A simple weighted essentially nonoscillatory limiter for Runge-Kutta discontinuous Galerkin methods[1]

Xinghui Zhong[2] and Chi-Wang Shu[3]

Abstract

In this paper, we investigate a simple limiter using weighted essentially non-oscillatory (WENO) methodology for the Runge-Kutta discontinuous Galerkin (RKDG) methods solving conservation laws, with the goal of obtaining a robust and high order limiting procedure to simultaneously achieve uniform high order accuracy and sharp, non-oscillatory shock transitions. The idea of this limiter is to reconstruct the entire polynomial, instead of reconstructing point values or moments in the classical WENO reconstructions. That is, the reconstruction polynomial on the target cell is a convex combination of polynomials on this cell and its neighboring cells and the nonlinear weights of the convex combination follow the classical WENO procedure. The main advantage of this limiter is its simplicity in implementation, especially for multi-dimensional meshes. Numerical results in one and two dimensions are provided to illustrate the behavior of this procedure.

Keywords: discontinuous Galerkin method, WENO limiter

[1] Research supported by DOE grant DE-FG02-08ER25863 and NSF grant DMS-1112700.

[2] Division of Applied Mathematics, Brown University, Providence, RI 02912. E-mail: xzhong@dam.brown.edu

[3] Division of Applied Mathematics, Brown University, Providence, RI 02912. E-mail: shu@dam.brown.edu

1 Introduction

In this paper, we consider the following hyperbolic conservation law

$$\begin{aligned} u_t + f(u)_x &= 0, \\ u(x,0) &= u_0(x), \end{aligned} \tag{1.1}$$

and its two-dimensional version

$$\begin{aligned} u_t + f(u)_x + g(u)_y &= 0, \\ u(x,y,0) &= u_0(x,y), \end{aligned} \tag{1.2}$$

where u, $f(u)$ and $g(u)$ can be either scalars or vectors. We investigate a simple limiter using weighted essentially non-oscillatory (WENO) methodology for the Runge-Kutta discontinuous Galerkin (RKDG) methods, with the goal of obtaining a robust and high order limiting procedure to simultaneously maintain uniform high order accuracy in smooth regions and control spurious numerical oscillations near discontinuities. The idea of this limiter is to reconstruct the entire polynomial based on the polynomials of the DG solution in the target and neighboring cells, instead of reconstructing point values or moments based on cell averages or lower order moments.

DG methods are a class of finite element methods using completely discontinuous piecewise polynomials as basis functions. The first DG method was introduced by Reed and Hill [25] in 1973, to solve the neutron transport equation. The type of DG methods we will discuss in this paper is the Runge-Kutta discontinuous Galerkin (RKDG) methods [7, 6, 5, 4, 8], which use explicit and nonlinearly stable high order Runge-Kutta method for time discretization and the DG method for space discretization. This method has several advantages, such as local conservation, the allowance of arbitrary triangulation, excellent parallel efficiency, the capability in *h*-*p* adaptivity and certain superconvergence properties.

The main difficulty in solving (1.1) and (1.2) is that solutions may contain discontinuities even if the initial conditions are smooth. DG methods can compute solutions to (1.1) and (1.2), which are either smooth or have weak shocks and other discontinuities, without further modification. However, for problems containing strong discontinuities, the scheme will generate significant oscillations and even nonlinear instability. We often need to apply nonlinear limiters to control these oscillations. Many such limiters exist in the literature, such as the minmod type total variation bounded (TVB) limiter [7, 6, 5, 4, 8], the moment-based limiter [2] and the more recent improved moment limiter [3]. 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. Qiu and Shu [23] and Zhu et al. [37] have made such an attempt using WENO methodology [13, 19, 15, 10, 14, 18, 20, 27] as limiters for the DG methods. They use the usual WENO reconstruction based on cell averages of neighboring cells as in [15, 14, 27], to reconstruct the values of the solutions at certain Gaussian quadrature points in the target cells, and then rebuild the solution polynomials from the original cell average and the reconstructed values at the Gaussian quadrature points through a numerical integration for the moments. This limiter needs to use the information from not only the immediate neighboring cells but also neighbors' neighbors, making it complicated to implement in multi-dimensions, especially for unstructured meshes [37, 14, 36]. The effort in [21, 24] attempts to construct Hermite type WENO approximations, which use the information of not only the cell averages but also the lower order moments such as slopes, to reduce the spread of reconstruction stencils. However for higher order methods the information of neighbors' neighbors is still needed.

In this paper, we use the WENO methodology to design a new and simpler limiter for the RKDG methods. We do not reconstruct the point values or moments individually and separately, but attempt to reconstruct the entire polynomial in one shot, using the information only from the target cell and its immediate neighbors. As we will see later, this approach simultaneously removes the problem of negative weights and reduces considerably the complexity of implementation, especially for multi-dimensional meshes including unstructured meshes, although in this paper simulation results on the structured meshes only are reported. Comparing with previous WENO type limiters for DG schemes, the limiter developed in this paper uses most fully the information of the complete polynomials which are already available for DG methods in the target and neighboring cells. Because of this richness of available information, the choice of linear weights are much less restrictive. Essentially, any choice of positive linear weights which add up to one is adequate for accuracy. We refer to [10, 16, 9] and the review paper [30] for a detailed discussion of such practice in choosing linear weights for WENO procedures in finite volume WENO schemes. For our new WENO limiter, following the practice in [9], we give the central cell with a larger linear weight compared to the neighboring cells since in smooth regions the central stencil should provide the most stable reconstruction together with the highest quality in accuracy.

This paper is organized as follows. We first review the RKDG algorithm formulation for the model problem in Section 2. In Section 3, we present the details of our WENO limiting procedure. In Section 4, numerical experiments are provided to verify the accuracy and stability of this approach. Finally, concluding remarks are provided 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) and (1.2).

One-dimensional case. To define the DG method for (1.1) in the one-dimensional case, we consider a partition of the computational domain $[a, b]$ in N cells as follows

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

We denote

$$I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}], \quad x_j = \frac{1}{2}\left(x_{j+\frac{1}{2}} + x_{j-\frac{1}{2}}\right)$$

as the cell and cell center, respectively. We again denote

$$\Delta x_j = x_{j+\frac{1}{2}} - x_{j-\frac{1}{2}}; \quad h = \max_{1 \le j \le N} \Delta x_j.$$

We assume that the mesh is regular, namely there is a constant $c > 0$ independent of h such that

$$\Delta x_j \geq ch, \quad 1 \leq j \leq N.$$

Define the approximation space as

$$V_h^k = \left\{ v : v|_{I_j} \in P^k(I_j);\ 1 \leq j \leq N \right\} \tag{2.1}$$

where $P^k(I_j)$ denotes the set of polynomials of degree up to k defined on the cell I_j. With a slight abuse of notation, the semi-discrete DG method for solving (1.1) is defined as follows: find the unique function $u = u(t) \in V_h^k$ such that, for $j = 1, \cdots, N$,

$$\int_{I_j} u_t v\, dx - \int_{I_j} f(u) v_x\, dx + \hat{f}_{j+\frac{1}{2}} v(x^-_{j+\frac{1}{2}}) - \hat{f}_{j-\frac{1}{2}} v(x^+_{j-\frac{1}{2}}) = 0 \tag{2.2}$$

holds for all test functions $v \in V_h^k$. Here and below u^-, u^+ denote the left and right limits of the function u at the cell interface, respectively. $\hat{f}_{j+\frac{1}{2}}$ is the the so-called monotone numerical fluxes (approximate or exact Riemann solvers in the system case).

Two-dimensional case. As before, we assume a rectangular mesh to cover the computational domain $[a, b]^2$, consisting of cells

$$I_{i,j} = \{(x, y) :\ x_{i-\frac{1}{2}} \leq x \leq x_{i+\frac{1}{2}},\ y_{j-\frac{1}{2}} \leq y \leq y_{j+\frac{1}{2}}\}$$

for $1 \leq i \leq N_x$ and $1 \leq j \leq N_y$. We also assume the mesh is regular and define a finite element space consisting of piecewise polynomials

$$W_h^k = \left\{ v : v|_{I_{i,j}} \in P^k(I_{i,j}); 1 \leq i \leq N_x,\ 1 \leq j \leq N_y \right\}$$

where $P^k(I_{i,j})$ denotes the set of polynomials of degree up to k defined on the cell $I_{i,j}$. The semi-discrete DG method for solving (1.2) is defined as follows: find the unique function $u \in W_h^k$ such that, for all test functions $v \in W_h^k$ and all $1 \leq i \leq N_x$ and $1 \leq j \leq N_y$, we have

$$\begin{aligned}
&\int_{I_{i,j}} u_t v\, dx\, dy = \\
&\int_{I_{i,j}} f(u) v_x\, dx\, dy - \int_{I_j} \hat{f}_{i+\frac{1}{2}}(y) v(x^-_{i+\frac{1}{2}}, y)\, dy + \int_{I_j} \hat{f}_{i-\frac{1}{2}}(y) v(x^+_{i-\frac{1}{2}}, y)\, dy \\
&+ \int_{I_{i,j}} g(u) v_y\, dx\, dy - \int_{I_i} \hat{g}_{j+\frac{1}{2}}(x) v(x, y^-_{j+\frac{1}{2}})\, dx + \int_{I_i} \hat{g}_{i-\frac{1}{2}}(x) v(x, y^+_{j-\frac{1}{2}})\, dx.
\end{aligned} \tag{2.3}$$

The semi-discrete scheme (2.2) and (2.3) can be written as

$$u_t = L(u),$$

where $L(u)$ is the spatial discretization operator. To discretize the time variable, we use the following total variation diminishing (TVD) third order Runge-Kutta method [31]:

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}). \end{aligned} \tag{2.4}$$

Other TVD, or strong stability preserving (SSP) time discretizations [11] can of course also be used.

For simplicity, we consider the forward Euler time discretization of the semi-discrete scheme (2.2). Starting from a solution $u^n \in V_h^k$ at time level n (for the initial condition, u^0 is taken as the L^2 projection of the analytical initial condition into V_h^k). We would like to "limit" it to obtain a new function $u^{n,new}$ before advancing it to next time level. That is: find $u^{n+1} \in V_h^k$, such that, for $j = 1, \cdots, N$,

$$\int_{I_j} \frac{u^{n+1} - u^{n,new}}{\Delta t} v\, dx - \int_{I_j} f(u^{n,new}) v_x\, dx + \hat{f}^{n,new}_{j+\frac{1}{2}} v(x^-_{j+\frac{1}{2}}) - \hat{f}^{n,new}_{j-\frac{1}{2}} v(x^+_{j-\frac{1}{2}}) = 0 \tag{2.5}$$

holds for all test functions $v \in V_h^k$. The limiting procedure to go from u^n to $u^{n,new}$ will be discussed in the following section.

