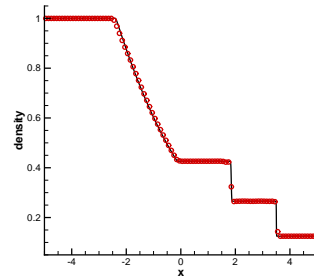
(c) $K = 5, M = 0.01$



(d) $K = 3, M = 5$



(e) $K = 4, M = 5$



(f) $K = 5, M = 10$

Figure 2: Sod problem at $t = 2$ with $N = 100$ cells.

Table 5: 1D Euler equation with initial condition $\rho(x, 0) = 1 + 0.2 \sin(x)$, $v(x, 0) = 1$, $p(x, 0) = 1$ at $t = 2\pi$. Errors for the density. Uniform meshes.

N	CPR without limiter				CPR with WENO limiter (M=0.01)				
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage
3rd order									
20	2.37E-05	–	1.00E-04	–	2.37E-05	–	1.00E-04	–	30.00%
40	2.71E-06	3.13	1.28E-05	2.96	2.71E-06	3.13	1.28E-05	2.96	20.00%
80	3.16E-07	3.10	1.60E-06	3.01	3.16E-07	3.10	1.60E-06	3.01	10.00%
160	3.78E-08	3.06	1.98E-07	3.01	3.78E-08	3.06	1.98E-07	3.01	5.63%
320	4.61E-09	3.04	2.46E-08	3.01	4.61E-09	3.04	2.46E-08	3.01	2.81%
4th order									
20	3.14E-07	–	3.26E-06	–	3.14E-07	–	3.26E-06	–	30.00%
40	1.95E-08	4.01	2.02E-07	4.01	1.95E-08	4.01	2.02E-07	4.01	17.50%
80	1.22E-09	4.00	1.26E-08	4.00	1.22E-09	4.00	1.27E-08	3.99	8.75%
160	7.60E-11	4.00	7.88E-10	4.00	7.60E-11	4.00	8.05E-10	3.98	4.38%
320	4.75E-12	4.00	4.93E-11	4.00	4.74E-12	4.00	5.19E-11	3.96	2.19%
5th order									
20	2.95E-08	–	1.35E-07	–	2.95E-08	–	1.35E-07	–	30.00%
40	9.21E-10	5.00	4.20E-09	5.01	9.21E-10	5.00	4.20E-09	5.01	15.00%
80	2.87E-11	5.00	1.29E-10	5.02	2.87E-11	5.00	1.29E-10	5.02	7.50%
160	8.93E-13	5.01	4.03E-12	5.00	9.14E-13	4.98	4.11E-12	4.98	3.75%

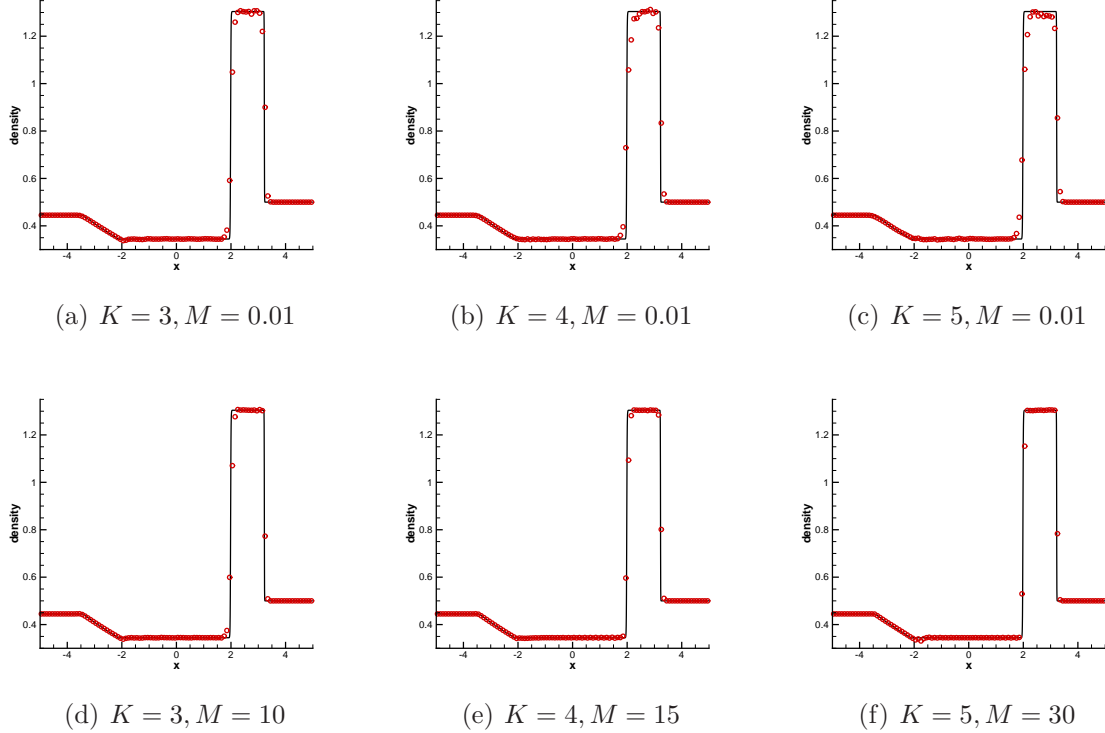


Figure 3: Lax problem at $t = 1.3$ with $N = 100$ cells.

Table 6: 1D Euler equation with initial condition $\rho(x, 0) = 1 + 0.2 \sin(x)$, $v(x, 0) = 1$, $p(x, 0) = 1$ at $t = 2\pi$. Errors for the density. Non-uniform meshes.

N	CPR without limiter				CPR with WENO limiter (M=0.01)				
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage
3rd order									
20	2.96E-05	–	1.38E-04	–	2.97E-05	–	1.38E-04	–	35.00%
40	3.19E-06	3.37	1.77E-05	3.11	3.19E-06	3.37	1.77E-05	3.11	20.00%
80	3.49E-07	2.93	1.95E-06	2.92	3.49E-07	2.93	1.95E-06	2.92	10.00%
160	4.22E-08	3.17	2.74E-07	2.95	4.22E-08	3.17	2.74E-07	2.95	5.63%
320	5.18E-09	3.08	3.72E-08	2.93	5.18E-09	3.08	3.72E-08	2.93	2.81%
4th order									
20	4.33E-07	–	1.26E-05	–	4.31E-07	–	1.26E-05	–	30.00%
40	1.84E-08	4.77	7.35E-07	4.29	1.84E-08	4.76	7.35E-07	4.29	17.50%
80	9.26E-10	3.96	3.80E-08	3.93	9.28E-10	3.96	3.80E-08	3.93	8.75%
160	5.87E-11	4.14	2.86E-09	3.89	5.87E-11	4.14	2.86E-09	3.89	4.38%
320	3.67E-12	4.07	2.06E-10	3.86	3.70E-12	4.06	2.04E-10	3.87	2.19%
5th order									
20	2.21E-08	–	3.48E-07	–	2.22E-08	–	3.48E-07	–	30.00%
40	7.74E-10	5.06	9.76E-09	5.40	7.74E-10	5.07	9.76E-09	5.40	15.00%
80	1.83E-11	4.97	2.47E-10	4.87	1.83E-11	4.97	2.47E-10	4.87	7.50%
160	6.46E-13	5.02	9.47E-12	4.90	6.63E-13	4.98	9.40E-12	4.91	3.75%

