J Sci Comput (2013) 55:606–644
DOI 10.1007/s10915-012-9649-9

# WENO Schemes and Their Application as Limiters for RKDG Methods Based on Trigonometric Approximation Spaces

Jun Zhu · Jianxian Qiu

Received: 14 January 2012 / Revised: 24 September 2012 / Accepted: 26 September 2012 /
Published online: 9 October 2012

**Abstract** In this paper, we present a class of finite volume trigonometric weighted essentially non-oscillatory (TWENO) schemes and use them as limiters for Runge-Kutta discontinuous Galerkin (RKDG) methods based on trigonometric polynomial spaces to solve hyperbolic conservation laws and highly oscillatory problems. As usual, the goal is to obtain a robust and high order limiting procedure for such a RKDG method to simultaneously achieve uniformly high order accuracy in smooth regions and sharp, non-oscillatory shock transitions. The major advantage of schemes which are based on trigonometric polynomial spaces is that they can simulate the wave-like and highly oscillatory cases better than the ones based on algebraic polynomial spaces. We provide numerical results in one and two dimensions to illustrate the behavior of these procedures in such cases. Even though we do not utilize optimal parameters for the trigonometric polynomial spaces, we do observe that the numerical results obtained by the schemes based on such spaces are better than or similar to those based on algebraic polynomial spaces.

**Keywords** Finite volume TWENO scheme · Runge-Kutta discontinuous Galerkin method · Orthogonal trigonometric polynomial space · Limiter

## 1 Introduction

In this paper, we investigate using trigonometric, instead of the usual algebraic, polynomial reconstruction in finite volume Weighted Essentially Non-Oscillatory (WENO)

The research was partially supported by NSFC grant No. 10931004, 11002071 and ISTCP of China Grant No. 2010DFR00700.

J. Zhu
College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, P.R. China
e-mail: zhujun@nuaa.edu.cn

J. Qiu (✉)
School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, P.R. China
e-mail: jxqiu@xmu.edu.cn

schemes, which we have termed TWENO schemes, and using the new TWENO as limiters for Runge-Kutta Discontinuous Galerkin methods (RKDG) on orthogonal trigonometric polynomial spaces to solve hyperbolic conservation laws. The major advantage of schemes, which are based on trigonometric polynomial spaces, is that they can simulate wave-like and highly oscillatory cases better than the ones based on algebraic polynomial spaces [14, 24, 26].

In order to achieve high order of accuracy in smooth regions, Harten and Osher [10] gave a weaker version of the Total Variation Diminishing (TVD) [9] criterion, and obtained the Essentially Non-Oscillatory (ENO) type schemes. The key idea of ENO schemes is applying the smoothest stencil among all candidate stencils to approximate appropriate variables at cell boundaries to a high order of accuracy in smooth regions, avoiding oscillations near discontinuities. In 1994, Liu et al. [12] proposed an $(r+2)$-th order accurate Weighted ENO (WENO) scheme that was constructed from the $(r+1)$-th order ENO scheme. Then in 1996, Jiang and Shu [11] proposed the framework to construct $(2r+1)$-th order accurate finite difference WENO schemes from the $(r+1)$-th order (in the $L^1$ norm sense) ENO schemes, gave a new way of measuring the smoothness indicators, and emulated the ideas of minimizing the total variation of the approximation. The key idea of WENO schemes is applying all the stencils to approximate the variables at cell boundaries to an even higher order of accuracy in smooth regions while avoiding oscillations near discontinuities. WENO type schemes have improved ENO type schemes in many aspects [18].

Trigonometric Essentially Non-Oscillatory schemes are more appropriate in simulating the wave-like phenomena or highly oscillatory problems that are encountered in nature. Christofi [3] constructed a trigonometric reconstruction methodology that could add points to the stencil one at a time that possessed necessary symmetries to be suitable for the ENO stencil selection idea. In 2010, Zhu and Qiu [25] extended this work that employed trigonometric basis functions to the more important (finite difference) WENO framework.