3 A new WENO limiter

In this section, we present the details of our new WENO limiting procedure for the RKDG methods. As in [23], we also adopt the following framework:

1. Identify the troubled cells, namely, those cells which might need the limiting procedure.

2. Replace the solution polynomials in the troubled cells with reconstructed polynomials, which keep the original cell averages, maintain the original high order of accuracy, but are less oscillatory.

3.1 Identification of the troubled cells

In this subsection, we discuss the identification of the troubled cells. This part is not the emphasis of our paper and we do not attempt to compare the pros and cons of various troubled cell identification procedures. We will simply use the TVB minmod limiter [7, 6, 5, 4, 8] to identify troubled cells. We emphasize that the goal of our limiter is to be insensitive to the troubled-cell indicators. That is, if more troubled cells are identified than they actually exist, the computational cost of the algorithm will increase but the original high order accuracy should be maintained.

We first provide an overview of the minmod limiter [6] in the one-dimensional scalar case. Denote the cell average of the solution u as

$$\bar{u}_j = \frac{1}{\Delta x_j} \int_{I_j} u\, dx \tag{3.1}$$

and further denote

$$\tilde{u}_j = u^-_{j+\frac{1}{2}} - \bar{u}_j, \quad \tilde{\tilde{u}}_j = \bar{u}_j - u^+_{j-\frac{1}{2}}. \tag{3.2}$$

$\tilde{u}_j$ and $\tilde{\tilde{u}}_j$ are modified either by the usual minmod limiter [12],

$$\tilde{u}^{(mod)}_j = m\left(\tilde{u}_j, \Delta_+\bar{u}_j, \Delta_-\bar{u}_j\right), \quad \tilde{\tilde{u}}^{(mod)}_j = m\left(\tilde{\tilde{u}}_j, \Delta_+\bar{u}_j, \Delta_-\bar{u}_j\right), \tag{3.3}$$

where

$$\Delta_+\bar{u}_j = \bar{u}_{j+1} - \bar{u}_j, \quad \Delta_-\bar{u}_j = \bar{u}_j - \bar{u}_{j-1},$$

with the minmod function m defined by

$$m(a_1, \cdots, a_l) = \begin{cases} s \min_{1\le j\le l} |a_j| & \text{if } s = sign(a_1) = \cdots = sign(a_l), \\ 0, & \text{otherwise,} \end{cases} \tag{3.4}$$

or by the TVB modified minmod function [28]

$$\tilde{m}(a_1, \cdots, a_l) = \begin{cases} a_1 & \text{if } |a_1| \le Mh^2, \\ m(a_1, \cdots, a_l), & \text{otherwise,} \end{cases} \tag{3.5}$$

where the TVB parameter M has to be chosen adequately depending on the solution of the problem. For more details, see [6].

To detect the troubled cells using the limiter described above, we declare that whenever one of the minmod functions (3.3) gets enacted (returns other than the first argument), this cell is marked as a troubled cell and is subject to WENO reconstructions. Of course, if too few cells are identified as troubled cells, oscillations and possible instability may not be avoided. If too many cells are identified as troubled cells, the computational cost associated with the second step will increase. Therefore, troubled-cell indicator is a very important issue for WENO limiters. However our main concern in this paper is on how to design the new WENO limiter. We refer the readers to [22] about the comparison among different troubled-cell indicators.

We have given the details of identifying troubled cells using the TVB minmod limiters for the one-dimensional scalar case. For two-dimensional scalar case on rectangular mesh, we perform the TVB minmod limiter in the x-direction and the y-direction separately as in the one-dimensional case. For more details and unstructured meshes, see, e.g. [4]. For one-dimensional and two-dimensional systems, we use the characteristic-wise TVB minmod limiters defined in [5] and [8], respectively.

3.2 Reconstruction of the new polynomials in the troubled cells using a WENO limiter: Scalar case

In this subsection, we present the details of the reconstruction procedure for the new polynomials in the troubled cells using our WENO limiter for scalar conservation laws. The idea of this WENO limiter is to reconstruct a new polynomial on the troubled cell I_j which is a convex combination of polynomials on this cell and its immediate neighboring cells, with necessary adjustments to keep the original cell average on the target cell. The construction of the nonlinear weights in the convex combination coefficients follows the classical WENO procedure.

We start with the one-dimensional scalar case. Assume that the cell I_j is a troubled cell. Denote the DG solution polynomial of u on the cells I_{j-1}, I_j, I_{j+1} as $p_0(x)$, $p_1(x)$, $p_2(x)$, respectively. In order to make sure that the reconstructed polynomial maintains the original cell average of p_1 in the target cell I_j, we make the following modifications:

$$\tilde{p}_0(x) = p_0(x) - \bar{\bar{p}}_0 + \bar{\bar{p}}_1, \qquad \tilde{p}_2(x) = p_2(x) - \bar{\bar{p}}_2 + \bar{\bar{p}}_1, \tag{3.6}$$

where

$$\bar{\bar{p}}_0 = \frac{1}{\Delta x_j}\int_{I_j} p_0(x)\,dx, \quad \bar{\bar{p}}_1 = \frac{1}{\Delta x_j}\int_{I_j} p_1(x)\,dx, \quad \bar{\bar{p}}_2 = \frac{1}{\Delta x_j}\int_{I_j} p_2(x)\,dx.$$

The final nonlinear WENO reconstruction polynomial $p_1^{new}(x)$ is now defined by a convex combination of these modified polynomials:

$$p_1^{new}(x) = \omega_0\tilde{p}_0(x) + \omega_1 p_1(x) + \omega_2\tilde{p}_2(x). \tag{3.7}$$

From (3.6) and (3.7), it is easy to prove that p_1^{new} has the same cell average and order of accuracy as p_1 if the weights satisfy $\omega_0 + \omega_1 + \omega_2 = 1$.

Following [15, 14, 1, 16], the normalized nonlinear weights are defined as

$$\omega_l = \frac{\bar{\omega}_l}{\sum_s \bar{w}_s}, \tag{3.8}$$

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

$$\bar{\omega}_l = \frac{\gamma_l}{(\varepsilon + \beta_l)^r}. \tag{3.9}$$

We use $\varepsilon = 10^{-6}$ and $r = 2$ in all computations in this paper. As in [15, 1], we use the following smoothness indicator for the one-dimensional case:

$$\beta_l = \sum_{s=1}^{k} \int_{I_j} \Delta x_j^{2s-1} \left(\frac{\partial^s}{\partial x^s} p_l(x)\right)^2 dx. \tag{3.10}$$

For the two-dimensional case and more details about this smoothness indicator, we refer to [15, 1, 29]. Notice that, because we have used the complete information of the three polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$ in the three cells I_{j-1}, I_j, I_{j+1}, we do not have extra requirements on the linear weights in order to maintain the original high order accuracy. The linear weights can be chosen to be any set of positive numbers adding up to one. 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, i.e.

$$\gamma_1 \gg \gamma_0, \qquad \gamma_1 \gg \gamma_2.$$

Lower values of the ratio $\frac{\gamma_1}{\gamma_0}$, $\frac{\gamma_1}{\gamma_2}$ yield better results on discontinuities while larger values are usually better for smooth solutions. In our numerical tests we take

$$\gamma_0 = 0.001,\ \gamma_1 = 0.998,\ \gamma_2 = 0.001, \tag{3.11}$$

which can maintain the original high order in smooth regions and can keep essentially non-oscillatory shock transitions in all our numerical examples.

We summarize the WENO limiting procedure for one-dimensional scalar conservation laws as follows. Assuming that DG solution at time level n is u^n, for $j = 1, \cdots, N$,

1. Use the minmod limiter described in Subsection 3.1 to detect whether I_j is a troubled cell or not.

2. If I_j is not a troubled cell, then $u^{n,new}|_{I_j} = u^n|_{I_j}$.

   If I_j is a troubled cell, then

   (a) Denote u^n on the cells I_{j-1}, I_j, I_{j+1} as $p_0(x)$, $p_1(x)$, $p_2(x)$, respectively and modify $p_0(x)$, $p_2(x)$ to $\tilde{p}_1(x)$, $\tilde{p}_2(x)$ using (3.6).

   (b) Determine the linear weights by (3.11).

   (c) Compute the smoothness indicators β_l using (3.10) for $l = 0, 1, 2$.

   (d) Compute the normalized nonlinear weights w_l using (3.8) and (3.9) for $l = 0, 1, 2$.

   (e) The reconstruction polynomial is given by (3.7), i.e. $u^{n,new}|_{I_j} = \omega_0\tilde{p}_0(x) + \omega_1 p_1(x) + \omega_2\tilde{p}_2(x)$.

For the two-dimensional scalar case with rectangular meshes considered in this paper, the limiting procedure is similar as described above. The WENO reconstruction polynomial on the troubled cell $I_{i,j}$ is a convex combination of the polynomials on this cell and its four neighboring cells $\{I_{i,j-1},\, I_{i,j+1},\, I_{i-1,j},\, I_{i+1,j}\}$ suitably modified to maintain the original cell average on the target cell $I_{i,j}$. For our numerical tests, we put a larger linear weight 0.996 on the troubled cell $I_{i,j}$ and the neighboring cells $\{I_{i,j-1},\, I_{i,j+1},\, I_{i-1,j},\, I_{i+1,j}\}$ get the smaller linear weight 0.001. The nonlinear weights of the convex combination follow the classical WENO procedure, with smoothness indicators still computed as the sums of L^2-norm squares of all the derivatives of the respective polynomials.

3.3 WENO limiting procedure for systems

In this subsection, we present the details of the WENO limiting procedure for systems.

One-dimensional systems. Consider equation (1.1) where u and f are vectors with m components. In order to achieve better non-oscillatory qualities, the WENO reconstruction limiter is applied with a local characteristic field decomposition, see, e.g. [29] for more details.

