Available online at www.sciencedirect.com
SCIENCE DIRECT®
Computers & Fluids 34 (2005) 642–663
computers & fluids
www.elsevier.com/locate/compfluid

Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: Two dimensional case

Jianxian Qiu [a,b,1], Chi-Wang Shu [c,*,2]

[a] *Department of Computational Science, National University of Singapore, Singapore 117543, Singapore*
[b] *Department of Mechanical Engineering, National University of Singapore, Singapore 119260, Singapore*
[c] *Division of Applied Mathematics, Brown University, Providence, RI 02912, USA*

Received 17 December 2003; received in revised form 1 May 2004; accepted 20 May 2004
Available online 11 September 2004

Abstract

A class of fifth-order weighted essentially non-oscillatory (WENO) schemes based on Hermite polynomials, termed HWENO (Hermite WENO) schemes, for solving one dimensional non-linear hyperbolic conservation law systems, was developed and applied as limiters for the Runge–Kutta discontinuous Galerkin (RKDG) methods in [J. Comput. Phys. 193 (2003) 115]. In this paper, we extend the method to solve two dimensional non-linear hyperbolic conservation law systems. The emphasis is again on the application of such HWENO finite volume methodology as limiters for RKDG methods to maintain compactness of RKDG methods. Numerical experiments for two dimensional Burgers' equation and Euler equations of compressible gas dynamics are presented to show the effectiveness of these methods.

MSC: 65M06; 65M60; 65M99; 35L65

* Corresponding author. Tel.: +1 401 863 2549; fax: +1 401 863 1355.
E-mail addresses: mpeqjx@nus.edu.sg (J. Qiu), shu@dam.brown.edu (C.-W. Shu).
[1] The research of this author is supported by NUS Research Project R-265-000-118-112.
[2] The research of this author is supported by NNSFC grant 10028103 while he is in residence at the Department of Mathematics, University of Science and Technology of China, Hefei, Anhui 230026, P.R. China. Additional support is provided by ARO grant DAAD19-00-1-0405 and NSF grant DMS-0207451.

0045-7930/$ - see front matter
doi:10.1016/j.compfluid.2004.05.005

1. Introduction

In [22], we constructed a class of fifth-order weighted essentially non-oscillatory (WENO) schemes based on Hermite polynomials, termed HWENO (Hermite WENO) schemes, for solving one dimensional non-linear hyperbolic conservation law systems, and applied this HWENO finite volume methodology as limiters for the Runge–Kutta discontinuous Galerkin (RKDG) methods. In this paper, we extend the method to solve two dimensional non-linear hyperbolic conservation law systems:

$$\begin{cases} u_t + f(u)_x + g(u)_y = 0, \\ u(x,y,0) = u_0(x,y). \end{cases} \tag{1.1}$$

WENO schemes have been designed in recent years as a class of high order finite volume or finite difference schemes to solve hyperbolic conservation laws with the property of maintaining both uniform high order accuracy and an essentially non-oscillatory shock transition. We have the third-order finite volume WENO schemes in one space dimension in [18], the third and fifth-order finite difference WENO schemes in multi-space dimensions with a general framework for the design of the smoothness indicators and non-linear weights in [16], finite difference WENO schemes of higher orders (seventh to eleventh order) in [1], and finite volume WENO schemes on unstructured and structured meshes in [12,15,17,20,24]. WENO schemes are designed based on the successful ENO schemes in [14,27,28]. Both ENO and WENO schemes use the idea of adaptive stencils in the reconstruction procedure based on the local smoothness of the numerical solution to automatically achieve high order accuracy and a non-oscillatory property near discontinuities. ENO uses just one (optimal in some sense) out of many candidate stencils when doing the reconstruction; while WENO uses a convex combination of all the candidate stencils, each being assigned a non-linear weight which depends on the local smoothness of the numerical solution based on that stencil. WENO improves upon ENO in robustness, better smoothness of fluxes, better steady state convergence, better provable convergence properties, and more efficiency. For a detailed review of ENO and WENO schemes, we refer to the lecture notes [26].

The main difference between the one dimensional Hermite WENO scheme designed in [22], see also related earlier work in [3,11,19,29], and the traditional WENO schemes is that the former has a more compact stencil than the latter for the same order of accuracy. This compactness is achieved by evolving both the function and its first derivative values in time and they are both used in the reconstruction in HWENO schemes. As a result, a fifth order one dimensional HWENO reconstruction uses only 3 points, while a fifth-order one dimensional WENO reconstruction would need to use 5 points.

One major emphasis of the HWENO methodology in [22] is its application as limiters for the RKDG (Runge–Kutta discontinuous Galerkin) methods. The discontinuous Galerkin (DG) method is a finite element method which evolves k degrees of freedom (in one dimension) per cell for a kth order accurate scheme, thus no reconstruction is needed. We refer to, e.g. [7,9,10,23] for the detailed description of the RKDG methods. An important component of RKDG methods for solving conservation laws (1.1) with strong shocks in the solutions is a non-linear limiter, which is applied to control spurious oscillations. Although many limiters

exist in the literature, e.g. [2,4–9], they tend to degenerate accuracy when mistakenly used in smooth regions of the solution. In [21], we initialized a study of using WENO methodology as limiters for RKDG methods. The idea is to first identify "troubled cells", namely those cells where limiting might be needed, then to abandon all moments in those cells except the cell averages and reconstruct those moments from the information of neighboring cells using a WENO methodology. This technique works quite well in our one and two dimensional test problems [21]. In [22], this approach is further improved by using HWENO rather than WENO methodology in the limiter so that a more compact stencil is used.

In this paper, we extend the HWENO methodology in [22] to solve two dimensional non-linear hyperbolic conservation law systems. The emphasis is again on the application of such HWENO finite volume methodology as the limiters for RKDG methods to maintain compactness of RKDG methods. Numerical experiments for two dimensional Burgers' equation and Euler equations of compressible gas dynamics are presented to show the effectiveness of these methods.

The organization of this paper is as follows. In Section 2, we describe in detail the construction and implementation of HWENO schemes with Runge–Kutta time discretizations, for two dimensional scalar and system equation (1.1). In Section 3, we investigate the usage of the HWENO finite volume methodology as limiters for RKDG methods, following the idea in [21,22], with the goal of obtaining a robust and high order limiting procedure to simultaneously obtain uniform high order accuracy and sharp, non-oscillatory shock transition for RKDG methods. In Section 4 we provide extensive numerical examples to demonstrate the behavior of the HWENO schemes and DG methods with HWENO limiters with Runge–Kutta time discretizations. Concluding remarks are given in Section 5.

2. The construction of Hermite WENO schemes

In this section we first consider two dimensional scalar conservation laws (1.1). For simplicity of presentation, we assume that the mesh is uniform with the cell size $x_{i+\frac{1}{2}} - x_{i-\frac{1}{2}} = \Delta x$, $y_{j+\frac{1}{2}} - y_{j-\frac{1}{2}} = \Delta y$ and cell centers $(x_i, y_j) = (\frac{1}{2}(x_{i+\frac{1}{2}} + x_{i-\frac{1}{2}}), \frac{1}{2}(y_{j+\frac{1}{2}} + y_{j-\frac{1}{2}}))$. We also denote the cells by $I_{ij} = [x_{i-\frac{1}{2}}, x_{i+\frac{1}{2}}] \times [y_{j-\frac{1}{2}}, y_{j+\frac{1}{2}}]$. Let $v = \frac{\partial u}{\partial x}$, $w = \frac{\partial u}{\partial y}$. Taking the derivatives of (1.1), we obtain:

$$\begin{cases} v_t + h_x + r_y = 0, \\ v(x, y, 0) = \frac{\partial u_0(x,y)}{\partial x}, \end{cases} \tag{2.1}$$

$$\begin{cases} w_t + q_x + s_y = 0, \\ w(x, y, 0) = \frac{\partial u_0(x,y)}{\partial y}, \end{cases} \tag{2.2}$$

where

$$h(u, v) = f'(u)v, \qquad r(u, v) = g'(u)v,$$

$$q(u, w) = f'(u)w, \qquad s(u, w) = g'(u)w.$$

We integrate Eqs. (1.1), (2.1) and (2.2) on a control volume I_{ij} to obtain the semi-discrete finite volume scheme as:

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{ij}^{(0)} = -\frac{1}{\Delta x\Delta y}\int_{\partial I_{ij}} F\cdot n\,\mathrm{d}s, \tag{2.3}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{ij}^{(1)} = -\frac{1}{\Delta y}\int_{\partial I_{ij}} H\cdot n\,\mathrm{d}s, \tag{2.4}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}u_{ij}^{(2)} = -\frac{1}{\Delta x}\int_{\partial I_{ij}} Q\cdot n\,\mathrm{d}s, \tag{2.5}$$

where

$$u_{ij}^{(0)} = \frac{1}{\Delta x\Delta y}\int_{I_{ij}} u\,\mathrm{d}x\,\mathrm{d}y, \quad u_{ij}^{(1)} = \frac{1}{\Delta y}\int_{I_{ij}} \frac{\partial u}{\partial x}\,\mathrm{d}x\,\mathrm{d}y, \quad u_{ij}^{(2)} = \frac{1}{\Delta x}\int_{I_{ij}} \frac{\partial u}{\partial y}\,\mathrm{d}x\,\mathrm{d}y$$

and

$$F = (f,g)^{\mathrm{T}}, \qquad H = (h,r)^{\mathrm{T}}, \qquad Q = (q,s)^{\mathrm{T}}.$$

The line integrals in (2.3), (2.4) and (2.5) are discretized by a q-point Gaussian integration formula

$$\int_{\partial I_{ij}} F\cdot n\,\mathrm{d}s \approx |\partial I_{ij}|\sum_{l=1}^{q}\omega_l F(u(G_l,t))\cdot n, \tag{2.6}$$

$$\int_{\partial I_{ij}} H\cdot n\,\mathrm{d}s \approx |\partial I_{ij}|\sum_{l=1}^{q}\omega_l H(u(G_l,t),v(G_l,t))\cdot n, \tag{2.7}$$

$$\int_{\partial I_{ij}} Q\cdot n\,\mathrm{d}s \approx |\partial I_{ij}|\sum_{l=1}^{q}\omega_l Q(u(G_l,t),w(G_l,t))\cdot n. \tag{2.8}$$

Since we are constructing schemes up to fourth-order accuracy, 2-point Gaussian will be used in each line integration, and $F(u(G_l,t))\cdot n$, $H(u(G_l,t),\ v(G_l,t))\cdot n$, $Q(u(G_l,t),\ w(G_l,t))\cdot n$ are replaced by numerical fluxes such as the Lax–Friedrichs fluxes:

$$f(u(G_l,t)) \approx \tfrac{1}{2}[f(u^-(G_l,t)) + f(u^+(G_l,t)) - \alpha(u^+(G_l,t) - u^-(G_l,t))], \tag{2.9}$$

$$\begin{aligned} h(u(G_l,t),v(G_l,t)) \approx \tfrac{1}{2}[&h(u^-(G_l,t),v^-(G_l,t)) + h(u^+(G_l,t),v^+(G_l,t)) \\ &- \alpha(v^+(G_l,t) - v^-(G_l,t))], \end{aligned} \tag{2.10}$$

$$\begin{aligned} q(u(G_l,t),w(G_l,t)) \approx \tfrac{1}{2}[&q(u^-(G_l,t),w^-(G_l,t)) + q(u^+(G_l,t),w^+(G_l,t)) \\ &- \alpha(w^+(G_l,t) - w^-(G_l,t))], \end{aligned} \tag{2.11}$$

for $G_l = (x_{i\pm 1/2}, y_{j\pm\sqrt{3}/6})$, $u^{\pm}(G_l,t)$, $v^{\pm}(G_l,t)$, $w^{\pm}(G_l,t)$ are the left and right limits of the solutions u, v, w at the cell interface G_l respectively; and

$$g(u(G_l,t)) \approx \tfrac{1}{2}[g(u^-(G_l,t)) + g(u^+(G_l,t)) - \alpha(u^+(G_l,t) - u^-(G_l,t))], \tag{2.12}$$

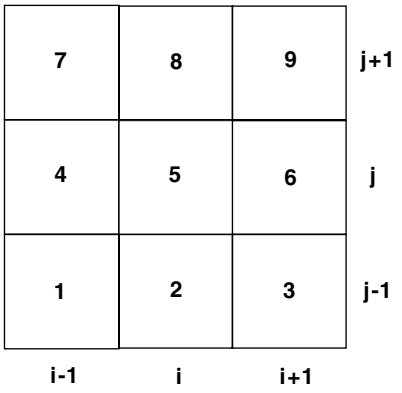

Fig. 1. The big stencil.

$$r(u(G_l,t),v(G_l,t)) \approx \tfrac{1}{2}[r(u^-(G_l,t),v^-(G_l,t)) + r(u^+(G_l,t),v^+(G_l,t)) - \alpha(v^+(G_l,t) - v^-(G_l,t))], \tag{2.13}$$

$$s(u(G_l,t),w(G_l,t)) \approx \tfrac{1}{2}[s(u^-(G_l,t),w^-(G_l,t)) + s(u^+(G_l,t),w^+(G_l,t)) - \alpha(w^+(G_l,t) - w^-(G_l,t))], \tag{2.14}$$

for $G_l = (x_{i\pm\sqrt{3}/6},\ y_{j\pm1/2})$, $u^{\pm}(G_l,t)$, $v^{\pm}(G_l,t)$, $w^{\pm}(G_l,t)$ are the bottom and top limits of the solutions u, v, w at the cell interface G_l respectively. What we want to do is to reconstruct $u^{\pm}(G_l,t)$, $v^{\pm}(G_l,t)$, $w^{\pm}(G_l,t)$ from $\{u_{ij}^{(0)}, u_{ij}^{(1)}, u_{ij}^{(2)}\}$.

For simplicity we relabel the cell I_{ij} and its neighboring cells as $I_1,\ldots,I_9$ as shown in Fig. 1.

We summarize the procedure to construct the fourth-order HWENO scheme below. Procedures to construct HWENO schemes of other orders of accuracy are similar.

Step 1. Reconstruction of $u^-(G_l,t)$ from $\{u_{ij}^{(0)}, u_{ij}^{(1)}, u_{ij}^{(2)}\}$ based on the Hermite type interpolation.

1. Given eight small stencils S_1, $S_2,\ldots,S_8$ shown in Fig. 2 and the big stencil $\mathscr{T} = \bigcup_{m=1}^{8} S_m$ shown in Fig. 1, and denoting the cells as I_m, $m = 1,\ldots,9$ as shown in the big stencil, Fig. 1, we construct Hermite quadratic reconstruction polynomials $p_1(x,y),\ldots,p_8(x,y)$ in the small stencils such that:

$$\frac{1}{\Delta x\Delta y}\int_{I_k} p_n(x,y)\,\mathrm{d}x\,\mathrm{d}y = u_k^{(0)},$$

$$\frac{1}{\Delta y}\int_{I_{k_x}} \frac{\partial p_n(x,y)}{\partial x}\,\mathrm{d}x\,\mathrm{d}y = u_{k_x}^{(1)}, \qquad \frac{1}{\Delta x}\int_{I_{k_y}} \frac{\partial p_n(x,y)}{\partial y}\,\mathrm{d}x\,\mathrm{d}y = u_{k_y}^{(2)}, \tag{2.15}$$

