

# A simple, high-order and compact WENO limiter for RKDG method<sup>☆</sup>

Hongqiang Zhu<sup>a</sup>, Jianxian Qiu<sup>b</sup>, Jun Zhu<sup>c,\*</sup>

<sup>a</sup> School of Natural Sciences, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210023, PR China

<sup>b</sup> School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, PR China

<sup>c</sup> College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, PR China



## ARTICLE INFO

### Article history:

Received 10 January 2018

Received in revised form 5 December 2018

Accepted 24 June 2019

Available online 18 July 2019

### Keywords:

Limiter

Runge–Kutta discontinuous Galerkin

Weighted essentially non-oscillatory

Conservation laws

## ABSTRACT

In this paper, a new limiter using weighted essentially non-oscillatory (WENO) methodology is investigated for the Runge–Kutta discontinuous Galerkin (RKDG) methods for solving hyperbolic conservation laws. The idea is to use the high-order DG solution polynomial itself in the target cell and the linear polynomials which are reconstructed by the cell averages of solution in the target cell and its neighboring cells to reconstruct a new high-order polynomial in a manner of WENO methodology. Since only the linear polynomials need to be prepared for reconstruction, this limiter is very simple and compact with a stencil including only the target cell and its immediate neighboring cells. Numerical examples of various problems show that the new limiting procedure can simultaneously achieve uniform high-order accuracy and sharp, non-oscillatory shock transitions.

© 2019 Elsevier Ltd. All rights reserved.

## 1. Introduction

In this paper, we consider the hyperbolic conservation laws

$$u_t + \nabla \cdot f(u) = 0, \quad (1.1a)$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}) \quad (1.1b)$$

where  $u$  and  $f(u)$  can be either scalars or vectors. For both one-dimensional ( $\mathbf{x} = x$ ) and two-dimensional ( $\mathbf{x} = (x, y)$ ) cases, we investigate a new limiter based on weighted essentially non-oscillatory (WENO) methodology for the Runge–Kutta discontinuous Galerkin (RKDG) methods, with the goal of obtaining a simple, robust, high-order and compact limiting procedure to simultaneously maintain uniform high-order accuracy in smooth regions and control spurious numerical oscillations near discontinuities.

Discontinuous Galerkin (DG) methods are a class of finite element methods using completely discontinuous piecewise polynomial space for the numerical solutions and the test functions. The first DG method was originally introduced in 1973 by Reed and Hill [1] for the neutron transport problem. A major development of this method was carried out by Cockburn

<sup>☆</sup> The research of H. Zhu is partially supported by NSFC grant 11871443 and the Natural Science Foundation for Colleges and Universities in Jiangsu Province, PR China (17KJB110013). The research of J. Zhu is partly supported by NSFC grant 11872210. The research of J. Qiu is partially supported by NSAF grant U1630247 and NSFC grant 11571290.

\* Corresponding author.

E-mail addresses: [zhuhq@njupt.edu.cn](mailto:zhuhq@njupt.edu.cn) (H. Zhu), [jxqiu@xmu.edu.cn](mailto:jxqiu@xmu.edu.cn) (J. Qiu), [zhujun@nuaa.edu.cn](mailto:zhujun@nuaa.edu.cn) (J. Zhu).

et al. in a series of papers [2–6], in which a framework to solve nonlinear time-dependent hyperbolic conservation laws was established. They adopted explicit, nonlinearly stable high-order Runge–Kutta time discretizations [7], DG space discretizations with exact or approximate Riemann solvers as interface fluxes and TVB (total variation bounded) nonlinear limiter [8] to achieve non-oscillatory properties. This method has the advantage of flexibility in handling complicated geometry,  $h$ - $p$  adaptivity, and efficiency of parallel implementation.

Solutions of nonlinear hyperbolic conservation laws usually develop discontinuities even though the initial conditions are smooth enough, which leads to great difficulty in numerical simulation. An important component of the RKDG method for solving conservation laws with strong shocks in the solution is a nonlinear limiter, which is applied to detect discontinuities and control spurious oscillations near the discontinuities. Many such limiters have been developed for the RKDG method. Cockburn et al. developed the minmod-type TVB limiter [2–6], which is a slope limiter using a technique borrowed from the finite volume methodology. Biswas et al. proposed a moment limiter [9] which works on the moments of the numerical solution. This moment limiter was later improved by Burbeau et al. [10]. Although these limiters can control spurious numerical oscillations near discontinuities, they tend to degrade accuracy when mistakenly used in smooth regions of the solution. It is usually difficult to design limiters to achieve both high-order accuracy and a non-oscillatory property near discontinuities. Such an attempt has been made in [11–13] using WENO methodology [14–17] as the limiter for the RKDG method. However, this limiter uses a reconstruction stencil of same width as the WENO methodology which includes not only the immediate neighboring cells but also neighbors' neighbors, making it complicated to implement in multi-dimensions, especially for unstructured meshes [12,13]. To reduce the spread of reconstruction stencils, Qiu and Shu [18,19] and Zhu and Qiu [13,20] constructed Hermite type WENO approximations on both structured and unstructured meshes, which use the information of not only the cell averages but also the lower order moments such as slopes. Unfortunately, information of neighbors' neighbors is still needed for higher-order methods. Later Zhong and Shu [21] proposed a simple WENO limiter with a very compact stencil including only the target cell and its immediate neighbors. Another advantage of their limiter is the simplicity in implementation, even for unstructured meshes [22]. This limiter was further modified in [23,24] to improve the robustness in the computation of problems with strong shocks or contact discontinuities, at the cost of adding a least-square process [25]. In addition to the aforementioned references, we would also like to bring the reader's attention to the recent interesting work [26–29] on limiters. Among them, it is worth mentioning the work of Dumbser et al. [26] where an alternative family of DG limiters was proposed. They introduced a novel *a posteriori* subcell limiter which involves recomputing the discrete solution from the old time level using a different and more robust numerical scheme on a subgrid level.

The current work determines on further improvement of existing WENO type limiters. We have designed a new WENO limiter on uniform meshes that still maintains the advantages of high-order accuracy, high compactness and simplicity in implementation as the simple WENO limiter [21,22], and is even easier and more efficient in the computation. WENO type limiters are closely connected with the development of WENO methods. This work is mainly attributed to a newly introduced WENO method by Zhu and Qiu [30–33] and Dumbser et al. [34], where the reconstruction is the combination of a high-order polynomial and linear polynomials. The ideas are also similar to the WENO works including [35–38] in which the reconstruction is the combination of a  $2r$  degree polynomial and  $r$  degree polynomials. In Zhu and Qiu's work [30–33], their WENO method uses linear constructed polynomials and only one high-order constructed polynomial to reconstruct point values so that it is more efficient and easier than classical WENO methods, yet it still maintains high-order accuracy. We will apply this method to the DG methods to design a new WENO limiter, expecting these good features.

This paper is organized as follows. We first review the RKDG method in Section 2 and then in Section 3 we present our new WENO limiter. Numerical results are reported in Section 4 and concluding remarks are followed in Section 5.

## 2. Review of the RKDG methods

In this section, we give an overview of the algorithm formulation of the RKDG method for solving hyperbolic conservation laws (1.1). For simplicity, we do that for  $u$  being a scalar. If  $u$  is vector-valued, one simply proceeds in the component by component way. For the detailed description of the RKDG method, the readers are referred to the review paper [39].

**One-dimensional DG space discretization.** To define the DG scheme for (1.1) in one dimension, the computational domain  $[a, b]$  is discretized with  $N$  cells as follows

$$a = x_{\frac{1}{2}} < x_{\frac{3}{2}} < \dots < x_{N+\frac{1}{2}} = b. \tag{2.1}$$

Denote the center of cell  $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$  by  $x_i$ , the length of cell  $I_i$  by  $\Delta x_i$ , and the maximum cell length by  $h = \max_i \Delta x_i$ .

The solution as well as the test function space is defined as  $V_h^k = \{v : v|_{I_i} \in P^k(I_i), 1 \leq i \leq N\}$ , where  $P^k(I_i)$  is the space of polynomials of degree up to  $k$  defined on the cell  $I_i$ . (In the computation of this paper  $k$  is at most three.) The semi-discrete DG scheme for solving (1.1) is defined as follows: find the unique function  $u = u(t) \in V_h^k$  such that

$$\int_{I_i} u_t v dx - \int_{I_i} f(u) v_x dx + \hat{f}_{i+\frac{1}{2}} v(x_{i+\frac{1}{2}}^-) - \hat{f}_{i-\frac{1}{2}} v(x_{i-\frac{1}{2}}^+) = 0, \quad i = 1, \dots, N \tag{2.2}$$

holds for all test functions  $v \in V_h^k$ . Here, the superscripts '±' indicate the left and right limits of the corresponding functions with respect to  $x$ . The integral terms are computed either exactly or by a numerical quadrature with sufficiently

**Table 2.1**  
Parameters of Runge–Kutta schemes (2.6).

DG	Order	$\alpha_{il}$				$\beta_{il}$			
$p^2$	3	1				1			
		$\frac{3}{4}$	$\frac{1}{4}$			0	$\frac{1}{4}$		
		$\frac{1}{3}$	0	$\frac{2}{3}$		0	0	$\frac{2}{3}$	
$p^3$	4	1				$\frac{1}{2}$			
		1	0			0	$\frac{1}{2}$		
		1	0	0		0	0	1	
		$-\frac{1}{3}$	$\frac{1}{3}$	$\frac{2}{3}$	$\frac{1}{3}$	0	0	0	$\frac{1}{6}$

high order of accuracy.  $\hat{f}_{i+\frac{1}{2}} = \hat{f}(u_{i+\frac{1}{2}}^-, u_{i+\frac{1}{2}}^+)$  is the so-called monotone numerical fluxes. In this paper we use the global Lax–Friedrichs flux

$$\hat{f}(u^-, u^+) = \frac{1}{2}[f(u^-) + f(u^+) - \alpha(u^+ - u^-)] \tag{2.3}$$

where  $\alpha = \max_u |f'(u)|$  with the maximum taken over the whole range of  $u$ .

**Two-dimensional DG space discretization.** Since rectangular cells are used in this work, it is required that the computational domain  $\Omega$  can be covered by a rectangular mesh without hanging nodes. For simplicity of notations and formulation of the scheme, we assume for this part that  $\Omega = [a_x, b_x] \times [a_y, b_y]$  is rectangular. Under this assumption, the mesh consists of cells

$$I_{i,j} = \{(x, y) : x_{i-\frac{1}{2}} \leq x \leq x_{i+\frac{1}{2}}, y_{j-\frac{1}{2}} \leq y \leq y_{j+\frac{1}{2}}\}, \quad i = 1, \dots, N_x, j = 1, \dots, N_y. \tag{2.4}$$

As before, we define a finite element space consisting of piecewise polynomials  $W_h^k = \{v : v|_{I_{i,j}} \in P^k(I_{i,j}), 1 \leq i \leq N_x, 1 \leq j \leq N_y\}$ , where  $P^k(I_{i,j})$  is the space of polynomials of degree up to  $k$  defined on the cell  $I_{i,j}$ . The semi-discrete DG scheme for solving (1.1) in two dimensions is defined as follows: find the unique function  $u = u(t) \in W_h^k$  such that, for all test functions  $v \in W_h^k$  and  $i = 1, \dots, N_x, j = 1, \dots, N_y$ ,

$$\begin{aligned} & \int_{I_{i,j}} u_t v dx dy - \int_{I_{i,j}} f(u) v_x dx dy + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{f}_{i+\frac{1}{2}}(y) v(x_{i+\frac{1}{2}}^-, y) dy - \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} \hat{f}_{i-\frac{1}{2}}(y) v(x_{i-\frac{1}{2}}^+, y) dy \\ & - \int_{I_{i,j}} g(u) v_y dx dy + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{g}_{j+\frac{1}{2}}(x) v(x, y_{j+\frac{1}{2}}^-) dx - \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} \hat{g}_{j-\frac{1}{2}}(x) v(x, y_{j-\frac{1}{2}}^+) dx = 0. \end{aligned} \tag{2.5}$$

Again, the integral terms are computed either exactly or by a numerical quadrature. The numerical fluxes  $\hat{f}$  and  $\hat{g}$  are computed by the global Lax–Friedrichs flux (2.3). The readers are referred to [5,6] for more details.

**Time discretization.** The semi-discrete schemes (2.2) and (2.5) are systems of ordinary differential equations, and can be written as  $\mathbf{u}_t = L(\mathbf{u})$  where  $L(\cdot)$  is the spatial discretization operator. To discretize the time variable, we apply the  $(k + 1)$ th order Runge–Kutta method [7,40] for  $P^k(2 \leq k \leq 3)$  DG discretization

$$\mathbf{u}^{(i)} = \sum_{l=0}^{i-1} [\alpha_{il} \mathbf{u}^{(l)} + \beta_{il} \Delta t L(\mathbf{u}^{(l)})], \quad i = 1, \dots, k + 1, \tag{2.6a}$$

$$\mathbf{u}^{(0)} = \mathbf{u}^n, \quad \mathbf{u}^{n+1} = \mathbf{u}^{(k+1)} \tag{2.6b}$$

with  $\mathbf{u}^n$  corresponding to the solution at time level  $t_n$ . The parameters in (2.6) are listed in Table 2.1.