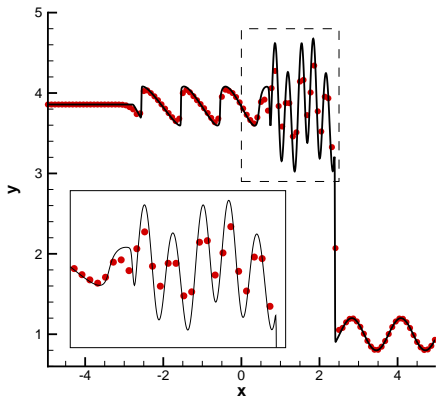
Example 5. To demonstrate the advantage of higher order methods, we use the Euler equation (64) with the initial condition

$$\mathbf{u}(x, 0) = \begin{cases} \mathbf{u}_L, & x < -4, \\ \mathbf{u}_R, & x \geq -4, \end{cases} \quad (85)$$

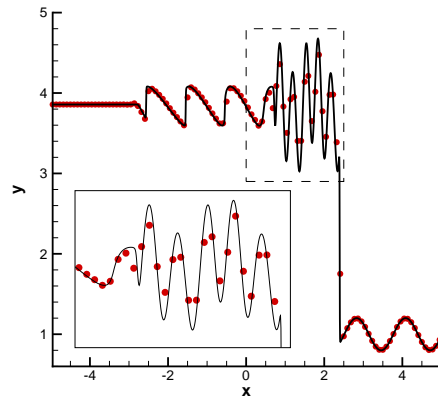
where $(\rho_L, v_L, p_L) = (3.857143, 2.629369, 10.333333)$ and $(\rho_R, v_R, p_R) = (1+0.2 \sin(5x), 0, 1)$.

This example describes the interaction of a Mach 3 shock with a density wave [34]. A Mach 3 shock is initially located at $x = -4$ and moves to the right. A sine wave is superimposed to the density in the right region of the shock. It contains both shocks and fine structures in smooth regions. The numerical results at $t = 1.8$ are shown in Figure 4. For $K = 4$ and $K = 5$, we add the positivity-preserving limiter to avoid negative density or negative pressure during the time evolution. The solid lines are the reference “exact” solution, which is a converged solution computed by the fifth order finite difference WENO scheme [16] with 2000 grid points. We use the TVB constants $M = 0.01$ and $M = 300$, respectively. For $M = 0.01$, we can observe some smearing.

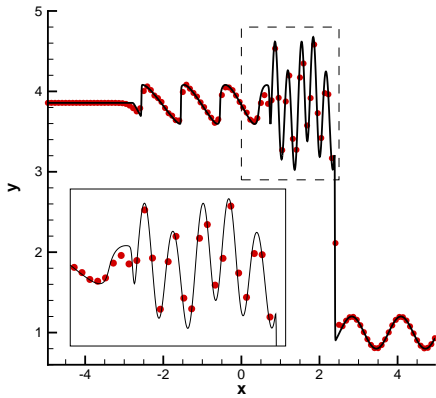
The results become better as M increases.



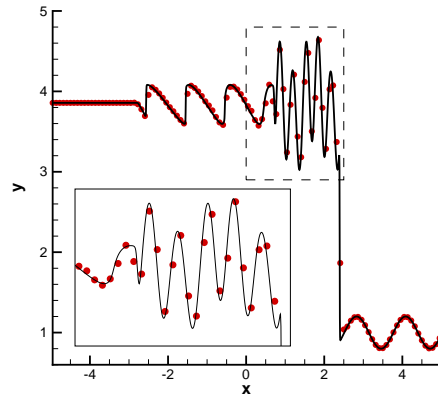
(a) $K = 3, M = 0.01$



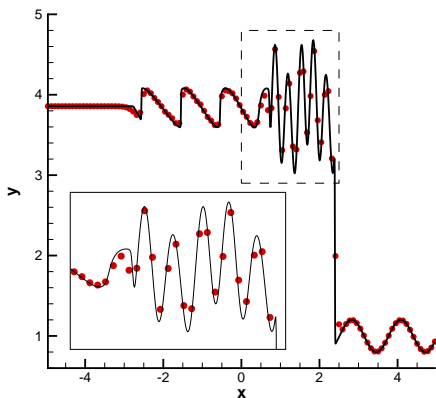
(b) $K = 3, M = 300$



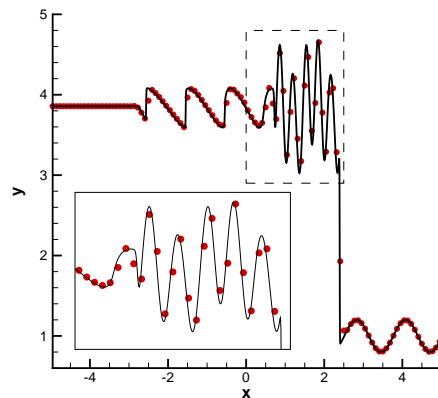
(c) $K = 4, M = 0.01$



(d) $K = 4, M = 300$



(e) $K = 5, M = 0.01$



(f) $K = 5, M = 300$

Figure 4: **The shock density wave interaction problem.** $t = 1.8$. $N = 110$.

Example 6. Now we consider the interaction of blast waves of the Euler equations (64) with the initial condition

$$\mathbf{u}(x, 0) = \begin{cases} \mathbf{u}_L, & 0 \leq x < 0.1, \\ \mathbf{u}_M, & 0.1 \leq x < 0.9, \\ \mathbf{u}_R, & 0.9 \leq x < 1, \end{cases} \quad (86)$$

where $(\rho_L, v_L, p_L) = (1, 0, 10^3)$, $(\rho_M, v_M, p_M) = (1, 0, 10^{-2})$ and $(\rho_R, v_R, p_R) = (1, 0, 10^2)$. A reflecting boundary condition is applied to both ends. The numerical results at $t = 0.038$ with $M = 0.01$ and $M = 200$ are plotted in Figure 5. The solid lines are the reference “exact” solution, which is a converged solution computed by the fifth order finite difference WENO scheme [16] with 16000 grid points. The positivity-preserving limiter is added to the schemes of all orders. We can see that the results are well resolved and we have not noticed any significant difference for taking M from 0.01 to 200.