Reed and Hill [16] introduced the first discontinuous Galerkin (DG) method in 1973, in the framework of neutron transport (steady state linear hyperbolic equations). Cockburn et al. carried out a major development of the DG method in a series of papers [4–8], in which they established a framework to easily solve nonlinear time dependent hyperbolic conservation laws (2.1). Specifically, to achieve non-oscillatory properties for strong shocks, they used explicit, nonlinearly stable, high order Runge-Kutta time [20] as well as spatial DG discretizations along with exact or approximate Riemann solvers as interface fluxes, and a total variation bounded (TVB) limiter [19]. Such schemes were termed RKDG methods. In 2006, Yuan and Shu [21] developed DG methods based on non-polynomial approximation spaces for numerically solving time dependent hyperbolic and parabolic as well as steady state hyperbolic and elliptic partial differential equations (PDEs). The algorithm was based on approximation spaces consisting of non-polynomial elementary functions, such as exponential and trigonometric functions. It relied on optimized best-fitting parameter(s) with the objective of obtaining better approximations for specific types of PDEs and initial and boundary conditions. It was shown that $L^2$ stability and error estimates can be obtained when the approximation space was suitably selected. Numerical examples also demonstrated that a careful selection of the approximation space to fit individual PDE and initial and boundary conditions often provides more accurate results than when the DG methods are based on polynomial approximation spaces of the same order of accuracy.

An important component of RKDG methods for solving the conservation laws (2.1) with strong shocks in the solutions is a nonlinear limiter, which is applied to both detect discontinuities and control spurious oscillations near these discontinuities. Many such limiters have been used in the literature for RKDG methods. Examples include the minmod type TVB limiter [4–8], which is a slope limiter that uses a technique borrowed from the finite volume methodology, as well as the moment based limiter [1] and an improved moment limiter [2], which work on the moments of the numerical solution and are specifically designed for DG methods. These limiters tend to degrade accuracy when mistakenly used in smooth regions of the solution. In [14, 26], Qiu et al. initiated a study of using the WENO methodology as limiters for RKDG methods on (un)structured meshes. The following two-step framework was adopted:

Step 1.1. Identify the "troubled cells", namely those cells which might need the limiting procedure.

Step 1.2. Replace the solution polynomials in those "troubled cells" by reconstructed polynomials using the WENO methodology which maintain the original cell averages (conservation), have the same orders of accuracy as before, but are less oscillatory.

This technique worked quite well in one and two dimensional test problems in [14] and in the follow up work [13, 15, 24] where the more compact Hermite WENO reconstruction [23] was used in the "troubled cells".

In this continuation paper, following the ideas in [3, 14, 18, 21, 22, 25], we extend the method to solve hyperbolic conservation laws on trigonometric polynomial spaces. We propose and use in this paper a different form of trigonometric basis functions than the one used in [25], as this time orthogonal basis are preferred. The form used in [25] was related to work done in [3] whereas the current form resembles more the ones in [21]. One difference between this paper and [21] is that our method does not rely on optimizing best-fitting parameter(s) to obtain good results in cases which exhibit wave-like and highly oscillatory features. We use the TWENO reconstruction based on the cell averages of neighboring cells to reconstruct the moments. This turns out to be a robust way to retain the original high order accuracy of the DG method. The organization of this paper is as follows: In Sect. 2, we review and construct finite volume TWENO schemes in detail. In Sect. 3, we investigate the usage of the finite volume TWENO methodology as limiters for the RKDG method. In Sect. 4, we present extensive numerical results to verify the accuracy and stability of this approach. Finally in Sect. 5, we give concluding remarks.