### 3. A new WENO limiter

In this section, we introduce our new WENO limiter in detail for the RKDG methods. We adopt the same framework of limiters as in [11], that is, we first identify the “troubled cells”, namely those cells which might need the limiting procedure; then we replace the solution polynomials in those troubled cells by reconstructed polynomials which should maintain the original cell averages (conservation) and order of accuracy, but should be less oscillatory. The first step, how to identify troubled cells by the troubled-cell indicator, is a very important issue for a limiter. However, the focus of this paper is on the new way to reconstruct the polynomial in a troubled cell (the second step). We have no intention of comparing the pros and cons of various troubled-cell indicators. We will simply use the shock detection technique developed in [41], termed as the KXRCF troubled-cell indicator, to identify the troubled cells. This is one of the recommended troubled-cell indicators in the work of comparison among different troubled-cell indicators [42]. Of course the other troubled-cell indicators can also be used.

### 3.1. Identification of troubled cells

The KXRCF troubled-cell indicator [41] works in the following way. We take the two-dimensional case as an example to describe the procedure. The procedure for the one-dimensional case can be obtained analogously. Firstly, the boundary of the target cell  $I_{i,j}$  is partitioned into two portions  $\partial I_{i,j}^-$  and  $\partial I_{i,j}^+$ , where the flow is into ( $\vec{v} \cdot \vec{n} < 0$ ,  $\vec{n}$  is the outward normal vector to  $\partial I_{i,j}$ ) and out of ( $\vec{v} \cdot \vec{n} > 0$ ) cell  $I_{i,j}$ , respectively. For the scalar case, we define  $\vec{v} = (f'(u), g'(u))$  with  $u$  being the solution. For the Euler system (4.3),  $\vec{v}$  is the vector of velocity. We compute  $\vec{v}$  based on cell averages. Then cell  $I_{i,j}$  is believed to contain a discontinuity and identified as a troubled cell if

$$\frac{\left| \int_{\partial I_{i,j}^-} (u^h|_{I_{i,j}} - u^h|_{I_i}) ds \right|}{h_{i,j}^{\frac{k+1}{2}} |\partial I_{i,j}^-| \|u^h|_{I_{i,j}}\|} > C_k. \tag{3.1}$$

Here,  $C_k$  is usually taken to be one, but it is also adjustable so that the number of troubled cells can be increased or decreased accordingly.  $h_{i,j}$  is the radius of the circumscribed circle in cell  $I_{i,j}$ .  $I_i$  denotes the neighboring cell of  $I_{i,j}$  on the side of  $\partial I_{i,j}^-$ . The norm  $\|u^h|_{I_{i,j}}\|$  is based on the maximum norm taken from the interior of  $I_{i,j}$ . The indicator variable  $u$  is the scalar solution itself for the scalar case and density and energy solution components for the Euler system (4.3).

### 3.2. WENO reconstruction of the new polynomials

In this subsection, we give the detailed procedure of reconstructing the new polynomials in the troubled cells using our new WENO limiter. The idea of this WENO limiter is to reconstruct a new polynomial in the target cell which is a combination of the polynomial of the target cell and the linear polynomials constructed from several small stencils using cell averages of the target cell and its immediate neighbors.

**One-dimensional scalar case.** Let  $I_i$  be the target cell which means that it is a troubled cell. Denote the DG solution on this cell by  $p_0(x)$ , which is a  $k$ th-degree polynomial. Choose two small stencils  $T_1 = \{I_{i-1}, I_i\}$  and  $T_2 = \{I_i, I_{i+1}\}$ . Their union forms the big stencil of the WENO limiter which consists of exactly the target cell and its immediate neighbors. Denote the cell average of the solution  $u$  in cell  $I_r$  as  $\bar{u}_r = \frac{1}{\Delta x_r} \int_{I_r} u dx$ . It is very easy to get the linear polynomial  $p_1(x)$  from stencil  $T_1$  which satisfies

$$\frac{1}{\Delta x_r} \int_{I_r} p_1(x) dx = \bar{u}_r, \quad r = i - 1, i. \tag{3.2}$$

Similarly, we can get the linear polynomial  $p_2(x)$  from stencil  $T_2$  which satisfies

$$\frac{1}{\Delta x_r} \int_{I_r} p_2(x) dx = \bar{u}_r, \quad r = i, i + 1. \tag{3.3}$$

Similar to the ideas in [30,31,34–38], for a given positive linear weights  $\gamma_0, \gamma_1, \gamma_2$  with their sum being unity, we rewrite  $p_0(x)$  as

$$p_0(x) = \gamma_0 \left( \frac{1}{\gamma_0} p_0(x) - \frac{\gamma_1}{\gamma_0} p_1(x) - \frac{\gamma_2}{\gamma_0} p_2(x) \right) + \gamma_1 p_1(x) + \gamma_2 p_2(x), \tag{3.4}$$

and introduce the modified polynomials

$$\tilde{p}_0(x) = \frac{1}{\gamma_0} p_0(x) - \frac{\gamma_1}{\gamma_0} p_1(x) - \frac{\gamma_2}{\gamma_0} p_2(x), \tag{3.5}$$

$$\tilde{p}_1(x) = p_1(x), \tag{3.6}$$

$$\tilde{p}_2(x) = p_2(x). \tag{3.7}$$

Then compute the smoothness indicators  $\beta_l, l = 0, 1, 2$ , which measure how smooth the functions  $\tilde{p}_l(x), l = 0, 1, 2$  are in the target cell. We use the same recipe for the smoothness indicators as in [15], that is,

$$\beta_l = \sum_{q=1}^k \int_{I_i} (\Delta x_i)^{2q-1} \left( \frac{\partial^q \tilde{p}_l(x)}{\partial x^q} \right)^2 dx, \quad l = 0, 1, 2. \tag{3.8}$$

Let

$$\tau = \left( \frac{|\beta_0 - \beta_1| + |\beta_0 - \beta_2|}{2} \right)^2 \tag{3.9}$$

and define the following nonlinear weights

$$\omega_l = \frac{\tilde{\omega}_l}{\sum_q \tilde{\omega}_q}, \quad \tilde{\omega}_l = \gamma_l \left( 1 + \frac{\tau}{\epsilon + \beta_l} \right), \quad l = 0, 1, 2. \tag{3.10}$$

Here  $\epsilon$  is a small positive number to avoid the denominator to become zero. Finally, the new reconstructed polynomial in the target cell is given by

$$p_0^{\text{new}}(x) = \omega_0 \tilde{p}_0(x) + \omega_1 \tilde{p}_1(x) + \omega_2 \tilde{p}_2(x). \tag{3.11}$$

**One-dimensional system case.** Consider Eq. (1.1) with  $u$  and  $f$  being vectors with  $m$  components. In order to achieve better non-oscillatory qualities, our WENO reconstruction is performed with a local characteristic field decomposition. Suppose that the Jacobian matrix is denoted by  $A_i = \frac{\partial f}{\partial u} |_{\bar{u}_i}$  and its left and right eigenvectors are denoted by  $l_i^{(p)}, r_i^{(p)}, p = 1, \dots, m$ . They are normalized so that  $l_i^{(p)} \cdot r_i^{(q)} = \delta_{pq}$ . Let  $R(\bar{u}_i)$  be the  $m \times m$  matrix with the right eigenvectors as columns, i.e.

$$R(\bar{u}_i) = \left( r_i^{(1)}, r_i^{(2)}, \dots, r_i^{(m)} \right). \tag{3.12}$$

Hence  $R^{-1}(\bar{u}_i)$  is a  $m \times m$  matrix with the left eigenvectors as rows, that is

$$R^{-1}(\bar{u}_i) = \left( l_i^{(1)}, l_i^{(2)}, \dots, l_i^{(m)} \right)^T. \tag{3.13}$$

The procedure of our WENO reconstruction for one-dimensional system case is presented below. We use the same notations as in the one-dimensional scalar case but all the variables are now vectors except the linear weights  $\gamma_0, \gamma_1, \gamma_2$ .

1. For  $l = 0, 1, 2$ , find  $\tilde{p}_l(x)$  component by component in the same way as the one-dimensional scalar case.
2. Project  $\tilde{p}_l(x)$  into the characteristic fields  $\mu_l(x) = R^{-1} \tilde{p}_l(x)$  for  $l = 0, 1, 2$ .
3. For each component of  $\mu_l(x), l = 0, 1, 2$ , compute smoothness indicators via (3.8), nonlinear weights via (3.10) and the reconstructed polynomial via (3.11). The resulting updated vector of  $\mu_0(x)$  is denoted by  $\mu_0^{\text{new}}(x)$ .
4. The final reconstructed polynomial of  $p_0(x)$  is  $p_0^{\text{new}}(x) = R \mu_0^{\text{new}}(x)$ .

**Remark 3.1.** Another equivalent procedure is that we can first project all the involved data (DG solution polynomial in the target cell and cell averages in its immediate neighboring cells) into the characteristic fields, and then perform the whole WENO reconstruction procedure in the one-dimensional scalar case componentwisely, and finally project back (Step 4) to obtain the new polynomial of WENO reconstruction.

**Two-dimensional scalar case on rectangular meshes.** Suppose that  $I_{i,j}$  is the target cell in which the DG solution is a  $k$ th degree polynomial  $p_0(x, y)$ . Four small stencils,

$$T_1 = \{I_{i,j}, I_{i,j-1}, I_{i-1,j}\}, \quad T_2 = \{I_{i,j}, I_{i,j-1}, I_{i+1,j}\}, \tag{3.14}$$

$$T_3 = \{I_{i,j}, I_{i,j+1}, I_{i-1,j}\}, \quad T_4 = \{I_{i,j}, I_{i,j+1}, I_{i+1,j}\}, \tag{3.15}$$

are carefully chosen so that we can define the linear polynomials  $p_l(x, y), l = 1, 2, 3, 4$  for reconstruction. Similar choices of such stencils and polynomial degrees can be found in [32,34,43].  $p_l(x, y)$  is defined on stencil  $T_l$  such that

$$\frac{1}{\Delta x_r \Delta y_s} \int_{I_{r,s}} p_l(x, y) dx dy = \bar{u}_{r,s} \tag{3.16}$$

where  $(r, s)$  goes over the indexes of all cells in stencil  $T_l$ , and  $\bar{u}_{r,s} = \frac{1}{\Delta x_r \Delta y_s} \int_{I_{r,s}} u dx dy$  is again the cell average of solution  $u$  in cell  $I_{r,s}$ . Similar to the procedure of the one-dimensional scalar case, for a given positive linear weights  $\gamma_l, l = 0, \dots, 4$  with their sum being unity, we rewrite  $p_0(x, y)$  as

$$p_0(x, y) = \gamma_0 \left( \frac{1}{\gamma_0} p_0(x, y) - \sum_{l=1}^4 \frac{\gamma_l}{\gamma_0} p_l(x, y) \right) + \sum_{l=1}^4 \gamma_l p_l(x, y), \tag{3.17}$$

and introduce the modified polynomials

$$\tilde{p}_0(x, y) = \frac{1}{\gamma_0} p_0(x, y) - \sum_{l=1}^4 \frac{\gamma_l}{\gamma_0} p_l(x, y), \tag{3.18}$$

$$\tilde{p}_l(x, y) = p_l(x, y), \quad l = 1, 2, 3, 4. \tag{3.19}$$

Then compute the smoothness indicators

$$\beta_l = \sum_{q_1+q_2=1}^k \int_{I_{i,j}} (\Delta x_i)^{2q_1-1} (\Delta y_j)^{2q_2-1} \left( \frac{\partial^{q_1+q_2} \tilde{p}_l(x, y)}{\partial x^{q_1} \partial y^{q_2}} \right)^2 dx dy, \quad l = 0, \dots, 4. \tag{3.20}$$

Define

$$\tau = \left( \frac{\sum_{l=1}^4 |\beta_0 - \beta_l|}{4} \right)^2 \tag{3.21}$$

**Table 3.1**  
Settings of linear weights.

Settings	One-dimensional case	Two-dimensional case
LW-1	$\gamma_0 = 0.98, \gamma_1 = \gamma_2 = 0.01$	$\gamma_0 = 0.96, \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.01$
LW-2	$\gamma_0 = 0.8, \gamma_1 = \gamma_2 = 0.1$	$\gamma_0 = 0.8, \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.05$
LW-3	$\gamma_0 = 0.5, \gamma_1 = \gamma_2 = 0.25$	$\gamma_0 = \frac{1}{3}, \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = \frac{1}{6}$
LW-4	$\gamma_0 = 0.02, \gamma_1 = \gamma_2 = 0.49$	$\gamma_0 = 0.04, \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.24$

and compute the nonlinear weights

$$\omega_l = \frac{\bar{\omega}_l}{\sum_q \bar{\omega}_q}, \quad \bar{\omega}_l = \gamma_l \left( 1 + \frac{\tau}{\epsilon + \beta_l} \right), \quad l = 0, \dots, 4. \tag{3.22}$$

The new reconstructed polynomial in the target cell is given by

$$p_0^{\text{new}}(x, y) = \sum_{l=0}^4 \omega_l \tilde{p}_l(x, y). \tag{3.23}$$

**Two-dimensional system case on rectangular meshes.** We consider the two-dimensional system (1.1) in which  $u, f(u)$  and  $g(u)$  are  $m$ -component vectors. Since there are two Jacobian matrices corresponding to fluxes in the  $x$  and  $y$  directions respectively and therefore two sets of eigenspaces, we need to be careful when using the characteristic-wise WENO reconstruction procedure. The recipe proposed in [21,23] is used. Suppose that  $I_{i,j}$  is a troubled cell in which the DG solution polynomial is  $p_0(x, y)$ ,

1. In the  $x$ -direction, based on the Jacobian matrix  $\frac{\partial f}{\partial u}$ , the involved data is first projected into the characteristic field. Then the two-dimensional scalar WENO reconstruction procedure is performed component by component. A new polynomial  $p_0^{x,\text{new}}(x, y)$  is finally obtained by projecting back the resulting reconstructed polynomial.
2. Similarly, in the  $y$ -direction, a new polynomial  $p_0^{y,\text{new}}(x, y)$  can be obtained separately based on the Jacobian matrix  $\frac{\partial g}{\partial u}$ .
3. The final reconstructed polynomial is  $p_0^{\text{new}}(x, y) = (p_0^{x,\text{new}}(x, y) + p_0^{y,\text{new}}(x, y))/2$ .