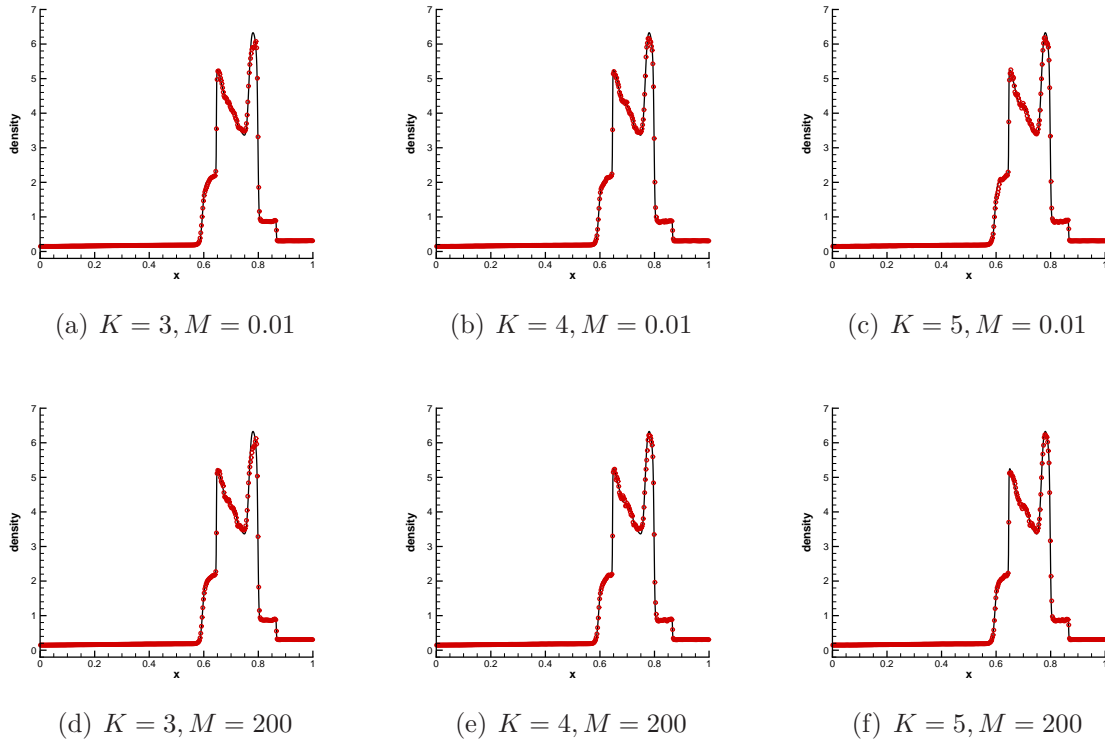


Figure 5: The blast wave problem at $t = 0.038$. $N = 400$.

Example 7. Consider the LeBlanc shock tube problem of the Euler equations (64) with

the initial condition

$$\mathbf{u}(x, 0) = \begin{cases} \mathbf{u}_L, & -10 \leq x < 0, \\ \mathbf{u}_R, & 0 \leq x < 10, \end{cases} \quad (87)$$

where $(\rho_L, v_L, p_L) = (2, 0, 10^9)$ and $(\rho_R, v_R, p_R) = (0.001, 0, 1)$. The positivity-preserving limiter is added to the schemes of all orders. The numerical solutions with $N = 2400$ and $N = 19200$ are shown in Figures 6-8. For $N = 2400$ grid points, the TVB constant $M = 200$ is used. For $N = 19200$ grid points, we use $M = 500$, $M = 800$ and $M = 10000$ for $K = 3$, $K = 4$ and $K = 5$ respectively. We can see that the results with $N = 2400$ are well resolved. As we increase the number of the grid point N and the TVB constant M , the results become better resolved. For this extreme Riemann problem, there are slight over- and under-shoots near the contact discontinuity, which also exist when using other WENO schemes, e.g. in [17].

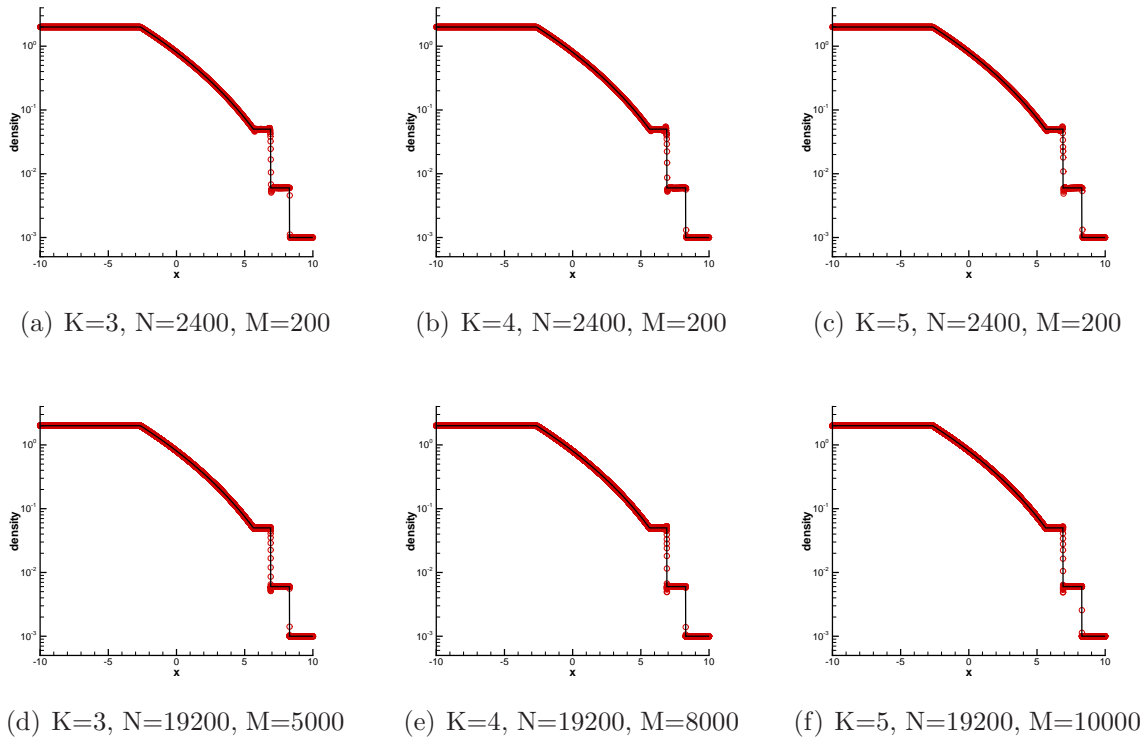
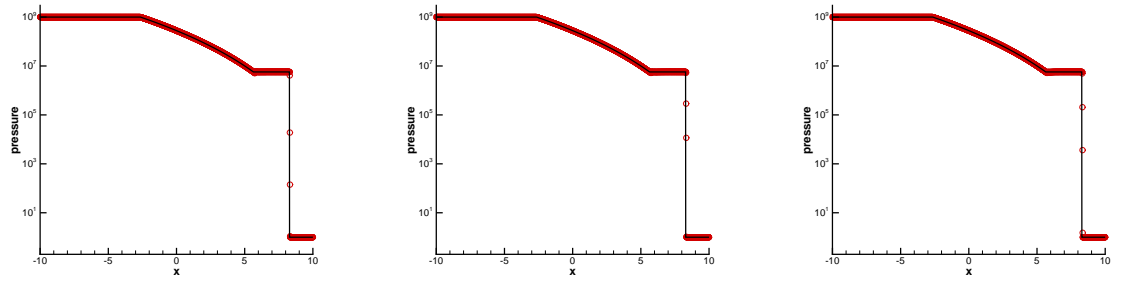


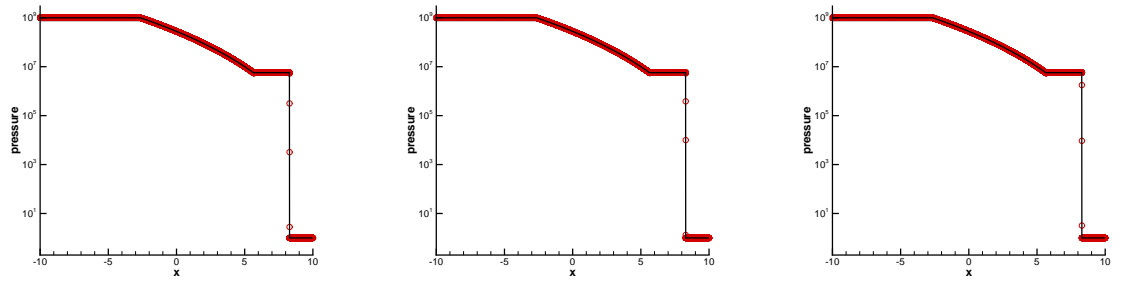
Figure 6: LeBlanc problem at $t = 0.0001$. Log plot of density.