## 2 The Construction of Finite Volume Trigonometric WENO Scheme

In this section, we first consider one dimensional hyperbolic conservation laws:

$$\begin{cases} \dfrac{\partial u}{\partial t} + \dfrac{\partial f(u)}{\partial x} = 0, \\[2ex] u(x,0) = u_0(x). \end{cases} \tag{2.1}$$

For simplicity of presentation, we assume that the mesh is uniformly distributed into several cells $I_i = [x_{i-1/2}, x_{i+1/2}]$, with the cell size $x_{i+1/2} - x_{i-1/2} = \Delta x = h$ and cell centers $x_i = \frac{1}{2}(x_{i+1/2} + x_{i-1/2})$. We denote the cell averages of $u$ as: $\bar{u}_i(t) = \frac{1}{h}\int_{I_i} u(x,t)dx$. Integrating (2.1) over the target cell $I_i$, we obtain the equivalent formulation of conservation laws:

$$\frac{d\bar{u}_i(t)}{dt} = -\frac{1}{h}\big(f\big(u(x_{i+1/2},t)\big) - f\big(u(x_{i-1/2},t)\big)\big). \tag{2.2}$$

We approximate (2.2) by the following conservative scheme:

$$\frac{d\bar{u}_i(t)}{dt} = -\frac{1}{h}\big(\hat{f}_{i+1/2} - \hat{f}_{i-1/2}\big), \tag{2.3}$$

where the numerical flux $\hat{f}_{i+1/2}$ is defined by:

$$\hat{f}_{i+1/2} = \hat{h}\big(u^-_{i+1/2}, u^+_{i+1/2}\big) \tag{2.4}$$

and $u^{\pm}_{i+1/2}$ are numerical approximations to the point values of $u(x_{i+1/2},t)$ from the left and right, respectively. In this paper, we use the following Lax-Friedrichs flux:

$$\hat{h}(a,b) = \frac{1}{2}\big(f(a) + f(b) - \bar{\alpha}(b-a)\big), \tag{2.5}$$

where $\bar{\alpha} = \max_u |f'(u)|$ is a constant. The maximum is taken over the relevant range of $u$.

In this paper, we reconstruct the point values $u^{\pm}_{i+1/2}$ by using the cell averages $\{\bar{u}_i\}$ on trigonometric polynomial spaces. The reconstruction should be both high order accurate and essentially non-oscillatory. Below, we give the fifth order accurate reconstruction procedures of $u^-_{i+1/2}$ and $u^+_{i-1/2}$:

*Step 2.1.* We start with the stencil: $\Gamma = \{I_{i-2},\ldots,I_{i+2}\}$ and construct a fourth-degree trigonometric polynomial $q(x) \in \mathrm{span}\{1, \sin(\alpha(x-x_i)), \cos(\alpha(x-x_i)) - \frac{\sin(h\alpha/2)}{h\alpha/2}, \sin((\alpha+1)(x-x_i)), \cos((\alpha+1)(x-x_i)) - \frac{\sin(h(\alpha+1)/2)}{h(\alpha+1)/2}\}$, such that:

$$\frac{1}{h}\int_{I_{i+j}} q(x)dx = \bar{u}_{i+j}, \quad j = -2,\ldots,2,$$

where $\alpha$ is an adjustable parameter. Unless we specify otherwise, we set $\alpha = 1.0$ in all 1D examples of Sect. 4. Then, we divide stencil $\Gamma$ into three smaller stencils that each includes the target cell $I_i$: $S_1 = \{I_{i-2}, I_{i-1}, I_i\}$, $S_2 = \{I_{i-1}, I_i, I_{i+1}\}$, $S_3 = \{I_i, I_{i+1}, I_{i+2}\}$ and construct the trigonometric polynomials $p_n(x) \in \mathrm{span}\{1, \sin(\alpha(x-x_i)), \cos(\alpha(x-x_i)) - \frac{\sin(h\alpha/2)}{h\alpha/2}\}$, for $n = 1,2,3$, such that:

$$\frac{1}{h}\int_{I_{i+j+n-1}} p_n(x)dx = \bar{u}_{i+j+n-1}, \quad j = -2,-1,0. \tag{2.6}$$

In the following text, we denote $u_i = \bar{u}_i$ for simplicity. The values of functions $p_n(x)$, at the end points $x_{i+1/2}$ and $x_{i-1/2}$ of target cell $I_i$ can be written as linear combinations of $\{u_i\}$.