For this WENO limiter procedure, the DG solution polynomial in the target cell, i.e.  $p_0$ , is the key component to obtain a high-order reconstructed polynomial. However, if  $p_0$  itself is a linear polynomial (the case of  $k = 1$ ), there is no benefit to include  $p_0$  in our WENO reconstruction procedure since the linear polynomials from the small stencils are sufficient to reconstruct a linear solution approximation. The WENO limiters in [11] and [12] are more efficient than our new WENO limiter when  $k = 1$ . Our WENO limiter is advantageous when  $k \geq 2$ . As a result, the case of  $k = 1$  is not considered.

To complete our WENO limiter, we need to discuss the specification of linear weights. For the WENO schemes in [30,31], the only requirements imposed on these weights are that they are positive and their sum equals one. As to our WENO limiter, based on our numerical tests, these requirements are sufficient when the solution is smooth. When the solution contains discontinuities and a limiter is needed to control spurious oscillations, we find that our limiter cannot control oscillations effectively if the value of  $\gamma_0$  is close enough to unity. Therefore,  $\gamma_0$  cannot be too close to unity in this discontinuous case, but such additional requirement is rather weak. Let us assume that equal values are assigned to the weights other than  $\gamma_0$ , which is reasonable and usual. According to our numerical tests, for most of the test examples in Section 4 the scheme works well even when  $\gamma_0$  reaches 0.98. The numerical results in this paper are obtained with four representative settings of linear weights listed in Table 3.1.  $\gamma_0$  is close to one in the setting of LW-1, close to zero in LW-4, 0.8 in LW-2 and twice as large as the other weights in LW-3. All of the four settings are used to run the accuracy and convergence test for smooth solutions. When the solution is continuous, the closer to one  $\gamma_0$  is, the better. When the solution contains discontinuities, it is better to assign relatively less weights to  $\gamma_0$ , so LW-2 is mainly used to generate the numerical results. More settings have been tested but the results are not included because of space limitation.

### 3.3. The entire algorithm

Let  $\mathbb{P}_{V_h^k}$  denote the operator of standard  $L^2$ -projection into the finite element space  $V_h^k$  and  $\mathbb{W}$  denote the operator of the above WENO limiter. Suppose that  $\{t_n\}_{n=0}^{N_t}$  is a temporal discretization with  $\Delta t^n = t^{n+1} - t^n$ . Introducing the WENO limiter to the time-marching scheme (2.6), the resulting entire algorithm can be described as follows:

- Set  $\mathbf{u}^0 = \mathbb{W}\{\mathbb{P}_{V_h^k}(u_0(\mathbf{x}))\}$ ;
- For  $n = 0, \dots, N_t - 1$  compute  $\mathbf{u}^{n+1}$  as follows:
  1. set  $\mathbf{u}^{(0)} = \mathbf{u}^n$ ;

2. for  $i = 1, \dots, k + 1$ , compute the intermediate quantities:

$$\mathbf{u}^{(i)} = \mathbb{W} \left\{ \sum_{l=0}^{i-1} [\alpha_{il} \mathbf{u}^{(l)} + \beta_{il} \Delta t^n L(\mathbf{u}^{(l)})] \right\};$$

3. set  $\mathbf{u}^{n+1} = \mathbf{u}^{(k+1)}$ .

The determination of time step  $\Delta t$  should satisfy the so called CFL condition for stability reason. In this paper, for the one-dimensional scalar case

$$\Delta t = \frac{CFL_k \Delta x}{\max |f'(u)|}. \tag{3.24}$$

For the two dimensional scalar case

$$\Delta t = \frac{CFL_k}{\max |f'(u)|/\Delta x + \max |g'(u)|/\Delta y}. \tag{3.25}$$

For both one-dimensional and two-dimensional system cases,  $\max |f'(u)|$  and  $\max |g'(u)|$  in (3.24) and (3.25) are replaced by the maximum value of the largest eigenvalue (in absolute value) of the corresponding Jacobian matrix. We take the CFL numbers  $CFL_2 = 0.18$  and  $CFL_3 = 0.1$  in the computation according to paper [39] where a table of such numbers for a wide variety of time and space discretizations is given.

#### 4. Numerical tests

In this section, we provide a series of numerical examples to demonstrate the performance of the new WENO limiter. All the results are obtained with uniform (rectangular in two dimensions) meshes.

As to the parameter  $\epsilon$  in the nonlinear weights (3.10) and (3.22), it is known that its choice can not only avoid zero denominators but also influence the order of convergence of the method. Kolb [44] discussed the advantage of choosing this parameter in a finite volume WENO scheme on uniform meshes, and Cravero and Semplice [45] extended the results to non-uniform meshes. In this paper, we will not carry out a deep analysis of the role of  $\epsilon$ . Instead, we just take  $\epsilon = 10^{-6}$  in all test examples except that in Example 4.1, we take two more choices [43–45],  $\epsilon = \Delta x$  and  $\epsilon = \Delta x^2$ , to make a brief comparison.

##### 4.1. Accuracy and convergence results for smooth solutions

In this subsection, we take various examples with smooth solutions to test the accuracy and convergence order of the new WENO limiter. In order to see the effect of the WENO limiter on the accuracy of the RKDG method, every cell is forcibly identified as a troubled cell. As a result, the WENO reconstruction procedure is applied to every cell.

**Example 4.1.** We solve the following one-dimensional Burgers equation

$$u_t + \left( \frac{u^2}{2} \right)_x = 0, \quad 0 \leq x \leq 2 \tag{4.1}$$

with the initial condition  $u_0(x) = 0.5 + \sin(\pi x)$  and periodic boundary conditions. When  $t = 0.5/\pi$  the solution is still smooth. The computation of this example was performed using 128-bit arithmetic (quadruple precision) so that the actual errors and orders can be obtained on very fine meshes. We first list the errors and numerical orders of accuracy with a fixed value of  $\epsilon = 10^{-6}$  in Tables 4.1 and 4.2. It can be seen that the WENO limiter maintains the optimal  $(k + 1)$ th order of accuracy for all four settings of linear weights in the  $P^k$  case, but only asymptotically for small values of  $\gamma_0$ . Since for smooth solutions the DG solution polynomial in the target cell is usually the best approximation,  $\gamma_0$  should be as close to unity as possible. When  $\gamma_0$  is away from unity, we can see in the tables that there is a decrease of order on coarse meshes. When the meshes are fine enough, the order is then recovered.

These results can be improved by choosing an  $h$ -dependent  $\epsilon$  as in [43–45]. We show the  $L^\infty$  errors and orders with  $\epsilon = \Delta x$  and  $\epsilon = \Delta x^2$  in Tables 4.3 and 4.4, respectively. Comparing these results with Table 4.2, we can find that both choices,  $\epsilon = \Delta x$  and  $\epsilon = \Delta x^2$ , can decrease the errors notably on coarse meshes and in general provide a much more regular pattern of error decay. Moreover, the former choice is better than the latter one on this smooth test. Similar conclusions can be found in [44,45].

**Remark 4.1.** The degradation of the  $L^\infty$  convergence order (Tables 4.2–4.4) in the  $P^2$  case also arises in the RKDG method without a limiter and hence is not caused by our WENO limiter.

**Table 4.1**

Example 4.1 (one-dimensional Burgers equation).  $L^1$  errors and orders at  $t = 0.5/\pi$  with  $\epsilon = 10^{-6}$ .

N	LW-1		LW-2		LW-3		LW-4						
	$L^1$ error	Order	$L^1$ error	Order	$L^1$ error	Order	$L^1$ error	Order					
$p^2$	20	1.69E-03			7.76E-03			9.72E-03			1.75E-02		
	40	4.30E-05	5.30		1.54E-04	5.65		1.09E-03	3.16		4.51E-03	1.96	
	80	6.14E-06	2.81		6.22E-06	4.63		1.26E-05	6.43		9.50E-04	2.25	
	160	8.70E-07	2.82		8.70E-07	2.84		8.70E-07	3.86		8.95E-05	3.41	
	320	1.21E-07	2.85		1.21E-07	2.85		1.21E-07	2.85		2.45E-06	5.19	
	640	1.65E-08	2.87		1.65E-08	2.87		1.65E-08	2.87		1.62E-08	7.24	
	1 280	2.23E-09	2.89		2.23E-09	2.89		2.23E-09	2.89		2.23E-09	2.87	
	2 560	2.99E-10	2.90		2.99E-10	2.90		2.99E-10	2.90		2.99E-10	2.90	
	5 120	3.97E-11	2.91		3.97E-11	2.91		3.97E-11	2.91		3.97E-11	2.91	
	10 240	5.22E-12	2.93		5.22E-12	2.93		5.22E-12	2.93		5.22E-12	2.93	
	20 480	6.81E-13	2.94		6.81E-13	2.94		6.81E-13	2.94		6.81E-13	2.94	
$p^3$	20	1.30E-02			1.70E-02			1.76E-02			1.77E-02		
	40	1.40E-05	9.87		3.36E-03	2.34		4.15E-03	2.08		4.54E-03	1.96	
	80	8.55E-08	7.35		9.77E-07	11.75		2.56E-04	4.02		1.08E-03	2.07	
	160	3.44E-09	4.64		4.33E-09	7.82		2.50E-08	13.32		2.08E-04	2.37	
	320	2.14E-10	4.01		2.14E-10	4.34		2.17E-10	6.85		1.44E-05	3.85	
	640	1.34E-11	4.00		1.34E-11	4.00		1.34E-11	4.02		5.69E-09	11.31	
	1 280	8.35E-13	4.00		8.35E-13	4.00		8.35E-13	4.00		8.17E-12	9.45	
	2 560	5.22E-14	4.00		5.22E-14	4.00		5.22E-14	4.00		5.34E-14	7.26	
	5 120	3.26E-15	4.00		3.26E-15	4.00		3.26E-15	4.00		3.26E-15	4.03	
	10 240	2.04E-16	4.00		2.04E-16	4.00		2.04E-16	4.00		2.04E-16	4.00	
	20 480	1.28E-17	4.00		1.28E-17	4.00		1.28E-17	4.00		1.28E-17	4.00	

**Table 4.2**

Example 4.1 (one-dimensional Burgers equation).  $L^\infty$  errors and orders at  $t = 0.5/\pi$  with  $\epsilon = 10^{-6}$ .

N	LW-1		LW-2		LW-3		LW-4						
	$L^\infty$ error	Order	$L^\infty$ error	Order	$L^\infty$ error	Order	$L^\infty$ error	Order					
$p^2$	20	9.01E-03			3.24E-02			4.29E-02			6.99E-02		
	40	5.24E-04	4.10		1.59E-03	4.35		7.68E-03	2.48		2.17E-02	1.69	
	80	1.02E-04	2.36		1.02E-04	3.96		2.13E-04	5.17		7.52E-03	1.53	
	160	1.65E-05	2.62		1.65E-05	2.62		1.65E-05	3.69		1.42E-03	2.40	
	320	2.52E-06	2.72		2.52E-06	2.72		2.52E-06	2.72		7.34E-05	4.28	
	640	3.56E-07	2.82		3.56E-07	2.82		3.56E-07	2.82		3.56E-07	7.69	
	1 280	4.81E-08	2.89		4.81E-08	2.89		4.81E-08	2.89		4.81E-08	2.89	
	2 560	6.34E-09	2.92		6.34E-09	2.92		6.34E-09	2.92		6.34E-09	2.92	
	5 120	8.94E-10	2.83		8.94E-10	2.83		8.94E-10	2.83		8.94E-10	2.83	
	10 240	1.37E-10	2.71		1.37E-10	2.71		1.37E-10	2.71		1.37E-10	2.71	
	20 480	2.19E-11	2.64		2.19E-11	2.64		2.19E-11	2.64		2.19E-11	2.64	
$p^3$	20	6.43E-02			7.05E-02			6.80E-02			6.74E-02		
	40	1.70E-04	8.57		1.89E-02	1.90		1.95E-02	1.80		2.20E-02	1.62	
	80	1.01E-06	7.38		2.60E-05	9.51		3.88E-03	2.33		7.83E-03	1.49	
	160	5.38E-08	4.24		5.43E-08	8.90		1.03E-06	11.88		2.17E-03	1.85	
	320	3.42E-09	3.98		3.42E-09	3.99		3.43E-09	8.23		1.67E-04	3.70	
	640	2.15E-10	3.99		2.15E-10	3.99		2.15E-10	3.99		2.14E-07	9.61	
	1 280	1.35E-11	3.99		1.35E-11	3.99		1.35E-11	4.00		5.71E-10	8.55	
	2 560	8.46E-13	4.00		8.46E-13	4.00		8.46E-13	4.00		1.07E-12	9.06	
	5 120	5.30E-14	4.00		5.30E-14	4.00		5.30E-14	4.00		5.30E-14	4.33	
	10 240	3.31E-15	4.00		3.31E-15	4.00		3.31E-15	4.00		3.31E-15	4.00	
	20 480	2.07E-16	4.00		2.07E-16	4.00		2.07E-16	4.00		2.07E-16	4.00	

**Example 4.2.** We solve the following two-dimensional Burgers equation

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0, \quad -2 \leq x, y \leq 2 \tag{4.2}$$

with the initial condition  $u_0(x, y) = 0.5 + \sin(\frac{\pi(x+y)}{2})$  and periodic boundary conditions. When  $t = 0.5/\pi$  the solution is still smooth. We list the errors and numerical orders of accuracy in Tables 4.5 and 4.6. The usual double precision was used to perform the computation. We can observe that the expected order of  $k + 1$  is actually achieved for different settings of linear weights.

