SIAM J. SCI. COMPUT.
Vol. 26, No. 3, pp. 907–929

RUNGE–KUTTA DISCONTINUOUS GALERKIN METHOD USING WENO LIMITERS*

JIANXIAN QIU† AND CHI-WANG SHU‡

Abstract. The Runge–Kutta discontinuous Galerkin (RKDG) method is a high order finite element method for solving hyperbolic conservation laws. It uses ideas from high resolution finite volume schemes, such as the exact or approximate Riemann solvers, total variation diminishing (TVD) Runge–Kutta time discretizations, and limiters. It has the advantage of flexibility in handling complicated geometry, *h-p* adaptivity, and efficiency of parallel implementation, and has been used successfully in many applications. However, the limiters used to control spurious oscillations in the presence of strong shocks are less robust than the strategies of essentially nonoscillatory (ENO) and weighted ENO (WENO) finite volume and finite difference methods. In this paper we investigate using WENO finite volume methodology as limiters for RKDG methods, with the goal of obtaining a robust and high order limiting procedure to simultaneously obtain uniform high order accuracy and sharp, nonoscillatory shock transition for RKDG methods. The traditional finite volume WENO framework based on cell averages is used to reconstruct point values of the solution at Gaussian-type points in those cells where limiting is deemed necessary, and the polynomial solutions in those cells are then rebuilt through numerical integration using these Gaussian points. Numerical results in one and two dimensions are provided to illustrate the behavior of this procedure.

Key words. Runge–Kutta discontinuous Galerkin method, limiters, WENO finite volume scheme, high order accuracy

AMS subject classifications. 65M60, 65M99, 35L65

DOI. 10.1137/S1064827503425298

1. Introduction.

In this paper, we investigate using weighted essentially nonoscillatory (WENO) finite volume methodology [18, 17, 12, 16, 21] as limiters for the Runge–Kutta discontinuous Galerkin (RKDG) finite element methods [9, 8, 7, 5, 10, 11], for solving nonlinear hyperbolic conservation laws

$$\begin{cases} u_t + \nabla \cdot f(u) = 0, \\ u(x,0) = u_0(x), \end{cases} \tag{1.1}$$

with the goal of obtaining a robust and high order limiting procedure to simultaneously obtain uniform high order accuracy and sharp, nonoscillatory shock transition for the RKDG methods.

The first discontinuous Galerkin (DG) method was introduced in 1973 by Reed and Hill [20] in the framework of neutron transport (steady state linear hyperbolic equations). A major development of the DG method was carried out by Cockburn et al. in a series of papers [9, 8, 7, 5, 10], in which they established a framework to easily solve *nonlinear* time dependent hyperbolic conservation laws (1.1) using explicit, nonlinearly stable high order Runge–Kutta time discretizations [24] and DG

*Received by the editors March 28, 2003; accepted for publication (in revised form) April 29, 2004; published electronically January 20, 2005.
http://www.siam.org/journals/sisc/26-3/42529.html

†Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, People's Republic of China (jxqiu@ustc.edu). The research of this author was supported in part by NNSFC grant 10028103. Current address: Department of Mechanical Engineering, National University of Singapore, Singapore 119260.

‡Division of Applied Mathematics, Brown University, Providence, RI 02912 (shu@dam.brown.edu). The research of this author was supported by NNSFC grant 10028103 while he was in residence at the Department of Mathematics, University of Science and Technology of China. Additional support was provided by ARO grant DAAD19-00-1-0405 and NSF grant DMS-0207451.

discretization in space with exact or approximate Riemann solvers as interface fluxes and total variation bounded (TVB) limiter [22] to achieve nonoscillatory properties for strong shocks. These schemes are termed RKDG methods.

Given a triangulation consisting of cells $\triangle_j$ (intervals in one dimension, triangles or quadrilaterals in two dimensions, etc.), a semidiscrete DG method for solving the conservation law (1.1) is obtained by multiplying (1.1) with a test function $v(x)$, integrating over a cell $\triangle_j$, and integrating by parts:

$$\frac{d}{dt}\int_{\triangle_j} u(x,t)v(x)dx - \int_{\triangle_j} f(u)\cdot\nabla v\,dx + \int_{\partial\triangle_j} f(u)\cdot nv\,ds = 0, \tag{1.2}$$

where n is the outward unit normal of the cell boundary $\partial\triangle_j$. We seek a piecewise polynomial u in $\mathbb{P}^k$ of degree at most k (k could actually change from cell to cell, but for simplicity we assume it is a constant over the whole triangulation) such that (1.2) holds for any test function v also in $\mathbb{P}^k$. The boundary integral in (1.2) is typically discretized by a Gaussian quadrature of sufficiently high order of accuracy

$$\int_{\partial\triangle_j} f\cdot n\,ds \approx |\partial\triangle_j|\sum_{k=1}^{q}\omega_k f(u(G_k,t))\cdot n$$

and $f(u(G_k,t))\cdot n$ is replaced with a monotone numerical flux (approximate or exact Riemann solvers in the system case). For example, one could use the simple Lax–Friedrichs flux, which is given by

$$f(u(G_k,t))\cdot n \approx \frac{1}{2}[(f(u^-(G_k,t)) + f(u^+(G_k,t)))\cdot n - \alpha(u^+(G_k,t) - u^-(G_k,t))],$$

where α is taken as an upper bound for $|f'(u)\cdot n|$ in the scalar case, or the absolute value of eigenvalues of the Jacobian in the n direction for the system case, and u^- and u^+ are the values of u inside the cell $\triangle_j$ and outside the cell $\triangle_j$ (inside the neighboring cell) at the Gaussian point G_k. The idea of using such a numerical flux is borrowed from a finite volume methodology. The test function v in the boundary integral in (1.2) is taken from inside the cell $\triangle_j$. The volume integral $\int_{\triangle_j} f(u)\cdot\nabla v\,dx$ in (1.2) is either computed exactly or by a numerical quadrature with sufficient accuracy; see [8, 5] for details. The semidiscrete scheme (1.2), written as

$$u_t = L(u),$$

is then discretized in time by a total variation diminishing (TVD) Runge–Kutta method [24], for example, the third order version given by

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n),\\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}),\\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}). \end{aligned} \tag{1.3}$$

This scheme, as briefly described above, has the typical advantage of finite element methods in that it can easily handle complicated geometry and arbitrary triangulations. It also has the added advantages, due to the discontinuous nature of the solution and the test function space, in an explicit time marching, of local communications (hence, high efficiency in parallel implementation [2]) and easy *h-p* adaptivity.

For these reasons, this scheme is widely used in applications; see, for example the survey paper [6], and other papers in that volume. The lecture notes [4], as well as the extensive review paper [11], are good references offering many details.

The method described above can compute solutions to (1.1), which are either smooth or have weak shocks and other discontinuities, without further modification. If the discontinuities are strong, however, the scheme will generate significant oscillations and even nonlinear instability. To avoid such difficulties, we again borrow a technique of a slope limiter from the finite volume methodology and use it after each Runge–Kutta inner stage (or after the complete Runge–Kutta time step) to control the numerical solution. Many such limiters exist in the literature; for example, the minmod-type limiters in [9, 8, 7, 5, 10], the moment-based limiters in [2], and the more recent limiter in [3]. Such limiters must be designed to control spurious oscillations and at the same time maintain accuracy in smooth regions in a robust way, which is usually difficult to achieve. It is no exaggeration to state that the design of good, robust limiters is one of the bottlenecks to the development of DG methods for solving conservation laws.

On the other hand, the ENO and WENO methods have been developed in the context of finite volume and finite difference frameworks to successfully achieve both uniform high order accuracy and sharp, ENO shock transitions; see, e.g., [14, 24, 25, 18, 17, 12, 16, 21]. The ENO/WENO methodology is more robust than the slope limiter methodology, especially for high order schemes. Thus it would be natural to try to use an ENO or WENO methodology as limiters for the discontinuous Galerkin methods. We make such an attempt in this paper, adopting the following framework:

1. First, we identify the "troubled cells," namely, those cells which might need the limiting procedure.

2. Second, we replace the solution polynomials in those troubled cells with reconstructed polynomials which maintain the original cell averages (conservation), have the same orders of accuracy as before, but are less oscillatory.

In this paper the first step above is achieved by the usual minmod-type TVB limiters as in [8, 5, 10]. That is, whenever the minmod limiter changes the slope, the cell is declared to be a troubled cell. 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. There is clearly room here for improvement and this is left for future study. We emphasize, however, that if the second step above can achieve its goal, then it is less crucial for the first step to identify only the really troubled cells. If a good cell is mistakenly identified as troubled, it will cost extra computational time for the correction in the second step, but no loss of accuracy (at least no loss in the order of accuracy) will result.

It is in the second step above that we use the finite volume WENO reconstruction procedure. The principle in this step is to abandon the solution polynomials in such troubled cells (except for their cell averages to keep conservation) and "reconstruct" new polynomials from the information of neighboring cells. A WENO scheme is used in such reconstructions. The main difficulty in this step is maintaining the original high order accuracy. It turns out that one has to be very careful in order not to lose the order of accuracy in such reconstructions.

In this paper we use the usual WENO reconstructions based on cell averages of neighboring cells, such as in [21, 16], to reconstruct the values of the solutions at certain Gaussian quadrature points in the troubled cells, and then rebuild the solution

polynomials in those troubled cells from the original cell averages and the reconstructed values at the Gaussian quadrature points through a numerical integration for the moments. This turns out to be a robust way to retain the original high order accuracy of the DG method. We describe the details of this procedure in section 2 and present extensive numerical results in section 3 to verify the accuracy and stability of this approach. For simplicity, we will present results only in rectangular elements, but the methodology clearly also applies to triangular elements, which is left for future work. Concluding remarks and suggestions for future work are given in section 4.