(a) $K=3, N=2400, M=200$

(b) $K=4, N=2400, M=200$

(c) $K=5, N=2400, M=200$

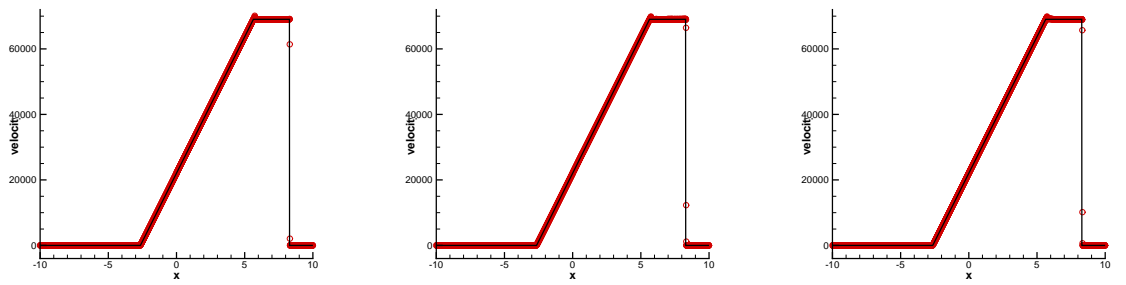


(d) $K=3, N=19200, M=5000$

(e) $K=4, N=19200, M=8000$

(f) $K=5, N=19200, M=10000$

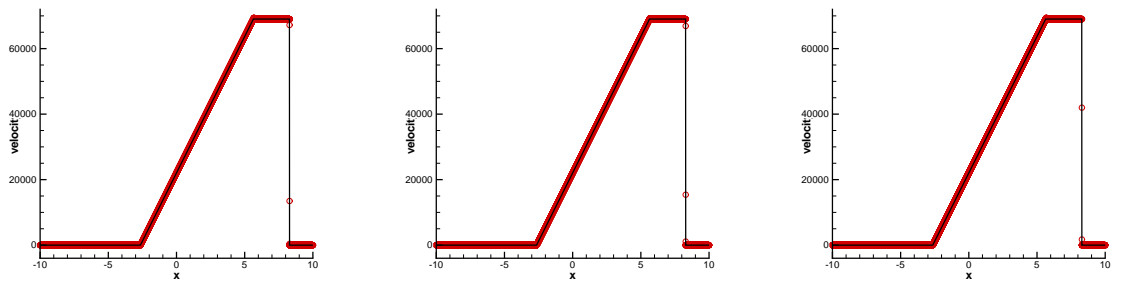
Figure 7: LeBlanc problem at $t = 0.0001$. Log plot of pressure.



(a) $K=3, N=2400, M=200$

(b) $K=4, N=2400, M=200$

(c) $K=5, N=2400, M=200$



(d) $K=3, N=19200, M=5000$

(e) $K=4, N=19200, M=8000$

(f) $K=5, N=19200, M=10000$

Figure 8: LeBlanc problem at $t = 0.0001$. Plot of velocity.

Example 8. Consider the two-dimensional linear equation

$$u_t + au_x + bu_y = 0, \quad 0 \leq x, y \leq 2\pi, \quad (88)$$

with the initial condition $u(x, y, 0) = \sin(x + y)$ and a 2π -periodic boundary condition. Here, a and b are constants. We take $a = 1$ and $b = -2$ in the computation. The errors at $t = 2\pi$ are listed in Tables 7 and 8. We can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the errors of the original CPR method.

Table 7: 2D linear equation with initial condition $u(x, y, 0) = \sin(x + y)$ at $t = 2\pi$. Uniform meshes.

$N \times N$	CPR without limiter				CPR with WENO limiter (M=0.01)					
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage	
3rd order										
20×20	8.13E-05	–	2.75E-04	–	8.13E-05	–	2.75E-04	–	23.46%	
40×40	8.50E-06	3.26	3.05E-05	3.17	8.50E-06	3.26	3.05E-05	3.17	11.92%	
80×80	9.70E-07	3.13	3.59E-06	3.09	9.70E-07	3.13	3.59E-06	3.09	6.00%	
160×160	1.16E-07	3.07	4.34E-07	3.05	1.16E-07	3.07	4.34E-07	3.05	3.00%	
320×320	1.41E-08	3.03	5.34E-08	3.02	1.41E-08	3.03	5.34E-08	3.02	1.50%	
4th order										
20×20	9.98E-07	–	2.06E-05	–	9.98E-07	–	2.06E-05	–	23.25%	
40×40	5.94E-08	4.07	1.29E-06	4.00	5.94E-08	4.07	1.29E-06	4.00	11.74%	
80×80	3.66E-09	4.02	8.05E-08	4.00	3.66E-09	4.02	8.06E-08	4.00	5.88%	
160×160	2.28E-10	4.01	5.03E-09	4.00	2.28E-10	4.01	5.07E-09	3.99	2.93%	
320×320	1.50E-11	3.93	3.15E-10	4.00	1.50E-11	3.92	3.21E-10	3.98	1.46%	
5th order										
20×20	7.04E-08	–	3.98E-07	–	7.04E-08	–	3.97E-07	–	23.13%	
40×40	2.19E-09	5.01	1.21E-08	5.04	2.19E-09	5.01	1.21E-08	5.04	11.68%	
80×80	6.85E-11	5.00	3.75E-10	5.01	6.85E-11	5.00	3.75E-10	5.01	5.85%	
160×160	2.58E-12	4.73	9.92E-12	5.24	2.63E-12	4.71	9.92E-12	5.24	2.92%	

Example 9. Consider the two-dimensional nonlinear Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0, \quad 0 \leq x, y \leq 2\pi, \quad (89)$$

with the initial condition $u(x, y, 0) = 0.5 + \sin(x + y)$ and periodic boundary conditions. We show the errors at $t = 0.25$ (smooth solution) in Tables 9 and 10. Again, we can see

Table 8: 2D linear equation with initial condition $u(x, y, 0) = \sin(x + y)$ at $t = 2\pi$. Non-uniform meshes.

$N \times N$	CPR without limiter				CPR with WENO limiter (M=0.01)					
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage	
3rd order										
20×20	1.05E-04	–	6.47E-04	–	1.05E-04	–	6.47E-04	–	23.46%	
40×40	1.02E-05	3.11	6.10E-05	3.15	1.02E-05	3.11	6.10E-05	3.15	11.92%	
80×80	1.23E-06	3.16	8.51E-06	2.94	1.23E-06	3.16	8.51E-06	2.94	6.00%	
160×160	1.50E-07	3.09	1.08E-06	3.03	1.50E-07	3.09	1.08E-06	3.03	3.00%	
320×320	1.83E-08	3.15	1.42E-07	3.05	1.83E-08	3.15	1.42E-07	3.05	1.50%	
4th order										
20×20	2.20E-06	–	7.25E-05	–	2.20E-06	–	7.25E-05	–	24.09%	
40×40	1.00E-07	4.12	3.95E-06	3.88	1.00E-07	4.12	3.95E-06	3.88	12.00%	
80×80	6.31E-09	4.13	2.45E-07	4.14	6.31E-09	4.13	2.45E-07	4.14	6.04%	
160×160	3.99E-10	4.05	1.68E-08	3.94	3.99E-10	4.05	1.68E-08	3.93	3.02%	
320×320	2.61E-11	4.08	1.19E-09	3.96	2.61E-11	4.08	1.19E-09	3.96	1.50%	
5th order										
20×20	5.88E-08	–	1.69E-06	–	5.90E-08	–	1.69E-06	–	23.75%	
40×40	1.31E-09	5.07	4.46E-08	4.85	1.32E-09	5.07	4.46E-08	4.85	11.86%	
80×80	4.48E-11	5.04	1.40E-09	5.16	4.48E-11	5.04	1.40E-09	5.16	5.98%	
160×160	1.74E-12	4.76	4.84E-11	4.94	1.79E-12	4.72	4.84E-11	4.94	2.99%	

that the WENO limiter maintains both the designed order of accuracy and the magnitude of the errors of the original CPR method. At $t = 0.5$, a shock begins to form. We plot the solution surfaces at $t = 0.75$ in Figure 9 and the solution cuts in the diagonal cells in Figure 10. We can see that the shocks are well resolved without spurious oscillations.