for

$$n=1,\ k=1,2,4,5,\ k_x=4,\ k_y=2; \quad n=2,\ k=2,3,5,6,\ k_x=6,\ k_y=2;$$

$$n=3,\ k=4,5,7,8,\ k_x=4,\ k_y=8; \quad n=4,\ k=5,6,8,9,\ k_x=6,\ k_y=8$$

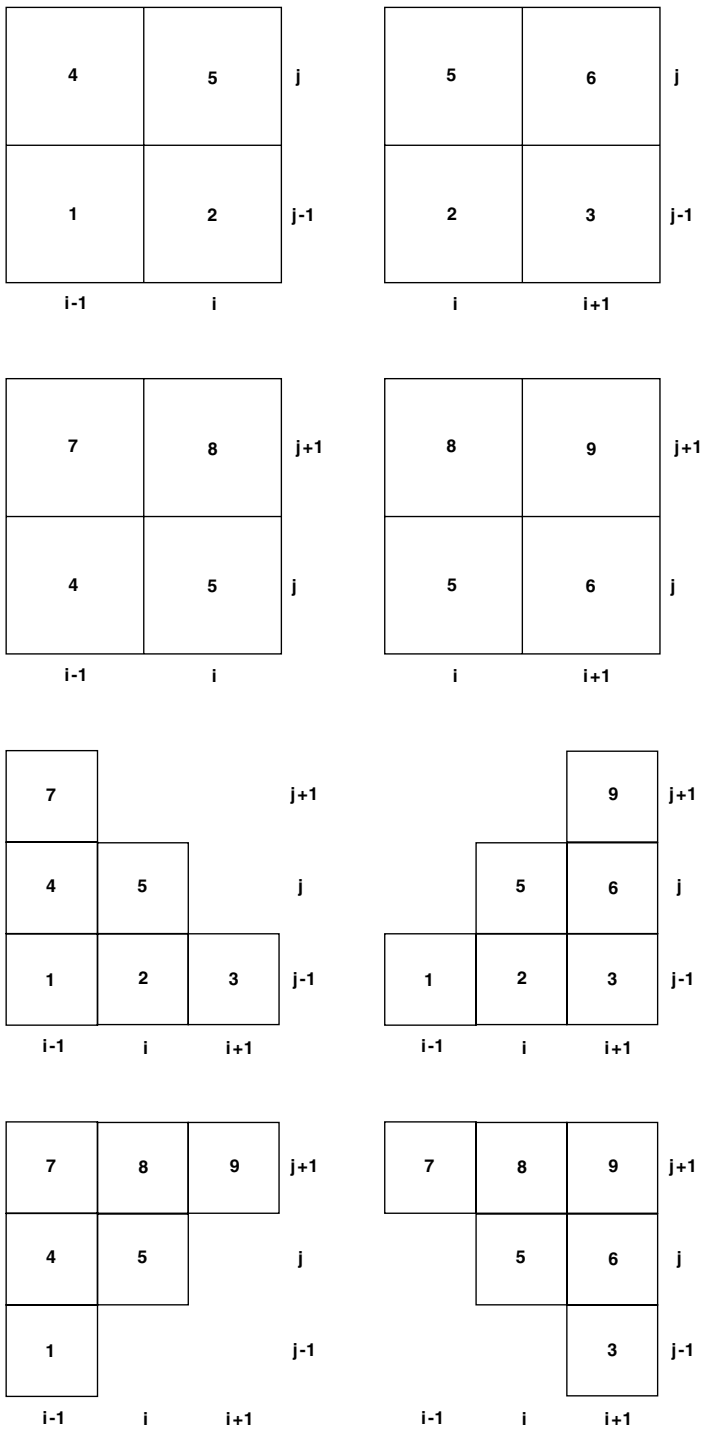

Fig. 2. The eight small stencils and information used for the reconstruction, from left to right and top to bottom are the stencils: $S_1, \ldots, S_8$ respectively.

and

$$\frac{1}{\Delta x \Delta y}\int_{I_k} p_n(x,y)\,\mathrm{d}x\,\mathrm{d}y = u_k^{(0)},$$

for

$$n=5,\ \ k=1,2,3,4,5,7; \qquad n=6,\ \ k=1,2,3,5,6,9;$$

$$n=7,\ \ k=1,4,5,7,8,9; \qquad n=8,\ \ k=3,5,6,7,8,9.$$

2. To combine the quadratic polynomials to obtain a fourth-order approximation of u at the point G_l.
   If we choose the linear weights denoted by $\gamma_1^{(l)},\ldots,\gamma_8^{(l)}$ such that

$$u(G_l) = \sum_{n=1}^{8} \gamma_n^{(l)} p_n(G_l) \tag{2.16}$$

   is valid for any polynomial u of degree at most 3, then we can obtain a fourth-order approximation of u at the point G_l for all sufficiently smooth functions u.
   Notice that (2.16) holds for any polynomial u of degree at most 2 if $\sum_{n=1}^{8}\gamma_n^{(l)} = 1$. This is because each individual $p_n(x,y)$ reconstructs quadratic polynomials exactly. There are four other constraints on the linear weights $\gamma_1^{(l)},\ldots,\gamma_8^{(l)}$ from requiring (2.16) to hold for $u = x^3$, x^2y, xy^2 and y^3 respectively. This leaves 3 free parameters in determining the linear weights $\gamma_1^{(l)},\ldots,\gamma_8^{(l)}$. These free parameters are uniquely determined by least square

$$\min\left(\sum_{n=1}^{8}(\gamma_n^{(l)})^2\right)$$

   subject to the constraints listed above. The linear weights $\gamma_1^{(l)},\ldots,\gamma_8^{(l)}$ chosen this way are positive.
3. We compute the smoothness indicator, denoted by β_n, for each stencil S_n, which measures how smooth the function $p_n(x,y)$ is in the target cell I_{ij}. The smaller this smoothness indicator β_n, the smoother the function $p_n(x,y)$ is in the target cell. We use the same recipe for the smoothness indicator as in [15]:

$$\beta_n = \sum_{|k|=1}^{2} |I_{ij}|^{2|k|-1} \int_{I_{ij}} \left(\frac{\partial^{|k|}}{\partial x^{k_1}\partial y^{k_2}} p_n(x,y)\right)^2 \mathrm{d}x\,\mathrm{d}y, \tag{2.17}$$

   where $k=(k_1,k_2)$.
4. We compute the non-linear weights based on the smoothness indicators:

$$\omega_n^{(l)} = \frac{\overline{\omega}_n^{(l)}}{\sum_k \overline{\omega}_k^{(l)}}, \qquad \overline{\omega}_k^{(l)} = \frac{\gamma_k^{(l)}}{(\varepsilon+\beta_k)^2}, \tag{2.18}$$

where $\gamma_k^{(l)}$ are the linear weights determined in Step 1.2 above, and ε is a small number to avoid the denominator to become 0. We use $\varepsilon = 10^{-6}$ in all the computation in this paper. The final HWENO reconstruction is then given by:

$$u^-(G_l) \approx \sum_{n=1}^{8} \omega_n^{(l)} p_n(G_l). \tag{2.19}$$

The reconstruction to $u^+(G_l)$ is similar to the above procedure.

Step 2. Reconstruction of $v^-(G_l, t)$ and $w^-(G_l, t)$ from $\{u_{ij}^{(0)}, u_{ij}^{(1)}, u_{ij}^{(2)}\}$ based on the Hermite type interpolation.

1. We construct Hermite reconstruction polynomials of the form

$$p_n(x, y) = a_0 + a_1 x + a_2 y + a_3 x^2 + a_4 xy + a_5 y^2 + a_6 x^3 + a_7 y^3, \quad n = 1, \ldots, 8$$

in the small stencils such that:

$$\frac{1}{\Delta x \Delta y} \int_{I_k} p_n(x, y)\, \mathrm{d}x\, \mathrm{d}y = u_k^{(0)},$$

$$\frac{1}{\Delta y} \int_{I_{k_x}} \frac{\partial p_n(x, y)}{\partial x}\, \mathrm{d}x\, \mathrm{d}y = u_{k_x}^{(1)}, \quad \frac{1}{\Delta x} \int_{I_{k_y}} \frac{\partial p_n(x, y)}{\partial y}\, \mathrm{d}x\, \mathrm{d}y = u_{k_y}^{(2)},$$

for

$$n = 1,\ k = 1, 2, 4, 5,\ k_x = 4, 5,\ k_y = 2, 5; \quad n = 2,\ k = 2, 3, 5, 6,\ k_x = 5, 6,\ k_y = 2, 5;$$

$$n = 3,\ k = 4, 5, 7, 8,\ k_x = 4, 5,\ k_y = 5, 8; \quad n = 4,\ k = 5, 6, 8, 9,\ k_x = 5, 6,\ k_y = 5, 8;$$

$$n = 5,\ k = 1, 2, 3, 4, 5, 7,\ k_x = 5,\ k_y = 5; \quad n = 6,\ k = 1, 2, 3, 5, 6, 9,\ k_x = 5,\ k_y = 5;$$

$$n = 7,\ k = 1, 4, 5, 7, 8, 9,\ k_x = 5,\ k_y = 5; \quad n = 8,\ k = 3, 5, 6, 7, 8, 9,\ k_x = 5,\ k_y = 5.$$

2. To combine the polynomials to obtain third-order approximation to u_x and u_y at the point G_l, we choose the linear weights denoted by $\gamma_{x1}^{(l)}, \ldots, \gamma_{x8}^{(l)}$, $\gamma_{y1}^{(l)}, \ldots, \gamma_{y8}^{(l)}$ such that

$$\frac{\partial}{\partial x} u(G_l) = \sum_{n=1}^{8} \gamma_{xn}^{(l)} \frac{\partial}{\partial x} p_n(G_l), \tag{2.20}$$

$$\frac{\partial}{\partial y} u(G_l) = \sum_{n=1}^{8} \gamma_{yn}^{(l)} \frac{\partial}{\partial y} p_n(G_l) \tag{2.21}$$

are valid for any polynomial u of degree at most 3, then we can obtain third-order approximations to u_x and u_y at the point G_l for all sufficiently smooth functions u.
Notice that (2.20) and (2.21) hold for any polynomial u which is a linear combination of 1, x, y, x^2, xy, y^2, x^3, y^3 if $\sum_{n=1}^{8} \gamma_{xn}^{(l)} = 1$ and $\sum_{n=1}^{8} \gamma_{yn}^{(l)} = 1$ respectively. This is because each individual $p_n(x, y)$ reconstructs such polynomials (hence their derivatives) exactly. There are two other constraints on each of the groups of linear weights $\gamma_{x1}^{(l)}, \ldots, \gamma_{x8}^{(l)}$ and $\gamma_{y1}^{(l)}, \ldots, \gamma_{y8}^{(l)}$, from requiring

(2.20) and (2.21) to hold for $u = x^2y$ and xy^2 respectively. This leaves 5 free parameters in determining each group of the linear weights, which are obtained uniquely by least square

$$\min\left(\sum_{n=1}^{8}(\gamma_{xn}^{(l)})^2\right), \quad \min\left(\sum_{n=1}^{8}(\gamma_{yn}^{(l)})^2\right)$$

subject to the constraints listed above. The linear weights chosen this way are positive.