2. WENO reconstruction as a limiter to the RKDG method. In this section we give the details of the procedure using the WENO reconstruction as a limiter to the RKDG method. We start with the description in the one-dimensional case and use the notation in [8]; however, we emphasize that the procedure described below does not depend on the specific basis chosen for the polynomials. We would like to solve the one-dimensional scalar conservation law

$$\begin{cases} u_t + f(u)_x = 0, \\ u(x,0) = u_0(x). \end{cases} \tag{2.1}$$

The points x_i are the centers of the cells $I_i = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}]$, and we denote the cell sizes by $\Delta x_i = x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}}$ and the maximum cell size by $h = \sup_i \Delta x_i$. The solution, as well as the test function space, is given by $V_h^k = \{p : p|_{I_i} \in P^k(I_i)\}$, where $P^k(I_i)$ is the space of polynomials of degree $\le k$ on the cell I_i. We adopt a local orthogonal basis over I_i, $\{v_l^{(i)}(x),\ l = 0, 1, \ldots, k\}$, namely, the scaled Legendre polynomials

$$v_0^{(i)}(x) = 1, \qquad v_1^{(i)}(x) = \frac{x - x_i}{\Delta x_i}, \qquad v_2^{(i)}(x) = \left(\frac{x - x_i}{\Delta x_i}\right)^2 - \frac{1}{12}, \ldots .$$

Then the numerical solution $u^h(x,t)$ in the space V_h^k can be written as

$$u^h(x,t) = \sum_{l=0}^{k} u_i^{(l)}(t) v_l^{(i)}(x) \quad \text{for } x \in I_i \tag{2.2}$$

and the degrees of freedom $u_i^{(l)}(t)$ are the moments defined by

$$u_i^{(l)}(t) = \frac{1}{a_l} \int_{I_i} u^h(x,t) v_l^{(i)}(x) dx, \quad l = 0, 1, \ldots, k,$$

where $a_l = \int_{I_i} (v_l^{(i)}(x))^2 dx$ are the normalization constants since the basis is not orthonormal. In order to determine the approximate solution, we would like to evolve the degrees of freedom $u_i^{(l)}$:

$$\begin{aligned} \frac{d}{dt} u_i^{(l)} + \frac{1}{a_l} \Bigg(& - \int_{I_i} f(u^h(x,t)) \frac{d}{dx} v_l^{(i)}(x) dx + \hat{f}(u^-_{i+1/2}, u^+_{i+1/2}) v_l^{(i)}(x_{i+1/2}) \\ & - \hat{f}(u^-_{i-1/2}, u^+_{i-1/2}) v_l^{(i)}(x_{i-1/2}) \Bigg) = 0, \quad l = 0, 1, \ldots, k, \end{aligned} \tag{2.3}$$

where $u^{\pm}_{i+1/2} = u^h(x^{\pm}_{i+1/2}, t)$ are the left and right limits of the discontinuous solution u^h at the cell interface $x_{i+1/2}$, $\hat{f}(u^-, u^+)$ is a monotone flux (nondecreasing in the first argument and nonincreasing in the second) for the scalar case and an exact or

approximate Riemann solver for the system case. The semidiscrete scheme (2.3) is discretized in time by a nonlinearly stable Runge–Kutta time discretization, e.g., the third order version (1.3). The integral term in (2.3) can be computed either exactly or by a suitable numerical quadrature accurate to at least $O(h^{k+l+2})$.

The limiter adopted in [8] is described below in some detail, as it is the one used in this paper to detect "troubled cells." Denote

$$u^-_{i+1/2} = u^{(0)}_i + \tilde{u}_i, \qquad u^+_{i-1/2} = u^{(0)}_i - \tilde{\tilde{u}}_i.$$

From (2.2) we can see that

$$\tilde{u}_i = \sum_{l=1}^{k} u^{(l)}_i v^{(i)}_l(x_{i+1/2}), \qquad \tilde{\tilde{u}}_i = -\sum_{l=1}^{k} u^{(l)}_i v^{(i)}_l(x_{i-1/2}).$$

These are modified by either the standard minmod limiter [13]

$$\tilde{u}^{(mod)}_i = m\left(\tilde{u}_i, \Delta_+ u^{(0)}_i, \Delta_- u^{(0)}_i\right), \qquad \tilde{\tilde{u}}^{(mod)}_i = m\left(\tilde{\tilde{u}}_i, \Delta_+ u^{(0)}_i, \Delta_- u^{(0)}_i\right),$$

where m is given by

(2.4)
$$m(a_1, a_2, \dots, a_n) = \begin{cases} s \cdot \min_{1\le j\le n} |a_j| & \text{if } \operatorname{sign}(a_1) = \operatorname{sign}(a_2) = \cdots = \operatorname{sign}(a_n) = s, \\ 0 & \text{otherwise,} \end{cases}$$

or by the TVB modified minmod function [22]

$$\tilde{m}(a_1, a_2, \dots, a_n) = \begin{cases} a_1 & \text{if } |a_1| \le Mh^2, \\ m(a_1, a_2, \dots, a_n) & \text{otherwise,} \end{cases} \tag{2.5}$$

where $M > 0$ is a constant. The choice of M depends on the solution of the problem. For scalar problems it is possible to estimate M by the initial condition as in [8] (M is proportional to the second derivative of the initial condition at smooth extrema); however, it is more difficult to estimate M for the system case. If M is chosen too small, accuracy may degenerate at smooth extrema of the solution; however, if M is chosen too large, oscillations will appear.

In this paper we use the limiter described above to identify "troubled cells"; i.e., if one of the minmod functions gets enacted (returns other than the first argument), this cell is declared "troubled" and marked for further reconstructions. Since the reconstruction described below maintains the high order accuracy in the troubled cells, it is less crucial to choose an accurate M. We present in section 3 numerical results obtained with different M's. Basically, if M is chosen too small, more good cells will be declared troubled cells and will be subject to unnecessary WENO reconstructions. This increases the computational cost but does not degrade the order of accuracy in these cells.

For the troubled cells, we would like to reconstruct the polynomial solution while retaining its cell average. In other words, we will reconstruct the degrees of freedom, or the moments, $u^{(l)}_i$ for the troubled cell I_i for $l = 1, \dots, k$ and retain only the cell average $u^{(0)}_i$.

We have experimented with several ways to perform this reconstruction and have settled on the following procedure. We will comment on an alternative procedure in Remark 2.2 below.

Step 1. Here we reconstruct point values of u at the Gauss or Gauss–Lobatto quadrature points. For the $\mathbb{P}^k$-based DG (which is $(k+1)$th order accurate), we need a Gauss or Gauss–Lobatto quadrature rule accurate to at least $O(h^{2k+2})$, and the order of accuracy for the WENO reconstruction must be at least $2k+1$. For this purpose, we would need to use the cell averages of the neighboring $2k+1$ cells $I_{i-k},\dots,I_{i+k}$ to reconstruct the point values of u at the Gauss or Gauss–Lobatto quadrature points. In particular, we have used the following quadrature points:

(1) For the $\mathbb{P}^1$ case, we use the two-point Gauss quadrature points $x_{i-\sqrt{3}/6}$ and $x_{i+\sqrt{3}/6}$.

(2) For the $\mathbb{P}^2$ case, we use the four-point Gauss–Lobatto quadrature points $x_{i-1/2}$, $x_{i-\sqrt{5}/10}$, $x_{i+\sqrt{5}/10}$, and $x_{i+1/2}$.

(3) For the $\mathbb{P}^3$ case, we use the four-point Gauss quadrature points $x_{i-\sqrt{525+70\sqrt{30}}/70}$, $x_{i-\sqrt{525-70\sqrt{30}}/70}$, $x_{i+\sqrt{525-70\sqrt{30}}/70}$ and $x_{i+\sqrt{525+70\sqrt{30}}/70}$.

The WENO reconstruction [18, 17, 1] is then performed:

Step 1.1. We identify $k+1$ small stencils S_j, $j=0,\dots,k$, such that I_i belongs to each of them. Here we set $S_j=\cup_{l=0}^{k} I_{i+j-l}$. We denote by $\mathcal{T}=\cup_{j=0}^{k} S_j$ the larger stencil which contains all the cells from the $k+1$ smaller stencils.

We have a kth degree polynomial reconstruction denoted by $p_j(x)$, associated with each of the stencils S_j, $j=0,\dots,k$, such that the cell average of $p_j(x)$ in each of the cells in the stencil S_j agrees with the given cell average of u, i.e., $\frac{1}{\Delta x_{i+j-l}}\int_{I_{i+j-l}} p_j(x)dx = u^{(0)}_{i+j-l}$, $l=0,\dots,k$. We also have a higher order $(2k)$th degree polynomial reconstruction denoted by $Q(x)$, associated with the larger stencil $\mathcal{T}$, such that $\frac{1}{\Delta x_{i+l}}\int_{I_{i+l}} Q(x)dx = u^{(0)}_{i+l}$, $l=-k,\dots,k$. The detail of the construction of $p_j(x)$ and $Q(x)$ can be found in [23].

Step 1.2. We find the combination coefficients, also called linear weights, denoted by $\gamma_0,\dots,\gamma_k$, which satisfy

$$Q(x_G)=\sum_{j=0}^{k}\gamma_j p_j(x_G),$$

where x_G is a Gauss quadrature point. Different quadrature points correspond to different linear weights. The value of the functions $Q(x)$ and $p_j(x)$, $j=0,\dots,k$, at a Gaussian point x_G can be written as a linear combination of $u_i^{(0)}$ in the stencil. For example, when $k=2$, with a uniform mesh, for $x_G=x_{i+1/2}$, we have

$$\begin{aligned}
p_0(x_G)&=\frac{1}{3}u^{(0)}_{i-2}-\frac{7}{6}u^{(0)}_{i-1}+\frac{11}{6}u^{(0)}_{i},\\
p_1(x_G)&=-\frac{1}{6}u^{(0)}_{i-1}+\frac{5}{6}u^{(0)}_{i}+\frac{1}{3}u^{(0)}_{i+1},\\
p_2(x_G)&=\frac{1}{3}u^{(0)}_{i}+\frac{5}{6}u^{(0)}_{i+1}-\frac{1}{6}u^{(0)}_{i+2},\\
Q(x_G)&=\frac{1}{30}u^{(0)}_{i-2}-\frac{13}{60}u^{(0)}_{i-1}+\frac{47}{60}u^{(0)}_{i}+\frac{9}{20}u^{(0)}_{i+1}-\frac{1}{20}u^{(0)}_{i+2},
\end{aligned}$$

and

$$\gamma_0=\frac{1}{10},\qquad \gamma_1=\frac{6}{10},\qquad \gamma_2=\frac{3}{10}.$$

For $x_G = x_{i+\sqrt{5}/10}$ we have

$$p_0(x_G) = \left(-\frac{1}{60} + \frac{\sqrt{5}}{20}\right) u_{i-2}^{(0)} + \left(\frac{1}{30} - \frac{\sqrt{5}}{5}\right) u_{i-1}^{(0)} + \left(\frac{59}{60} + \frac{3\sqrt{5}}{20}\right) u_i^{(0)},$$

$$p_1(x_G) = \left(-\frac{1}{60} - \frac{\sqrt{5}}{20}\right) u_{i-1}^{(0)} + \frac{31}{30} u_i^{(0)} + \left(-\frac{1}{60} + \frac{\sqrt{5}}{20}\right) u_{i+1}^{(0)},$$

$$p_2(x_G) = \left(\frac{59}{60} - \frac{3\sqrt{5}}{20}\right) u_i^{(0)} + \left(\frac{1}{30} + \frac{\sqrt{5}}{5}\right) u_{i+1}^{(0)} + \left(-\frac{1}{60} - \frac{\sqrt{5}}{20}\right) u_{i+2}^{(0)},$$

$$Q(x_G) = \frac{1+6\sqrt{5}}{600} u_{i-2}^{(0)} - \frac{7+21\sqrt{5}}{300} u_{i-1}^{(0)} + \frac{313}{300} u_i^{(0)} + \frac{-7+21\sqrt{5}}{300} u_{i+1}^{(0)} + \frac{1-6\sqrt{5}}{600} u_{i+2}^{(0)},$$

and

$$\gamma_0 = \frac{91+9\sqrt{5}}{440}, \qquad \gamma_1 = \frac{129}{220}, \qquad \gamma_2 = \frac{91-9\sqrt{5}}{440}.$$

The linear combination coefficients of the values of the functions $Q(x)$ and $p_j(x)$, $j = 0, \ldots, k$, and the linear weights for the Gaussian points $x_{i-1/2}$ and $x_{i-\sqrt{5}/10}$ are mirror symmetric with respect to those at $x_{i+1/2}$ and $x_{i+\sqrt{5}/10}$, respectively.