Denote the Jacobian matrix by $A_j = \frac{\partial f}{\partial u}\big|_{\bar{u}_j}$. Denote the left and right eigenvectors of A_j by $l_j^{(p)}$, $r_j^{(p)}$, $p = 1, \cdots, m$, 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.

$$R(\bar{u}_j) = \left(r_j^{(1)}, r_j^{(2)}, \cdots, r_j^{(m)}\right). \tag{3.12}$$

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

$$R^{-1}(\bar{u}_j) = \left(l_j^{(1)}, l_j^{(2)}, \cdots, l_j^{(m)}\right)^T. \tag{3.13}$$

The WENO limiting procedure for one-dimensional system case is given as follows. Assuming that the DG solution at time level n is u^n (for simplicity, we will use the notation u for the following description), for $j = 1, \cdots, N$,

1. Compute $R = R(\bar{u}_j)$ and R^{-1} as defined in (3.12) and (3.13).

2. Compute $\Delta_+ \bar{v}_j = R^{-1}(\bar{u}_{j+1} - \bar{u}_j)$, $\Delta_- \bar{v}_j = R^{-1}(\bar{u}_j - \bar{u}_{j-1})$, $\tilde{v}_j = R^{-1}\left(u^-_{j+\frac{1}{2}} - \bar{u}_j\right)$ and $\tilde{\tilde{v}}_j = R^{-1}\left(\bar{u}_j - u^+_{j-\frac{1}{2}}\right)$ respectively.

3. Compute $\tilde{v}_j^{(mod)} = \tilde{m}(\tilde{v}_j, \Delta_+\bar{v}_j, \Delta_-\bar{v}_j)$ and $\tilde{\tilde{v}}_j^{(mod)} = \tilde{m}(\tilde{\tilde{v}}_j, \Delta_+\bar{v}_j, \Delta_-\bar{v}_j)$ with the modified minmod function $\tilde{m}$ define in (3.5) for each component of the vectors.

4. If

$$\tilde{v}_j^{(mod)} = \tilde{v}_j \quad \& \quad \tilde{\tilde{v}}_j^{(mod)} = \tilde{\tilde{v}}_j, \tag{3.14}$$

then I_j is not a troubled cell and $u^{new}|_{I_j} = u|_{I_j}$. Otherwise,

(i) Denote the DG polynomial u on the cells I_{j+l} as u_{j+l}. Project u_{j+l} into the characteristic fields $v_{j+l} = R^{-1}u_{j+l}$, for $l = -1, 0, 1$. Note that v_{j+l} is a m-component vector and each component is a polynomial.

(ii) Perform Steps (a), (b), (c), (d), (e) as in the one-dimensional scalar case for each component of v_j which is a troubled component, namely the corresponding component makes the condition (3.14) not satisfied. The updated vector v_j is denoted as v_j^{new}.

(iii) $u^{new}|_{I_j} = Rv_j^{new}$.

Two-dimensional systems. Considering the following two-dimensional system (1.2) where u, $f(u)$ and $g(u)$ are vectors with m components. The DG scheme with Euler forward time discretization for solving (1.2) on a rectangular mesh is

$$\begin{aligned}
&\int_{I_{i,j}} \frac{u^{n+1} - u}{\Delta t} v \, dx \, dy = \\
&\int_{I_{i,j}} f(u)v_x \, dx \, dy - \int_{I_j} \hat{f}_{i+\frac{1}{2}}(y)v(x^-_{i+\frac{1}{2}}, y) \, dy + \int_{I_j} \hat{f}_{i-\frac{1}{2}}(y)v(x^+_{i-\frac{1}{2}}, y) \, dy \\
&+ \int_{I_{i,j}} g(u)v_y \, dx \, dy - \int_{I_i} \hat{g}_{j+\frac{1}{2}}(x)v(x, y^-_{j+\frac{1}{2}}) \, dx + \int_{I_i} \hat{g}_{i-\frac{1}{2}}(x)v(x, y^+_{j-\frac{1}{2}}) \, dx,
\end{aligned} \tag{3.15}$$

where $u = u^n$ and $\hat{f}$ and $\hat{g}$ are monotone numerical fluxes.

For two-dimensional systems, we need to be more careful when using the characteristic-wise WENO limiting procedure, since there are two Jacobian matrices corresponding to fluxes in the x and y directions respectively, and therefore two sets of eigenspaces. We perform the characteristic-wise WENO reconstruction in the x-direction and y-direction separately.

Assume that $I_{i,j}$ is a troubled cell,

1. In the x-direction, we choose the polynomials on the cells $I_{i-1,j}$, $I_{i,j}$, $I_{i+1,j}$ to reconstruct a new polynomial $u_{i,j}^{x,new}$ using the characteristic-wise WENO limiting procedure with the Jacobian matrix $\frac{\partial f}{\partial u}$ as in the one-dimensional system case.

2. Similarly, in the y-direction, we choose the polynomials on the cells $I_{i,j-1}$, $I_{i,j}$, $I_{i,j+1}$ to reconstruct a new polynomial $u_{i,j}^{y,new}$ using the characteristic-wise WENO limiting procedure with the Jacobian matrix $\frac{\partial g}{\partial u}$.

3. $u^{new}|_{I_{i,j}} = \frac{1}{2}\left(u_{i,j}^{x,new} \ + u_{i,j}^{y,new}\right)$.

After limiting DG solution, we advance it to the next time level for the Euler forward time discretization by

$$\int_{I_{i,j}} \frac{u^{n+1} - u^{new}}{\Delta t} v \, dx \, dy =$$

$$\int_{I_{i,j}} f(u^{x,new}) v_x \, dx \, dy - \int_{I_j} \hat{f}(u^{x,new})_{i+\frac{1}{2}}(y) v(x^-_{i+\frac{1}{2}}, y) \, dy + \int_{I_j} \hat{f}(u^{x,new})_{i-\frac{1}{2}}(y) v(x^+_{i-\frac{1}{2}}, y) \, dy$$

$$+ \int_{I_{i,j}} g(u^{y,new}) v_y \, dx \, dy - \int_{I_i} \hat{g}(u^{y,new})_{j+\frac{1}{2}}(x) v(x, y^-_{j+\frac{1}{2}}) \, dx + \int_{I_i} \hat{g}(u^{y,new})_{j-\frac{1}{2}}(x) v(x, y^+_{j-\frac{1}{2}}) \, dx.$$

The Runge-Kutta time discretization is just a convex combination of such Euler forward steps.

4 Numerical experiments

In this section, we provide numerical experiments to demonstrate the performance of the WENO limiter for RKDG methods described in Section 3. Even though the advantage of simplicity is most evident for higher-dimensional unstructured meshes, we show in this paper only results for one and two dimensional structured meshes.

For all the computational results, we use the local Lax-Friedrichs flux. For the one-dimensional examples, the CFL number is set to be 0.3 for the P^1 case, 0.15 for the P^2 case and 0.1 for the P^3 case (for the P^3 case Δt is further reduced in the accuracy test).

We have used both uniform and nonuniform meshes in the numerical experiments, obtaining similar results. The nonuniform meshes are obtained from a 20% random perturbation of each node of the uniform mesh. Take the one-dimensional case as an example. The cell boundary point is now $x_{j+\frac{1}{2}} + 20\%(r_{j+\frac{1}{2}} - 0.5)\Delta x$, where $x_{j+\frac{1}{2}}$ and Δx are taking values from the uniform mesh and $r_{j+\frac{1}{2}}$ is a random number from the uniform distribution over the range $(0, 1)$. We will only show results with accuracy tests on nonuniform meshes as representative tests.

For all the accuracy tests (Tables 4.1-4.8), in order to see the effect of the WENO limiter on the accuracy of the RKDG 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. For Figures 4.1-4.12, the solid lines are for the exact solutions or grid converged solutions, and the symbols "+" are for the numerical solutions (just one point per cell is plotted).

4.1 Scalar conservation laws

Example 4.1 We consider the Burgers equation:

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

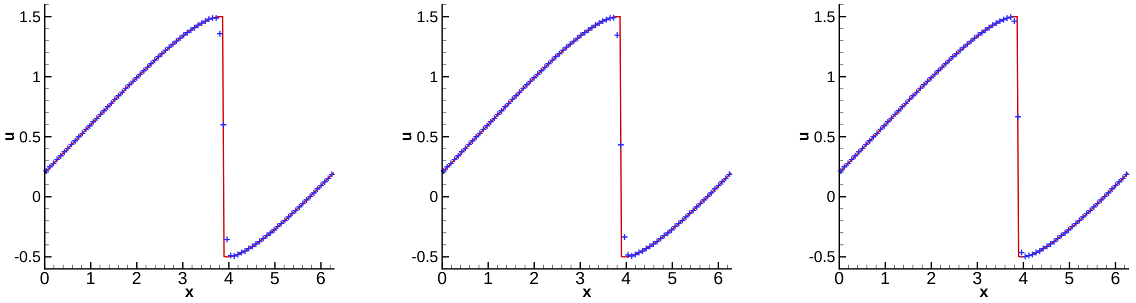
with the initial condition $u(x, 0) = 0.5 + \sin x$ and periodic boundary conditions. The exact solution is smooth up to $t = 1$, then it develops a moving shock which also acts with the rarefaction waves. We can get the exact solution by Newton iteration. For details, see [13]. The errors at t = 0.5 when the solution is smooth are listed in Tables 4.1 and 4.2. We can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the errors of the original RKDG method. In Figure 4.1, we show the RKDG solutions with a WENO limiter at $t = 1.5$ using 80 cells. We can see that the schemes of all orders perform well in capturing this discontinuity without oscillations.