3. We compute the smoothness indicator, denoted by β_n:

$$\beta_n = \sum_{|k|=2}^{3} |I_{ij}|^{2|k|-1} \int_{I_{ij}} \left(\frac{\partial^{|k|}}{\partial x^{k_1}\partial y^{k_2}} p_n(x,y)\right)^2 \mathrm{d}x\,\mathrm{d}y, \tag{2.22}$$

where $k = (k_1, k_2)$.

4. We compute the non-linear weights based on the smoothness indicators by formula (2.18). The final HWENO reconstruction to $v^-(G_l)$ and $w^-(G_l)$ are given by:

$$v^-(G_l) \approx \sum_{n=1}^{8} \omega_{xn} \frac{\partial}{\partial x} p_n(G_l), \qquad w^-(G_l) \approx \sum_{n=1}^{8} \omega_{yn} \frac{\partial}{\partial y} p_n(G_l)$$

respectively. The reconstructions to $v^+(G_l)$ and $w^+(G_l)$ are similar to the above procedure.

For systems of conservation laws, such as the Euler equations of gas dynamics, all of the reconstructions are performed in the local characteristic directions to avoid oscillations. For details of such local characteristic decompositions, see, e.g. [26].

The semi-discrete scheme (2.3)–(2.5), written as

$$u_t = L(u)$$

is then discretized in time by a total variation diminishing (TVD) Runge–Kutta method [27], for example the third-order version given by

$$\begin{aligned} u^{(1)} &= u^n + \Delta t L(u^n), \\ u^{(2)} &= \frac{3}{4}u^n + \frac{1}{4}u^{(1)} + \frac{1}{4}\Delta t L(u^{(1)}), \\ u^{n+1} &= \frac{1}{3}u^n + \frac{2}{3}u^{(2)} + \frac{2}{3}\Delta t L(u^{(2)}). \end{aligned} \tag{2.23}$$

3. HWENO reconstruction as limiters for the discontinuous Galerkin method

In the discontinuous Galerkin method, the solution as well as the test function space is given by $V_h^k = \{p : p|_{I_{ij}} \in P^k(I_{ij})\}$, where $P^k(I_{ij})$ is the space of polynomials of degree $\leq k$ on the cell I_{ij}. We adopt a local orthogonal basis over I_{ij}, $\{v_l^{(ij)}(x,y),\ l = 0, 1, \ldots, K;\ K = \frac{1}{2}(k+1)(k+2) - 1\}$:

$$v_0^{(ij)}(x,y) = 1, \quad v_1^{(ij)}(x,y) = \frac{x - x_i}{\Delta x}, \quad v_2^{(ij)}(x,y) = \frac{y - y_j}{\Delta y},$$

$$v_3^{(ij)}(x) = \left(\frac{x-x_i}{\Delta x}\right)^2 - \frac{1}{12}, \quad v_4^{(ij)}(x,y) = \frac{(x-x_i)(y-y_j)}{\Delta x \Delta y}, \quad v_5^{(ij)}(y) = \left(\frac{y-y_j}{\Delta y}\right)^2 - \frac{1}{12}, \ldots .$$

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

$$u^h(x,y,t) = \sum_{l=0}^{K} u_{ij}^{(l)}(t) v_l^{(ij)}(x,y), \quad \text{for } (x,y) \in I_{ij} \tag{3.1}$$

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

$$u_{ij}^{(l)}(t) = \frac{1}{a_l} \int_{I_{ij}} u^h(x,y,t)\ v_l^{(ij)}(x,y)\,\mathrm{d}x\,\mathrm{d}y, \quad l = 0,1,\ldots,K,$$

where $a_l = \int_{I_{ij}} (v_l^{(ij)}(x,y))^2\,\mathrm{d}x\,\mathrm{d}y$ are the normalization constants since the basis is not orthonormal. In order to determine the approximate solution, what we would like to do is to evolve the degrees of freedom $u_{ij}^{(l)}$:

$$\begin{aligned}
&\frac{\mathrm{d}}{\mathrm{d}t} u_{ij}^{(l)} + \frac{1}{a_l} \left(- \int_{I_{ij}} \left(f(u^h(x,y,t)) \frac{\mathrm{d}}{\mathrm{d}x} v_l^{(ij)}(x,y) + g(u^h(x,y,t)) \frac{\mathrm{d}}{\mathrm{d}y} v_l^{(ij)}(x,y) \right) \mathrm{d}x\,\mathrm{d}y \right. \\
&\quad + \int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} (f(u^h(x_{i+\frac{1}{2}},y)) v_l^{(ij)}(x_{i+\frac{1}{2}},y) - f(u^h(x_{i+\frac{1}{2}},y)) v_l^{(ij)}(x_{i-\frac{1}{2}},y))\,\mathrm{d}y \\
&\quad \left. + \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} (g(u^h(x,y_{j+\frac{1}{2}})) v_l^{(ij)}(x,y_{j+\frac{1}{2}}) - g(u^h(x,y_{j-\frac{1}{2}})) v_l^{(ij)}(x,y_{j-\frac{1}{2}}))\,\mathrm{d}x \right) = 0, \quad l = 0,1,\ldots,K.
\end{aligned} \tag{3.2}$$

In (3.2) the first integral term can be computed either exactly or by a suitable numerical quadrature accurate to at least $O(h^{2k+2})$. The second and third integral terms can also be computed by suitable numerical quadratures, but the flux functions f and g would need to be replaced by monotone numerical fluxes (or approximate Riemann solvers in the system case) because they are computed at cell interfaces. For $k \leqslant 2$, we can use the following 2-point Gaussian quadrature:

$$\int_{I_{ij}} f(u^h(x,y,t)) \frac{\mathrm{d}}{\mathrm{d}x} v_l^{(ij)}(x,y)\,\mathrm{d}x\,\mathrm{d}y \approx \frac{1}{4}\Delta x \Delta y \sum_{n=1}^{2}\sum_{m=1}^{2} f(u^h(x_{in},y_{jm},t)) \frac{\mathrm{d}}{\mathrm{d}x} v_l^{(ij)}(x_{in},y_{jm}),$$

$$\int_{I_{ij}} g(u^h(x,y,t)) \frac{\mathrm{d}}{\mathrm{d}y} v_l^{(ij)}(x,y)\,\mathrm{d}x\,\mathrm{d}y \approx \frac{1}{4}\Delta x \Delta y \sum_{n=1}^{2}\sum_{m=1}^{2} g(u^h(x_{in},y_{jm},t)) \frac{\mathrm{d}}{\mathrm{d}y} v_l^{(ij)}(x_{in},y_{jm}),$$

$$\int_{y_{j-\frac{1}{2}}}^{y_{j+\frac{1}{2}}} f\left(u^h\left(x_{i+\frac{1}{2}},y\right)\right) v_l^{(ij)}\left(x_{i+\frac{1}{2}},y\right)\mathrm{d}y \approx \frac{1}{2}\Delta y \sum_{m=1}^{2} \hat{f}\left(u^-_{i+\frac{1}{2},jm}, u^+_{i+\frac{1}{2},jm}\right) v_l^{(ij)}\left(x_{i+\frac{1}{2}},y_{jm}\right),$$

$$\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} g\left(u^h\left(x,y_{j+\frac{1}{2}}\right)\right) v_l^{(ij)}\left(x,y_{j+\frac{1}{2}}\right)\mathrm{d}x \approx \frac{1}{2}\Delta x \sum_{n=1}^{2} \hat{g}\left(u^-_{in,j+\frac{1}{2}}, u^+_{in,j+\frac{1}{2}}\right) v_l^{(ij)}\left(x_{in},y_{j+\frac{1}{2}}\right),$$

where $x_{i1} = x_i - \frac{\sqrt{3}}{6}\Delta x, x_{i2} = x_i + \frac{\sqrt{3}}{6}\Delta x, y_{j1} = y_j - \frac{\sqrt{3}}{6}\Delta y, y_{j2} = y_j + \frac{\sqrt{3}}{6}\Delta y$ are Gaussian quadrature points, $u^{\pm}_{i+\frac{1}{2},jm} = u^h(x^{\pm}_{i+\frac{1}{2}}, y_{jm}, t)$ are the left and right limits of the discontinuous solution u^h at the cell interface $(x_{i+\frac{1}{2}}, y_{jm})$ and $u^{\pm}_{in,j+\frac{1}{2}} = u^h(x_{in}, y^{\pm}_{j+\frac{1}{2}}, t)$ are the bottom and top limits of the discontinuous solution u^h at the cell interface $(x_{in}, y_{j+\frac{1}{2}})$. The fluxes $\hat{f}(u^-, u^+)$ and $\hat{g}(u^-, u^+)$ are monotone fluxes (non-decreasing in the first argument and non-increasing in the second argument) for the scalar case and an exact or approximate Riemann solver for the system case. The semi-discrete scheme (3.2) is discretized in time by a non-linearly stable Runge–Kutta time discretization, e.g. the third-order version (2.23).