Step 1.3. We compute the smoothness indicator, denoted by β_j, for each stencil S_j, which measures how smooth the function $p_j(x)$ is in the target cell I_i. The smaller this smoothness indicator β_j is, the smoother the function $p_j(x)$ is in the target cell. The smoothness indicators are the same for the reconstruction at all Gauss points in the same cell, thus significantly reducing the computational cost. As in [17, 1], we use the following smoothness indicator:

$$\beta_j = \sum_{l=1}^{k} \int_{I_i} \Delta x_i^{2l-1} \left(\frac{\partial^l}{\partial x^l} p_j(x)\right)^2 dx. \tag{2.6}$$

In the actual numerical implementation the smoothness indicators β_j are written out explicitly as quadratic forms of the cell averages of u in the stencil; see [17, 1, 23] for details.

Step 1.4. We compute the nonlinear weights based on the smoothness indicators,

$$\omega_j = \frac{\overline{\omega}_j}{\sum_l \overline{\omega}_l}, \quad \overline{\omega}_j = \frac{\gamma_j}{\sum_l (\varepsilon + \beta_l)^2}, \tag{2.7}$$

where γ_j are the linear weights determined in Step 1.2 above, and ε is a small number to avoid the denominator becoming zero. We use $\varepsilon = 10^{-6}$ in all computations in this paper. The final WENO approximation is then given by

$$u_G \approx \sum_{j=0}^{k} \omega_j p_j(x_G). \tag{2.8}$$

Step 2. We obtain the reconstructed moments based on the reconstructed point values $u(x_G)$ at the Gauss or Gauss–Lobatto quadrature points x_G and a numerical integration

$$u_i^{(l)} = \frac{\Delta x_i}{a_l} \sum_G w_G u(x_G) v_l^{(i)}(x_G), \quad l = 1, \ldots, k.$$

Here w_G are the Gaussian quadrature weights for the Gaussian points x_G. The polynomial solution in this cell I_i is then obtained by (2.2) with these reconstructed moments $u_i^{(l)}$ for $l = 1, \ldots, k$ and the original cell average $u_i^{(0)}$.

Remark 2.1. For the $\mathbb{P}^2$ case, we can also reconstruct values of u at the three Gauss quadrature points by the fifth order WENO. But the linear weights at the middle Gaussian point x_j are negative. Although such a negative weight case can be treated by the technique developed in [21], we have opted to use the four-point Gauss–Lobatto quadrature to guarantee positive linear weights.

Remark 2.2. It would seem to be more natural to reconstruct the moments $u_i^{(l)}$ for $l = 1, \ldots, k$ directly from the cell averages of neighboring cells. The procedure is similar to what is described above, with Step 1.2 replaced by the following.

Step 1.2′. We find the combination coefficients, also called linear weights, denoted by $\gamma_0, \ldots, \gamma_k$, which satisfy

$$\int_{I_i} Q(x) v_l^{(i)}(x) dx = \sum_{j=0}^{k} \gamma_j \int_{I_i} p_j(x) v_l^{(i)}(x) dx, \quad l = 1, \ldots, k.$$

The final WENO approximation to the moments are then given by

$$u_i^{(l)} \approx \frac{1}{a_l} \sum_{j=0}^{k} \omega_j \int_{I_i} p_j(x) v_l^{(i)}(x) dx, \quad l = 1, \ldots, k,$$

and Step 2 is no longer needed.

Indeed, this approach works well for the $\mathbb{P}^1$ and $\mathbb{P}^2$ cases (results will not be presented in this paper). Unfortunately, the linear weights for such reconstructions do not exist for the $\mathbb{P}^3$ case.

For the system cases, in order to achieve better qualities at the price of more complicated computations, the WENO reconstruction limiter is always used with a local characteristic field decomposition; see, e.g., [23] for details. To find "troubled cells" we could use either a componentwise minmod TVB limiter or a characteristic one. It turns out that, even though a componentwise minmod TVB limiter saves CPU time, it tends to give false alarms for many cells (i.e., declaring too many good cells as troubled cells), and thus the WENO reconstruction is performed in many more cells. We have thus used a characteristic-based minmod TVB limiter to detect troubled cells.

For two spatial dimensions we choose to reconstruct values of the function u in troubled cells at the tensor product Gauss or Gauss–Lobatto points for the rectangular elements considered in this paper. The actual WENO reconstruction we use is the one in [21]. For triangular elements (not considered in this paper), suitable quadrature points such as the ones in [15] and the WENO reconstruction procedure in [16] should be used.

We have given the details of the minmod TVB limiters used to identify troubled cells for the one-dimensional scalar cases in this section (see also [8]). For the one-dimensional system case, we use the characteristic-based limiter procedure [7]. For the two-dimensional scalar case we use the TVB limiter defined in [5] and for the two-dimensional system case, we use the characteristic-based procedure described in [10].

3. Numerical results. In this section we provide extensive numerical experimental results to demonstrate the performance of the WENO reconstruction limiter for the RKDG methods described in section 2.

TABLE 1

Burgers equation $u_t + (u^2/2)_x = 0$. Initial condition $u(x,0) = 0.5 + \sin(\pi x)$ and periodic boundary condition. RKDG with a WENO limiter ($M = 0.01$) compared to RKDG without limiter. Lax–Friedrichs flux. $t = 0.5/\pi$. L_1 and L_∞ errors. Nonuniform meshes with N cells.

		DG with WENO limiter				DG with no limiter			
	N	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
P^1	10	4.61E-02		2.25E-01		1.39E-02		1.06E-01	
	20	1.16E-02	1.99	1.30E-01	0.79	3.53E-03	1.97	3.27E-02	1.70
	40	2.14E-03	2.44	2.53E-02	2.37	8.60E-04	2.04	9.10E-03	1.84
	80	2.85E-04	2.91	3.19E-03	2.99	2.11E-04	2.03	2.53E-03	1.85
	160	5.95E-05	2.26	7.42E-04	2.10	5.27E-05	2.00	6.73E-04	1.91
	320	1.45E-05	2.03	1.79E-04	2.05	1.31E-05	2.01	1.69E-04	1.99
P^2	10	5.98E-03		5.98E-02		1.95E-03		2.87E-02	
	20	5.52E-04	3.44	5.06E-03	3.56	2.72E-04	2.84	4.83E-03	2.57
	40	4.61E-05	3.58	8.97E-04	2.50	4.30E-05	2.66	8.39E-04	2.52
	80	6.22E-06	2.89	1.60E-04	2.48	6.23E-06	2.79	1.76E-04	2.26
	160	8.92E-07	2.80	2.55E-05	2.65	8.94E-07	2.80	2.52E-05	2.80
	320	1.28E-07	2.81	3.78E-06	2.75	1.27E-07	2.81	4.14E-06	2.60
P^3	10	5.19E-03		3.77E-02		1.44E-04		1.78E-03	
	20	9.69E-05	5.74	1.71E-03	4.46	1.89E-05	2.93	4.22E-04	2.08
	40	1.37E-06	6.14	4.56E-05	5.23	1.00E-06	4.24	4.56E-05	3.21
	80	5.84E-08	4.56	2.36E-06	4.27	5.59E-08	4.16	2.36E-06	4.27
	160	3.60E-09	4.02	2.04E-07	3.54	3.48E-09	4.01	2.04E-07	3.54
	320	2.28E-10	3.98	1.10E-08	4.22	2.18E-10	4.00	1.10E-08	4.22

3.1. Accuracy test. We first test the accuracy of the schemes on scalar and system problems. For all accuracy tests we have used the TVB minmod limiter with a small $M = 0.01$ to identify troubled cells (this is close to a TVD limiter with $M = 0$), resulting in many good cells identified as troubled cells. In this way we can clearly see the effect of the WENO reconstruction limiter on the accuracy of the RKDG method; namely, the order of accuracy is maintained after the application of this limiter. As a comparison, the RKDG method, with the TVB minmod limiter itself using such a small $M = 0.01$, would degenerate to second order accuracy in the L^1 norm (first order in the L^∞ norm) for these smooth problems.

We have tested many standard problems for accuracy, such as one- and two-dimensional linear advection, one- and two-dimensional nonlinear Burgers equations, and one- and two-dimensional nonlinear Euler equations. Both uniform and nonuniform meshes are used. The nonuniform meshes are obtained by randomly perturbing the cell boundaries of a uniform mesh within 10%. We present only the results of the one- and two-dimensional Burgers equations and two-dimensional nonlinear Euler equations on nonuniform meshes as representative examples.

Example 3.1. We solve the following nonlinear scalar Burgers equation:

$$u_t + \left(\frac{u^2}{2}\right)_x = 0 \tag{3.1}$$

with the initial condition $u(x,0) = 0.5+\sin(\pi x)$, with 2-periodic boundary conditions. When $t = 0.5/\pi$ the solution is still smooth. The errors and numerical orders of accuracy for the RKDG method with a WENO limiter compared to the original RKDG method without a limiter are shown in Table 1. We can see that the WENO limiter keeps both the designed order and the magnitude of accuracy of the original

TABLE 2

Burgers equation $u_t+(u^2/2)_x+(u^2/2)_y=0$. Initial condition $u(x,y,0)=0.5+\sin(\pi(x+y)/2)$ and periodic boundary conditions. RKDG with WENO limiter ($M=0.01$) compared to RKDG without limiter. Lax–Friedrichs flux. $t=0.5/\pi$. L_1 and L_∞ errors. Nonuniform meshes with $N\times N$ cells.