**Table 4.3**

**Example 4.1** (one-dimensional Burgers equation).  $L^\infty$  errors and orders at  $t = 0.5/\pi$  with  $\epsilon = \Delta x$ .

$N$	LW-1		LW-2		LW-3		LW-4		
	$L^\infty$ error	Order							
$p^2$	20	3.13E-03		3.38E-03		6.00E-03		3.27E-02	
	40	5.24E-04	2.58	5.25E-04	2.69	5.26E-04	3.51	5.83E-03	2.49
	80	1.02E-04	2.36	1.02E-04	2.37	1.02E-04	2.37	1.14E-03	2.35
	160	1.65E-05	2.62	1.65E-05	2.62	1.65E-05	2.62	9.79E-05	3.54
	320	2.52E-06	2.72	2.52E-06	2.72	2.52E-06	2.72	2.51E-06	5.28
	640	3.56E-07	2.82	3.56E-07	2.82	3.56E-07	2.82	3.56E-07	2.82
	1280	4.81E-08	2.89	4.81E-08	2.89	4.81E-08	2.89	4.81E-08	2.89
	2560	6.34E-09	2.92	6.34E-09	2.92	6.34E-09	2.92	6.34E-09	2.92
	5120	8.94E-10	2.83	8.94E-10	2.83	8.94E-10	2.83	8.94E-10	2.83
	10240	1.37E-10	2.71	1.37E-10	2.71	1.37E-10	2.71	1.37E-10	2.71
20480	2.19E-11	2.64	2.19E-11	2.64	2.19E-11	2.64	2.19E-11	2.64	
$p^3$	20	2.93E-04		1.95E-03		2.81E-02		3.71E-02	
	40	1.48E-05	4.31	1.64E-05	6.90	3.98E-05	9.46	8.48E-03	2.13
	80	8.43E-07	4.13	8.50E-07	4.27	8.64E-07	5.53	1.74E-03	2.28
	160	5.37E-08	3.97	5.37E-08	3.98	5.38E-08	4.01	4.66E-04	1.90
	320	3.42E-09	3.98	3.42E-09	3.98	3.42E-09	3.98	1.69E-06	8.11
	640	2.15E-10	3.99	2.15E-10	3.99	2.15E-10	3.99	1.18E-09	10.49
	1280	1.35E-11	3.99	1.35E-11	3.99	1.35E-11	3.99	1.35E-11	6.45
	2560	8.46E-13	4.00	8.46E-13	4.00	8.46E-13	4.00	8.46E-13	4.00
	5120	5.30E-14	4.00	5.30E-14	4.00	5.30E-14	4.00	5.30E-14	4.00
	10240	3.31E-15	4.00	3.31E-15	4.00	3.31E-15	4.00	3.31E-15	4.00
20480	2.07E-16	4.00	2.07E-16	4.00	2.07E-16	4.00	2.07E-16	4.00	

**Table 4.4**

**Example 4.1** (one-dimensional Burgers equation).  $L^\infty$  errors and orders at  $t = 0.5/\pi$  with  $\epsilon = \Delta x^2$ .

$N$	LW-1		LW-2		LW-3		LW-4		
	$L^\infty$ error	Order							
$p^2$	20	3.19E-03		3.97E-03		8.74E-03		4.71E-02	
	40	5.24E-04	2.60	5.25E-04	2.92	5.28E-04	4.05	1.24E-02	1.92
	80	1.02E-04	2.36	1.02E-04	2.37	1.02E-04	2.38	1.74E-03	2.84
	160	1.65E-05	2.62	1.65E-05	2.62	1.65E-05	2.62	1.97E-04	3.14
	320	2.52E-06	2.72	2.52E-06	2.72	2.52E-06	2.72	1.25E-05	3.97
	640	3.56E-07	2.82	3.56E-07	2.82	3.56E-07	2.82	3.56E-07	5.14
	1280	4.81E-08	2.89	4.81E-08	2.89	4.81E-08	2.89	4.81E-08	2.89
	2560	6.34E-09	2.92	6.34E-09	2.92	6.34E-09	2.92	6.34E-09	2.92
	5120	8.94E-10	2.83	8.94E-10	2.83	8.94E-10	2.83	8.94E-10	2.83
	10240	1.37E-10	2.71	1.37E-10	2.71	1.37E-10	2.71	1.37E-10	2.71
20480	2.19E-11	2.64	2.19E-11	2.64	2.19E-11	2.64	2.19E-11	2.64	
$p^3$	20	3.92E-04		4.30E-02		4.35E-02		4.58E-02	
	40	1.55E-05	4.66	5.42E-05	9.63	2.65E-04	7.36	1.24E-02	1.88
	80	8.48E-07	4.19	1.04E-06	5.70	2.02E-06	7.03	2.67E-03	2.22
	160	5.38E-08	3.98	5.42E-08	4.27	5.51E-08	5.20	5.54E-04	2.27
	320	3.42E-09	3.98	3.42E-09	3.99	3.43E-09	4.01	7.03E-05	2.98
	640	2.15E-10	3.99	2.15E-10	3.99	2.15E-10	3.99	1.99E-07	8.46
	1280	1.35E-11	3.99	1.35E-11	3.99	1.35E-11	3.99	5.41E-10	8.52
	2560	8.46E-13	4.00	8.46E-13	4.00	8.46E-13	4.00	1.11E-12	8.93
	5120	5.30E-14	4.00	5.30E-14	4.00	5.30E-14	4.00	5.30E-14	4.39
	10240	3.31E-15	4.00	3.31E-15	4.00	3.31E-15	4.00	3.31E-15	4.00
20480	2.07E-16	4.00	2.07E-16	4.00	2.07E-16	4.00	2.07E-16	4.00	

**Example 4.3.** We solve the following two-dimensional Euler system

$$\begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E + p) \end{pmatrix}_x + \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E + p) \end{pmatrix}_y = 0 \tag{4.3}$$

in which  $\rho, E$  and  $p$  represent the density, total energy and pressure, and  $u$  and  $v$  are the velocity components in the  $x$ - and  $y$ -directions, respectively. The system is completed by the equation of state

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

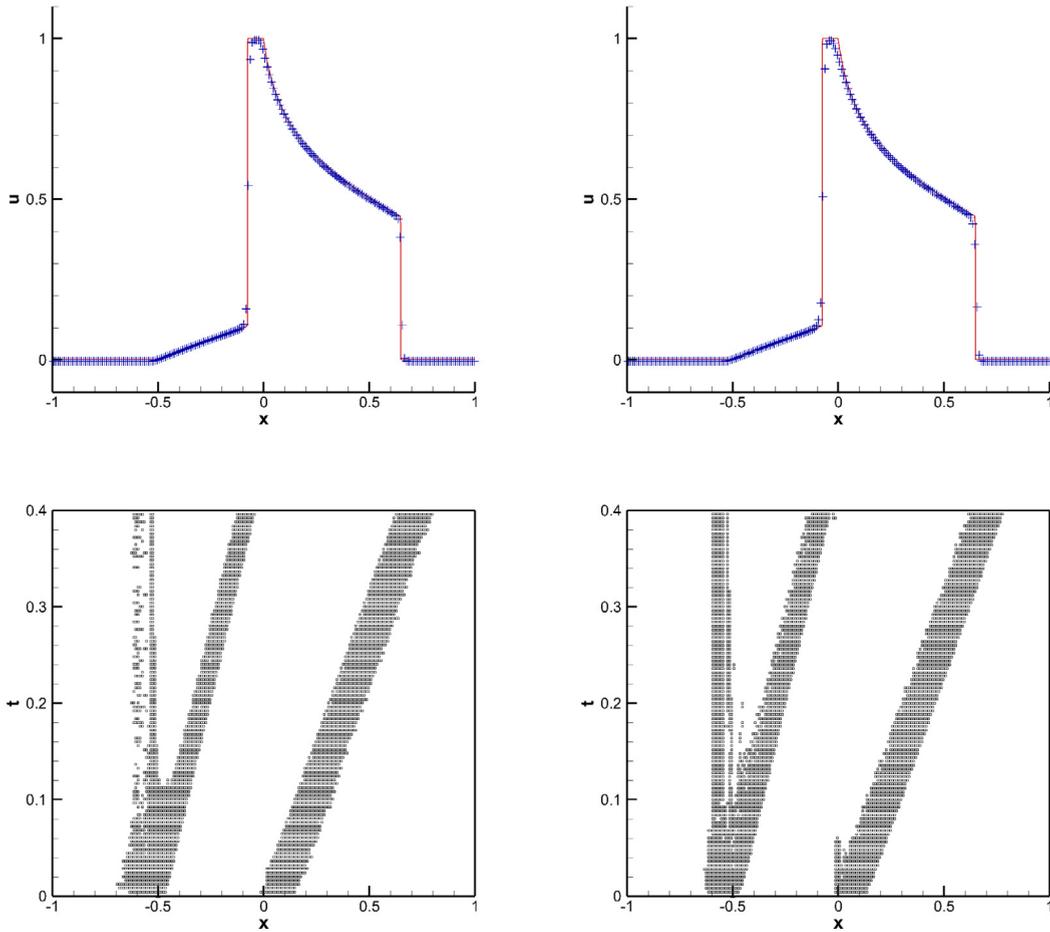


Fig. 4.1. Example 4.4 (Buckley-Leverett problem), solutions (upper row) at  $t = 0.4$  and time history of the troubled cells (lower row),  $N = 200$ ,  $P^2$  (left) and  $P^3$  (right).

Table 4.5  
Example 4.2 (two-dimensional Burgers equation),  $L^1$  errors and orders at  $t = 0.5/\pi$ .