Table 4.1: 1D Burgers Equation. $t = 0.5$. Uniform mesh with N cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	3.81E-03		3.82E-02		7.65E-03		6.86E-02	
	40	9.46E-04	2.01	1.03E-02	1.90	1.81E-03	2.08	2.01E-02	1.77
	80	2.35E-04	2.01	2.65E-03	1.95	3.06E-04	2.57	3.10E-03	2.70
	160	5.89E-05	2.00	6.73E-04	1.98	5.93E-05	2.37	6.73E-04	2.20
	320	1.47E-05	2.00	1.70E-04	1.98	1.47E-05	2.01	1.70E-04	1.98
P^2	20	2.73E-04		5.07E-03		2.68E-04		5.08E-03	
	40	4.22E-05	2.69	8.96E-04	2.50	4.17E-05	2.68	8.96E-04	2.50
	80	6.17E-06	2.77	1.60E-04	2.48	6.17E-06	2.76	1.60E-04	2.48
	160	8.86E-07	2.80	2.55E-05	2.65	8.92E-07	2.79	2.55E-05	2.65
	320	1.25E-07	2.82	3.79E-06	2.75	1.27E-07	2.81	3.79E-06	2.75
P^3	20	1.85E-05		3.64E-04		2.17E-05		3.66E-04	
	40	1.00E-06	4.20	3.69E-05	3.30	1.03E-06	4.40	3.69E-05	3.31
	80	6.08E-08	4.05	2.22E-06	4.06	6.17E-08	4.06	2.22E-06	4.06
	160	3.76E-09	4.02	1.44E-07	3.95	3.83E-09	4.01	1.44E-07	3.95
	320	2.33E-10	4.01	9.19E-09	3.97	2.45E-10	3.97	9.19E-09	3.97

Table 4.2: 1D Burgers Equation. $t = 0.5$. Nonuniform mesh with N cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20	3.84E-03		4.01E-02		6.35E-03		7.39E-02	
	40	9.85E-04	1.96	1.28E-02	1.65	1.93E-03	1.72	2.07E-02	1.84
	80	2.44E-04	2.02	3.09E-03	2.05	3.16E-04	2.61	3.16E-03	2.71
	160	6.14E-05	1.99	8.28E-04	1.90	6.22E-05	2.35	8.28E-04	1.93
	320	1.53E-05	2.00	2.36E-04	1.81	1.53E-05	2.02	2.36E-04	1.81
P^2	20	2.86E-04		4.87E-03		2.79E-04		4.88E-03	
	40	4.43E-05	2.69	7.68E-04	2.66	4.38E-05	2.67	7.68E-04	2.67
	80	6.37E-06	2.80	1.61E-04	2.25	6.37E-06	2.78	1.61E-04	2.25
	160	9.08E-07	2.81	2.83E-05	2.51	9.16E-07	2.80	2.83E-05	2.51
	320	1.30E-07	2.81	4.12E-06	2.78	1.32E-07	2.80	4.12E-06	2.78
P^3	20	1.85E-05		3.91E-04		1.85E-05		3.95E-04	
	40	1.01E-06	4.20	3.20E-05	3.61	1.03E-06	4.17	3.20E-05	3.62
	80	6.42E-08	3.97	2.79E-06	3.52	6.55E-08	3.97	2.79E-06	3.52
	160	4.40E-09	3.87	2.80E-07	3.32	4.42E-09	3.89	2.80E-07	3.32
	320	2.72E-10	4.02	1.66E-08	4.07	2.77E-10	4.00	1.66E-08	4.07



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.1: Burgers equation at $t = 1.5$ with $N = 80$ cells.

Example 4.2 A two-dimensional version of Example 4.1

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

is tested with initial condition $u(x, y, 0) = 0.5 + \sin(x + y)$ and periodic boundary conditions. The exact solution is one-dimensional depending only on $\xi = x + y$; however, our meshes are rectangular in the (x, y) coordinates, and thus this example is a truly two-dimensional test problem. As in Example 4.1, we collect the L_1 and L_∞ errors at $t = 0.25$ (smooth solution) in Tables 4.3 and 4.4. At $t = 0.5$, a shock begins to form. We compute the solutions of the RKDG methods using P^k polynomials with a WENO limiter with 80×80 meshes until $t = 0.75$ and plot the solution surfaces in Figure 4.2 and the solution cuts in the diagonal cells in Figure 4.3. Again, we can see that the WENO limiter obtains uniform high order accuracy and sharp, non-oscillatory shock transitions for the RKDG methods.

Example 4.3 Our last scalar example is the Buckley-Leverett problem that is governed by the equation

$$u_t + \left(\frac{4u^2}{4u^2 + (1-u)^2}\right)_x = 0, \tag{4.3}$$

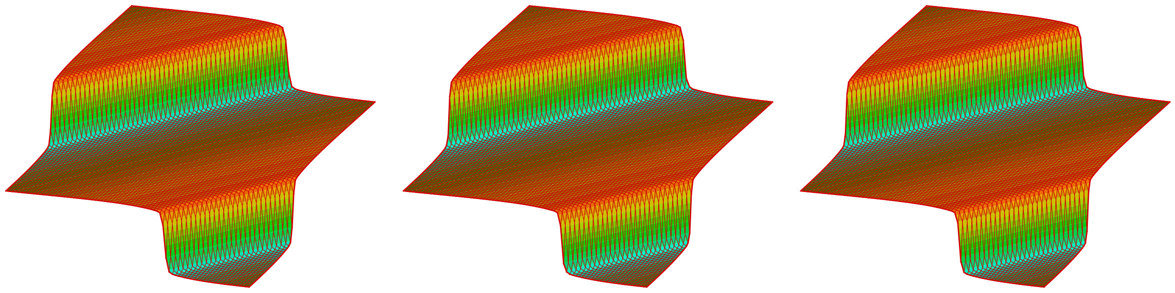
with the initial condition $u = 1$ for $-\frac{1}{2} \le x \le 0$ and $u = 0$ elsewhere. The exact solution is a shock-rarefaction-contact discontinuity mixture. The solution is computed up to $t = 0.4$. Figure 4.4 shows the numerical solutions of RKDG methods with a WENO limiter using $N = 80$ cells. Again, all schemes perform similarly well for this example.

Table 4.3: 2D Burgers equation at $t = 0.25$. Uniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	$N \times N$	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20×20	7.56E-03		1.22E-01		1.11E-02		2.07E-01	
	40×40	1.92E-03	1.98	3.55E-02	1.78	3.09E-03	1.85	6.57E-02	1.65
	80×80	4.79E-04	2.00	9.36E-03	1.92	6.60E-04	2.23	9.36E-03	2.81
	160×160	1.20E-04	2.00	2.39E-03	1.97	1.23E-04	2.43	2.39E-03	1.97
	320×320	2.99E-05	2.00	6.04E-04	1.98	2.99E-05	2.04	6.04E-04	1.98
P^2	20×20	8.62E-04		4.37E-02		8.62E-04		4.37E-02	
	40×40	1.16E-04	2.89	6.08E-03	2.84	1.16E-04	2.89	6.08E-03	2.84
	80×80	1.50E-05	2.96	9.60E-04	2.66	1.50E-05	2.96	9.60E-04	2.66
	160×160	1.90E-06	2.98	1.35E-04	2.83	1.90E-06	2.98	1.35E-04	2.83
	320×320	2.43E-07	2.97	1.83E-05	2.88	2.44E-07	2.96	1.83E-05	2.88

Table 4.4: 2D Burgers equation at $t = 0.25$. Nonuniform mesh with $N \times N$ cells

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	$N \times N$	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20×20	7.70E-03		1.38E-01		1.24E-02		2.82E-01	
	40×40	1.99E-03	1.96	4.62E-02	1.58	3.48E-03	1.84	8.75E-02	1.69
	80×80	4.93E-04	2.01	1.21E-02	1.93	6.86E-04	2.35	1.21E-02	2.85
	160×160	1.24E-04	2.00	3.38E-03	1.84	1.28E-04	2.42	3.38E-03	1.84
	320×320	3.10E-05	1.99	8.50E-04	1.99	3.10E-05	2.05	8.50E-04	1.99
P^2	20×20	8.95E-04		5.48E-02		8.92E-04		5.48E-02	
	40×40	1.21E-04	2.89	8.18E-03	2.74	1.21E-04	2.88	8.18E-03	2.74
	80×80	1.55E-05	2.96	1.37E-03	2.57	1.57E-05	2.95	1.37E-03	2.57
	160×160	1.95E-06	3.00	1.98E-04	2.79	1.99E-06	2.97	1.98E-04	2.79
	320×320	2.50E-07	2.96	2.77E-05	2.84	2.60E-07	2.94	2.77E-05	2.84

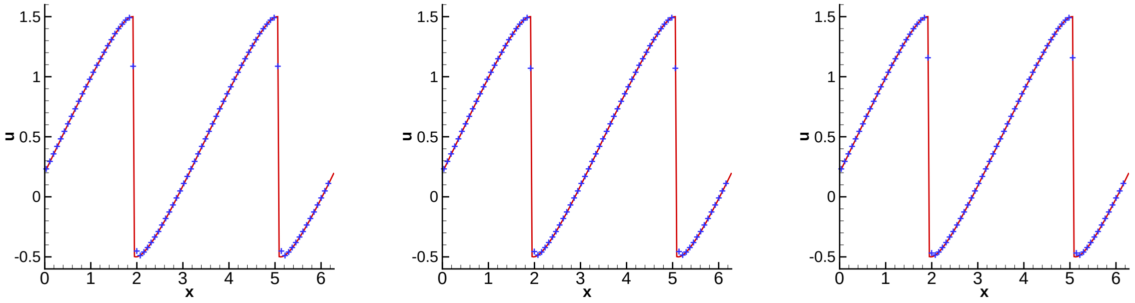


(a) $k = 1$

(b) $k = 2$

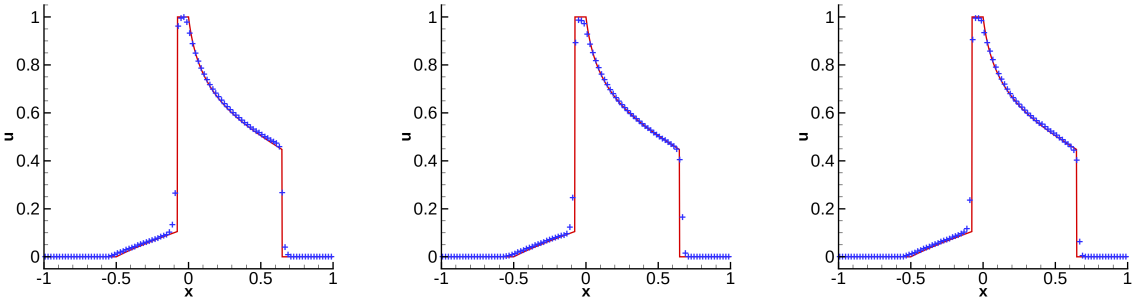
(c) $k = 3$

Figure 4.2: 2D Burgers solution at $t = 0.75$ with 80×80 cells.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.3: 2D Burgers solution that cuts along the diagonal at $t = 0.75$ with 80×80 cells.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.4: Buckley-Leverett problem at $t = 0.4$ with 80 cells.