The limiter adopted in [5,9] is described below in some detail, and in this paper we use it to detect "troubled cells". The limiting is performed on the first moment $u^{(1)}_{ij}$ and $u^{(2)}_{ij}$, using the differences of means. For a scalar equation, $u^{(1)}_{ij}$ would be limited by either the standard minmod limiter [13]:

$$m\left(\frac{1}{2}u^{(1)}_{ij}, u^{(0)}_{i+1,j} - u^{(0)}_{ij}, u^{(0)}_{ij} - u^{(0)}_{i-1,j}\right),$$

where m is given by:

$$m(a_1, a_2, \ldots, a_n) = \begin{cases} s \cdot \min_{1\le j\le n}|a_j| & \text{if } \operatorname{sign}(a_1) = \operatorname{sign}(a_2) = \cdots = \operatorname{sign}(a_n) = s, \\ 0 & \text{otherwise,} \end{cases} \tag{3.3}$$

or by the TVB modified minmod function [25]

$$\tilde{m}(a_1, a_2, \ldots, a_n) = \begin{cases} a_1 & \text{if } |a_1| \le M\Delta x^2, \\ m(a_1, a_2, \ldots, a_n) & \text{otherwise,} \end{cases} \tag{3.4}$$

where $M > 0$ is the TVB 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 [7] (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.

Similarly, $u^{(2)}_{ij}$ is limited by

$$m\left(\frac{1}{2}u^{(2)}_{ij}, u^{(0)}_{i,j+1} - u^{(0)}_{ij}, u^{(0)}_{ij} - u^{(0)}_{i,j-1}\right)$$

or

$$\tilde{m}\left(\frac{1}{2}u^{(2)}_{ij}, u^{(0)}_{i,j+1} - u^{(0)}_{ij}, u^{(0)}_{ij} - u^{(0)}_{i,j-1}\right)$$

with a change of Δx to Δy in (3.4).

For systems, we perform the limiting in local characteristic variables as in [9].

Similar to [21,22], in this paper we use the limiter described above to identify "troubled cells", namely, 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 HWENO reconstruction maintains 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 as troubled cells and will be subject to

unnecessary HWENO reconstructions. This does increase 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, $u_{ij}^{(l)}$ for any troubled cell I_{ij} for $l = 1, \ldots, K$ and retain only the cell average $u_{ij}^{(0)}$.

For the third-order $k = 2$ case, we summarize the procedure to reconstruct the first and second-order moments $u_{ij}^{(1)}, \ldots, u_{ij}^{(5)}$ for a troubled cell I_{ij} using HWENO reconstruction. For simplicity, we relabel the "troubled cell" and its neighboring cells as $I_1, \ldots, I_9$ as shown in Fig. 1.

1. We construct Hermite quadratic reconstruction polynomials $p_1(x,y), \ldots, p_8(x,y)$ in the small stencils satisfying the conditions in Step 1.1, Section 2, with (2.15) replaced by

$$\int_{I_{k_x}} p_n(x,y) v_1^{(k_x)}(x,y)\,\mathrm{d}x\,\mathrm{d}y = u_{k_x}^{(1)} a_1, \quad \int_{I_{k_y}} p_n(x,y) v_2^{(k_y)}(x,y)\,\mathrm{d}x\,\mathrm{d}y = u_{k_y}^{(2)} a_2.$$

2. We find the combination coefficients, also called linear weights, denoted by $\gamma_1^{(l)}, \ldots, \gamma_8^{(l)}$, satisfying:

$$\int_{I_5} u v_l^{(5)}(x,y)\,\mathrm{d}x\,\mathrm{d}y = \sum_{n=1}^{8} \gamma_n^{(l)} \int_{I_5} p_n(x,y) v_l^{(5)}(x,y)\,\mathrm{d}x\,\mathrm{d}y, \quad l = 1, \ldots, 5,$$

for $u = v_1^{(5)}, \ldots, v_9^{(5)}$, and ask for

$$\min\left(\sum_{n=1}^{8} (\gamma_n^{(l)})^2\right)$$

under these restrictions. For $l = 1, 2$, these lead to

$$\gamma_1^{(l)} = \gamma_2^{(l)} = \gamma_3^{(l)} = \gamma_4^{(l)} = \frac{11}{76}; \quad \gamma_5^{(l)} = \gamma_6^{(l)} = \gamma_7^{(l)} = \gamma_8^{(l)} = \frac{2}{19}$$

and for $l = 3, 4, 5$ we obtain:

$$\gamma_n^{(l)} = \frac{1}{8}, \qquad n = 1, \ldots, 8.$$

3. Compute the smoothness indicators:

   (a) For the reconstruction of $u_{ij}^{(1)}$, we compute the smoothness indicator β_n as:

$$\beta_n = \sum_{k=1}^{2} |I_{ij}|^{2k-1} \int_{I_{ij}} \left(\frac{\partial^k}{\partial x^k} p_n(x,y)\right)^2 \mathrm{d}x\,\mathrm{d}y. \tag{3.5}$$

   (b) For the reconstruction of $u_{ij}^{(2)}$, we compute the smoothness indicator β_n as:

$$\beta_n = \sum_{k=1}^{2} |I_{ij}|^{2k-1} \int_{I_{ij}} \left(\frac{\partial^k}{\partial y^k} p_n(x,y)\right)^2 \mathrm{d}x\,\mathrm{d}y. \tag{3.6}$$

(c) For the reconstruction of $u_{ij}^{(3)}$, $u_{ij}^{(4)}$ and $u_{ij}^{(5)}$, we compute the smoothness indicator β_n as:

$$\beta_n = \sum_{|k|=2}^{2} |I_{ij}|^{2k-1} \int_{I_{ij}} \left(\frac{\partial^{|k|}}{\partial x^{k_1} \partial y^{k_2}} p_n(x,y) \right)^2 \mathrm{d}x\,\mathrm{d}y \tag{3.7}$$

and compute the non-linear weights based on the smoothness indicators by (2.18). The moments of the reconstructed polynomial is then given by:

$$u_{ij}^{(l)} = \frac{1}{a_l} \sum_{n=1}^{8} \omega_n^{(l)} \int_{I_{ij}} p_n(x,y) v_n^{(ij)}(x,y)\,\mathrm{d}x\,\mathrm{d}y, \qquad l = 1,2,3,4,5. \tag{3.8}$$

For the system cases, to find out the "troubled cells", we could either use 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) that the HWENO reconstruction would be performed in many more cells. We have thus used a characteristic based minmod TVB limiter to detect troubled cells. In order to achieve better qualities at the price of more complicated computations, the reconstructions of two first moments $u_{ij}^{(1)}, u_{ij}^{(2)}$ are performed in the local characteristic directions to avoid oscillations, and the reconstructions of the three second-order moments $u_{ij}^{(3)}, u_{ij}^{(4)}, u_{ij}^{(5)}$ are performed componentwise.

4. Numerical results

In this section we present the results of our numerical experiments for the fourth-order HWENO schemes with the third-order TVD Runge–Kutta method (HWENO4-RK3) and the third-order DG method with HWENO limiter (DG3-HWENO4-RK3) developed in the previous sections. A uniform mesh with $N_x \times N_y$ cells is used for all the test cases, the CFL number is taken as 0.6 for HWENO4 and 0.18 for DG3-HWENO4-RK3 except for some accuracy tests where a suitably reduced time step is used to guarantee that spatial error dominates.

4.1. Accuracy test

We first test the accuracy of the schemes on scalar and system problems. For all accuracy tests of DG3-HWENO4-RK3 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 HWENO reconstruction limiter on the accuracy of the RKDG method.

We have tested many standard problems for accuracy, such as linear advection, non-linear Burgers' equation, and non-linear Euler equations. We present only the results of Burgers' equations and non-linear Euler equations on uniform meshes as representative examples to save space.