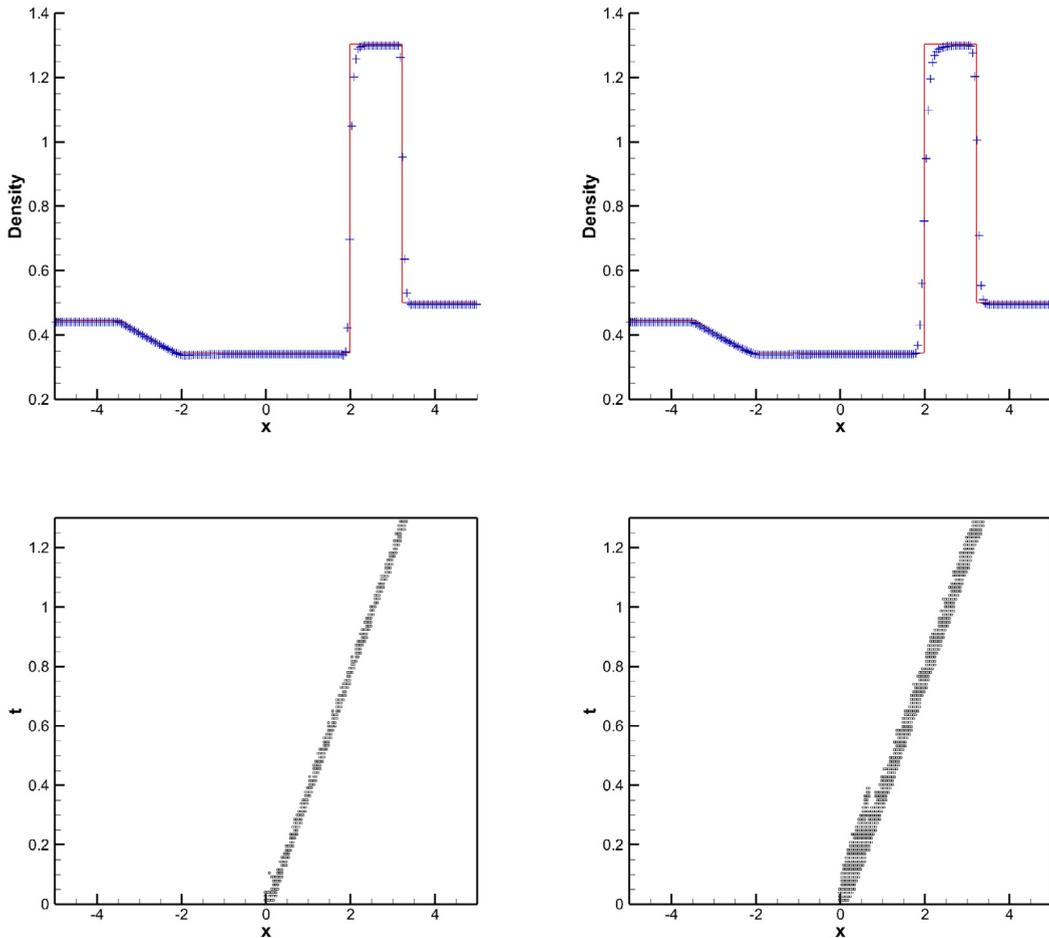
$N$	LW-1		LW-2		LW-3		LW-4		
	$L^1$ error	Order							
$P^2$	200	1.62E-05		1.62E-05		1.65E-05		5.31E-04	
	400	2.10E-06	2.94	2.10E-06	2.94	2.10E-06	2.97	5.71E-06	6.54
	600	6.42E-07	2.92	6.42E-07	2.92	6.42E-07	2.92	6.45E-07	5.38
	800	2.77E-07	2.92	2.77E-07	2.92	2.77E-07	2.92	2.76E-07	2.95
	1000	1.44E-07	2.92	1.44E-07	2.92	1.44E-07	2.92	1.44E-07	2.90
	1200	8.48E-08	2.92	8.48E-08	2.92	8.48E-08	2.92	8.48E-08	2.92
	1400	5.41E-08	2.92	5.41E-08	2.92	5.41E-08	2.92	5.40E-08	2.92
	1600	3.66E-08	2.92	3.66E-08	2.92	3.66E-08	2.92	3.66E-08	2.92
1800	2.59E-08	2.92	2.59E-08	2.92	2.59E-08	2.92	2.59E-08	2.92	
$P^3$	200	2.35E-07		2.49E-07		2.74E-06		1.29E-03	
	400	1.48E-08	3.99	1.48E-08	4.07	1.55E-08	7.46	7.14E-05	4.18
	600	2.93E-09	3.99	2.93E-09	3.99	2.93E-09	4.11	1.28E-07	15.60
	800	9.30E-10	3.99	9.30E-10	3.99	9.30E-10	3.99	7.19E-09	10.01
	1000	3.81E-10	4.00	3.81E-10	4.00	3.81E-10	4.00	9.81E-10	8.93
	1200	1.84E-10	4.00	1.84E-10	4.00	1.84E-10	4.00	2.56E-10	7.37
	1400	9.94E-11	4.00	9.94E-11	4.00	9.94E-11	4.00	1.07E-10	5.65
	1600	5.83E-11	4.00	5.83E-11	4.00	5.83E-11	4.00	5.86E-11	4.53
	1800	3.64E-11	4.00	3.64E-11	4.00	3.64E-11	4.00	3.63E-11	4.08

with gas constant  $\gamma = 1.4$ . We solve (4.3) in the domain  $[0, 2] \times [0, 2]$  up to  $t = 2$  with the initial condition  $(\rho, u, v, p)^T = (1.0 + 0.2 * \sin(\pi(x + y)), 0.5, 0.5, 1)^T$  and periodic boundary conditions. The exact solution of  $\rho$  is

**Table 4.6**

**Example 4.2** (two-dimensional Burgers equation).  $L^\infty$  errors and orders at  $t = 0.5/\pi$ .

$N$	LW-1		LW-2		LW-3		LW-4		
	$L^\infty$ error	Order							
$p^2$	200	2.27E-05		2.27E-05		2.27E-05		6.20E-04	
	400	3.07E-06	2.88	3.07E-06	2.88	3.07E-06	2.88	9.21E-06	6.07
	600	9.37E-07	2.93	9.37E-07	2.93	9.37E-07	2.93	9.37E-07	5.64
	800	4.02E-07	2.94	4.02E-07	2.94	4.02E-07	2.94	4.02E-07	2.94
	1000	2.08E-07	2.96	2.08E-07	2.96	2.08E-07	2.96	2.08E-07	2.96
	1200	1.21E-07	2.96	1.21E-07	2.96	1.21E-07	2.96	1.21E-07	2.96
	1400	7.67E-08	2.97	7.67E-08	2.97	7.67E-08	2.97	7.67E-08	2.97
	1800	5.15E-08	2.97	5.15E-08	2.97	5.15E-08	2.97	5.15E-08	2.97
$p^3$	200	5.84E-07		5.84E-07		7.38E-06		9.20E-04	
	400	3.77E-08	3.96	3.77E-08	3.96	3.77E-08	7.61	1.56E-04	2.56
	600	7.52E-09	3.97	7.52E-09	3.97	7.52E-09	3.97	2.72E-07	15.67
	800	2.39E-09	3.98	2.39E-09	3.98	2.39E-09	3.98	1.03E-08	11.38
	1000	9.83E-10	3.99	9.83E-10	3.99	9.83E-10	3.99	1.11E-09	9.98
	1200	4.75E-10	3.99	4.75E-10	3.99	4.75E-10	3.99	4.73E-10	4.68
	1400	2.57E-10	3.99	2.57E-10	3.99	2.57E-10	3.99	2.56E-10	3.97
	1800	1.51E-10	3.99	1.51E-10	3.99	1.51E-10	3.99	1.51E-10	3.98
		9.42E-11	3.99	9.42E-11	3.99	9.42E-11	3.99	9.42E-11	3.99



**Fig. 4.2.** Example 4.5 (Lax problem), density solutions (upper row) at  $t = 1.3$  and time history of the troubled cells (lower row),  $N = 200$ ,  $p^2$  (left) and  $p^3$  (right).

$\rho(x, t) = 1.0 + 0.2 * \sin(\pi(x + y - t))$ . The computation was also performed in double precision. In Tables 4.7 and 4.8 we present the errors and numerical orders of accuracy of  $\rho$ . We can see that the WENO limiter maintains the optimal

**Table 4.7**

**Example 4.3** (two-dimensional Euler system).  $L^1$  errors and orders of the density at  $t = 2$ .

$N$	LW-1		LW-2		LW-3		LW-4		
	$L^1$ error	Order							
$p^2$	10	3.73E-02		8.02E-02		2.68E-01		3.89E-01	
	20	1.14E-03	5.03	5.55E-03	3.85	4.57E-02	2.55	9.96E-02	1.96
	40	4.39E-05	4.70	8.12E-05	6.09	2.13E-03	4.42	2.71E-02	1.88
	80	5.33E-06	3.04	5.35E-06	3.93	1.94E-05	6.78	1.99E-03	3.77
	160	6.63E-07	3.01	6.63E-07	3.01	6.67E-07	4.87	3.86E-05	5.69
	320	8.28E-08	3.00	8.28E-08	3.00	8.28E-08	3.01	1.05E-07	8.53
	640	1.03E-08	3.00	1.03E-08	3.00	1.03E-08	3.00	1.03E-08	3.34
$p^3$	10	3.03E-01		3.56E-01		3.89E-01		3.94E-01	
	20	2.61E-02	3.54	4.49E-02	2.99	9.56E-02	2.02	1.10E-01	1.84
	40	1.78E-05	10.52	1.84E-04	7.93	9.54E-03	3.33	3.25E-02	1.76
	80	6.57E-08	8.08	4.16E-07	8.79	4.40E-05	7.76	4.72E-03	2.78
	160	2.65E-09	4.63	2.68E-09	7.28	3.45E-08	10.32	3.21E-04	3.88
	320	1.65E-10	4.00	1.65E-10	4.02	1.66E-10	7.70	6.77E-08	12.21
	640	1.03E-11	4.00	1.03E-11	4.00	1.03E-11	4.00	1.89E-11	11.81

**Table 4.8**

**Example 4.3** (two-dimensional Euler system).  $L^\infty$  errors and orders of the density at  $t = 2$ .

$N$	LW-1		LW-2		LW-3		LW-4		
	$L^\infty$ error	Order							
$p^2$	10	2.00E-02		4.70E-02		1.16E-01		1.58E-01	
	20	9.73E-04	4.36	4.29E-03	3.46	2.32E-02	2.33	5.88E-02	1.43
	40	4.22E-05	4.53	8.50E-05	5.66	2.58E-03	3.16	1.61E-02	1.87
	80	5.46E-06	2.95	5.46E-06	3.96	3.99E-05	6.02	2.43E-03	2.73
	160	6.88E-07	2.99	6.88E-07	2.99	6.88E-07	5.86	1.12E-04	4.44
	320	8.62E-08	3.00	8.62E-08	3.00	8.62E-08	3.00	1.22E-07	9.84
	640	1.08E-08	3.00	1.08E-08	3.00	1.08E-08	3.00	1.08E-08	3.50
$p^3$	10	1.28E-01		1.46E-01		1.58E-01		1.60E-01	
	20	1.54E-02	3.06	2.32E-02	2.66	5.61E-02	1.50	6.20E-02	1.37
	40	3.77E-05	8.67	3.73E-04	5.96	7.88E-03	2.83	1.94E-02	1.68
	80	1.74E-07	7.76	1.53E-06	7.93	1.41E-04	5.80	4.18E-03	2.22
	160	2.97E-09	5.87	3.99E-09	8.58	1.75E-07	9.65	4.73E-04	3.14
	320	1.80E-10	4.04	1.80E-10	4.47	2.03E-10	9.75	2.48E-07	10.90
	640	1.13E-11	3.99	1.13E-11	3.99	1.13E-11	4.17	3.73E-11	12.70

$(k + 1)$ th order of accuracy for LW-1, LW-2 and LW-3 cases. For LW-4 case, we can expect that the correct orders will be recovered on finer meshes.

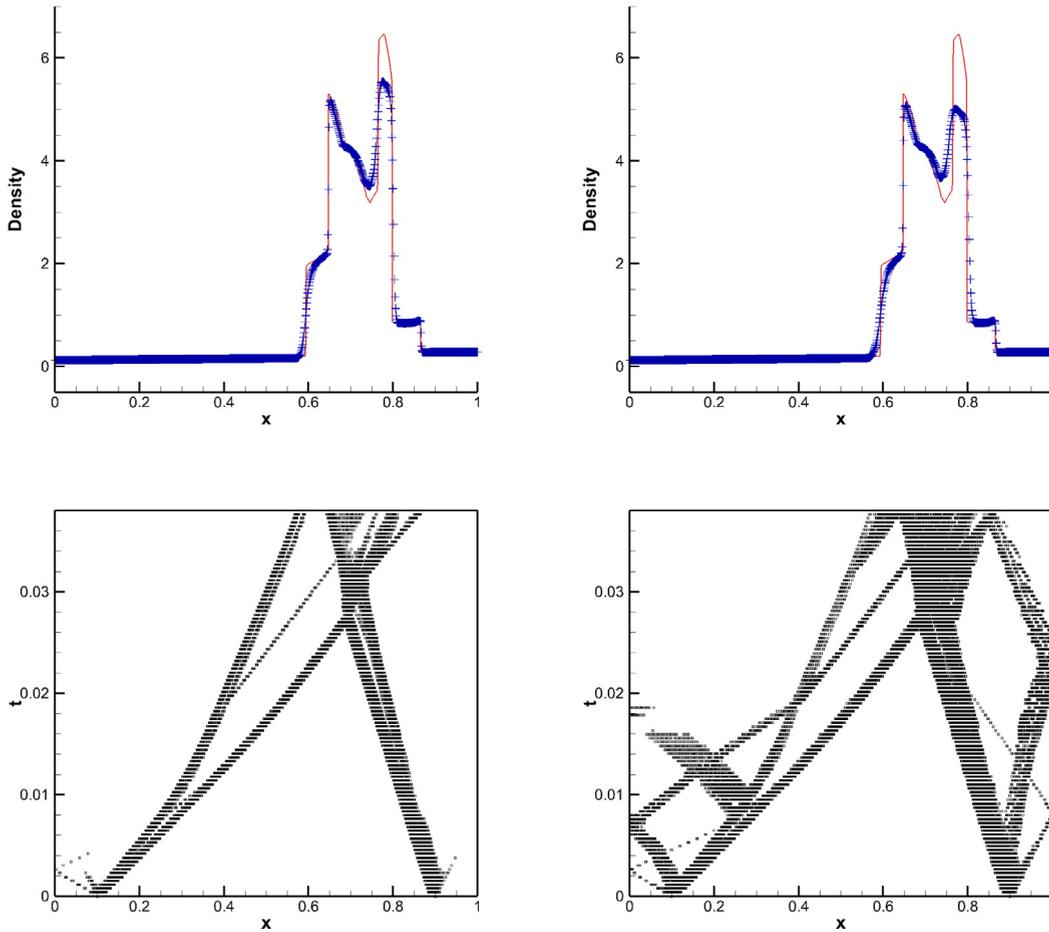
4.2. Numerical results for discontinuous solutions

This subsection gives the numerical results of classical test examples that have discontinuous solutions. Note that the linear weights LW-2 ( $\gamma_0 = 0.8$ ) is used unless otherwise specified. Although the KXRCF parameter  $C_k$  in Eq. (3.1) was taken to be one originally [41], some later numerical results [29,42,46] indicated that this troubled-cell indicator with fixed  $C_k$  tended to mark more troubled cells than necessary for higher-order DG methods. When more cells than necessary are limited, higher-order DG method may produce inferior solutions. This problem also arise in the test examples in this section. One remedy is to adjust the value of  $C_k$  for different values of  $k$  and different test examples. However, we will discuss this in detail in the blast wave problem (Example 4.6) alone as an example because this is not the emphasis of the paper and the space is limited. In the following test examples  $C_k$  is equal to one if it is not stated explicitly.