		DG with WENO limiter				DG with no limiter			
	$N\times N$	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
P^1	10×10	6.17E-02		6.33E-01		3.20E-02		3.51E-01	
	20×20	1.42E-02	2.12	2.51E-01	1.33	7.81E-03	2.04	1.16E-01	1.60
	40×40	3.03E-03	2.23	4.80E-02	2.39	1.92E-03	2.02	3.68E-02	1.65
	80×80	5.42E-04	2.48	1.02E-02	2.23	4.70E-04	2.03	1.02E-02	1.85
	160×160	1.26E-04	2.10	2.72E-03	1.91	1.16E-04	2.02	2.72E-03	1.91
	320×320	3.15E-05	2.01	6.99E-04	1.96	2.89E-05	2.01	6.99E-04	1.96
P^2	10×10	1.31E-02		1.75E-01		5.61E-03		1.69E-01	
	20×20	1.16E-03	3.49	4.95E-02	1.83	9.00E-04	2.64	4.94E-02	1.77
	40×40	1.31E-04	3.15	7.09E-03	2.80	1.22E-04	2.88	7.09E-03	2.80
	80×80	1.62E-05	3.01	1.12E-03	2.67	1.57E-05	2.96	1.12E-03	2.67
	160×160	2.05E-06	2.99	1.66E-04	2.75	1.98E-06	2.98	1.66E-04	2.75
	320×320	2.64E-07	2.95	2.27E-05	2.87	2.54E-07	2.97	2.27E-05	2.87
P^3	10×10	7.98E-03		1.75E-01		1.96E-03		8.66E-02	
	20×20	2.37E-04	5.07	1.32E-02	3.73	1.29E-04	3.93	1.04E-02	3.06
	40×40	9.36E-06	4.67	9.17E-04	3.84	9.23E-06	3.80	9.18E-04	3.51
	80×80	5.84E-07	4.00	8.08E-05	3.51	5.85E-07	3.98	8.08E-05	3.51
	160×160	3.71E-08	3.98	5.52E-06	3.87	3.71E-08	3.98	5.52E-06	3.87
	320×320	2.34E-09	3.99	3.58E-07	3.94	2.34E-09	3.99	3.58E-07	3.94

RKDG method.

Example 3.2. We solve the following nonlinear scalar Burgers equation in two dimensions:

$$u_t+\left(\frac{u^2}{2}\right)_x+\left(\frac{u^2}{2}\right)_y=0 \tag{3.2}$$

with the initial condition $u(x,y,0)=0.5+\sin(\pi(x+y)/2)$ and a 4-periodic boundary condition in both directions. When $t=0.5/\pi$ the solution is still smooth. The errors and numerical orders of accuracy for the RKDG method with a WENO limiter compared to the original RKDG method without a limiter are shown in Table 2. We can see that, again, the WENO limiter keeps both the designed order and the magnitude of accuracy of the original RKDG method.

Example 3.3. We solve the following nonlinear system of Euler equations:

$$\xi_t+f(\xi)_x+g(\xi)_y=0 \tag{3.3}$$

with

$$\xi=(\rho,\rho u,\rho v,E)^T,\quad f(\xi)=(\rho u,\rho u^2+p,\rho uv,u(E+p))^T,$$
$$g(\xi)=(\rho v,\rho uv,\rho v^2+p,v(E+p))^T.$$

Here ρ is the density, (u,v) is the velocity, E is the total energy, p is the pressure, which is related to the total energy by $E=\frac{p}{\gamma-1}+\frac{1}{2}\rho(u^2+v^2)$ with $\gamma=1.4$. The initial condition is set to be $\rho(x,y,0)=1+0.2\sin(\pi(x+y))$, $u(x,y,0)=0.7$, $v(x,y,0)=0.3$,

TABLE 3

Euler equations. Initial condition $\rho(x,y,0) = 1 + 0.2\sin(\pi(x+y))$, $u(x,y,0) = 0.7$, $v(x,y,0) = 0.3$, $p(x,y,0) = 1$ *and periodic boundary conditions. RKDG with WENO limiter* ($M = 0.01$) *compared to RKDG without limiter. Lax–Friedrichs flux.* $t = 2.0$. L_1 *and* L_∞ *errors for the density* ρ. *Nonuniform meshes with* $N \times N$ *cells.*

		DG with WENO limiter				DG with no limiter			
	$N \times N$	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
P^1	10×10	3.48E-02		7.34E-02		1.11E-02		3.15E-02	
	20×20	6.89E-03	2.34	2.74E-02	1.42	1.91E-03	2.55	1.04E-02	1.60
	40×40	1.21E-03	2.51	7.36E-03	1.89	3.70E-04	2.37	2.93E-03	1.83
	80×80	2.33E-04	2.37	2.02E-03	1.87	8.15E-05	2.18	7.82E-04	1.91
	160×160	5.19E-05	2.17	6.45E-04	1.65	1.93E-05	2.08	2.04E-04	1.94
P^2	10×10	1.26E-03		8.22E-03		5.95E-04		8.89E-03	
	20×20	9.97E-05	3.66	1.21E-03	2.76	6.88E-05	3.11	1.18E-03	2.91
	40×40	9.61E-06	3.38	1.50E-04	3.02	8.41E-06	3.03	1.50E-04	2.98
	80×80	1.10E-06	3.12	1.90E-05	2.98	1.04E-06	3.01	1.90E-05	2.98
	160×160	1.34E-07	3.04	2.40E-06	2.98	1.30E-07	3.00	2.40E-06	2.98
P^3	10×10	8.10E-04		2.04E-03		4.35E-05		9.88E-04	
	20×20	6.68E-06	6.92	5.44E-05	5.23	2.73E-06	3.99	7.04E-05	3.81
	40×40	5.24E-07	3.67	6.24E-06	3.12	1.71E-07	4.00	4.43E-06	3.99
	80×80	3.44E-08	3.93	5.69E-07	3.46	1.07E-08	4.00	2.88E-07	3.94
	160×160	2.05E-09	4.07	5.05E-08	3.49	6.66E-10	4.00	1.86E-08	3.96

$p(x,y,0) = 1$, with a 2-periodic boundary condition. The exact solution is $\rho(x,y,t) = 1 + 0.2\sin(\pi(x+y-(u+v)t))$, $u = 0.7$, $v = 0.3$, $p = 1$. We compute the solution up to $t = 2$. The errors and numerical orders of accuracy for the RKDG method with a WENO limiter compared to the original RKDG method without a limiter are shown in Table 3. We can see that, again, the WENO limiter keeps both the designed order and the magnitude of accuracy of the original RKDG method.

3.2. Test cases with shocks. We now test the performance of the RKDG method with WENO limiters for problems containing shocks. We have used both uniform and nonuniform meshes in the numerical experiments, obtaining similar results. We will only show results with uniform meshes. We have also computed many more problems, such as the two-dimensional forward facing step problem, but will not present all the results. For a direct comparison with the RKDG method using the original minmod TVB limiter, we refer to the results in [8, 7, 5, 10]. In general, the results are comparable when M is chosen adequately. When M is chosen too small, however, the RKDG method with the new limiter produces much better results than with the original minmod TVB limiter.

Example 3.4. We solve the same nonlinear Burgers equation (3.1) as in Example 3.1 with the same initial condition $u(x,0) = 0.5 + \sin(\pi x)$, except that we now plot the results at $t = 1.5/\pi$ when a shock has already appeared in the solution. In Figure 1, the solutions of RKDG with a WENO limiter using $N = 80$ cells are shown. We can see that schemes of all orders give good nonoscillatory shock transitions for this problem.

Example 3.5. We solve the nonlinear nonconvex scalar Buckley–Leverett problem

$$u_t + \left(\frac{4u^2}{4u^2 + (1-u)^2}\right)_x = 0 \tag{3.4}$$

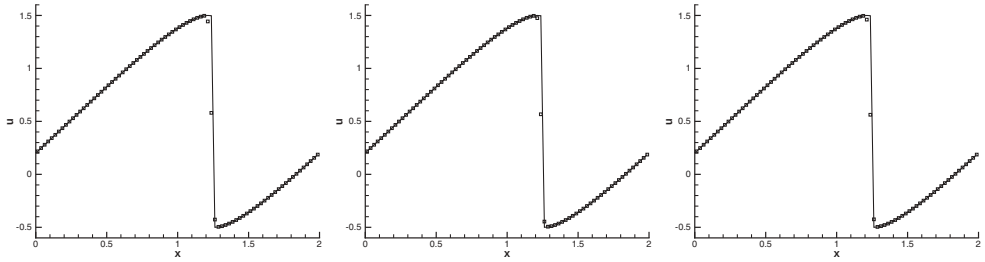

FIG. 1. *Burgers equation.* $u(x,0) = 0.5 + \sin(\pi x)$. $t = 1.5/\pi$. $N = 80$ *cells. RKDG with WENO limiters. Solid line: the exact solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

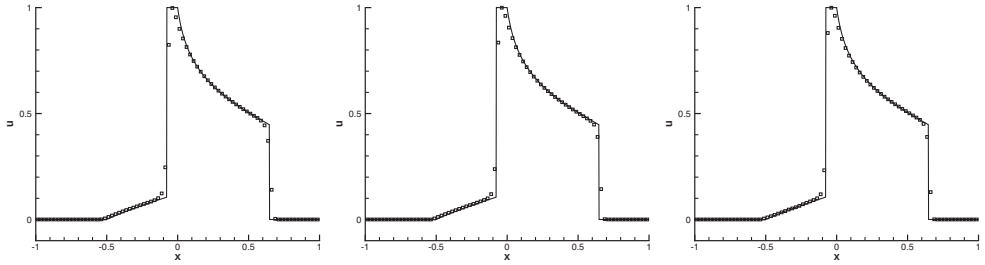

FIG. 2. *The Buckley–Leverett problem.* $t = 0.4$. $N = 80$ *cells. RKDG with WENO limiters. Solid line: the exact solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

with the initial data $u = 1$ when $-\frac{1}{2} \le x \le 0$ and $u = 0$ elsewhere. The solution is computed up to $t = 0.4$. The exact solution is a shock-rarefaction-contact discontinuity mixture. We remark that some high order schemes may fail to converge to the correct entropy solution for this problem. In Figure 2, the solutions of RKDG with a WENO limiter using $N = 80$ cells are shown. We can see that schemes of all orders give good nonoscillatory resolutions to the correct entropy solution for this problem.