Example 4.1. We solve the following non-linear scalar Burgers' equation in two dimensions:

$$u_t + \left(\frac{u^2}{2}\right)_x + \left(\frac{u^2}{2}\right)_y = 0 \tag{4.1}$$

Table 1
Burgers' equation $u_t + (u^2/2)_x + (u^2/2)_y = 0$

$N \times N$	HWENO4-RK3			
	L_1 error	Order	L_∞ error	Order
10×10	1.13E−02		3.46E−02	
20×20	9.12E−04	3.63	5.97E−03	2.53
40×40	6.09E−05	3.90	4.56E−04	3.71
80×80	3.09E−06	4.30	3.59E−05	3.67
160×160	1.70E−07	4.19	2.95E−06	3.60

$N \times N$	DG with HWENO limiter				DG with no limiter			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10×10	2.98E−02		5.03E−01		5.19E−03		1.83E−01	
20×20	1.81E−03	4.04	4.09E−02	3.62	8.29E−04	2.65	4.16E−02	2.14
40×40	1.73E−04	3.38	6.04E−03	2.76	1.12E−04	2.88	6.04E−03	2.79
80×80	2.07E−05	3.06	1.00E−03	2.59	1.44E−05	2.96	1.00E−03	2.59
160×160	2.49E−06	3.06	1.38E−04	2.87	1.83E−06	2.98	1.38E−04	2.87

Initial condition $u(x, y, 0) = 0.5 + \sin(\pi(x + y)/2)$ and periodic boundary conditions. HWENO4-RK3 and DG3-HWENO4-RK3 ($M = 0.01$) comparing with RKDG without limiter. Local Lax–Friedrichs flux. $t = 0.5/\pi$. L_1 and L_∞ errors. Uniform meshes with $N \times N$ cells.

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 HWENO4-RK3 and DG3-HWENO4-RK3 comparing with the original RKDG method without a limiter are shown in Table 1. We can see that the HWENO limiter keeps both the designed order and the magnitude of accuracy of the original RKDG method.

Example 4.2. We solve the following non-linear system of Euler equations:

$$\xi_t + f(\xi)_x + g(\xi)_y = 0 \tag{4.2}$$

with

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

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 $\rho(x, y, 0) = 1 + 0.2\sin(\pi(x + y))$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$, $p(x, y, 0) = 1$, 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 HWENO4-RK3 and DG3-HWENO4-RK3 ($M = 0.01$) comparing with RKDG without

Table 2
Euler equations

$N \times N$	HWENO4-RK3			
	L_1 error	Order	L_∞ error	Order
10×10	2.14E−02		3.34E−02	
20×20	2.39E−04	6.49	5.19E−04	6.01
40×40	2.74E−05	3.12	5.22E−05	3.31
80×80	2.62E−06	3.39	4.28E−06	3.61
160×160	1.88E−07	3.80	3.20E−07	3.74

$N \times N$	DG with HWENO limiter				DG with no limiter			
	L_1 error	Order	L_∞ error	Order	L_1 error	Order	L_∞ error	Order
10×10	7.15E−03		5.26E−02		7.94E−04		5.53E−03	
20×20	2.67E−04	4.75	1.10E−03	5.59	1.03E−04	2.95	8.79E−04	2.65
40×40	2.66E−05	3.32	1.29E−04	3.08	1.26E−05	3.03	1.28E−04	2.78
80×80	2.36E−06	3.49	1.71E−05	2.92	1.50E−06	3.07	1.71E−05	2.91
160×160	2.19E−07	3.43	2.17E−06	2.97	1.81E−07	3.05	2.17E−06	2.97

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. HWENO4-RK3 and DG3-HWENO4-RK3 ($M = 0.01$) comparing with RKDG without limiter. Local Lax–Friedrichs flux. $t = 2.0$. L_1 and L_∞ errors for the density ρ. Uniform meshes with $N \times N$ cells.

limiter are shown in Table 2. We can see that again the HWENO limiter keeps both the designed order and the magnitude of accuracy of the original RKDG method.

4.2. Test cases with shocks

We now test the performance of HWENO4-RK3 scheme and DG3-HWENO4-RK3 for problems containing shocks. We have also computed many more problems, but will not present all the results to save space.

Example 4.3. We solve the same non-linear Burgers' equation (4.1) as in Example 4.1 with the same initial condition $u(x,y,0) = 0.5 + \sin(\pi(x+y)/2)$, except that we now plot the results at $t = 1.5/\pi$ when a shock has already appeared in the solution. In Fig. 3, the solutions of HWENO4-RK3 and DG3-HWENO4-RK3 using 80×80 cells are shown. We can see that both schemes give good non-oscillatory shock transitions for this problem.

Example 4.4. Double Mach reflection. This problem is originally from [30]. 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 post-shock condition is imposed for the part from $x = 0$ to $x = \frac{1}{6}$ and a reflective boundary condition is used for the rest. At the top boundary, the flow values are set to describe the exact motion of a Mach 10 shock. We compute the solution

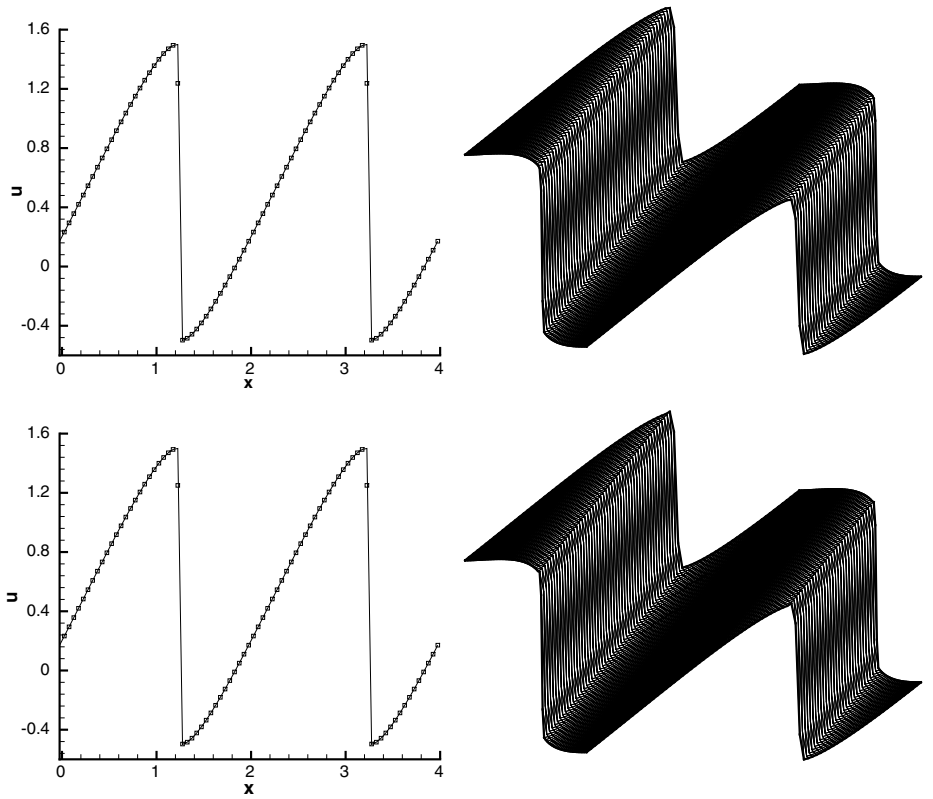

Fig. 3. 2D Burgers' equation: $u(x,0) = 0.5 + \sin(\pi(x + y)/2)$, $t = 1.5/\pi$. 80×80 grid points, by HWENO4-RK3 (top) and DG3-HWENO4-RK3 (bottom). Left: a cut of the solution at $x = y$, where the solid line is the exact solution and the squares are the computed solution; right: the surface of the solution.

Table 3
The maximal percentage of troubled cells subject to HWENO limiters in the double Mach reflection problem

M	0.01	10	100
960×240	18.57	10.76	4.65
1920×480	16.76	11.22	5.23

up to $t = 0.2$. In Table 3 we document the maximal percentage of cells declared to be "troubled cells" for different TVB constant M in the minmod limiter to identify troubled cells. We can see that only a small percentage of cells are declared as "troubled cells". Three different uniform meshes, with 480×120, 960×240 and 1920×480 cells, as well as three different values of the TVB constant, $M = 0.01$, $M = 10$ and $M = 100$, are used in the numerical experiments. To save space, we show only the simulation results on the most refined mesh with 1920×480 cells in Fig. 4, and the "zoomed-in" figures around the double Mach stem to show more details, for the cases with 960×240 and 1920×480 cells in Fig. 5. All the figures are showing 30 equally spaced density contours from 1.5 to 22.7.