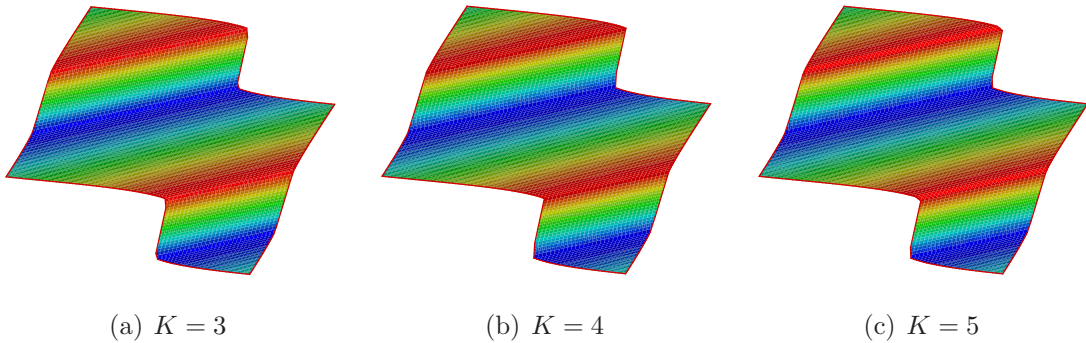


Figure 9: **2D Burgers solution at $t = 0.75$ with 80×80 cells.**

Table 9: 2D Burgers equation at $t = 0.25$. Uniform meshes.

$N \times N$	CPR without limiter				CPR with WENO limiter (M=0.01)					
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage	
3rd order										
20×20	2.24E-04	–	2.06E-03	–	2.24E-04	–	2.06E-03	–	24.83%	
40×40	2.87E-05	2.96	4.78E-04	2.11	2.87E-05	2.96	4.78E-04	2.11	12.46%	
80×80	3.63E-06	2.99	7.88E-05	2.60	3.63E-06	2.99	7.88E-05	2.60	6.21%	
160×160	4.53E-07	3.00	1.17E-05	2.75	4.53E-07	3.00	1.17E-05	2.75	3.08%	
320×320	5.62E-08	3.01	1.57E-06	2.89	5.62E-08	3.01	1.57E-06	2.89	1.54%	
4th order										
20×20	2.08E-05	–	6.07E-04	–	2.08E-05	–	6.07E-04	–	24.50%	
40×40	1.28E-06	4.02	8.42E-05	2.85	1.28E-06	4.02	8.42E-05	2.85	12.09%	
80×80	7.84E-08	4.03	7.39E-06	3.51	7.84E-08	4.03	7.39E-06	3.51	5.97%	
160×160	4.65E-09	4.08	5.54E-07	3.74	4.65E-09	4.08	5.54E-07	3.74	2.97%	
320×320	2.76E-10	4.07	3.81E-08	3.86	2.76E-10	4.07	3.81E-08	3.86	1.47%	
5th order										
20×20	2.39E-06	–	6.65E-05	–	2.39E-06	–	6.65E-05	–	24.10%	
40×40	8.14E-08	4.88	6.05E-06	3.46	8.14E-08	4.88	6.05E-06	3.46	11.94%	
80×80	2.35E-09	5.11	2.59E-07	4.55	2.35E-09	5.11	2.59E-07	4.55	5.89%	
160×160	6.84E-11	5.10	1.13E-08	4.52	6.84E-11	5.10	1.13E-08	4.52	2.93%	
320×320	2.04E-12	5.07	4.17E-10	4.76	2.04E-12	5.07	4.17E-10	4.76	1.46%	

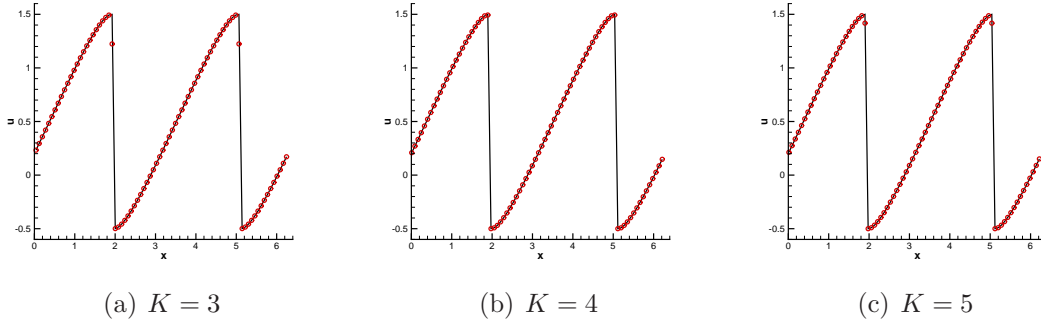


Figure 10: **2D Burgers solution that cuts along the diagonal at $t = 0.75$ with 80×80 cells.**

Example 10. Consider the two-dimensional Euler system described in Equation (72). The initial condition is set to be $\rho(x, y, 0) = 1 + 0.2 \sin(x + y)$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$ and $p(x, y, 0) = 1$. The boundary conditions are periodic. The exact solution is $\rho(x, y, t) = 1 + 0.2 \sin(x + y - t)$, $u(x, y, t) = 0.7$, $v(x, y, t) = 0.3$ and $p(x, y, t) = 1$. Table 11 shows the L_1 and L_∞ errors for the density at $t = 2\pi$. We can see that the WENO

Table 10: 2D Burgers equation at $t = 0.25$. Non-uniform meshes.

$N \times N$	CPR without limiter				CPR with WENO limiter (M=0.01)				
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage
3rd order									
20×20	2.86E-04	–	4.62E-03	–	2.86E-04	–	4.62E-03	–	26.25%
40×40	3.49E-05	2.81	7.63E-04	2.40	3.49E-05	2.81	7.63E-04	2.40	12.78%
80×80	4.57E-06	3.03	1.51E-04	2.42	4.57E-06	3.03	1.51E-04	2.42	6.43%
160×160	5.71E-07	3.05	2.54E-05	2.61	5.71E-07	3.05	2.54E-05	2.61	3.22%
320×320	6.99E-08	3.15	3.62E-06	2.92	6.99E-08	3.15	3.62E-06	2.92	1.61%
4th order									
20×20	3.45E-05	–	1.45E-03	–	3.45E-05	–	1.45E-03	–	25.72%
40×40	1.88E-06	3.88	1.64E-04	2.91	1.88E-06	3.88	1.64E-04	2.91	12.45%
80×80	1.24E-07	4.05	1.83E-05	3.27	1.24E-07	4.05	1.83E-05	3.27	6.18%
160×160	7.72E-09	4.08	1.64E-06	3.55	7.72E-09	4.08	1.64E-06	3.55	3.07%
320×320	4.71E-10	4.19	1.13E-07	4.01	4.71E-10	4.19	1.13E-07	4.01	1.53%
5th order									
20×20	6.95E-06	–	2.34E-04	–	6.95E-06	–	2.34E-04	–	25.30%
40×40	1.81E-07	4.87	1.24E-05	3.92	1.81E-07	4.87	1.24E-05	3.92	12.23%
80×80	6.17E-09	5.04	8.80E-07	3.95	6.17E-09	5.04	8.80E-07	3.95	6.07%
160×160	1.98E-10	5.05	4.38E-08	4.40	1.98E-10	5.05	4.38E-08	4.40	3.02%
320×320	6.54E-12	5.11	1.63E-09	4.93	6.54E-12	5.11	1.63E-09	4.93	1.50%

limiter maintains both the designed order of accuracy and the magnitude of the error of the original CPR method.