4.2 Euler system in one dimension

The 1D Euler system is given by

$$\boldsymbol{u}_t + \boldsymbol{f}(\boldsymbol{u})_x = 0, \tag{4.4}$$

where $\boldsymbol{u} = (\rho,\, \rho v,\, E)^T$ and $\boldsymbol{f}(\boldsymbol{u}) = (\rho v, \rho v^2 + p, v(E+p))^T$. Here ρ is the density, v is the velocity, E is the total energy, and p is the pressure, with

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho v^2\right); \tag{4.5}$$

$\gamma = 1.4$ is used in the computation. For details of the Jacobian, its eigenvalues, eigenvectors, etc., see [13, 26]. We consider the following typical examples.

Example 4.4 We consider here two well-known problems of the Euler equation (4.4) which have the following Riemann type initial conditions:

$$\boldsymbol{u}(x,0) = \begin{cases} \boldsymbol{u}_L, & x < 0, \\ \boldsymbol{u}_R, & x > 0. \end{cases} \tag{4.6}$$

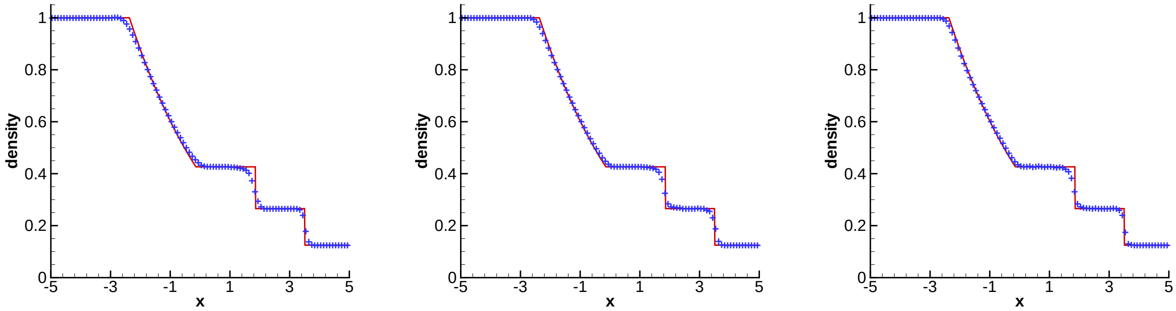
The first one is the Sod problem [33]. The initial data are

$$(\rho_L,\, v_L,\, p_L) = (1,\, 0,\, 1); \quad (\rho_R,\, v_R,\, p_R) = (0.125,\, 0,\, 0.1). \tag{4.7}$$

The second one is the Lax problem [17], with the initial data

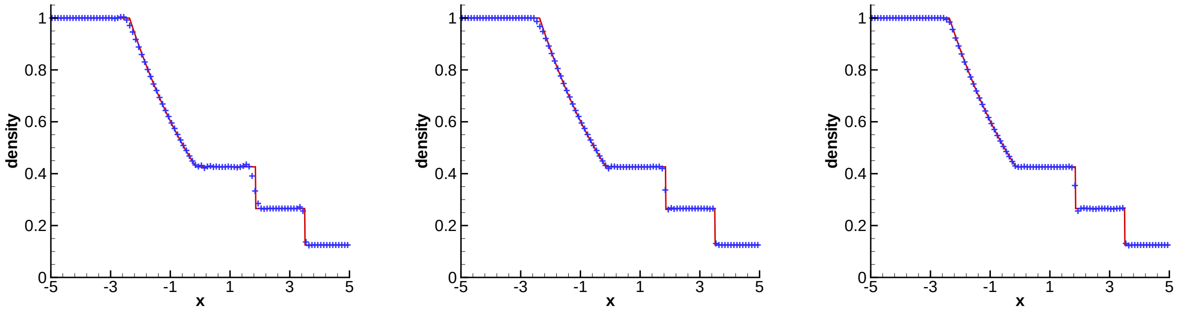
$$(\rho_L,\, v_L,\, p_L) = (0.445,\, 0.698,\, 3.528); \quad (\rho_R,\, v_R,\, p_R) = (0.5,\, 0,\, 0.571). \tag{4.8}$$

The numerical results with the WENO limiter are in Figures 4.5-4.8. To save space, we only show the plots for density. The figures for velocity and pressure are not shown. For Figures 4.5 and 4.7, we use the TVB constant $M = 0.01$. We can see that there are no oscillations near the discontinuities, however we also observe rather severe smearing, especially for the contact discontinuity, due to this strong limiter. For the Sod problem, relaxing the limiting by taking $M = 30$ improves the smearing at the price of slight over- and under-shoots, comparing Figure 4.6 with 4.5. The Lax problem is more sensitive to the parameter M, we relax the limiting by taking $M = 7$, 30, 50 for $k = 1, k = 2, k = 3$ respectively, see Figure 4.8.



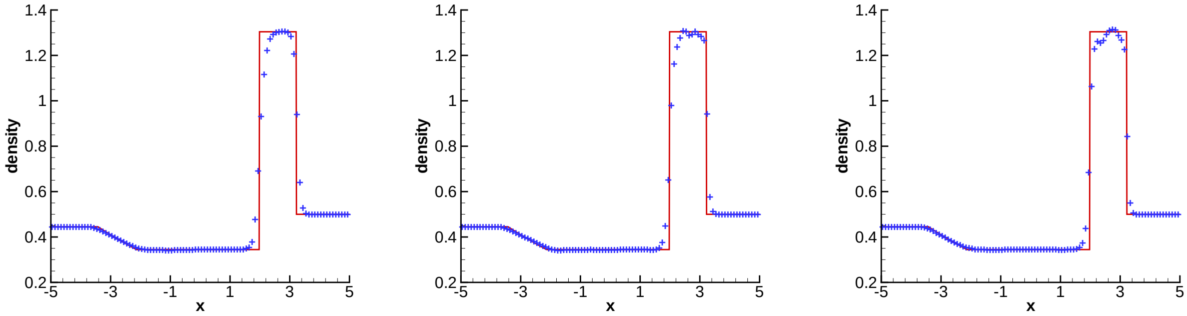
(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.5: Sod problem. $t = 2$. $M = 0.01$. $N = 100$. Density.



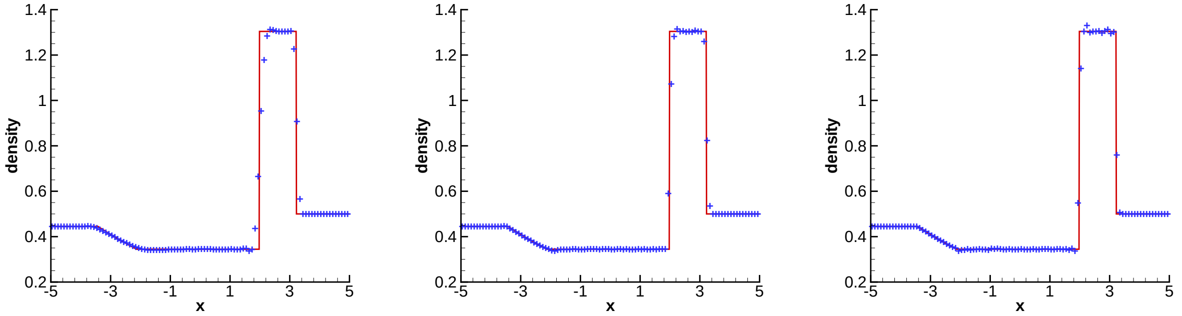
(a) $k=1$ (b) $k=2$ (c) $k=3$

Figure 4.6: Sod problem. $t=2$. $M=30$. $N=100$. Density.



(a) $k=1$ (b) $k=2$ (c) $k=3$

Figure 4.7: Lax problem. $t=1.3$. $M=0.01$. $N=100$. Density.



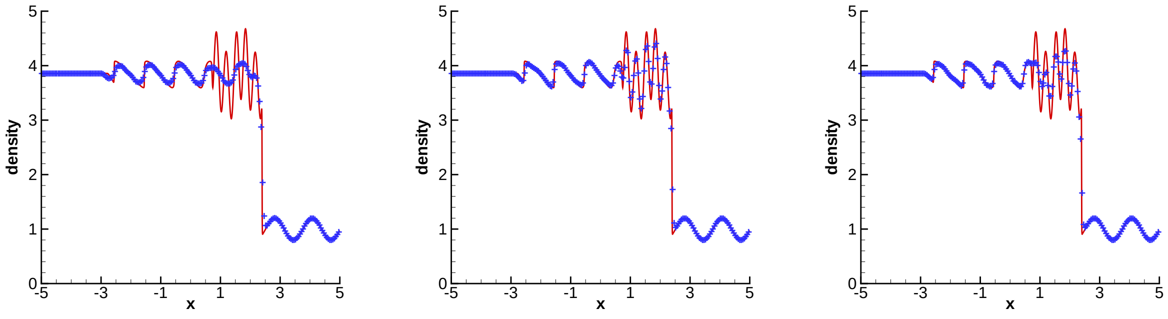
(a) $k=1, M=7$ (b) $k=2, M=30$ (c) $k=3, M=50$

Figure 4.8: Lax problem. $t=1.3$. $N=100$. Density.

Example 4.5 To demonstrate the advantage of higher order methods, we use the Euler equation (4.4) with initial condition

$$\begin{array}{ll} (\rho_L,\, v_L,\, p_L) = (3.857143,\, 2.629369,\, 10.333333), & \text{when } x < -4, \\ (\rho_R,\, v_R,\, p_R) = (1 + 0.2\sin(5x),\, 0,\, 1), & \text{when } x \geq -4. \end{array} \tag{4.9}$$

This example was used in [32]. It describes the interaction of a Mach 3 shock with a density wave. 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. Our results are shown in Figures 4.9-4.10. The solid lines are the referenced "exact" solution, which is a converged solution computed by the fifth order finite difference WENO scheme [15] with 2000 grid points. Again, we explore the effect of the TVB constant M in the minmod limiter to identify troubled cells. If M is adjusted adequately, schemes of all orders can perform extremely well, see Figure 4.10.