Example 3.6. We solve the following one-dimensional nonlinear system of Euler equations:

$$u_t + f(u)_x = 0 \tag{3.5}$$

with

$$u = (\rho, \rho v, E)^T, \quad f(u) = (\rho v, \rho v^2 + p, v(E + p))^T.$$

Here ρ is the density, v is the velocity, E is the total energy, and p is the pressure, related to the total energy by $E = \frac{p}{\gamma - 1} + \frac{1}{2}\rho v^2$ with $\gamma = 1.4$.

We use the following Riemann initial condition for the Lax problem:

$$(\rho, v, p) = (0.445, 0.698, 3.528) \quad \text{for } x \le 0; \qquad (\rho, v, p) = (0.5, 0, 0.571) \quad \text{for } x > 0.$$

The computed density ρ is plotted at $t = 1.3$ against the exact solution. In this example we explore the effect of the TVB constant M in the minmod limiter to identify troubled cells. We observe that, with an increased M, we have fewer cells identified as troubled cells and subject to WENO limiting, and the resolution of the contact discontinuity improves with an increased M, indicating that the RKDG

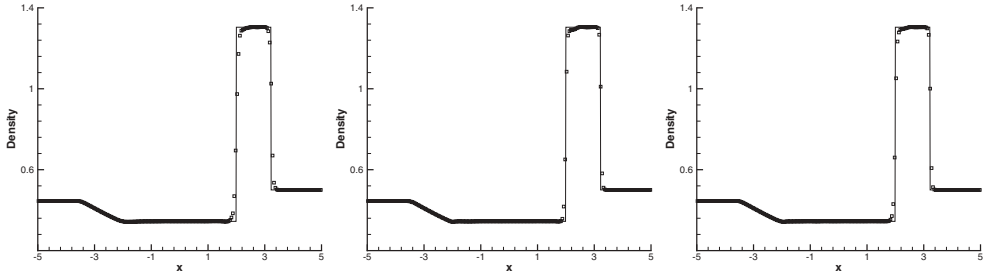

FIG. 3. *Lax problem.* $t = 1.3$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 0.01$. *Density. Solid line: the exact solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

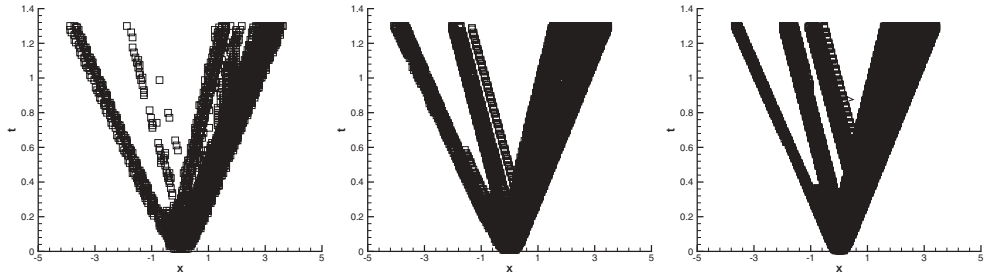

FIG. 4. *Lax problem.* $t = 1.3$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 0.01$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

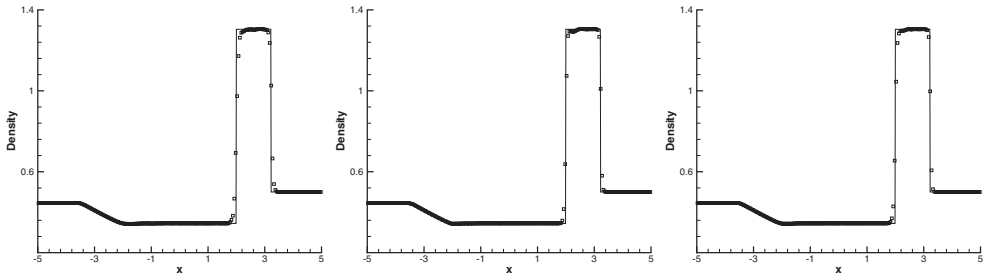

FIG. 5. *Lax problem.* $t = 1.3$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 1$. *Density. Solid line: the exact solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

method does a better job than the WENO reconstruction in keeping sharp contact discontinuities. Thus we might want to choose a larger M within the range allowed by stability to minimize the number of troubled cells subject to WENO limiting, both to save computational cost and to improve resolution at contact discontinuities. In Figures 3–8, we plot the densities by RKDG with WENO limiters using $N = 200$ cells, and the time history of the "troubled cells," for the $M = 0.01$, $M = 1$, and $M = 50$ cases.

Example 3.7. A higher order scheme would show its advantage when the solution contains both shocks and complex smooth region structures. A typical example for this is the problem of shock interaction with entropy waves [25]. We solve the Euler equations (3.5) with a moving Mach = 3 shock interacting with sine waves in density,

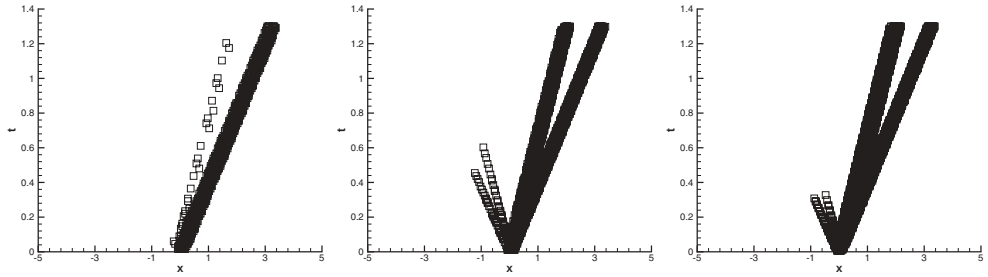

FIG. 6. *Lax problem.* $t = 1.3$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 1$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

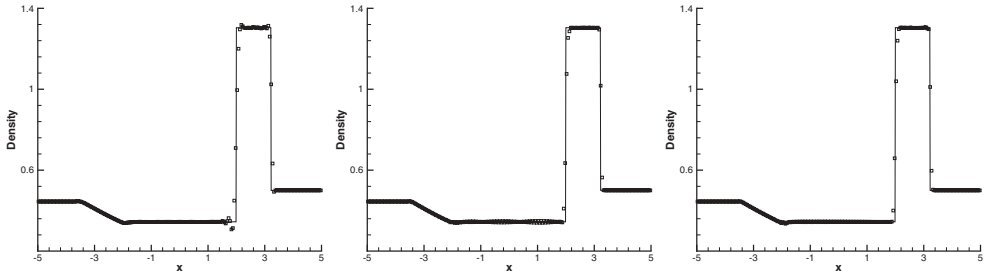

FIG. 7. *Lax problem.* $t = 1.3$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 50$. *Density. Solid line: the exact solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

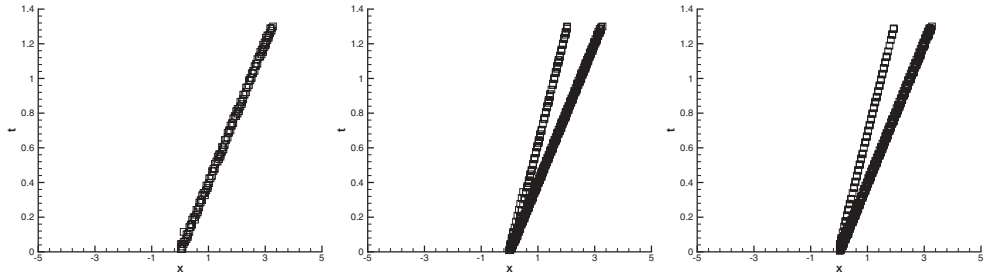

FIG. 8. *Lax problem.* $t = 1.3$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 50$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

i.e., initially

$$\begin{aligned} (\rho, v, p) &= (3.857143, 2.629369, 10.333333) \quad \text{for } x < -4; \\ (\rho, v, p) &= (1 + \varepsilon \sin(5x), 0, 1) \qquad\qquad\quad \text{for } x \geq -4. \end{aligned}$$

Here we take $\varepsilon = 0.2$. The computed density ρ is plotted at $t = 1.8$ against the referenced "exact" solution, which is a converged solution computed by the fifth order finite difference WENO scheme [17] with 2000 grid points.

We again explore the effect of the TVB constant M in the minmod limiter to identify troubled cells. As before, we observe that, with an increased M, we have fewer cells identified as troubled cells and subject to WENO limiting, and the resolution of the detailed structures in the solution behind the shock improves with an increased M,

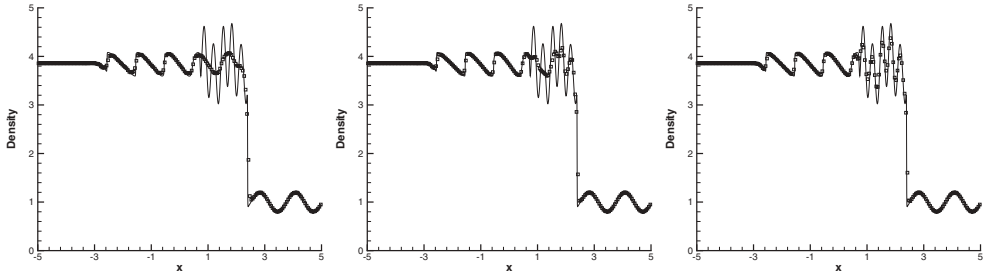

FIG. 9. *The shock density wave interaction problem.* $t = 1.8$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 0.01$. *Density. Solid line: the "exact" solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

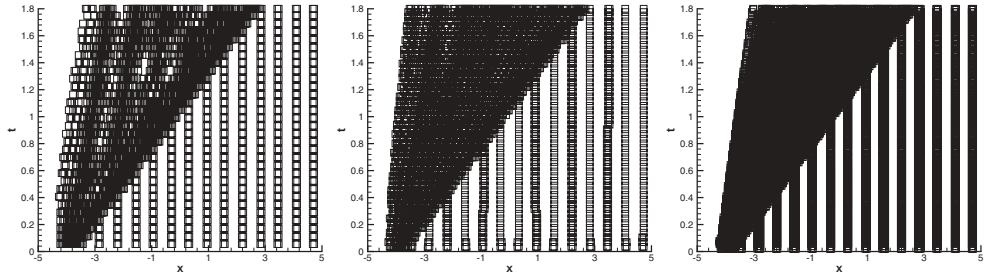

FIG. 10. *The shock density wave interaction problem.* $t = 1.8$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 0.01$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