Example 11. Consider the two-dimensional vortex evolution problem, which is an idealized problem for the 2D Euler equations [32]. The setup of this problem is: The mean flow is $\rho = 1$, $p = 1$ and $(u, v) = (1, 1)$ (diagonal flow). We add, to this mean flow, an isentropic vortex (perturbation in (u, v) and the temperature $T = \frac{p}{\rho}$, no perturbation in the entropy $S = \frac{p}{\rho^\gamma}$):

$$(\delta u, \delta v) = \frac{\epsilon}{2\pi} e^{0.5(1-t^2)} (-\bar{y}, \bar{x}), \quad \delta T = -\frac{(\gamma-1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad \delta S = 0, \quad (90)$$

where $(\bar{x}, \bar{y}) = (x - 7, y - 7)$, $r^2 = \bar{x}^2 + \bar{y}^2$, and the vortex strength $\epsilon = 5$. Since the mean flow is in the diagonal direction, the vortex movement is not aligned with the mesh direction. The computational domain is taken as $[0, 14] \times [0, 14]$, extended periodically in both directions. It is clear that the exact solution of the Euler equation with the above initial and boundary conditions is just the passive convection of the vortex with

Table 11: 2D Euler equation with initial condition $\rho(x, y, 0) = 1 + 0.2 \sin(x + y)$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$ and $p(x, y, 0) = 1$ at $t = 2\pi$. Uniform meshes.

$N \times N$	CPR without limiter				CPR with WENO limiter (M=0.01)				
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage
3rd order									
20×20	1.83E-05	–	8.90E-05	–	2.29E-05	–	9.63E-05	–	23.46%
40×40	2.14E-06	3.10	1.09E-05	3.02	2.43E-06	3.24	1.14E-05	3.07	11.91%
80×80	2.58E-07	3.05	1.36E-06	3.01	2.76E-07	3.14	1.39E-06	3.04	6.00%
160×160	3.16E-08	3.03	1.69E-07	3.01	3.27E-08	3.08	1.71E-07	3.02	2.96%
4th order									
20×20	4.68E-07	–	8.79E-06	–	3.26E-07	–	8.13E-06	–	23.13%
40×40	2.07E-08	4.50	5.17E-07	4.09	1.97E-08	4.05	5.11E-07	3.99	11.70%
80×80	1.20E-09	4.11	3.18E-08	4.03	1.23E-09	4.00	3.24E-08	3.98	5.85%
160×160	7.44E-11	4.01	1.99E-09	4.00	7.75E-11	3.99	2.09E-09	3.96	2.91%
5th order									
20×20	3.24E-08	–	1.88E-07	–	9.42E-08	–	2.44E-07	–	22.93%
40×40	1.03E-09	4.97	6.06E-09	4.95	2.91E-09	5.02	7.32E-09	5.06	11.61%
80×80	3.14E-11	5.04	1.79E-10	5.08	9.01E-11	5.01	2.23E-10	5.03	5.82%
160×160	9.75E-13	5.01	5.74E-12	4.96	2.81E-12	5.00	7.16E-12	4.96	2.89%

the mean velocity. The errors and orders of accuracy for the density at $t = 0.2$ are shown in Table 12. We can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the error of the original CPR method.

Example 12. We consider the double Mach reflection problem. The computational domain is set to be $[0, 4] \times [0, 1]$. The reflection wall lies at the bottom of the computational domain starting from $x = \frac{1}{6}$. Initially a right-moving Mach 10 shock is positioned at $x = \frac{1}{6}, y = 0$ and makes a 60° angle with the x -axis. For the bottom boundary, the exact post-shock condition is imposed for the part from $x = 0$ to $x = \frac{1}{6}$, and a reflective boundary condition is used for the rest. At the top boundary of the computational domain, the flow values are set to describe the exact motion of the Mach 10 shock. The initial pre-shock condition is

$$(\rho, p, u, v) = (1.4, 1, 0, 0), \quad (91)$$

Table 12: 2D Euler system of the smooth vortex evolution problem at $t = 0.2$. Uniform meshes.

$N \times N$	CPR without limiter				CPR with WENO limiter (M=0.01)					
	L^1 norm	order	L^∞ norm	order	L^1 norm	order	L^∞ norm	order	percentage	
3rd order										
20×20	6.90E-05	–	3.21E-03	–	8.18E-05	–	4.01E-03	–	16.46%	
40×40	8.98E-06	2.94	4.50E-04	2.83	8.98E-06	3.19	4.46E-04	3.17	7.90%	
80×80	1.19E-06	2.92	6.51E-05	2.79	1.19E-06	2.92	6.51E-05	2.78	4.90%	
160×160	1.37E-07	3.12	7.99E-06	3.03	1.37E-07	3.12	7.99E-06	3.03	3.33%	
320×320	1.52E-08	3.17	1.04E-06	2.94	1.52E-08	3.17	1.04E-06	2.94	2.47%	
4th order										
20×20	1.46E-05	–	3.15E-03	–	1.84E-05	–	4.03E-03	–	16.38%	
40×40	8.69E-07	4.07	1.32E-04	4.57	8.72E-07	4.40	1.32E-04	4.93	7.96%	
80×80	4.05E-08	4.42	9.25E-06	3.84	4.05E-08	4.43	9.25E-06	3.84	4.89%	
160×160	2.16E-09	4.23	8.00E-07	3.53	2.16E-09	4.23	8.00E-07	3.53	3.33%	
320×320	1.15E-10	4.24	5.13E-08	3.96	1.15E-10	4.24	5.13E-08	3.96	2.46%	
5th order										
20×20	1.45E-06	–	2.85E-04	–	6.97E-06	–	7.04E-03	–	16.35%	
40×40	3.95E-08	5.20	8.83E-06	5.01	3.94E-08	7.47	8.82E-06	9.64	7.90%	
80×80	1.23E-09	5.00	3.26E-07	4.76	1.23E-09	5.00	3.26E-07	4.76	4.86%	
160×160	3.17E-11	5.28	9.64E-09	5.08	3.17E-11	5.28	9.57E-09	5.09	3.32%	
320×320	7.98E-13	5.31	2.95E-10	5.03	8.01E-13	5.31	2.94E-10	5.02	2.45%	

and the post-shock condition is

$$(\rho, p, u, v) = (8, 116.5, 8.25 \cos(30^\circ), -8.25 \sin(30^\circ)). \quad (92)$$

We compute the solution of the 3rd order scheme up to $t = 0.2$ with 960×240 grid points. The density with TVB parameter $M = 0.01$ is plotted in Figure 11 and the density with $M = 200$ is plotted in Figure 12. In all the plots, we use 29 contours equally distributed from $\rho = 1.3$ to 23. Figures 13 and 14 show the blown-up portion around the double Mach region. We can see that the resolution around the double Mach region is high. Also, the resolution is slightly better as M increases from $M = 0.01$ to $M = 200$, however, this difference is not significant.