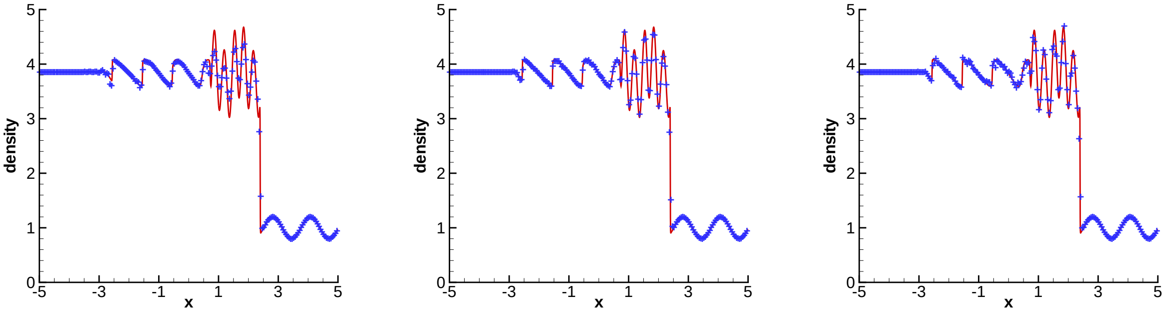
(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.9: The shock density wave interaction problem. $t = 1.8$. $M = 0.01$. $N = 200$.

Example 4.6 We consider the interaction of blast waves of the Euler equation (4.4) with the initial condition

$$\boldsymbol{u}(x,0) = \begin{cases} \boldsymbol{u}_L, & 0 \leq x < 0.1, \\ \boldsymbol{u}_M, & 0.1 \leq x < 0.9, \\ \boldsymbol{u}_R, & 0.9 \leq x < 1, \end{cases} \tag{4.10}$$

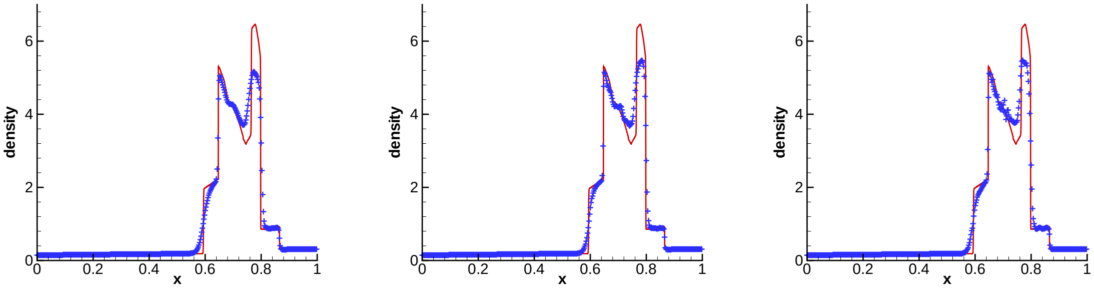
where $\rho_L = \rho_M = \rho_R = 1$, $v_L = v_M = v_R = 0$, $p_L = 10^3$, $p_M = 10^{-2}$, $p_R = 10^2$. A reflecting boundary condition is applied to both ends. See [35, 13]. The computed density ρ is plotted



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

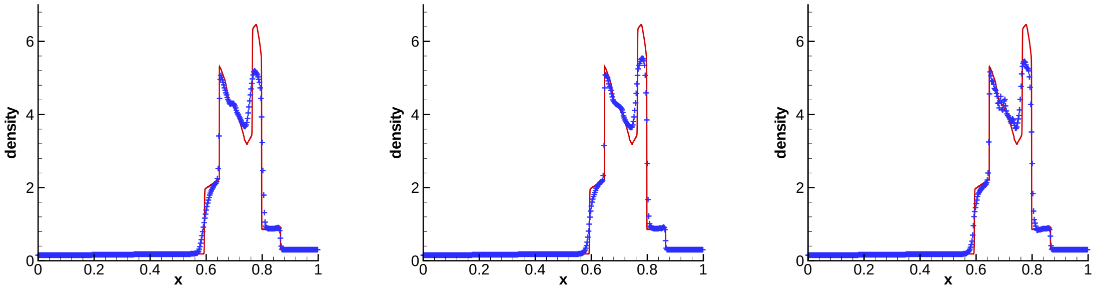
Figure 4.10: The shock density wave interaction problem. $t = 1.8$. $M = 300$. $N = 200$.

at $t = 0.038$ against the reference "exact" solution, which is a converged solution computed by the fifth order finite difference WENO scheme [15] with 16000 grid points. The results are in Figures 4.11-4.12. For the P^3 case, we add the positivity-preserving limiter [34] to avoid negative density or negative pressure during the time evolution. We can see that the pictures are satisfactory, except for the smearing of contact discontinuities, which seems more serious for this problem. For this problem, we have not noticed any significant difference for taking M from 0.01 to 300.



(a) $k = 1$ (b) $k = 2$ (c) $k = 3$

Figure 4.11: The blast wave problem. $t = 0.038$. $M = 0.01$. $N = 400$.



(a) $k = 1$

(b) $k = 2$

(c) $k = 3$

Figure 4.12: The blast wave problem. $t = 0.038$. $M = 200$. $N = 400$.

4.3 Euler system in two dimension

The 2D Euler system is given by

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

$$\boldsymbol{u} = \begin{pmatrix} \rho \\ \rho u \\ \rho v \\ E \end{pmatrix}, \quad \boldsymbol{f}(\boldsymbol{u}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ u(E+p) \end{pmatrix}, \quad \boldsymbol{g}(\boldsymbol{u}) = \begin{pmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ v(E+p) \end{pmatrix}. \tag{4.11}$$

Here ρ is the density, (u, v) is the velocity, E is the total energy, and p is the pressure, with

$$p = (\gamma - 1)\left(E - \frac{1}{2}\rho(u^2 + v^2)\right); \tag{4.12}$$

$\gamma = 1.4$ is used in the computation.

Example 4.7 This example is to test order of accuracy for RKDG methods with our WENO limiter. The initial condition is set to be $\rho(x, y, 0) = 1 + 0.2\sin(x + y)$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$, $p(x, y, 0) = 1$ and the boundary conditions are periodic. The exactly 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 $p(x, y, t) = 1$. We collect the L_1 and L_∞ errors at $t = 2\pi$ in Tables 4.5 and 4.6. Again, we can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the error of the original RKDG method.

Example 4.8 We consider the two-dimensional vortex evolution problem [29, 14], which is an idealized problem for the 2D Euler equations (4.11). The set up of this problem is as

Table 4.5: 2D Euler equation with initial condition $\rho(x, y, 0) = 1+0.2\sin(x+y)$, $u(x, y, 0) = 0.7$, $v(x, y0) = 0.3$, $p(x, y, 0) = 1$ at $t = 2\pi$. Uniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	$N \times N$	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20×20	2.67E-03		6.33E-03	2.68	8.31E-03		2.36E-02	
	40×40	3.35E-04	3.00	1.84E-03	1.78	7.63E-04	3.45	3.92E-03	2.59
	80×80	5.92E-05	2.50	5.38E-04	1.77	6.20E-05	3.62	5.97E-04	2.72
	160×160	1.40E-05	2.08	1.44E-04	1.90	1.64E-05	1.92	1.54E-04	1.95
P^2	20×20	8.92E-05		7.41E-04		9.73E-05		7.42E-04	
	40×40	1.08E-05	3.04	1.06E-04	2.81	1.14E-05	3.10	1.06E-04	2.81
	80×80	1.29E-06	3.08	1.39E-05	2.93	1.33E-06	3.09	1.39E-05	2.93
	160×160	1.56E-07	3.05	1.76E-06	2.98	1.60E-07	3.06	1.76E-06	2.98

Table 4.6: 2D Euler equation with initial condition $\rho(x, y, 0) = 1+0.2\sin(x+y)$, $u(x, y, 0) = 0.7$, $v(x, y0) = 0.3$, $p(x, y, 0) = 1$ at $t = 2\pi$. Nonuniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	$N \times N$	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20×20	2.76E-03		7.20E-03		8.14E-03		2.62E-02	
	40×40	3.52E-04	2.97	2.70E-03	1.41	8.28E-04	3.30	4.51E-03	2.54
	80×80	6.21E-05	2.50	6.87E-04	1.98	7.01E-05	3.56	7.60E-04	2.57
	160×160	1.49E-05	2.06	2.09E-04	1.72	1.49E-05	2.23	2.12E-04	1.85
P^2	20×20	9.25E-05		8.32E-04		1.08E-04		8.53E-04	
	40×40	1.12E-05	3.05	1.27E-04	2.71	1.22E-05	3.14	1.27E-04	2.74
	80×80	1.31E-06	3.09	1.59E-05	3.01	1.40E-06	3.13	1.59E-05	3.01
	160×160	1.58E-007	3.05	2.05E-06	2.95	1.65E-07	3.09	2.05E-06	2.95

follows: The mean flow is $\rho = 1$, $p = 1$, and $(u, v) = (1, 1)$ (diagonal flow). We add, to this mean flow, an isentropic vortex (perturbations in (u, v) and the temperature $T = \frac{p}{\rho}$, no perturbation in the entropy $S = \frac{p}{\rho^\gamma}$:

$$(\delta u, \delta v) = \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(-\bar{y}, \bar{x}), \quad \delta T = -\frac{(\gamma - 1)\epsilon^2}{8\gamma\pi^2} e^{1-r^2}, \quad \delta S = 0, \tag{4.13}$$

where $(\bar{x}, \bar{y}) = (x-5, y-5)$, $r^2 = \bar{x}^2 + \bar{y}^2$, 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 $[-5, 15] \times [-5, 15]$, 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 the mean velocity. We compute the solution to $t = 2$ for the accuracy test. The accuracy results are shown in Tables 4.7 and 4.8. Again, we can see that the WENO limiter maintains both the designed order of accuracy and the magnitude of the error of the original RKDG method.