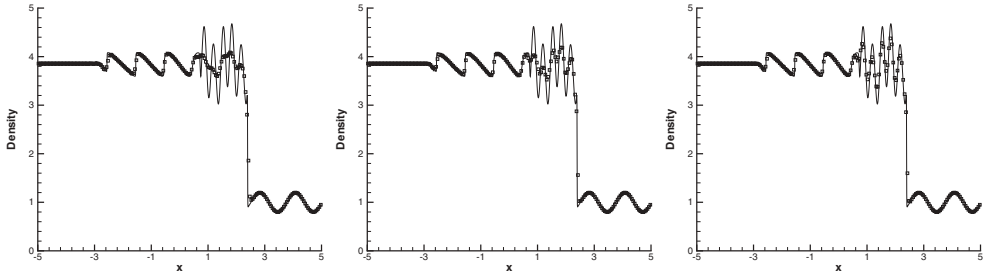

FIG. 11. *The shock density wave interaction problem.* $t = 1.8$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 10$. *Density. Solid line: the "exact" solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

indicating that the RKDG method does a better job than the WENO reconstruction in keeping such resolution. Thus we might want to choose a larger M within the range allowed by stability to minimize the number of troubled cells subject to WENO limiting, both to save computational cost and to improve resolution. In Figures 9–14, we plot the densities by RKDG with WENO limiters using $N = 200$ cells, and the time history of the "troubled cells," for the $M = 0.01$, $M = 10$, and $M = 300$ cases. We note that when $M = 300$, no "troubled cells" have been identified near the weak shocks in Figure 14, thus introducing small spurious oscillations near the weak shocks in Figure 13. This indicates that $M = 300$ might be too large.

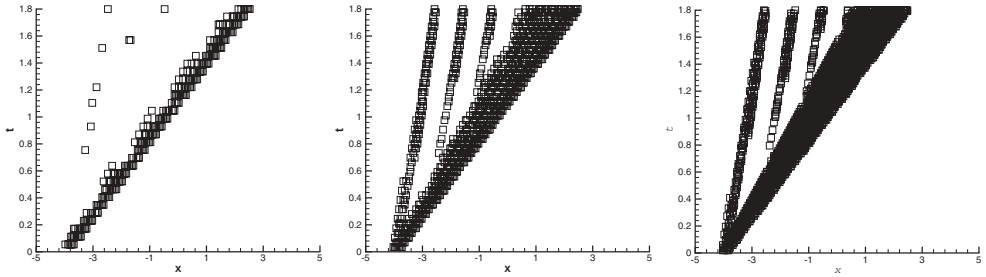

FIG. 12. *The shock density wave interaction problem.* $t = 1.8$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 10$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

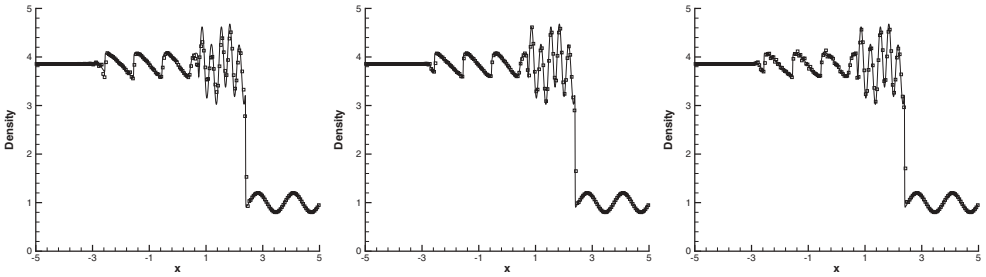

FIG. 13. *The shock density wave interaction problem.* $t = 1.8$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 300$. *Density. Solid line: the "exact" solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

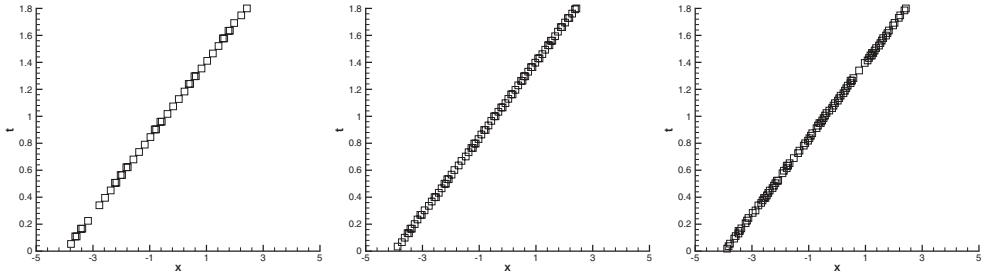

FIG. 14. *The shock density wave interaction problem.* $t = 1.8$. *RKDG with WENO limiters.* 200 *cells. TVB constant* $M = 300$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

Example 3.8. We consider the interaction of blast waves of Euler equation (3.5) with the initial conditions

$$\begin{aligned}
(\rho, v, p) &= (1, 0, 1000) \quad \text{for } 0 \le x < 0.1; \\
(\rho, v, p) &= (1, 0, 0.01) \quad \text{for } 0.1 \le x < 0.9; \\
(\rho, v, p) &= (1, 0, 100) \quad \text{for } 0.9 \le x.
\end{aligned}$$

A reflecting boundary condition is applied to both ends. See [26, 14]. The computed density ρ is plotted at $t = 0.038$ against the reference "exact" solution, which is a converged solution computed by the fifth order finite difference WENO scheme [17] with 2000 grid points.

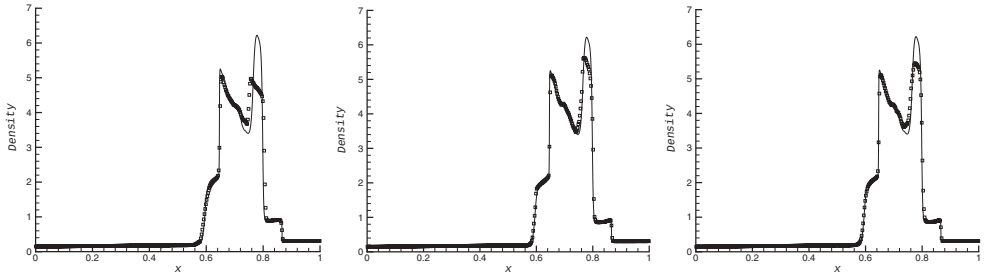

FIG. 15. *The blast wave problem.* $t = 0.038$. *RKDG with WENO limiters.* 400 *cells. TVB constant* $M = 0.01$. *Density. Solid line: the "exact" solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

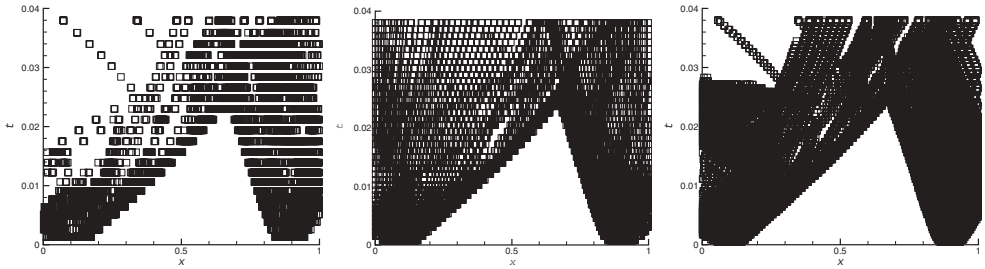

FIG. 16. *The blast wave problem.* $t = 0.038$. *RKDG with WENO limiters.* 400 *cells. TVB constant* $M = 0.01$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

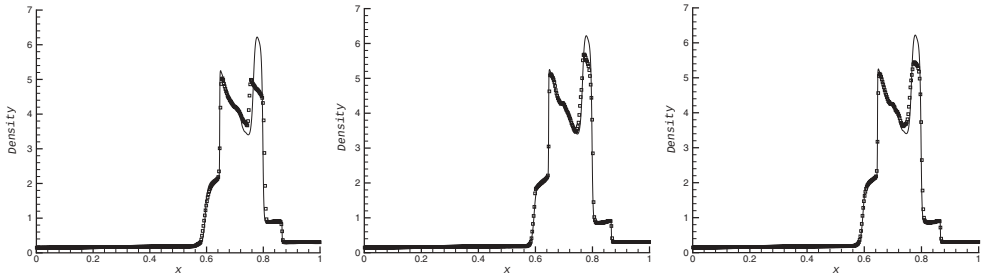

FIG. 17. *The blast wave problem.* $t = 0.038$. *RKDG with WENO limiters.* 400 *cells. TVB constant* $M = 10$. *Density. Solid line: the "exact" solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

In Figures 15–20, we plot the densities by RKDG with WENO limiters using $N = 400$ cells, and the time history of the "troubled cells," for the $M = 0.01$, $M = 10$, and $M = 300$ cases. As before, we explore the effect of the TVB constant M in the minmod limiter to identify troubled cells. We observe the same pattern as before, namely, with an increased M we have fewer cells identified as troubled cells and subject to WENO limiting, and the resolution of the numerical solution improves with an increased M up to a certain value (compare, for example, the resolution of $M = 10$ in Figure 17 with that of $M = 0.01$ in Figure 15), indicating that the RKDG method does a better job than the WENO reconstruction in keeping such resolution. Thus we might want to choose a larger M within the range allowed by stability to minimize the number of troubled cells subject to WENO limiting, both to save computational

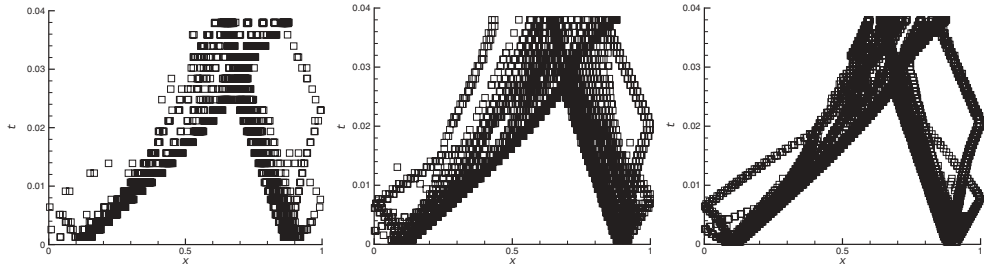

FIG. 18. *The blast wave problem.* $t = 0.038$. *RKDG with WENO limiters.* 400 *cells. TVB constant* $M = 10$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