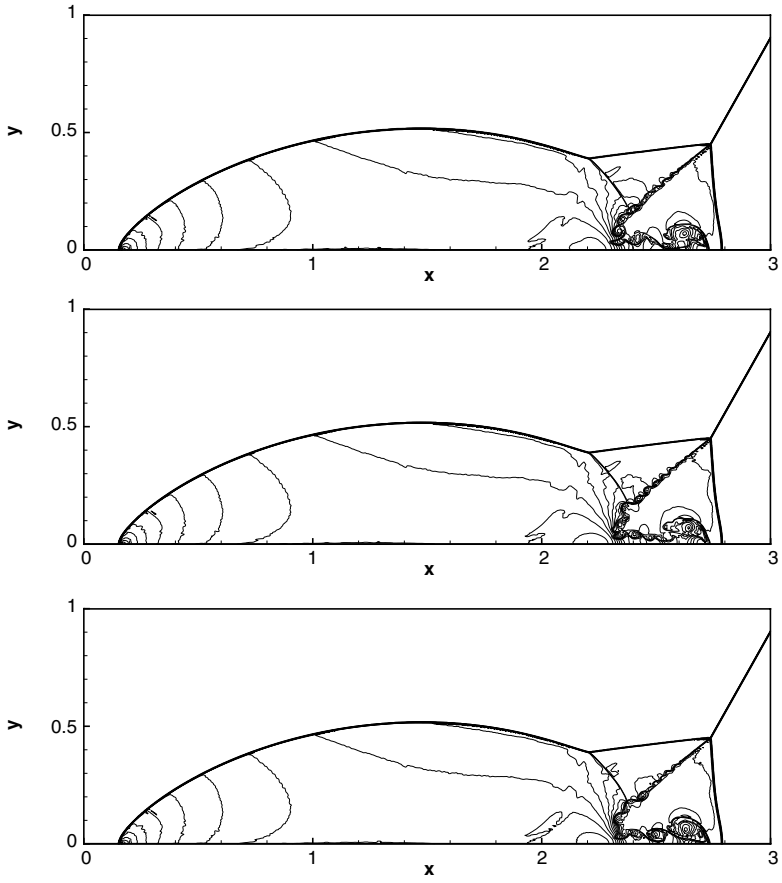

Fig. 4. Double Mach reflection problem. RKDG with HWENO limiters. 1920×480 cells. 30 equally spaced density contours from 1.5 to 22.7. TVB constant top: $M = 0.01$; middle: $M = 10$; bottom: $M = 100$.

Example 4.5. A Mach 3 wind tunnel with a step. This model problem is also originally from [30]. The setup of the problem is as follows. The wind tunnel is 1 length unit wide and 3 length units long. The step is 0.2 length units high and is located 0.6 length units from the left-hand end of the tunnel. The problem is initialized by a right-going Mach 3 flow. Reflective boundary conditions are applied along the wall of the tunnel and inflow/out flow boundary conditions are applied at the entrance/exit. The corner of the step is a singular point and we treat it the same way as in [30], which is based on the assumption of a nearly steady flow in the region near the corner. In Table 4 we show the maximum percentage of HWENO limiter used during the calculation. In Figs. 6, 7, we show 30 equally spaced density contours from 0.32 to 6.15 computed by the scheme DG3-HWENO4-RK3 at time $t = 4$. We use uniform meshes with 2 meshes of 240×80 and 480×160 cells respectively.

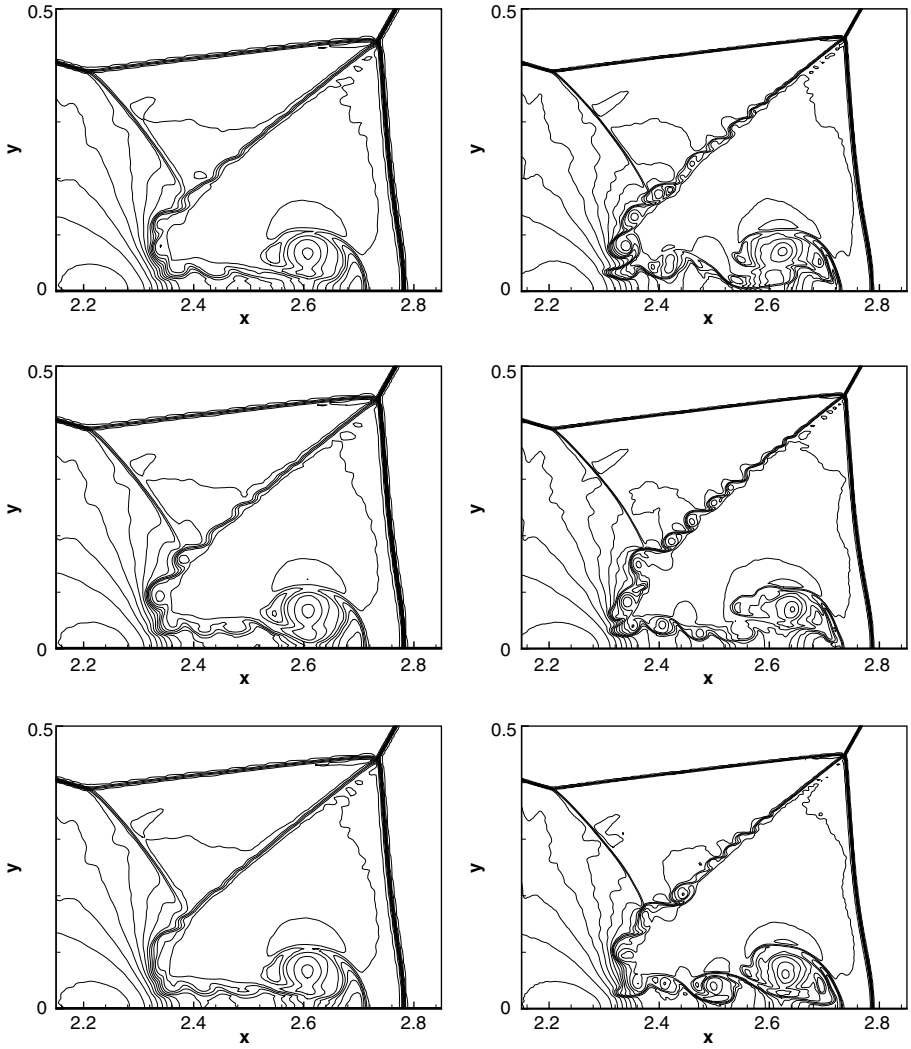

Fig. 5. Double Mach reflection problem. RKDG with HWENO limiters. 960×240 cells (left) and 1920×480 cells (right). Zoomed-in region to show more details. TVB constant top: $M = 0.01$; middle: $M = 10$; Bottom: $M = 100$.

Table 4
The maximal percentage of troubled cells subject to HWENO limiters in the forward step problem

M	0.01	10	100
240×80	39.1	21.65	8.20
480×160	34.71	19.99	8.25

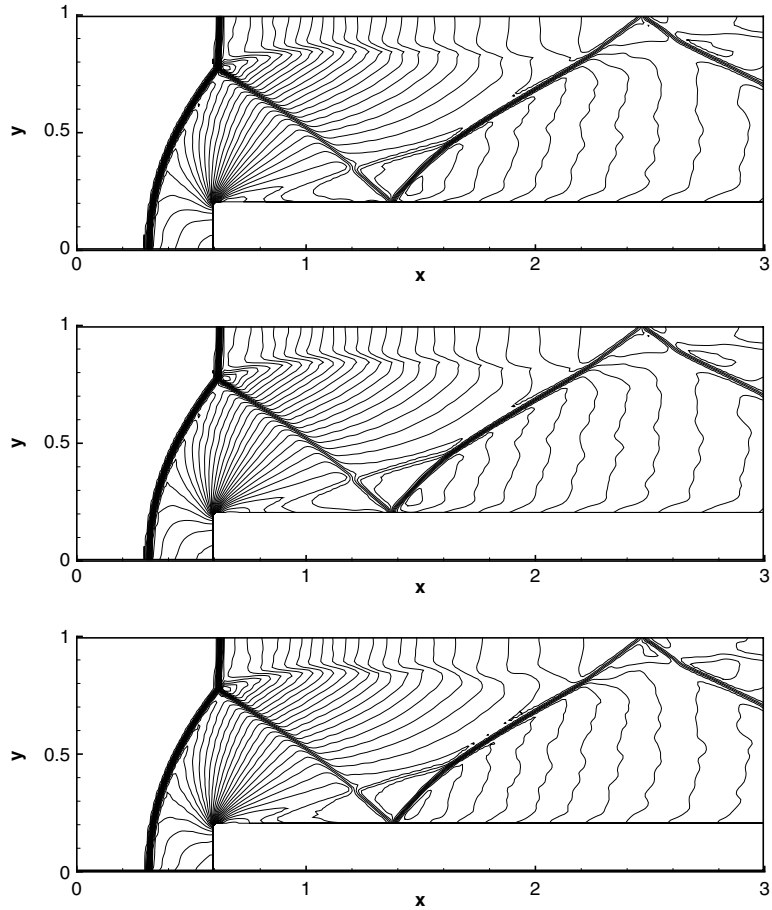

Fig. 6. Forward step problem by DG3-HWENO4-RK3 with 240×80 cells. 30 equally spaced density contours from 0.32 to 6.15. TVB constant top: $M = 0.01$; middle: $M = 10$; bottom: $M = 100$.