**Example 4.4.** We solve the following Buckley–Leverett problem

$$u_t + \left( \frac{4u^2}{4u^2 + (1 - u)^2} \right)_x = 0 \tag{4.4}$$

with the initial condition  $u_0 = 1$  for  $-0.5 \leq x \leq 0$  and  $u_0 = 0$  elsewhere. We plot the numerical solutions using 200 cells at  $t = 0.4$  in Fig. 4.1. Here and below, the solid line is for the exact solution or grid converged solution, and the symbols “+” are for the numerical solution (cell averages). We also present the time history of the troubled cells (denoted by tiny squares) in Fig. 4.1. Note that such plots in this paper display only the troubled cells on at most 100 time-levels. From the figure we can see that the KXRCF troubled-cell indicator marks the cells as troubled cells not only near the shocks but also near the rarefaction corners, and the solutions of both orders are oscillatory free.



**Fig. 4.3.** Example 4.6 (blast wave problem), density solutions (upper row) at  $t = 0.038$  and time history of the troubled cells (lower row),  $N = 800$ ,  $P^2$  (left) and  $P^3$  (right).

**Example 4.5.** In this example we solve the Lax problem which is governed by the one-dimensional Euler system of gas dynamics

$$\begin{pmatrix} \rho \\ \rho v \\ E \end{pmatrix}_t + \begin{pmatrix} \rho v \\ \rho v^2 + p \\ v(E + p) \end{pmatrix}_x = 0 \tag{4.5}$$

where  $\rho$  is the density,  $v$  is the velocity,  $E$  is the total energy,  $p$  is the pressure, related to the total energy by  $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho v^2$  with  $\gamma = 1.4$ . The initial condition is given by

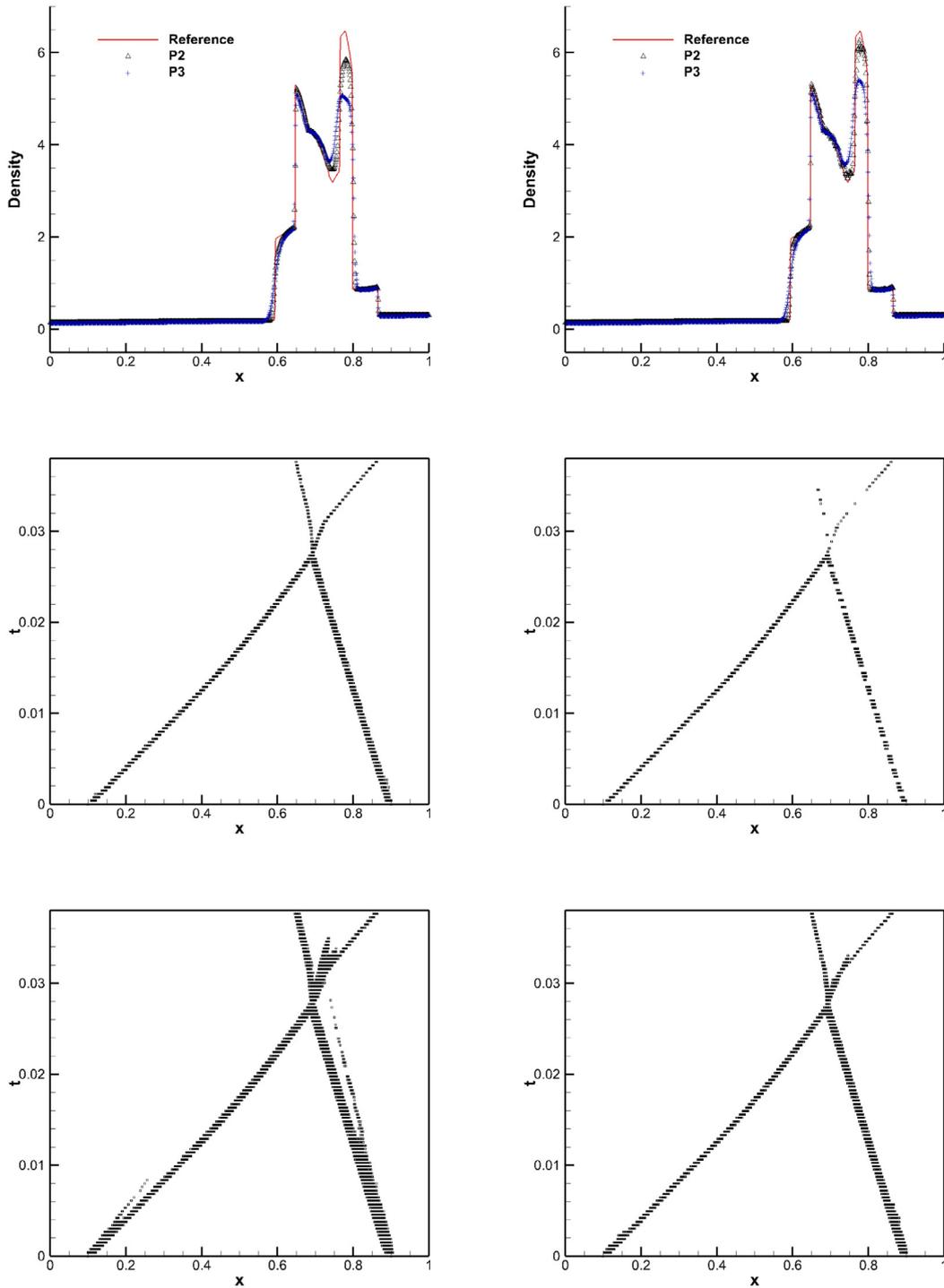
$$(\rho, v, p) = \begin{cases} (0.445, 0.698, 3.528) & \text{if } x \leq 0, \\ (0.5, 0, 0.571) & \text{if } x > 0. \end{cases} \tag{4.6}$$

Numerical solutions of density at  $t = 1.3$  with 200 cells and the corresponding time history of the troubled cells are presented in Fig. 4.2. We again see the good performance of our schemes in controlling the oscillations.

**Example 4.6.** We solve the blast wave problem which is also governed by the one-dimensional Euler system (4.5). The initial condition is

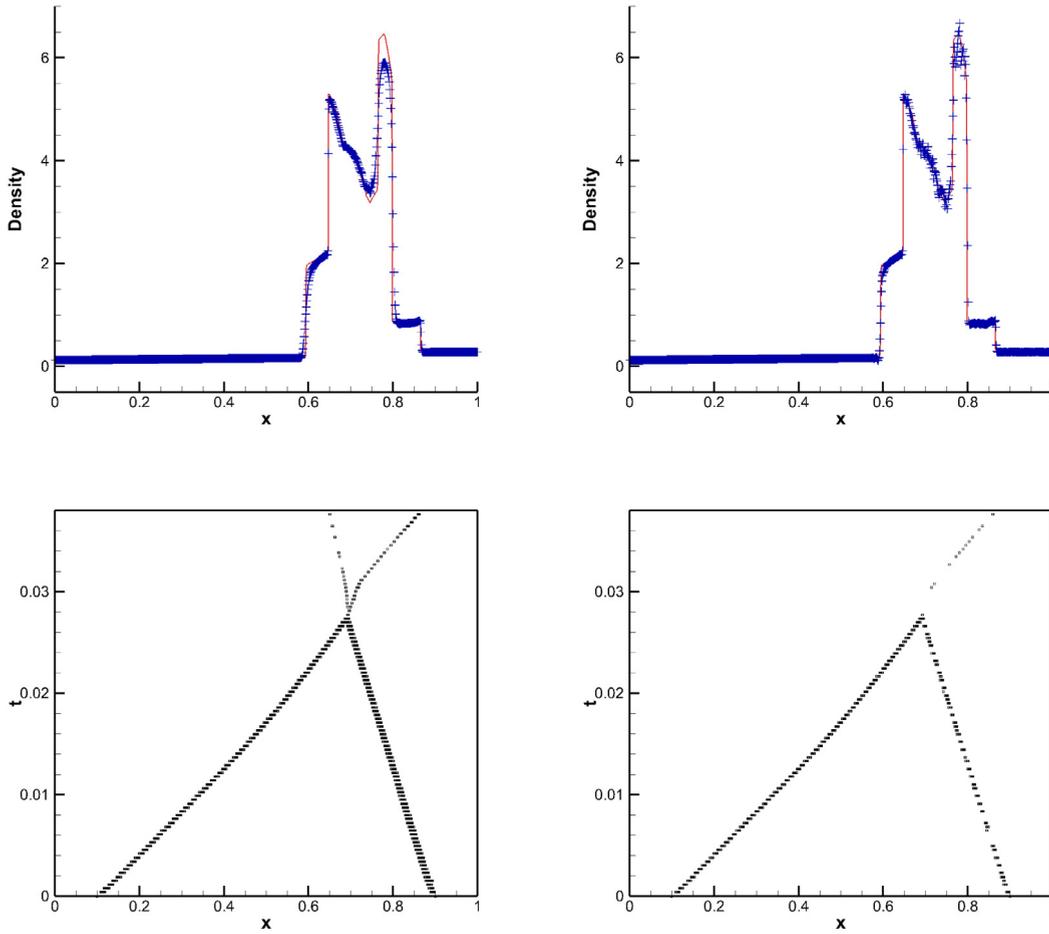
$$(\rho, v, p) = \begin{cases} (1, 0, 1000) & \text{if } 0 \leq x < 0.1, \\ (1, 0, 0.01) & \text{if } 0.1 \leq x < 0.9, \\ (1, 0, 100) & \text{if } 0.9 \leq x \leq 1. \end{cases} \tag{4.7}$$

A reflecting boundary condition is applied to both ends. Numerical solutions of density at  $t = 0.038$  and the time history of the troubled cells using 800 cells are presented in Fig. 4.3. The numerical oscillations are well controlled by both  $P^2$



**Fig. 4.4.** Example 4.6 (blast wave problem), density solutions (top row) at  $t = 0.038$  and time history of the troubled cells (middle row for  $P^2$  and bottom row for  $P^3$ ),  $N = 800$ ,  $C_t = 10^2$  (left column) and  $10^3$  (right column).

and  $P^3$  schemes. However, the  $P^3$  solution is worse than the  $P^2$  one which is mainly due to the overidentification of the troubled cells. Similar phenomena were reported in [4] and can be seen in some other examples in this paper. We can see in Fig. 4.3 that the KXRFC troubled-cell indicator identifies many more troubled cells in the  $P^3$  scheme than the  $P^2$  one.



**Fig. 4.5.** Example 4.6 (blast wave problem), density solutions (upper row) at  $t = 0.038$  and time history of the troubled cells (lower row) for  $P^3$ ,  $N = 800$ ,  $C_k = 10^4$  (left column) and  $10^5$  (right column).

**Table 4.9**

Example 4.6 (blast wave problem). Data on troubled cells with  $N = 800$ .

Order	$C_k = 1$		$C_k = 100$		$C_k = 1000$	
	$P^2$	$P^3$	$P^2$	$P^3$	$P^2$	$P^3$
$\eta_1$	8.70	17.26	2.22	4.59	1.33	2.88
$\eta_2$	0.15	4.24	1.47E-4	0.23	8.71E-5	0.23

Increasing the value of  $C_k$  in Eq. (3.1) can decrease the number of troubled cells and hence improve the solution quality. We show both  $P^2$  and  $P^3$  results with  $C_k = 10^2$  and  $10^3$  in Fig. 4.4, and  $P^3$  results with  $C_k = 10^4$  and  $10^5$  in Fig. 4.5 ( $P^2$  scheme with  $C_k = 10^4$  or higher blew up). As  $C_k$  goes larger, fewer troubled cells are identified and better solutions are obtained. When  $C_k$  becomes too large such as  $10^5$  for  $P^3$  scheme with 800 cells, oscillations start to appear. With the same choice of  $C_k$  we always observe that the  $P^3$  solution is poorer than the  $P^2$  one because  $P^3$  scheme identifies more troubled cells. To be more precise, we give the average (over time) percentage of troubled cells  $\eta_1$  in Table 4.9. We also give  $\eta_2$ , the average percentage of troubled cells in which solutions are actually limited to linear order by the WENO limiter. By saying that a troubled cell is limited to linear order, we mean that the second and higher-order degrees of freedom of  $\rho$  in this cell are limited to close to zero, which is  $\rho_i^{(l)} \leq 10^{-4}$  for  $l = 2, \dots, k$  in our computation. In Table 4.9, both  $\eta_1$  and  $\eta_2$  are greater in the  $P^3$  scheme than in the  $P^2$  one. It means that the  $P^3$  scheme always identifies more troubled cells, and more cells that are actually untroubled but are mistakenly identified as troubled cells are unnecessarily limited in the  $P^3$  scheme and hence the whole solution is affected.