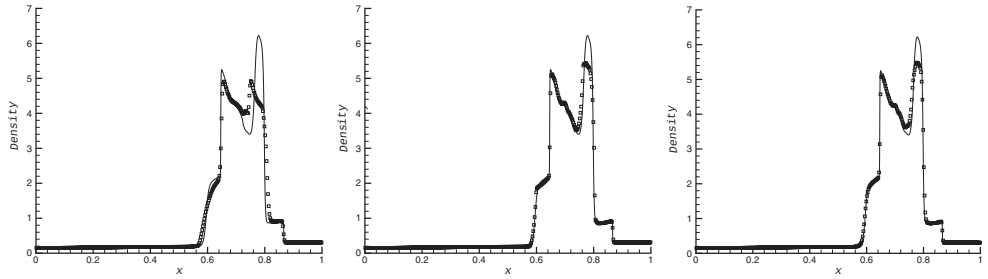

FIG. 19. *The blast wave problem.* $t = 0.038$. *RKDG with WENO limiters.* 400 *cells. TVB constant* $M = 300$. *Density. Solid line: the "exact" solution. Squares: numerical solution (one point per cell). Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

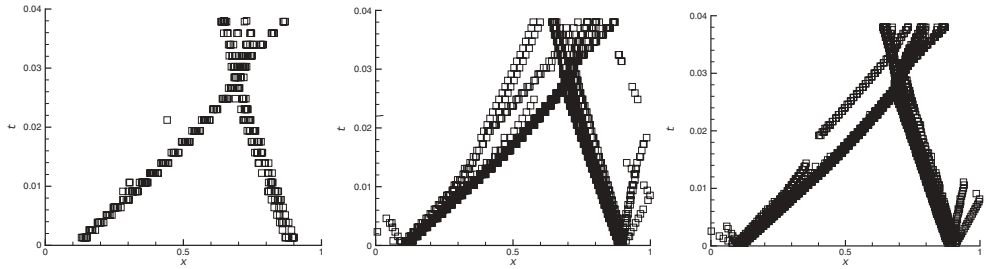

FIG. 20. *The blast wave problem.* $t = 0.038$. *RKDG with WENO limiters.* 400 *cells. TVB constant* $M = 300$. *Time history of the "troubled cells." Squares denote cells which are identified as "troubled cells" subject to WENO limiting. Left:* $k = 1$. *Middle:* $k = 2$. *Right:* $k = 3$.

cost and to improve resolution. However, if M is chosen too large, the improvement of resolution is not clear for this example. There is even some degradation of resolution in Figure 19 for $M = 300$ compared with Figure 17 for $M = 10$. We also notice that the resolution improves from P^1 to P^2, but deteriorates slightly from P^2 to P^3 for this problem.

Example 3.9. *Double Mach reflection.* This problem is originally from [26]. The computational domain for this problem is $[0, 4] \times [0, 1]$. The reflecting wall lies at the bottom, starting from $x = \frac{1}{6}$. Initially a right-moving Mach 10 shock is positioned 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

TABLE 4
Percentage of troubled cells subject to WENO limiters in the double Mach reflection problem.

Schemes	$k=1$		$k=2$		$k=3$	
M	0.01	100	0.01	100	0.01	100
480×120	14.24	3.05	20.75	4.93	25.52	6.97
960×240	10.80	2.18	18.91	4.80	23.65	8.84
1920×480	8.70	1.65	17.40	5.96	21.99	11.68

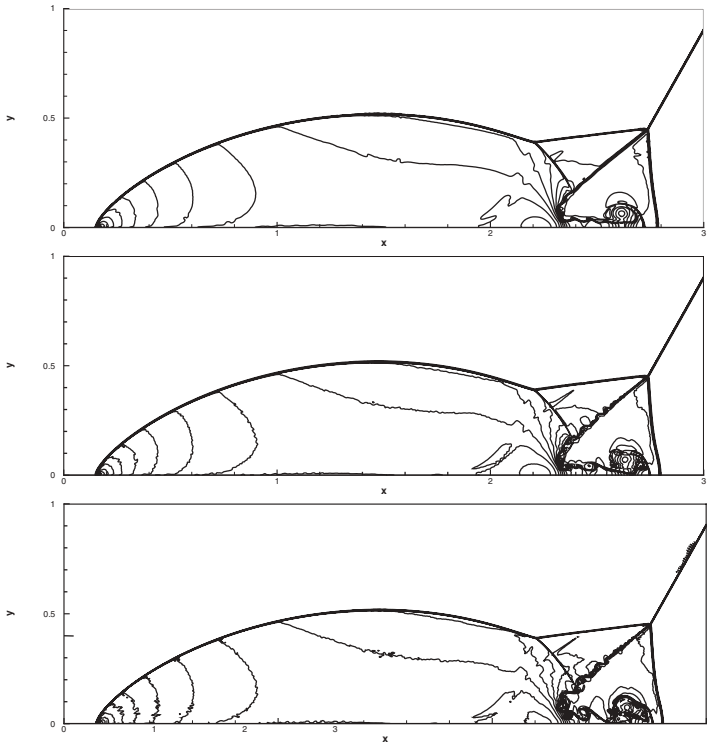

FIG. 21. *Double Mach reflection problem. RKDG with WENO limiters.* 1920×480 *cells. TVB constant* $M = 0.01$. *Thirty equally spaced density contours from* 1.5 *to* 22.7. *Top:* $k = 1$. *Middle:* $k = 2$. *Bottom:* $k = 3$.

to describe the exact motion of a Mach 10 shock. We compute the solution up to $t = 0.2$. In Table 4, we document the percentage of cells declared to be "troubled cells" for different orders of accuracy and different TVB constants M in the minmod limiter. We can see that only a small percentage of cells are declared "troubled cells." Three different uniform meshes, with 480×120, 960×240, and 1920×480 cells, and three different orders of accuracy for the RKDG with WENO limiters, from $k = 1$ to $k = 3$ (second to fourth order), as well as three different values of the TVB constant, $M = 0.01$, $M = 1$, and $M = 100$, are used in the numerical experiments. In Figures 21 and 22, we show only the simulation results on the most refined mesh with 1920×480 cells with $M = 0.01$ and $M = 100$, and in Figures 23 and 24 show the more detailed "zoomed-in" figures around the double Mach stem for the cases with 960×240 and

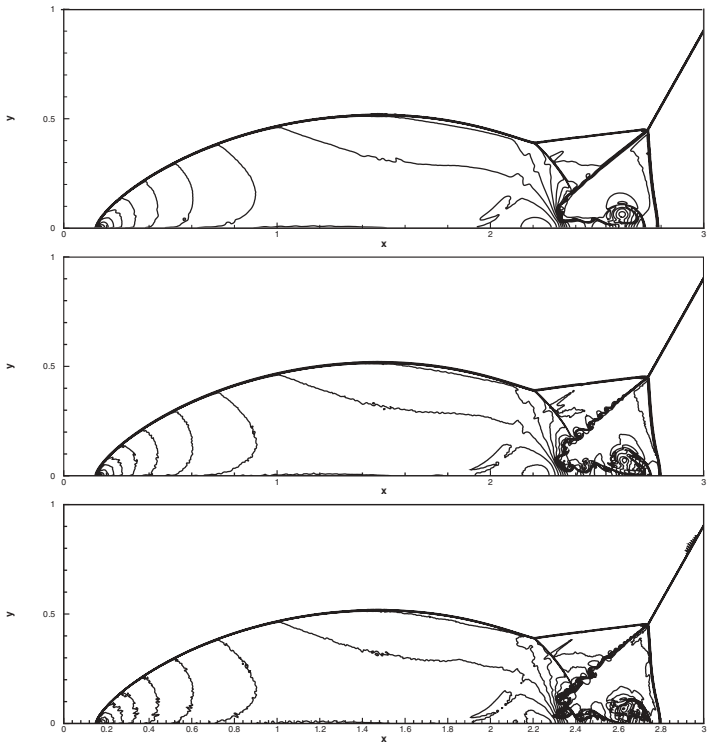

FIG. 22. *Double Mach reflection problem. RKDG with WENO limiters.* 1920×480 *cells. TVB constant* $M = 100$. *Thirty equally spaced density contours from* 1.5 *to* 22.7. *Top:* $k = 1$. *Middle:* $k = 2$. *Bottom:* $k = 3$.

1920×480 cells with $M = 0.01$ and $M = 100$. All the figures show 30 equally spaced density contours from 1.5 to 22.7. The results are comparable with those obtained in [10] using TVB minmod limiters with $M = 50$. Clearly, the resolution improves with an increasing k on the same mesh. Also, the resolution is slightly better for $M = 100$ than for $M = 0.01$; however, this difference is not significant.

4. Concluding remarks. We have developed a new limiter for the RKDG methods solving hyperbolic conservation laws using finite volume high order WENO reconstructions. The idea is to first identify troubled cells subject to the WENO limiting, using a TVB minmod-type limiter, then reconstruct the polynomial solution inside the troubled cells by WENO reconstruction using the cell averages of neighboring cells, while maintaining the original cell averages of the troubled cells. Numerical results are provided to show that the method is stable, accurate, and robust in maintaining accuracy. Improving the procedure for identifying troubled cells, improving the WENO reconstruction using more compact stencils (Hermite-type reconstructions), using ENO rather than WENO for the reconstruction (as the linear weights often fail to exist for some choices of the reconstruction [19]), and implementing the method for more general meshes and three-dimensional problems constitute ongoing research.