5. Concluding remarks

In this paper, we extend the HWENO schemes and their application as limiters for the Runge–Kutta discontinuous Galerkin (RKDG) methods developed in [22] to solve two dimensional non-linear hyperbolic conservation law systems. The emphasis is on the application of such HWENO finite volume methodology as the limiters for RKDG methods to maintain compactness of RKDG methods. HWENO schemes evolve both the solution and its derivatives, thus requiring more storage and evolution CPU time per grid point than regular WENO schemes. However, when used as limiters for the RKDG methods, the cost of HWENO reconstruction is about the same as the regular WENO reconstruction and the former has a more compact stencil in the reconstruction. Numerical experiments for two dimensional Burgers' equation and Euler equations of compressible gas dynamics are presented to show the effectiveness of the methods.

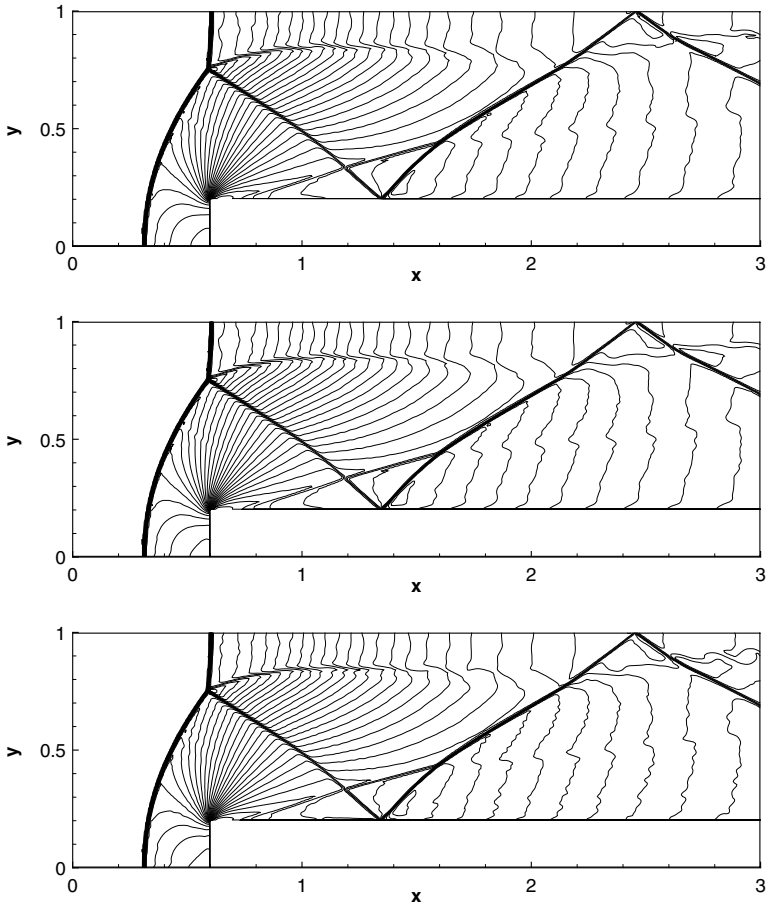

Fig. 7. Forward step problem by DG3-HWENO4-RK3 with 480×160 cells. 30 equally spaced density contours from 0.32 to 6.15. TVB constant top: $M = 0.01$; middle: $M = 10$; bottom: $M = 100$.

Acknowledgement

The first author would like to thank Professor G.W. Wei and Professor B.C. Khoo for their support and help. This research was partially performed when the first author was visiting the Division of Applied Mathematics, Brown University. We would like to thank Dr. Yong-Tao Zhang for helpful discussions at Brown University.

References

[1] Balsara DS, Shu C-W. Monotonicity preserving weighted essentially non-oscillatory schemes with increasingly high order of accuracy. J Comput Phys 2000;160:405–52.
[2] Biswas R, Devine KD, Flaherty J. Parallel, adaptive finite element methods for conservation laws. Appl Numer Math 1994;14:255–83.

[3] Bouchut F, Bourdarias C, Perthame B. A MUSCL method satisfying all the numerical entropy inequalities. Math Comput 1996;65:1439–61.
[4] Burbeau A, Sagaut P, Bruneau CH. A problem-independent limiter for high-order Runge–Kutta discontinuous Galerkin methods. J Comput Phys 2001;169:111–50.
[5] Cockburn B, Hou S, Shu C-W. The Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. Math Comput 1990;54:545–81.
[6] Cockburn B, Lin S-Y, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. J Comput Phys 1989;84:90–113.
[7] Cockburn B, Shu C-W. TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math Comput 1989;52:411–35.
[8] Cockburn B, Shu C-W. The Runge–Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws. Math Model Numer Anal (M2 AN) 1991;25:337–61.
[9] Cockburn B, Shu C-W. The Runge–Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J Comput Phys 1998;141:199–224.
[10] Cockburn B, Shu C-W. Runge–Kutta discontinuous Galerkin method for convection-dominated problems. J Scientific Comput 2001;16:173–261.
[11] Dougherty RL, Edelman AS, Hyman JM. Nonnegativity-, monotonicity-, or convexity-preserving cubic and quintic Hermite interpolation. Math Comput 1989;52:471–94.
[12] Friedrichs O. Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids. J Comput Phys 1998;144:194–212.
[13] Harten A. High resolution schemes for hyperbolic conservation laws. J Comput Phys 1983;49:357–93.
[14] Harten A, Engquist B, Osher S, Chakravathy S. Uniformly high order accurate essentially non-oscillatory schemes, III. J Comput Phys 1987;71:231–303.
[15] Hu C, Shu C-W. Weighted essentially non-oscillatory schemes on triangular meshes. J Comput Phys 1999;150:97–127.
[16] Jiang G, Shu C-W. Efficient implementation of weighted ENO schemes. J Comput Phys 1996;126:202–28.
[17] Levy D, Puppo G, Russo G. Central WENO schemes for hyperbolic systems of conservation laws. Math Model Numer Anal 1999;33:547–71.
[18] Liu X, Osher S, Chan T. Weighted essentially non-oscillatory schemes. J Comput Phys 1994;115:200–12.
[19] Nakamura T, Tanaka R, Yabe T, Takizawa K. Exactly conservative semi-Lagrangian scheme for multi-dimensional hyperbolic equations with directional splitting technique. J Comput Phys 2001;174:171–207.
[20] Qiu J, Shu C-W. On the construction, comparison, and local characteristic decomposition for high order central WENO schemes. J Comput Phys 2002;183:187–209.
[21] Qiu J, Shu C-W. Runge–Kutta discontinuous Galerkin method using WENO limiters. SIAM J. Scientific Comput, in press.
[22] Qiu J, Shu C-W. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one dimensional case. J Comput Phys 2003;193:115–35.
[23] Reed WH, Hill TR. Triangular mesh methods for neutron transport equation, Tech Report LA-UR-73-479, Los Alamos Scientific Laboratory, 1973.
[24] Shi J, Hu C, Shu C-W. A technique of treating negative weights in WENO schemes. J Comput Phys 2002;175:108–27.
[25] Shu C-W. TVB uniformly high-order schemes for conservation laws. Math Comput 1987;49:105–21.
[26] Shu C-W. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In: Cockburn B, Johnson C, Shu C-W, Tadmor E. Advanced numerical approximation of nonlinear hyperbolic equations. In: Quarteroni A, editor. Lecture Notes in Mathematics, vol. 1697. Berlin: Springer; 1998. p. 325–432.
[27] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock-capturing schemes. J Comput Phys 1988;77:439–71.
[28] Shu C-W, Osher S. Efficient implementation of essentially non-oscillatory shock capturing schemes II. J Comput Phys 1989;83:32–78.

[29] Takewaki H, Nishiguchi A, Yabe T. Cubic interpolated pseudoparticle method (CIP) for solving hyperbolic type equations. J Comput Phys 1985;61:261–8.
[30] Woodward P, Colella P. The numerical simulation of two-dimensional fluid flow with strong shocks. J Comput Phys 1984;54:115–73.