Table 4.7: 2D Euler System of Smooth Vortex Evolution at $t = 2$. Uniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20×20	2.11E-03		1.63E-01		4.05E-03		3.33E-01	
	40×40	4.57E-04	2.20	4.04E-02	2.01	8.74E-04	2.21	7.52E-02	2.15
	80×80	8.88E-05	2.36	1.28E-02	1.66	1.37E-04	2.67	1.54E-02	2.29
	160×160	1.67E-05	2.41	2.90E-03	2.14	2.05E-05	2.74	3.24E-03	2.25
P^2	20×20	5.44E-04		1.01E-01		1.95E-03		1.18E-01	
	40×40	6.19E-05	3.14	9.99E-03	3.34	8.72E-05	4.48	9.87E-03	3.58
	80×80	7.70E-06	3.01	1.28E-03	2.97	1.24E-05	2.81	1.59E-03	2.63
	160×160	1.08E-06	2.84	1.78E-04	2.85	1.41E-06	3.14	2.21E-04	2.85

Example 4.9 We consider the double Mach reflection problem [35]. It contains strong shock waves and contact discontinuity which is a good example to test the numerical scheme to show the ability to capture strong shock wave and the resolution for small scale structure. The computational domain for this problem is chosen to be $[0, 4] \times [0, 1]$. The reflecting wall lies at the bottom, starting from $x = \frac{1}{6}$. Initially a right-moving Mach 10 shock is positioned

Table 4.8: 2D Euler System of Smooth Vortex Evolution at $t = 2$. Nonuniform mesh with $N \times N$ cells.

		DG without limiter				DG with WENO limiter($M = 0.01$)			
	N	L_1 error	order	L_∞ error	order	L_1 error	order	L_∞ error	order
P^1	20×20	2.15E-03		1.67E-01		4.16E-03		3.31E-01	
	40×40	4.81E-04	2.16	4.08E-02	2.04	9.29E-04	2.16	7.69E-02	2.10
	80×80	9.89E-05	2.28	1.21E-02	1.75	1.51E-04	2.62	1.80E-02	2.10
	160×160	1.91E-05	2.38	2.79E-03	2.12	2.34E-05	2.69	3.29E-03	2.45
P^2	20×20	5.68E-04		9.37E-02		2.37E-03		1.46E-01	
	40×40	6.67E-05	3.09	1.10E-02	3.09	1.03E-04	4.53	1.14E-02	3.68
	80×80	8.09E-06	3.04	1.13E-03	3.28	1.52E-05	2.75	1.63E-03	2.80
	160×160	1.11E-06	2.86	1.67E-04	2.77	1.62E-06	3.23	2.42E-04	2.76

at $x = \frac{1}{6}$, $y = 0$ and makes a 60° angle with the x-axis. For the bottom boundary, the exact postshock 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, the flow values are set to describe the exact motion of a Mach 10 shock. We compute the solution up to $t = 0.2$. As in [35], only results in $[0, 3] \times [0, 1]$ are shown. Two different uniform meshes, with 480×120 and 960 cells, and three different values of the TVB constant, $M = 0.01$, $M = 100$ and $M = 200$, are used in the numerical experiments. The density is plotted in Figure 4.13 for $M = 0.01$ and in Figure 4.14 for $M = 100$. In all the plots, we use 29 contours equally distributed from $\rho = 1.3$ to 23. It is not easy to observe any significant difference among these results in the picture. However, if we show a "blown-up" portion around the double Mach region, as in Figures 4.15 and 4.16, it is clear that one observes an increased resolution with an increasing k on the same mesh. Also, the resolution is slightly better as M increases from $M = 0.01$ to $M = 200$; however, this difference is not significant. Finally, we list in Table 4.9 the percentage of troubled-cells among all the cells.

Table 4.9: Percentage of troubled cells in the double Mach problem

N	$k = 1$			$k = 2$		
	$M = 0.01$	$M = 100$	$M = 200$	$M = 0.01$	$M = 100$	$M = 200$
480×120	20.31	5.65	4.53	25.84	18.78	16.11
960×240	14.35	4.69	3.79	23.31	19.62	18.05

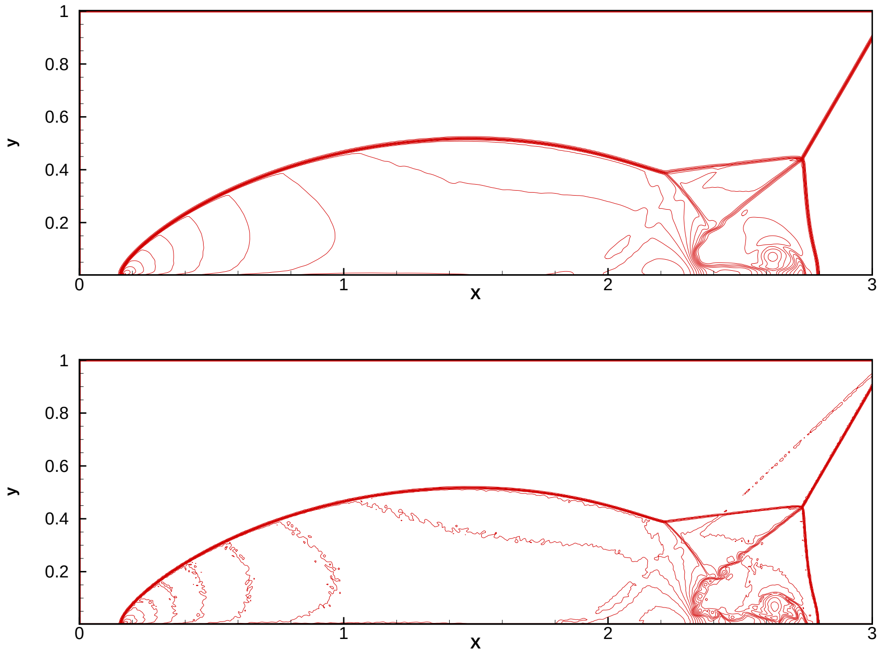


Figure 4.13: Double Mach reflection problem. $M = 0.01$. 960×240 cells. Twenty nine equally spaced density contours from 1.3 to 23. Top: $k = 1$. Bottom: $k = 2$.

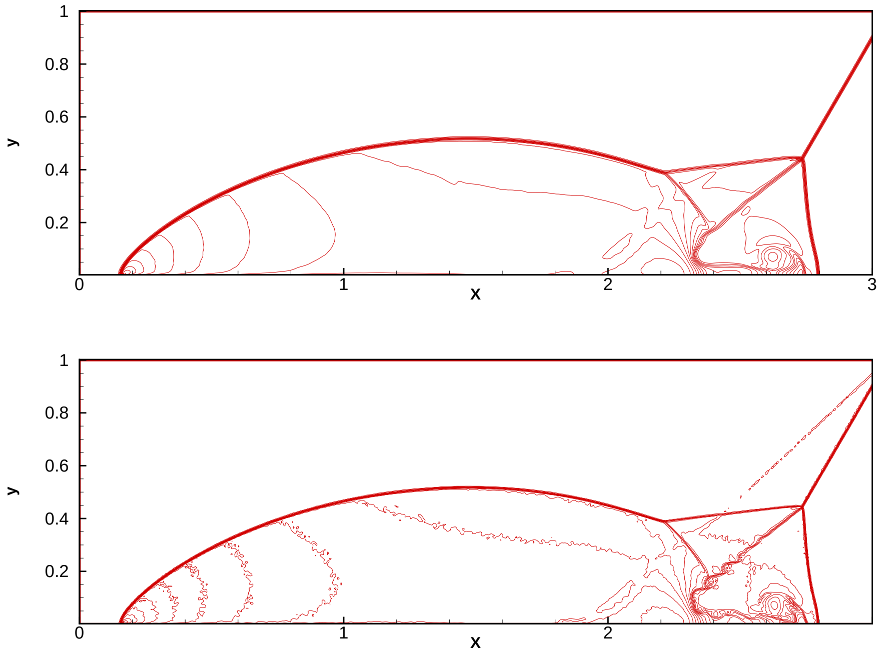


Figure 4.14: Double Mach reflection problem. $M = 200$. 960×240 cells. Twenty nine equally spaced density contours from 1.3 to 23. Top: $k = 1$. Bottom: $k = 2$.