Specifically, we have that:

$$\begin{aligned}
p_1(x_{i+1/2}) &= \chi\Bigg((u_{i-1}-u_{i-2})\cos(h\alpha)+(u_{i-1}-u_i)\cos(2h\alpha)+(u_{i-2}+u_i)\frac{\sin(h\alpha)}{h\alpha} \\
&\qquad - u_{i-1}\frac{\sin(2h\alpha)}{h\alpha}\Bigg), \\
p_2(x_{i+1/2}) &= -\chi\Bigg(u_{i-1}-u_i-(u_i-u_{i+1})\cos(h\alpha)-(u_{i-1}+u_{i+1})\frac{\sin(h\alpha)}{h\alpha} \\
&\qquad + u_i\frac{\sin(2h\alpha)}{h\alpha}\Bigg), \\
p_3(x_{i+1/2}) &= \chi\Bigg(u_{i+1}-u_{i+2}-(u_i-u_{i+1})\cos(h\alpha)+(u_i+u_{i+2})\frac{\sin(h\alpha)}{h\alpha} \\
&\qquad - u_{i+1}\frac{\sin(2h\alpha)}{h\alpha}\Bigg),
\end{aligned} \tag{2.7}$$

$$\begin{aligned}
p_1(x_{i-1/2}) &= \chi\Bigg(u_{i-1}-u_{i-2}+(u_{i-1}-u_i)\cos(h\alpha)+(u_{i-2}+u_i)\frac{\sin(h\alpha)}{h\alpha} \\
&\qquad - u_{i-1}\frac{\sin(2h\alpha)}{h\alpha}\Bigg), \\
p_2(x_{i-1/2}) &= \chi\Bigg(u_i-u_{i+1}-(u_{i-1}-u_i)\cos(h\alpha)+(u_{i-1}+u_{i+1})\frac{\sin(h\alpha)}{h\alpha} \\
&\qquad - u_i\frac{\sin(2h\alpha)}{h\alpha}\Bigg), \\
p_3(x_{i-1/2}) &= \chi\Bigg((u_{i+1}-u_{i+2})\cos(h\alpha)-(u_i-u_{i+1})\cos(2h\alpha)+(u_i+u_{i+2})\frac{\sin(h\alpha)}{h\alpha} \\
&\qquad - u_{i+1}\frac{\sin(2h\alpha)}{h\alpha}\Bigg),
\end{aligned} \tag{2.8}$$

where $\chi = \frac{h\alpha}{8\sin(h\alpha/2)^3\cos(h\alpha/2)}$.

*Step 2.2*. We may now find the linear weights, such that $q(x_{i+1/2}) = \sum_{n=1}^{3}\gamma_n p_n(x_{i+1/2})$. Specifically, we have that:

$$\begin{aligned}
\gamma_1 &= \big(\cos\big(h(-1+\alpha)\big)-\cos\big(h(2+\alpha)\big)+\cos\big(h(2+3\alpha)\big)-\cos(h+3h\alpha) \\
&\quad + 4h\sin(h\alpha)+4h\alpha\sin(h\alpha)-2h\sin(2h\alpha)-2h\alpha\sin(2h\alpha)-4h\alpha\sin\big(h(1+\alpha)\big) \\
&\quad + 2h\alpha\sin\big(2h(1+\alpha)\big)\big)\big/\big(\chi\left(h\alpha\cos(h\alpha)-\sin(h\alpha)\right)\big), \\
\gamma_2 &= 1-\gamma_1-\gamma_3, \\
\gamma_3 &= \big(-\cos\big(h(-1+\alpha)\big)+\cos\big(h(2+\alpha)\big)-\cos\big(h(2+3\alpha)\big)+\cos(h+3h\alpha) \\
&\quad + 2h\sin(h)+4h\alpha\sin(h)+h\sin\big(h(-1+\alpha)\big)+h\alpha\sin\big(h(-1+\alpha)\big) \\
&\quad - h\alpha\sin\big(h(2+\alpha)\big)-h\alpha\sin\big(h(2+3\alpha)\big)-2h\sin(h+2h\alpha) \\
&\quad + h\sin(h+3h\alpha)+h\alpha\sin(h+3h\alpha)\big)\big/\big(\chi\left(-h\alpha\cos(h\alpha)+\sin(h\alpha)\right)\big),
\end{aligned} \tag{2.9}$$