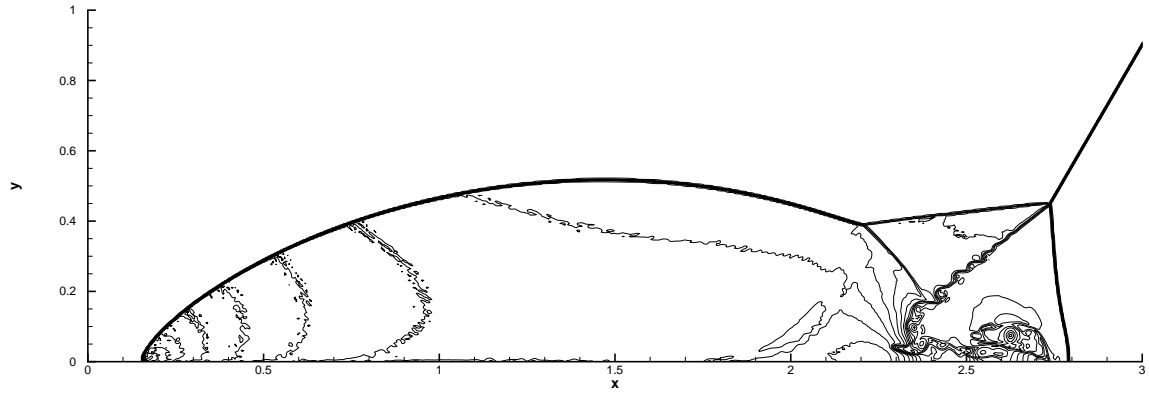


Figure 11: Double Mach reflection problem. 960×240 cells. Twenty-nine equally spaced density contours from 1.3 to 23. $M = 0.01$.

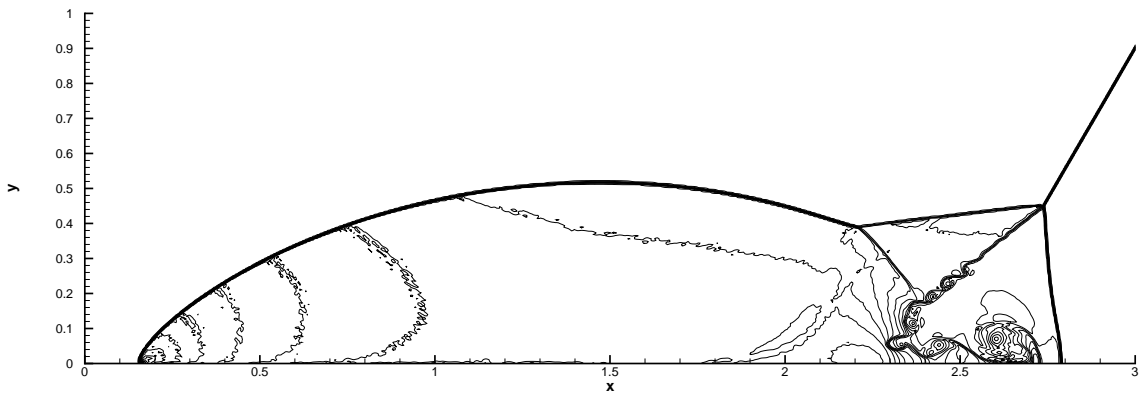


Figure 12: Double Mach reflection problem. 960×240 cells. Twenty-nine equally spaced density contours from 1.3 to 23. $M = 200$.

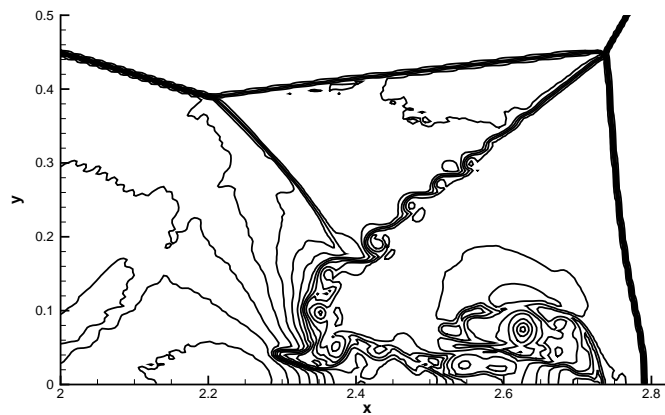


Figure 13: Zoomed-in figure. Double Mach reflection problem. 960×240 cells. Twenty-nine equally spaced density contours from 1.3 to 23. $M = 0.01$.

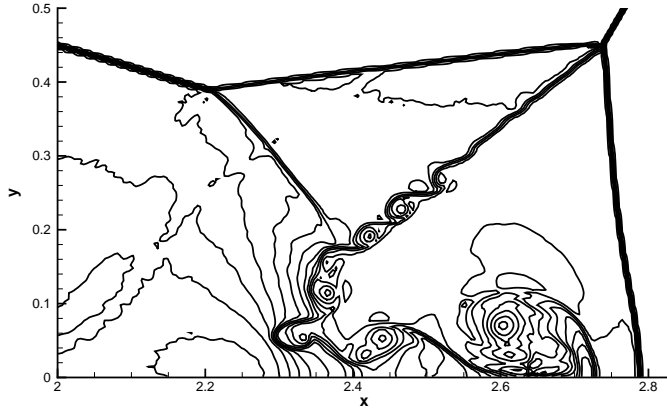


Figure 14: **Zoomed-in figure. Double Mach reflection problem. 960×240 cells. Twenty-nine equally spaced density contours from 1.3 to 23. $M = 200$**

6 Concluding remarks

In this paper, we adapt a WENO limiter [45] to the CPR framework solving hyperbolic conservation laws to make it more robust for shocked flows and uniformly high order accurate. Also, we extend the positivity-preserving limiter in [43, 44, 36] to the CPR framework. On each time level, we first use the WENO limiter to reconstruct the solution polynomials on those troubled cells, then use the positivity-preserving limiter to modify the solution polynomials in each cell if necessary. Finally, we update the numerical values at the solution points, and perform the normal CPR procedure to march to the next time level. Since this WENO limiter uses information only from immediate neighbors, it is very simple to implement and can maintain the compactness of the CPR framework. Also, we only perform this WENO limiter on the solution polynomials which can be discontinuous among adjacent cells, thus the conservativeness of the CPR framework will not be harmed. Numerical results in one and two dimensions are provided to show that this WENO limiting procedure can simultaneously maintain uniform high order accuracy of the CPR framework in smooth regions and control spurious numerical oscillations near discontinuities. In future work we will extend the WENO limiter to CPR framework on unstructured meshes along the lines of [47].