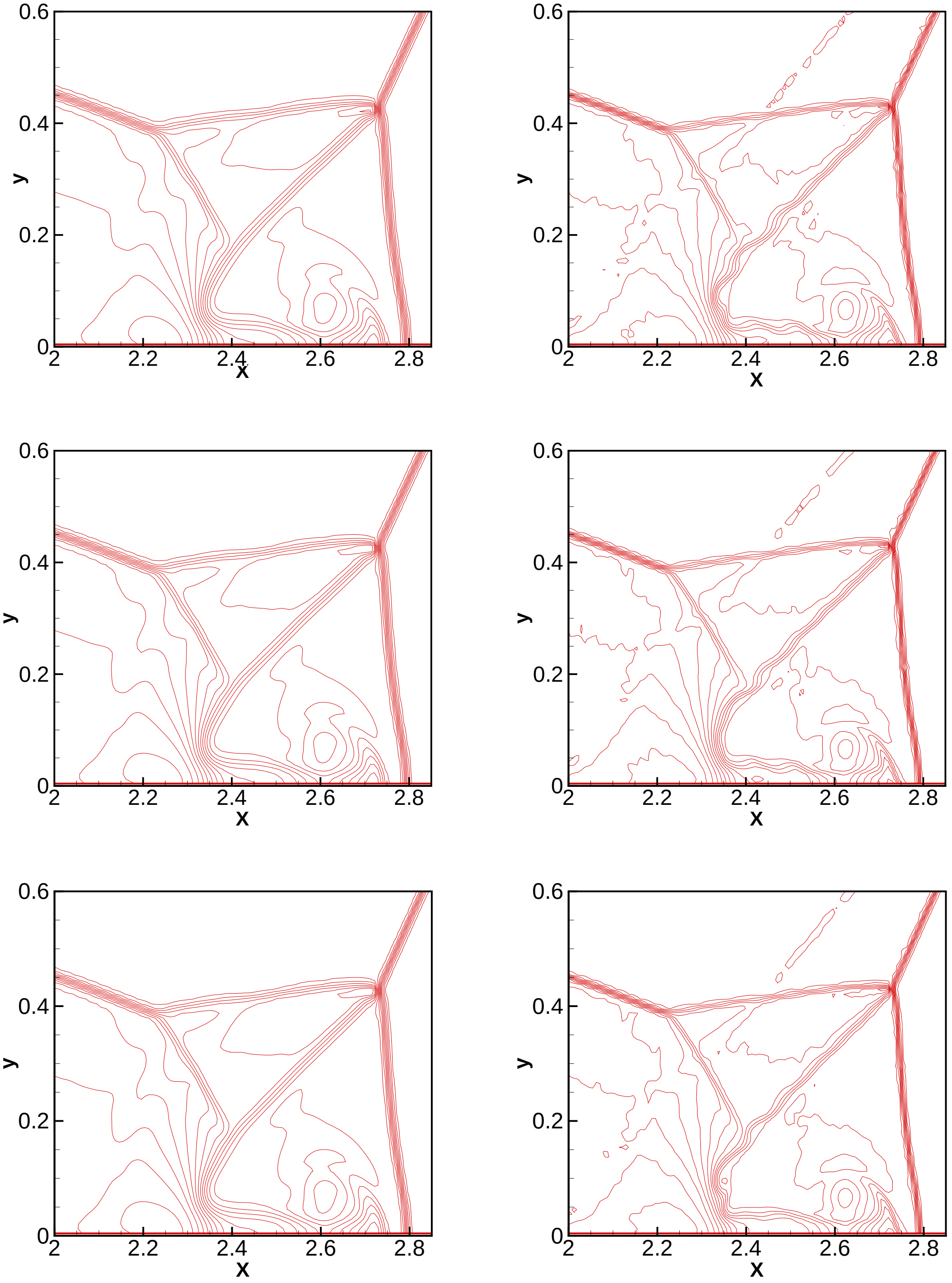


Figure 4.15: Double Mach reflection problem. 480 × 120 cells. $M = 0.01$ (top), $M = 100$ (middle) and $M = 200$ (bottom). Twenty nine equally spaced density contours from 1.3 to 23. Left: $k = 1$. Right: $k = 2$.

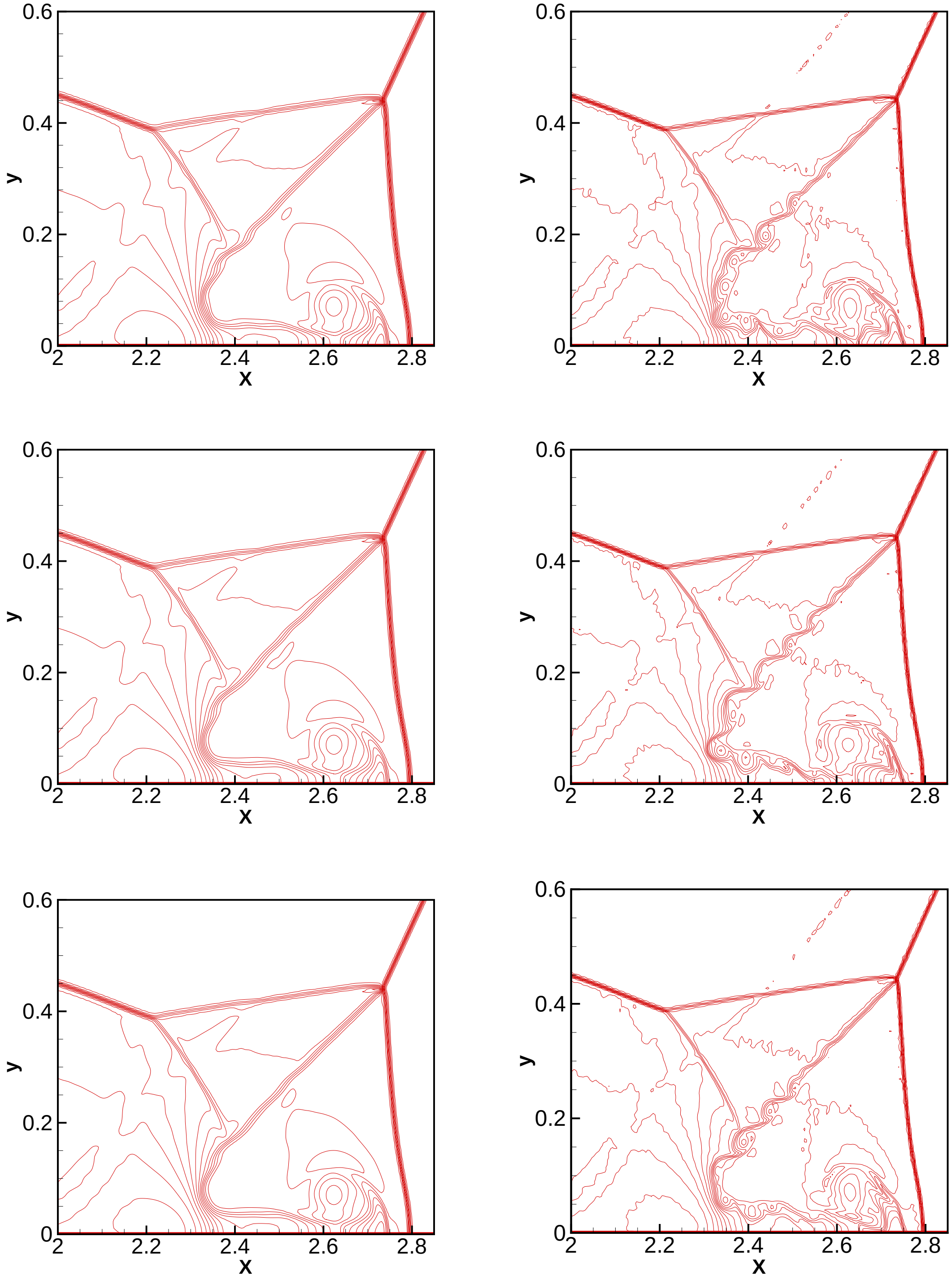


Figure 4.16: Double Mach reflection problem. 960 × 240 cells. $M = 0.01$ (top), $M = 100$ (middle) and $M = 200$ (bottom). Twenty nine equally spaced density contours from 1.3 to 23. Left: $k = 1$. Right: $k = 2$.

5 Conclusion

In this paper, we have developed a new limiter for the RKDG method solving hyperbolic conservation laws using the WENO methodology, which is particularly simple to implement. The general framework is to first identify troubled cells subject to the WENO limiting procedure (we use a TVB minmod limiter in this paper, but other troubled cell identification techniques can of course also be used), then reconstruct a new polynomial on the troubled cell by a WENO reconstruction. The idea of this simple WENO limiter is that the reconstruction polynomial on the troubled cell is a convex combination of the polynomials on this cell and its immediate neighboring cells, with the nonlinear weights of the linear combination following the classical WENO procedure. Since this procedure uses information only from immediate neighbors and simple positive linear weights, it is significantly simpler to implement than previous WENO type limiters for RKDG methods. Numerical results are provided to show that this limiting procedure can obtain both uniform high order accuracy and sharp, non-oscillatory shock transitions for the RKDG methods. The numerical performance is similar to previous WENO limiters which are much more complicated to implement. Generalization of this procedure to unstructured meshes is ongoing. Improving the procedure for identifying troubled cells and implementing the limiter for higher dimensional problems also constitute ongoing work.

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*, 160:405–452, 2000.

[2] R. Biswas, K. D. Devine and J. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14:255–283, 1994.

[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*, 169:111–150, 2001.

[4] 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*, 54:545–581, 1990.

[5] 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*, 84:90–113, 1989.

[6] 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*, 52:411–435, 1989.

[7] B. Cockburn and C.-W. Shu. The Runge-Kutta local projection P^1-discontinuous Galerkin method for scalar conservation laws. *Mathematical Modelling and Numerical Analysis*, 25:337–361, 1991.

[8] B. Cockburn and C.-W. Shu. The Runge-Kutta discontinuous Galerkin method for conservation laws V: Multidimensional systems. *Journal of Computational Physics*, 141:199–224, 1998.

[9] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221:693–723, 2007.

[10] O. Friedrich. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, 144:194–212, 1998.

[11] S. Gottlieb, D. I. Ketcheson and C.-W. Shu. High order strong stability preserving time discretizations. *Journal of Scientific Computing*, 38:251–289, 2009.

[12] A. Harten. High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics*, 49:357–393, 1983.

[13] A. Harten, B. Engquist, S. Osher and S. R. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, III. *Journal of Computational Physics*, 71:231–303, 1987.

[14] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150:97–127, 1999.

[15] G. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202–228, 1995.

[16] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205:486–508, 2005.

[17] P. D. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computation. *Communications on Pure and Applied Mathematics*, 7:159–193, 1954.

[18] D. Levy, G. Puppo and G. Russo. Central WENO Schemes for hyperbolic systems of conservation laws. *Mathematical Modelling and Numerical Analysis (M^2AN)*, 33:547–571, 2001.

[19] X. Liu, S. Osher and T. Chen. Weighted essentially non-oscillatory schemes. *Journal of Computational Physics*, 115:200–212, 1994.

[20] J. Qiu and C.-W. Shu. On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes. *Journal of Computational Physics*, 183:187–209, 2002.

[21] 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*, 193:115–135, 2003.

[22] 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*, 27:995–1013, 2005.

[23] J. Qiu and C.-W. Shu. Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM Journal on Scientific Computing*, 26:907–929, 2005.

[24] 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*, 34:642–663, 2005.

[25] W. H. Reed and T. R. Hill. Triangular mesh methods for the neutron transport equation. Technical Report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, 1973.

[26] P. L. Roe. Approximate Riemann solvers, parameter vectors, and difference schemes. *Journal of Computational Physics*, 135:250–258, 1997.

[27] J. Shi, C. Hu and C.-W. Shu. A technique of treating negative weights in WENO schemes. *Journal of Computational Physics*, 175:108–127, 2002.

[28] C.-W. Shu. TVB uniformly high-order schemes for conservation laws. *Mathematics of Computation*, 49:105–121, 1987.

[29] 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, 1998, 325-432.

[30] C.-W. Shu. High order weighted essentially non-oscillatory schemes for convection dominated problems. *SIAM Review*, 51:82–126, 2009.

[31] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77:439–471, 1988.

[32] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83:32–78, 1989.

[33] G. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *Journal of Computational Physics*, 27:1–31, 1978.

[34] 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*, 231:653-665, 2012.

[35] P. Woodward and P. Colella. The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics*, 54:115–173, 1984.

[36] Y.-T. Zhang and C.-W. Shu. Third order WENO scheme on three dimensional tetrahedral meshes. *Communications in Computational Physics*, 5:836–848, 2009.

[37] 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*, 227:4330–4353, 2008.