where $\chi = 32\sin((h(1+\alpha))/2)^3\cos((h(1+\alpha))/2)(-\cos(h\alpha)+\cos(h(1+\alpha)))$.

And for $q(x_{i-1/2}) = \sum_{n=1}^{3} \gamma_n p_n(x_{i-1/2})$. Specifically, we have that:

$$\begin{aligned}
\gamma_1 = {} & \bigl(-\cos\bigl(h(-1+\alpha)\bigr) + \cos\bigl(h(2+\alpha)\bigr) - \cos\bigl(h(2+3\alpha)\bigr) + \cos(h+3h\alpha) \\
& + 2h\sin(h) + 4h\alpha\sin(h) + h\sin\bigl(h(-1+\alpha)\bigr) + h\alpha\sin\bigl(h(-1+\alpha)\bigr) \\
& - h\alpha\sin\bigl(h(2+\alpha)\bigr) - h\alpha\sin\bigl(h(2+3\alpha)\bigr) - 2h\sin(h+2h\alpha) \\
& + h\sin(h+3h\alpha) + h\alpha\sin(h+3h\alpha)\bigr)\big/\bigl(\chi\,(-h\alpha\cos(h\alpha)+\sin(h\alpha))\bigr), \\
\gamma_2 = {} & 1 - \gamma_1 - \gamma_3, \\
\gamma_3 = {} & \bigl(\cos\bigl(h(-1+\alpha)\bigr) - \cos\bigl(h(2+\alpha)\bigr) + \cos\bigl(h(2+3\alpha)\bigr) - \cos(h+3h\alpha) \\
& + 4h\sin(h\alpha) + 4h\alpha\sin(h\alpha) - 2h\sin(2h\alpha) - 2h\alpha\sin(2h\alpha) - 4h\alpha\sin\bigl(h(1+\alpha)\bigr) \\
& + 2h\alpha\sin\bigl(2h(1+\alpha)\bigr)\bigr)\big/\bigl(\chi\,(h\alpha\cos(h\alpha)-\sin(h\alpha))\bigr),
\end{aligned} \tag{2.10}$$

where $\chi$ is same as above.

*Step 2.3*. For the smaller stencils $S_n$, $n = 1, 2, 3$, we compute the smoothness indicators, denoted by $\beta_n$, which measure how smooth the functions $p_n(x)$ are in the target cell $I_i$. The smaller these smoothness indicators, the smoother the functions are in the target cell. We use the same recipe for the smoothness indicators as in [11]:

$$\beta_n = \sum_{\eta=1}^{2} \int_{I_i} h^{2\eta-1} \left( \frac{d^\eta p_n(x)}{dx^\eta} \right)^2 dx, \quad n = 1, 2, 3, \tag{2.11}$$

to obtain the following relations for $\beta_1$, $\beta_2$, $\beta_3$:

$$\begin{aligned}
\beta_1 = {} & \chi\bigl(2h\alpha\bigl(2u_{i-1}^2 + u_{i-2}^2 + u_i^2 - 2u_{i-1}(u_{i-2}+u_i)\bigr) \\
& + 4h\alpha(u_{i-1}-u_{i-2})(u_{i-1}-u_i)\cos(h\alpha) + 2(u_{i-1}-u_{i-2})(u_{i-1}-u_i)\sin(h\alpha) \\
& - (u_{i-2}-u_i)(-2u_{i-1}+u_{i-2}+u_i)\sin(2h\alpha) - 2(u_{i-1}-u_{i-2})(u_{i-1}-u_i)\sin(3h\alpha) \\
& - (u_{i-1}-u_i)^2\sin(4h\alpha) + h^2\alpha^2\bigl(2h\alpha\bigl(2u_{i-1}^2 + u_{i-2}^2 + u_i^2 - 2u_{i-1}(u_{i-2}+u_i)\bigr) \\
& + 4h\alpha(u_{i-1}-u_{i-2})(u_{i-1}-u_i)\cos(h\alpha) - 2(u_{i-1}-u_{i-2})(u_{i-1}-u_i)\sin(h\alpha) \\
& + (u_{i-2}-u_i)(-2u_{i-1}+u_{i-2}+u_i)\sin(2h\alpha) \\
& + 2(u_{i-1}-u_{i-2})(u_{i-1}-u_i)\sin(3h\alpha) + (u_{i-1}-u_i)^2\sin(4h\alpha)\bigr)\bigr)/256, \\
\beta_2 = {} & \chi\bigl(-4h\alpha\bigl(1+h^2\alpha^2\bigr)(u_{i-1}-u_i)(u_i-u_{i+1})\cos(h\alpha) \\
& - 4\bigl(-1+h^2\alpha^2\bigr)(u_{i-1}-u_i)(u_i-u_{i+1})\sin(h\alpha) \\
& + \bigl(u_{i-1}^2 - 2u_{i-1}u_i + 2u_i^2 - 2u_iu_{i+1} + u_{i+1}^2\bigr)\bigl(2h\alpha\bigl(1+h^2\alpha^2\bigr) \\
& + \bigl(-1+h^2\alpha^2\bigr)\sin(2h\alpha)\bigr)\bigr)/256, \\
\beta_3 = {} & \chi\bigl(2h\alpha u_i^2 + 2h^3\alpha^3u_i^2 - 4h\alpha u_iu_{i+1} - 4h^3\alpha^3u_iu_{i+1} + 4h\alpha u_{i+1}^2 + 4h^3\alpha^3u_{i+1}^2 \\
& - 4h\alpha u_{i+1}u_{i+2} - 4h^3\alpha^3u_{i+1}u_{i+2} + 2h\alpha u_{i+2}^2 + 2h^3\alpha^3u_{i+2}^2 \\
& - 4h\alpha\bigl(1+h^2\alpha^2\bigr)(u_i-u_{i+1})(u_{i+1}-u_{i+2})\cos(h\alpha) \\
& + 2\bigl(-1+h^2\alpha^2\bigr)(u_i-u_{i+1})(u_{i+1}-u_{i+2})\sin(h\alpha) + u_i^2\sin(2h\alpha)
\end{aligned} \tag{2.12}$$

$$
\begin{aligned}
&- h^2\alpha^2 u_i^2 \sin(2h\alpha) - 2u_i u_{i+1} \sin(2h\alpha) + 2h^2\alpha^2 u_i u_{i+1} \sin(2h\alpha) \\
&+ 2u_{i+1}u_{i+2} \sin(2h\alpha) - 2h^2\alpha^2 u_{i+1}u_{i+2} \sin(2h\alpha) - u_{i+2}^2 \sin(2h\alpha) \\
&+ h^2\alpha^2 u_{i+2}^2 \sin(2h\alpha) + 2u_i u_{i+1} \sin(3h\alpha) - 2h^2\alpha^2 u_i u_{i+1} \sin(3h\alpha) - 2u_{i+1}^2 \sin(3h\alpha) \\
&+ 2h^2\alpha^2 u_{i+1}^2 \sin(3h\alpha) - 2u_i u_{i+2} \sin(3h\alpha) + 2h^2\alpha^2 u_i u_{i+2} \sin(3h\alpha) \\
&+ 2u_{i+1}u_{i+2} \sin(3h\alpha) - 2h^2\alpha^2 u_{i+1}u_{i+2} \sin(3h\alpha) - u_i^2 \sin(4h\alpha) \\
&+ h^2\alpha^2 u_i^2 \sin(4h\alpha) + 2u_i u_{i+1} \sin(4h\alpha) - 2h^2\alpha^2 u_i u_{i+1} \sin(4h\alpha) - u_{i+1}^2 \sin(4h\alpha) \\
&+ h^2\alpha^2 u_{i+1}^2 \sin(4h\alpha)\big)/256,
\end{aligned}
$$