References

- [1] D.S. Balsara and C.-W. Shu, *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, Journal of Computational Physics, 160 (2000), 405–452.
- [2] R. Biswas, K.D. Devine and J. Flaherty, *Parallel adaptive finite element methods for conservation laws*, Applied Numerical Mathematics, 14 (1994), 255–283.
- [3] A. Burbeau, P. Sagaut and Ch.H. Bruneau, *A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods*, Journal of Computational Physics, 169 (2001), 111–150.
- [4] M.H. Carpenter and D. Gottlieb, *Spectral methods on arbitrary grids*, Journal of Computational Physics, 129 (1996), 74–86.
- [5] B. Cockburn, S. Hou and C.-W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case*, Mathematics of Computation, 54 (1990), 545–581.
- [6] B. Cockburn, S.-Y. Lin and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems*, Journal of Computational Physics, 84 (1989), 90–113.
- [7] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework*, Mathematics of Computation, 52 (1989), 411–435.
- [8] B. Cockburn and C.-W. Shu, *The Runge-Kutta local projection P1-discontinuous Galerkin method for scalar conservation laws*, Mathematical Modelling and Numerical Analysis, 25 (1991), 337–361.

- [9] B. Cockburn and C.-W. Shu, *The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems*, Journal of Computational Physics, 141 (1998), 199–224.
- [10] H. Gao and Z.J. Wang, *A high-order lifting collocation penalty formulation for the Navier-Stokes equations on 2D mixed grids*, AIAA Paper 2009–3784.
- [11] D. Gottlieb and S.A. Orszag, *Numerical analysis of spectral method: Theory and Applications*, Society for Industrial and Applied Mathematics, 26 (1977).
- [12] T. Haga, H. Gao and Z.J. Wang, *A high-order unifying discontinuous formulation for the Navier-Stokes equations on 3D mixed grids*. Mathematical Modelling of Natural Phenomena, 6 (2011), 28–56.
- [13] J.S. Hesthaven and D. Gottlieb, *Stable spectral methods for conservation laws on triangles with unstructured grids*, Computer Methods in Applied Mechanics and Engineering, 175 (1999), 361–381.
- [14] H.T. Huynh, *A flux reconstruction approach to high-order schemes including discontinuous Galerkin methods*, AIAA Paper 2007–4079.
- [15] H.T. Huynh, *A reconstruction approach to high-order schemes including discontinuous Galerkin for diffusion*, AIAA Paper 2009–403.
- [16] G. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, Journal of Computational Physics, 126 (1995), 202–228.
- [17] Y. Jiang, C.-W. Shu and M. Zhang, *An alternative formulation of finite difference weighted ENO schemes with Lax-Wendroff time discretization for conservation laws*, SIAM Journal on Scientific Computing, 35 (2013), A1137–A1160.
- [18] G.E. Karniadakis and S.J. Sherwin, *Spectral-hp Element Methods*, Oxford University Press, 1999.

- [19] D.A. Kopriva, *A conservative staggered-grid Chebyshev multidomain method for compressible flows. II: A semi-structured method*, Journal of Computational Physics, 128 (1996), 475–488.
- [20] D.A. Kopriva, *A staggered-grid multidomain spectral method for the compressible Navier Stokes equations*, Journal of Computational Physics, 143 (1998), 125–158.
- [21] D.A. Kopriva and J.H. Kolas, *A conservative staggered-grid Chebyshev multidomain method for compressible flows*, Journal of Computational Physics, 125 (1996), 244–261.
- [22] X.-D. Liu and S. Osher, *Non-oscillatory high order accurate self similar maximum principle satisfying shock capturing schemes*, SIAM Journal on Numerical Analysis, 33 (1996), 760–779.
- [23] X.-D. Liu, S. Osher and T. Chan, *Weighted essentially non-oscillatory schemes*, Journal of Computational Physics, 115 (1994), 200–212.
- [24] Y. Liu, M. Vinokur and Z.J. Wang, *Spectral difference method for unstructured grids I: Basic formulation*, Journal of Computational Physics, 216 (2006), 780–801.
- [25] G. May and A. Jameson, *A spectral difference method for the Euler and Navier-Stokes equations on unstructured meshes*, AIAA Paper 2006–304.
- [26] J. Qiu and C.-W. Shu, *Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: one-dimensional case*, Journal of Computational Physics, 193 (2003), 115–135.
- [27] J. Qiu and C.-W. Shu, *Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two dimensional case*, Computers and Fluids, 34 (2005), 642–663.

- [28] J. Qiu and C.-W. Shu, *Runge-Kutta discontinuous Galerkin method using WENO limiters*, SIAM Journal on Scientific Computing, 26 (2005), 907–929.
- [29] J. Qiu and C.-W. Shu, *A comparison of troubled-cell indicators for Runge-Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters*, SIAM Journal on Scientific Computing, 27 (2005), 995–1013.
- [30] W.H. Reed, T.R. Hill, *Triangular mesh methods for the neutron transport equation*, Technical Report LA-UR-73-479, 1973, Los Alamos Scientific Laboratory, Los Alamos.
- [31] J. Shi, C. Hu and C.-W. Shu, *A technique of treating negative weights in WENO schemes*, Journal of Computational Physics, 175 (2002), 108–127.
- [32] C.-W. Shu, *Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws*, in *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), Lecture Notes in Mathematics, volume 1697, Springer, Berlin, 1998, pp.325–432.
- [33] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, Journal of Computational Physics, 77 (1988), 439–471.
- [34] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes, II*, Journal of Computational Physics, 83 (1989), 32–78.
- [35] K. Van den Abeele, C. Lacor and Z.J. Wang, *On the connection between the spectral volume and the spectral difference method*, Journal of Computational Physics, 227 (2007), 877–885.
- [36] C. Wang, X. Zhang, C.-W. Shu and J. Ning, *Robust high order discontinuous Galerkin schemes for two-dimensional gaseous detonations*, Journal of Computational Physics, 231 (2012), 653–665.

- [37] Z.J. Wang, *Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation*, Journal of Computational Physics, 178 (2002), 210–251.
- [38] Z.J. Wang and H. Gao, *A unifying lifting collocation penalty formulation including the discontinuous Galerkin, spectral volume/difference methods for conservation laws on mixed grids*, Journal of Computational Physics, 228 (2009), 8161–8186.
- [39] Z.J. Wang and Y. Liu, *Spectral (finite) volume method for conservation laws on unstructured grids II: extension to two-dimensional scalar equation*, Journal of Computational Physics, 179 (2002), 665–697.
- [40] Z.J. Wang and Y. Liu, *Spectral (finite) volume method for conservation laws on unstructured grids III: one-dimensional systems and partition optimization*, Journal of Scientific Computing, 20 (2004), 137–157.
- [41] Z.J. Wang, Y. Liu, G. May and A. Jameson, *Spectral Difference Method for Unstructured Grids II: Extension to the Euler Equations*, Journal of Scientific Computing, 32 (2007), 45–71.
- [42] Z.J. Wang, L. Zhang and Y. Liu, *Spectral (finite) volume method for conservation laws on unstructured grids IV: extension to two-dimensional Euler equations*, Journal of Computational Physics, 194 (2004), 716–741.
- [43] X. Zhang and C.-W. Shu, *On maximum-principle-satisfying high order schemes for scalar conservation laws*, Journal of Computational Physics, 229 (2010), 3091–3120.
- [44] X. Zhang and C.-W. Shu, *On positivity preserving high order discontinuous Galerkin schemes for compressible Euler equations on rectangular meshes*, Journal of Computational Physics, 229 (2010), 8918–8934.
- [45] X. Zhong, C.-W. Shu, *A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods*, Journal of Computational Physics, 232 (2013), 397–415.

- [46] J. Zhu, J. Qiu, C.-W. Shu and M. Dumbser, *Runge-Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes*, Journal of Computational Physics, 227 (2008), 4330–4353.
- [47] J. Zhu, X. Zhong, C.-W. Shu and J.-X. Qiu, *Runge-Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes*, Journal of Computational Physics, 248 (2013), 200–220.