To sum up, the performance of our new WENO limiter depends on the accuracy of the troubled-cell indicator. Since the emphasis of this paper is not on the troubled-cell indicator, we will try no more for a better troubled-cell indicator to improve the solution quality.

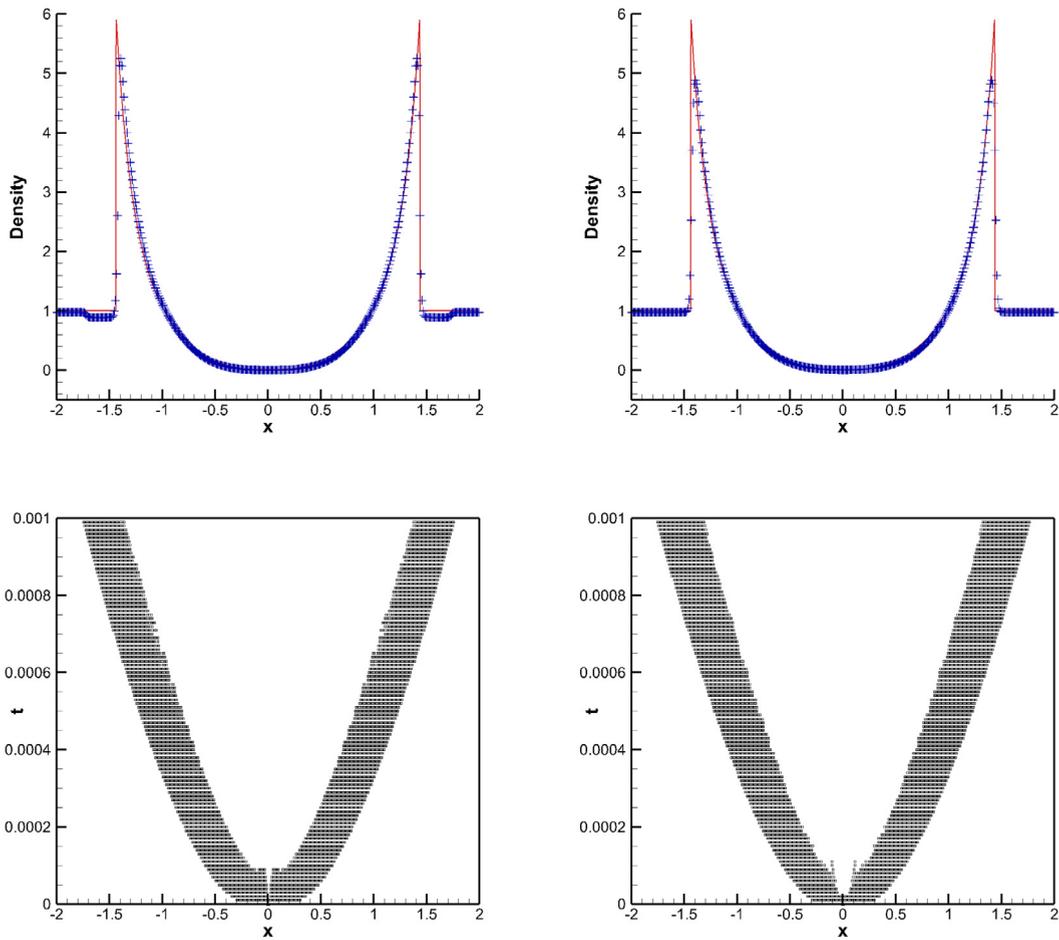


Fig. 4.6. Example 4.7 (Sedov blast wave problem), density solutions (upper row) at  $t = 0.001$  and time history of the troubled cells (lower row),  $N = 400$ ,  $\gamma_0 = 0.8$ ,  $P^2$  (left) and  $P^3$  (right).

**Example 4.7.** Consider the Sedov blast wave problem which is a typical low density problem involving shocks. We solve the one-dimensional Euler system (4.5) in the domain  $[-2, 2]$  till  $t = 0.001$ . The initial conditions are  $(\rho, v, E) = (1, 0, 10^{-12})$  everywhere except that the energy  $E$  in the center cell is the constant  $\frac{3 \cdot 200,000}{\Delta x}$  (emulating a  $\delta$ -function at the center). The exact solution formula can be found in [47,48]. Numerical solutions of density with 400 cells are presented in Fig. 4.6, along with the time history of the troubled cells. We can see that the solutions are overall fine except that a dent appears in both of the flat regions next to the discontinuities in the  $P^2$  solution. Decreasing the value of  $\gamma_0$  can overcome this problem, see the solutions in Fig. 4.7 obtained with  $\gamma_0 = 0.02$ .

**Example 4.8.** This test example is an extreme Riemann problem called the Leblanc shock tube problem. We again solve the one-dimensional Euler system (4.5) in the domain  $[-10, 10]$  with the initial condition

$$(\rho, v, p) = \begin{cases} (2, 0, 10^9) & \text{if } x \leq 0, \\ (0.001, 0, 1) & \text{if } x > 0. \end{cases} \tag{4.8}$$

In the computation, the scheme ends up with a blow-up when  $\gamma_0 \geq 0.57$ . We are able to obtain a converged solution with LW-3 ( $\gamma_0 = 0.5$ ) although small oscillations are spotted, see Fig. 4.8 where the numerical solutions of density at  $t = 0.0001$  with 6400 cells and the time history of the troubled cells are presented.

**Example 4.9.** Consider the double Mach reflection problem [49] which is governed by the two-dimensional Euler system (4.3). We use exactly the same setup as in [49]. The computational domain is  $[0, 4] \times [0, 1]$ . The reflecting 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  $x = \frac{1}{6}, y = 0$ , making a  $60^\circ$  angle with the  $x$ -axis. The undisturbed air ahead of the shock has a density of 1.4 and a pressure of 1. On the left and right boundaries, the inflow and outflow boundary conditions are used, respectively. For

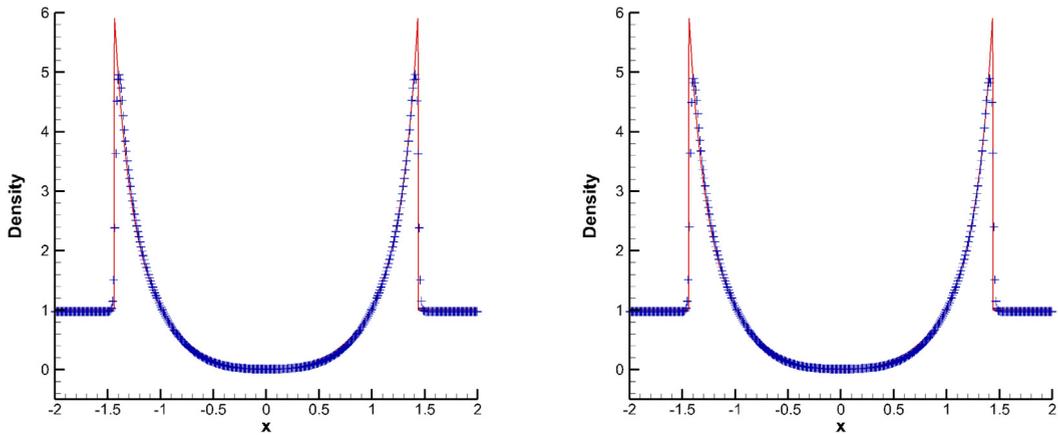


Fig. 4.7. Example 4.7 (Sedov blast wave problem), density solutions at  $t = 0.001$ ,  $N = 400$ ,  $\gamma_0 = 0.02$ ,  $P^2$  (left) and  $P^3$  (right).

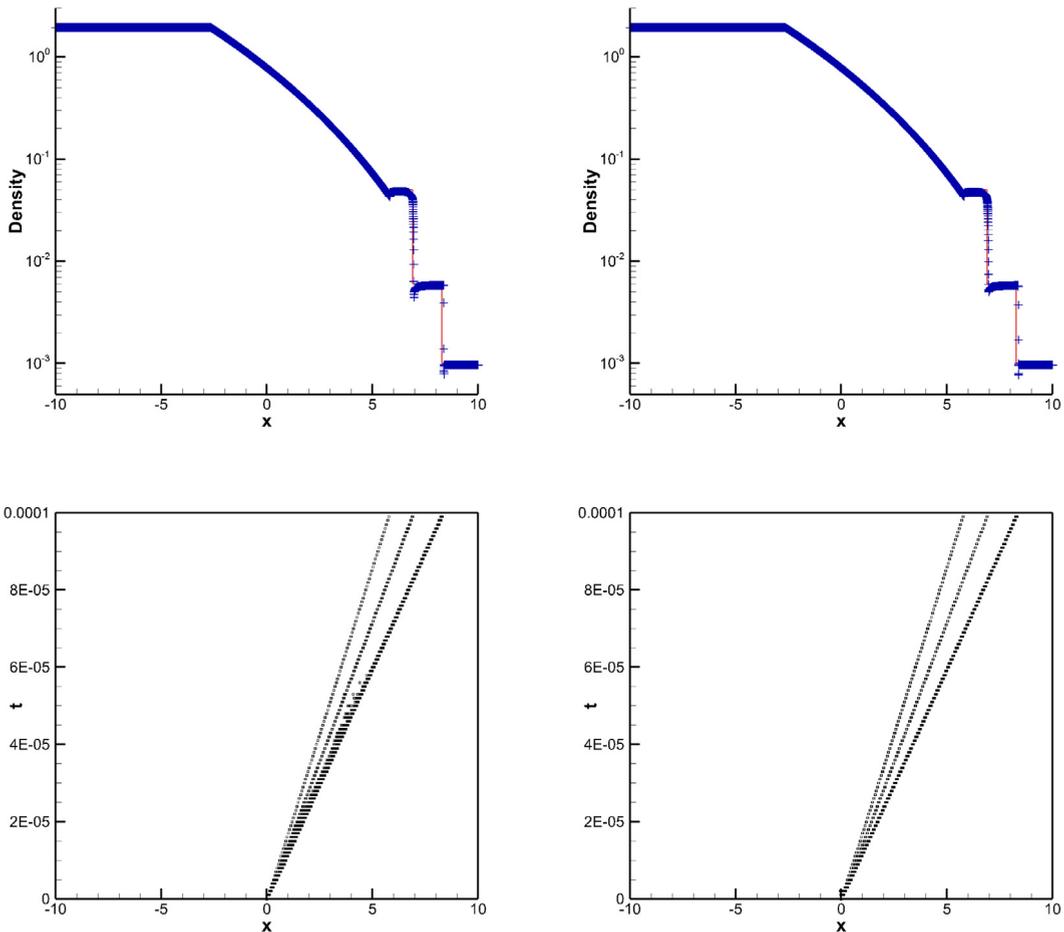
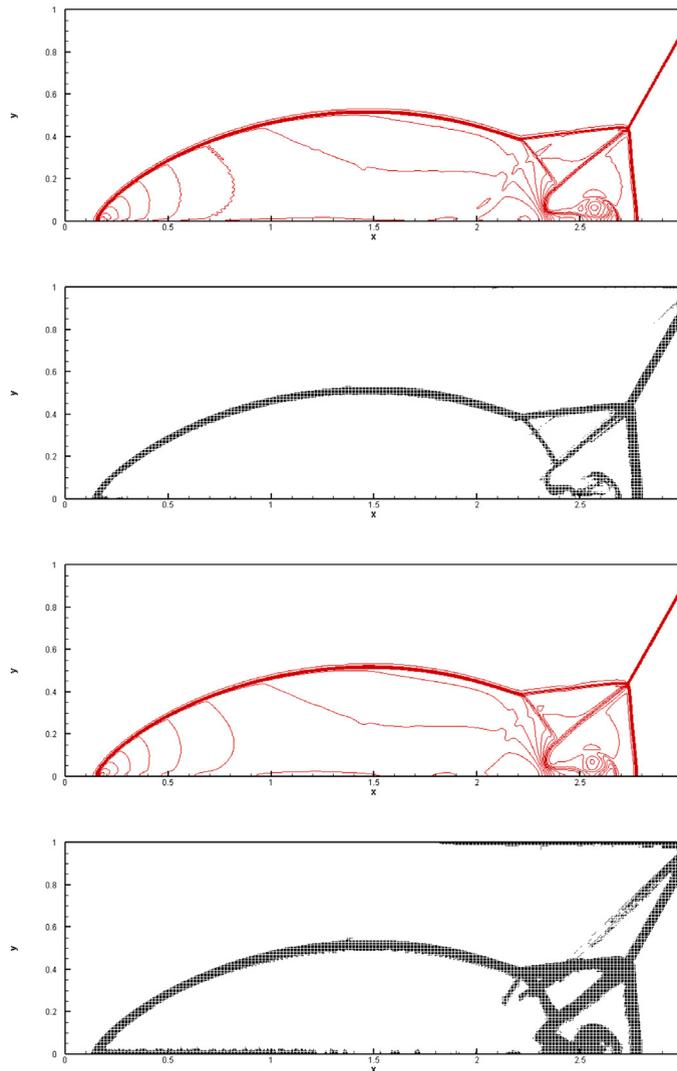


Fig. 4.8. Example 4.8 (Leblanc shock tube problem), density solutions (upper row) at  $t = 0.0001$  and time history of the troubled cells (lower row),  $N = 6400$ ,  $\gamma_0 = 0.5$ ,  $P^2$  (left) and  $P^3$  (right).

the bottom boundary, the reflective boundary condition is applied at the wall and the exact post-shock condition is used for the rest. Boundary conditions at the top correspond to the exact motion of a Mach 10 shock. The problem is run till a simulation time of 0.2 is reached. Density contours and the troubled cells are plotted in Fig. 4.9 using  $960 \times 240$  cells. We can see that both schemes with different orders produce non-oscillatory solutions.