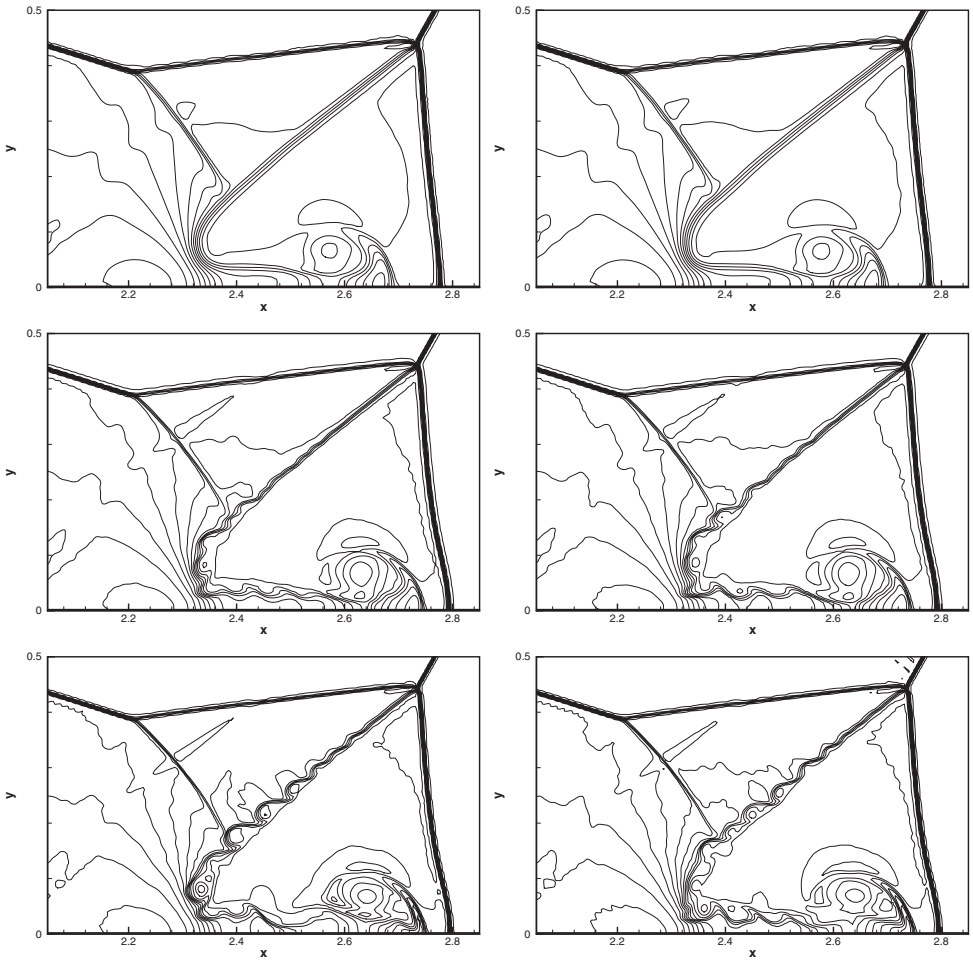

FIG. 23. *Double Mach reflection problem. RKDG with WENO limiters.* 960×240 *cells. Zoomed-in region shown for more details. TVB constant* $M = 0.01$ *(left) and* $M = 100$ *(right). Thirty equally spaced density contours from* 1.5 *to* 22.7. *Top:* $k = 1$. *Middle:* $k = 2$. *Bottom:* $k = 3$.

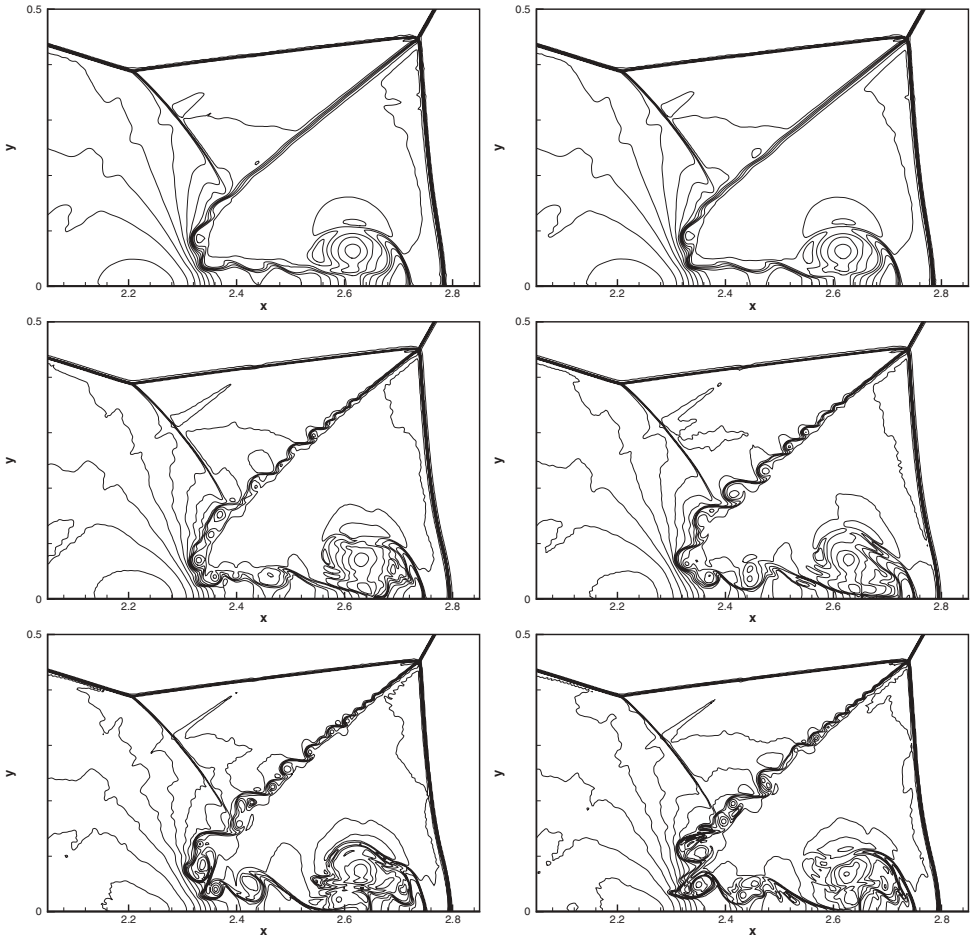

FIG. 24. *Double Mach reflection problem. RKDG with WENO limiters.* 1920×480 *cells. Zoomed-in region shown for more details. TVB constant* $M = 0.01$ *(left) and* $M = 100$ *(right). Thirty equally spaced density contours from* 1.5 *to* 22.7. *Top:* $k = 1$. *Middle:* $k = 2$. *Bottom:* $k = 3$.

REFERENCES

[1] D.S. Balsara and C.-W. Shu, *Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy*, J. Comput. Phys., 160 (2000), pp. 405–452.

[2] R. Biswas, K.D. Devine, and J. Flaherty, *Parallel, adaptive finite element methods for conservation laws*, Appl. Numer. Math., 14 (1994), pp. 255–283.

[3] A. Burbeau, P. Sagaut, and C.H. Bruneau, *A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods*, J. Comput. Phys., 169 (2001), pp. 111–150.

[4] B. Cockburn, *Discontinuous Galerkin methods for convection-dominated problems*, in High-Order Methods for Computational Physics, T.J. Barth and H. Deconinck, eds., Lecture Notes in Comput. Sci. Engrg. 9, Springer, New York, 1999, pp. 69–224.

[5] B. Cockburn, S. Hou, and C.-W. Shu, *The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws* IV: *The multidimensional case*, Math. Comp., 54 (1990), pp. 545–581.

[6] B. Cockburn, G. Karniadakis, and C.-W. Shu, *The development of discontinuous Galerkin methods*, in Discontinuous Galerkin Methods: Theory, Computation and Applications, B. Cockburn, G. Karniadakis, and C.-W. Shu, eds., Lecture Notes in Comput. Sci. Engrg. 11, Springer, Berlin, 2000, pp. 3–50.

[7] 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*, J. Comput. Phys., 84 (1989), pp. 90–113.

[8] B. Cockburn and C.-W. Shu, *TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws* II: *General framework*, Math. Comp., 52 (1989), pp. 411–435.

[9] B. Cockburn and C.-W. Shu, *The Runge-Kutta local projection $P1$-discontinuous Galerkin finite element method for scalar conservation laws*, RAIRO Modél. Math. Anal. Numér., 25 (1991), pp. 337–361.

[10] B. Cockburn and C.-W. Shu, *The Runge-Kutta discontinuous Galerkin method for conservation laws* V: *Multidimensional systems*, J. Comput. Phys., 141 (1998), pp. 199–224.

[11] B. Cockburn and C.-W. Shu, *Runge-Kutta discontinuous Galerkin method for convection-dominated problems*, J. Sci. Comput., 16 (2001), pp. 173–261.

[12] O. Friedrichs, *Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids*, J. Comput. Phys., 144 (1998), pp. 194–212.

[13] A. Harten, *High resolution schemes for hyperbolic conservation laws*, J. Comput. Phys., 49 (1983), pp. 357–393.

[14] A. Harten, B. Engquist, S. Osher, and S. Chakravathy, *Uniformly high order accurate essentially non-oscillatory schemes.* III, J. Comput. Phys., 71 (1987), pp. 231–303.

[15] J.S. Hesthaven and C.-H. Teng, *Stable spectral methods on tetrahedral elements*, SIAM J. Sci. Comput., 21 (2000), pp. 2352–2380.

[16] C. Hu and C.-W. Shu, *Weighted essentially non-oscillatory schemes on triangular meshes*, J. Comput. Phys., 150 (1999), pp. 97–127.

[17] G. Jiang and C.-W. Shu, *Efficient implementation of weighted ENO schemes*, J. Comput. Phys., 126 (1996), pp. 202–228.

[18] X. Liu, S. Osher, and T. Chan, *Weighted essentially non-oscillatory schemes*, J. Comput. Phys., 115 (1994), pp. 200–212.

[19] J. Qiu and C.-W. Shu, *On the construction, comparison, and local characteristic decomposition for high order central WENO schemes*, J. Comput. Phys., 183 (2002), pp. 187–209.

[20] W.H. Reed and T.R. Hill, *Triangular Mesh Methods for the Neutron Transport Equation*, Tech. report LA-UR-73-479, Los Alamos Scientific Laboratory, Los Alamos, NM, 1973.

[21] J. Shi, C. Hu, and C.-W. Shu, *A technique of treating negative weights in WENO schemes*, J. Comput. Phys., 175 (2002), pp. 108–127.

[22] C.-W. Shu, *TVB uniformly high-order schemes for conservation laws*, Math. Comp., 49 (1987), pp. 105–121.

[23] 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 (A. Quarteroni, ed.), Lecture Notes in Math. 1697, Springer, Berlin, 1998, pp. 325–432.

[24] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock-capturing schemes*, J. Comput. Phys., 77 (1988), pp. 439–471.

[25] C.-W. Shu and S. Osher, *Efficient implementation of essentially non-oscillatory shock capturing schemes* II, J. Comput. Phys., 83 (1989), pp. 32–78.

[26] P. Woodward and P. Colella, *The numerical simulation of two-dimensional fluid flow with strong shocks*, J. Comput. Phys., 54 (1984), pp. 115–173.