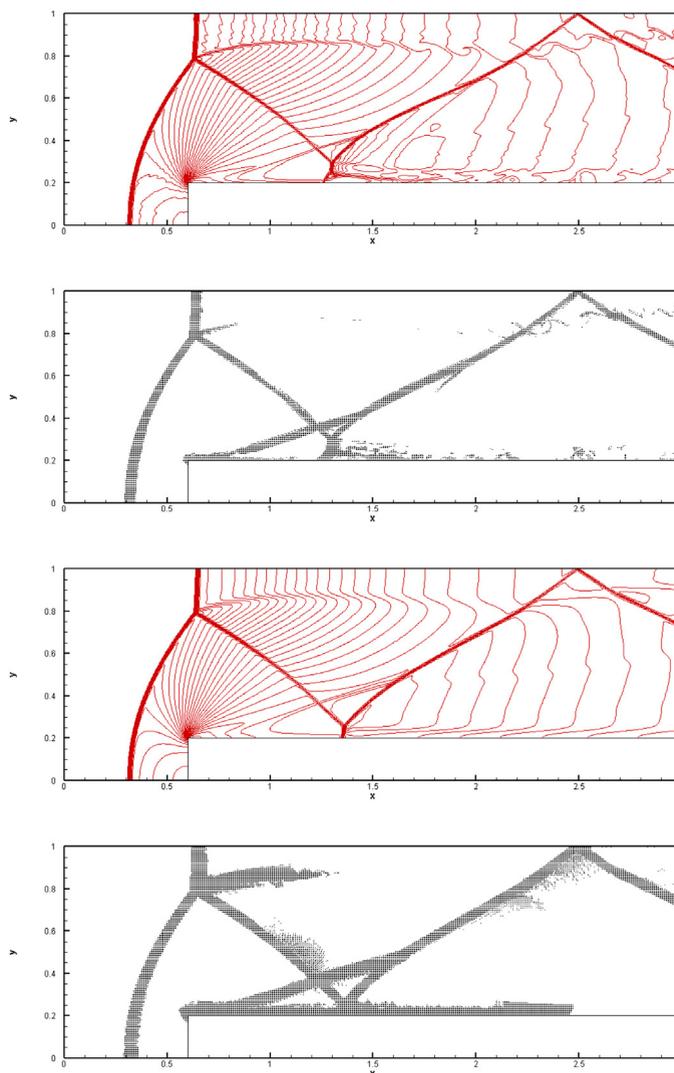


**Fig. 4.9.** Example 4.9 (double Mach reflection problem), thirty equally spaced density contours from 1.3 to 23 and the troubled cells at  $t = 0.2$ ,  $N_x = 960, N_y = 240, P^2$  (the upper two) and  $P^3$  (the lower two).

**Example 4.10.** Our last example is the forward facing step problem [49] which is also governed by the two-dimensional Euler equations (4.3). The problem starts with a uniform, right-going Mach 3 flow in a wind tunnel of 3 units long and 1 unit wide which contains a 0.2 units high step located 0.6 units from the left-hand end of the tunnel. Reflecting boundary conditions are used along the walls of the tunnel, and inflow and outflow boundary conditions are applied on the left and right boundaries, respectively. The problem is run until a simulation time of 4.0. We plot the density contours and the troubled cells in Fig. 4.10 with  $N_x = 600$  and  $N_y = 200$ . Once again, it is shown that the shocks are well captured without oscillations.

## 5. Conclusion

In this paper, we have developed a new limiter for the RKDG method solving hyperbolic conservation laws using the WENO methodology. This limiter employs a general framework of limiters where troubled cells are first identified by the KXRFC troubled-cell indicator, and then the newly proposed WENO solution reconstruction procedure is applied in these troubled cells to control the spurious oscillations. Our WENO limiter uses the high-order DG solution polynomial in the target cell and the linear polynomials obtained from small stencils to reconstruct a new high-order polynomial. As a result, this limiter has a very compact stencil which includes only the target cell and its immediate neighbors, and still maintains the high-order accuracy and advantage of simplicity. These are demonstrated by extensive numerical tests including both one- and two-dimensional problems and both scalar and systematic problems.



**Fig. 4.10.** Example 4.10 (forward facing step problem), thirty equally spaced density contours from 0.32 to 6.15 and the troubled cells at  $t = 4$ ,  $N_x = 600$ ,  $N_y = 200$ ,  $P^2$  (the upper two) and  $P^3$  (the lower two).

Future work of this subject should certainly look at extending this WENO limiter to unstructured triangular meshes for two-dimensional problems and to tetrahedral or parallelepipedal meshes for three-dimensional problems. The reconstruction procedure will be similar to those in [32–34].

## References

- [1] W.H. Reed, T.R. Hill, Triangular mesh methods for neutron transport equation. Technical report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.
- [2] B. Cockburn, C.-W. Shu, The Runge-Kutta local projection  $P^1$ -discontinuous Galerkin finite element method for scalar conservation laws, *Math. Model. Numer. Anal. (M<sup>2</sup>AN)* 25 (1991) 337–361.
- [3] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: General framework, *Math. Comp.* 52 (1989) 411–435.
- [4] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems, *J. Comput. Phys.* 84 (1989) 90–113.
- [5] B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case, *Math. Comp.* 54 (1990) 545–581.
- [6] B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws v: Multidimensional systems, *J. Comput. Phys.* 141 (1998) 199–224.
- [7] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, *J. Comput. Phys.* 77 (1988) 439–471.
- [8] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, *Math. Comp.* 49 (1987) 105–121.

- [9] R. Biswas, K. Devine, J. Flaherty, Parallel, adaptive finite element methods for conservation laws, *Appl. Numer. Math.* 14 (1994) 255–283.
- [10] A. Burbeau, P. Sagaut, C.H. Bruneau, A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods, *J. Comput. Phys.* 169 (2001) 111–150.
- [11] J. Qiu, C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, *SIAM J. Sci. Comput.* 26 (2005) 907–929.
- [12] J. Zhu, J. Qiu, C.-W. Shu, M. Dumbser, Runge-Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes, *J. Comput. Phys.* 227 (2008) 4330–4353.
- [13] J. Zhu, J. Qiu, Runge-Kutta discontinuous Galerkin method using WENO type limiters: Three dimensional unstructured meshes, *Commun. Comput. Phys.* 11 (2012) 985–1005.
- [14] X.-D. Liu, S. Osher, T. Chan, Weighted essentially nonoscillatory schemes, *J. Comput. Phys.* 115 (1994) 200–212.
- [15] G. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1996) 202–228.
- [16] O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, *J. Comput. Phys.* 144 (1998) 194–212.
- [17] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, *J. Comput. Phys.* 150 (1999) 97–127.
- [18] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: One dimensional case, *J. Comput. Phys.* 193 (2003) 115–135.
- [19] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: Two dimensional case, *Comp. Fluid.* 34 (2005) 642–663.
- [20] J. Zhu, J. Qiu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method III: Unstructured meshes, *J. Sci. Comput.* 39 (2009) 293–321.
- [21] X. Zhong, C.-W. Shu, A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods, *J. Comput. Phys.* 232 (2013) 397–415.
- [22] J. Zhu, X. Zhong, C.-W. Shu, J. Qiu, Runge-Kutta discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes, *J. Comput. Phys.* 248 (2013) 200–220.
- [23] J. Zhu, X. Zhong, C.-W. Shu, J. Qiu, Runge-Kutta discontinuous Galerkin method with a simple and compact hermite WENO limiter, *Commun. Comput. Phys.* 19 (2016) 944–969.
- [24] J. Zhu, X. Zhong, C.-W. Shu, J. Qiu, Runge-Kutta discontinuous Galerkin method with a simple and compact hermite WENO limiter on unstructured meshes, *Commun. Comput. Phys.* 21 (2017) 623–649.
- [25] M. Dumbser, D.S. Balsara, E.F. Toro, C.D. Munz, A unified framework for the construction of one-step finite-volume and discontinuous Galerkin schemes on unstructured meshes, *J. Comput. Phys.* 227 (2008) 8209–8253.
- [26] M. Dumbser, O. Zanotti, R. Loubère, S. Diot, A posteriori subcell limiting of the discontinuous Galerkin finite element method for hyperbolic conservation laws, *J. Comput. Phys.* 278 (2014) 47–75.
- [27] M.J. Vuik, J.K. Ryan, Multiwavelet troubled-cell indicator for discontinuity detection of discontinuous Galerkin schemes, *J. Comput. Phys.* 270 (2014) 138–160.
- [28] M.J. Vuik, J.K. Ryan, Automated parameters for troubled-cell indicators using outlier detection, *SIAM J. Sci. Comput.* 38 (2016) A84–A104.
- [29] G. Fu, C.-W. Shu, A new troubled-cell indicator for discontinuous Galerkin methods for hyperbolic conservation laws, *J. Comput. Phys.* 347 (2017) 305–327.
- [30] J. Zhu, J. Qiu, A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws, *J. Comput. Phys.* 318 (2016) 110–121.
- [31] J. Zhu, J. Qiu, A new type of finite volume WENO schemes for hyperbolic conservation laws, *J. Sci. Comput.* 73 (2017) 1338–1359.
- [32] J. Zhu, J. Qiu, New finite volume weighted essentially nonoscillatory schemes on triangular meshes, *SIAM J. Sci. Comput.* 40 (2018) A903–A928.
- [33] J. Zhu, J. Qiu, A new third order finite volume weighted essentially non-oscillatory scheme on tetrahedral meshes, *J. Comput. Phys.* 349 (2017) 220–232.
- [34] M. Dumbser, W. Boscheri, M. Semplice, G. Russo, Central weighted ENO schemes for hyperbolic conservation laws on fixed and moving unstructured meshes, *SIAM J. Sci. Comput.* 39 (2017) A2564–A2591.
- [35] D. Levy, G. Puppo, G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, *Math. Model. Numer. Anal.* 33 (1999) 547–571.
- [36] D. Levy, G. Puppo, G. Russo, Compact central WENO schemes for multidimensional conservation laws, *SIAM J. Sci. Comput.* 22 (2000) 656–672.
- [37] I. Cravero, G. Puppo, M. Semplice, G. Visconti, Cool WENO schemes, *Comp. Fluid* 169 (2018) 71–86.
- [38] I. Cravero, G. Puppo, M. Semplice, G. Visconti, CWENO: Uniformly accurate reconstructions for balance laws, *Math. Comp.* 87 (2018) 1689–1719.
- [39] B. Cockburn, C.-W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems, *J. Sci. Comput.* 16 (2001) 173–261.
- [40] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, II, *J. Comput. Phys.* 83 (1989) 32–78.
- [41] L. Krivodonova, J. Xin, J.-F. Remacle, N. Chevaugeon, J. Flaherty, Shock detection and limiting with discontinuous Galerkin methods for hyperbolic conservation laws, *Appl. Numer. Math.* 48 (2004) 323–338.
- [42] J. Qiu, C.-W. Shu, A comparison of troubled-cell indicators for Runge-Kutta discontinuous Galerkin methods using weighted essentially nonoscillatory limiters, *SIAM J. Sci. Comput.* 27 (2005) 995–1013.
- [43] M. Semplice, A. Coco, G. Russo, Adaptive mesh refinement for hyperbolic systems based on third-order compact WENO reconstruction, *J. Sci. Comput.* 66 (2016) 692–724.
- [44] O. Kolb, On the full and global accuracy of a compact third order WENO scheme, *SIAM J. Numer. Anal.* 52 (2014) 2335–2355.
- [45] I. Cravero, M. Semplice, On the accuracy of WENO and CWENO reconstructions of third order on nonuniform meshes, *J. Sci. Comput.* 67 (2016) 1219–1246.
- [46] H. Zhu, J. Qiu, An  $h$ -adaptive RKDG method with troubled-cell indicator for two-dimensional hyperbolic conservation laws, *Adv. Comput. Math.* 39 (2013) 445–463.
- [47] V.P. Korobeinikov, *Problems of Point-Blast Theory*, American Institute of Physics, 1991.
- [48] L.I. Sedov, *Similarity and Dimensional Methods in Mechanics*, Academic Press, New York, 1959.
- [49] P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks, *J. Comput. Phys.* 54 (1984) 115–173.