where $\chi = \frac{h^3\alpha^3}{\sin((h\alpha)/2)^6 \cos((h\alpha)/2)^2}$.

*Step 2.4.* Then we compute the nonlinear weights based on the linear weights and smoothness indicators [18]:

$$
\omega_n = \frac{\overline{\omega}_n}{\sum_{k=1}^3 \overline{\omega}_k}, \quad \overline{\omega}_n = \frac{\gamma_n}{\sum_{k=1}^3 (\varepsilon + \beta_k)^2}, \quad n = 1, 2, 3, \tag{2.13}
$$

where $\gamma_n$ are the linear weights determined in the above step, and $\varepsilon$ is a small positive number to avoid division by 0. In this paper, we use $\varepsilon = 10^{-6}$ in all the computations. The final approximations are given by:

$$
u^{\mp}_{i\pm\frac{1}{2}} \approx \sum_{n=1}^{3} \omega_n p_n(x_{i\pm\frac{1}{2}}).
$$

*Step 2.5.* The semi-discrete scheme (2.3) is then discretized in time by a Runge-Kutta method [20], for example the fourth order version given by:

$$
\begin{cases}
u^{(1)} = u^n + \frac{1}{2}\Delta t L(u^n), \\
u^{(2)} = u^n + \frac{1}{2}\Delta t L(u^{(1)}), \\
u^{(3)} = u^n + \Delta t L(u^{(2)}), \\
u^{n+1} = -\frac{1}{3}u^n + \frac{1}{3}u^{(1)} + \frac{2}{3}u^{(2)} + \frac{1}{3}u^{(3)} + \frac{1}{6}\Delta t L(u^{(3)}).
\end{cases} \tag{2.14}
$$

*Remark*

1. 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.
2. For the two dimensional case, we construct the finite volume TWENO schemes on uniform meshes by a "dimension by dimension" procedure. We consider the quadrature point $(x_G, y_G)$. First, we perform a one-dimensional TWENO reconstruction in the $y$-direction, in order to get the one-dimensional cell averages (in the $x$-direction) $w(\bullet, y_G)$. Then, we perform another one-dimensional TWENO reconstruction to $w$ in the $x$-direction, to obtain the final reconstructed point value at $(x_G, y_G)$. The details of "dimension by dimension" procedure of TWENO is similar to that of WENO which can be found in [22, 23].
3. The procedure for construction of any $(2r + 1)$-th order TWENO scheme is similar to that for the fifth order scheme.

## 3 TWENO Reconstruction as Limiters for the DG Method

In [14, 26], we have started the study of using WENO reconstruction methodology as limiters for the RKDG method with algebraic polynomial basis. The first step in the procedure is to identify the "troubled cells", namely those cells which might need the limiting procedure. In this paper, we use the minmod type TVB limiters as in [19] to identify the "troubled cells". That is, whenever the minmod limiter changes the slope, the cell is declared to be a "troubled cell". The second step is to replace the solution polynomials in the "troubled cells" by reconstructed polynomials, which maintain the original cell averages (for conservation) and have the same order of accuracy as before, but are less oscillatory. In [14, 26], finite volume type WENO reconstruction based on cell averages of neighbors is used for the second step. In this section, we apply the TWENO reconstruction while maintaining the same high order accuracy.

Given a division of cells $\Delta_j$ (intervals in 1D, quadrilaterals in 2D, etc.), a semi-discrete discontinuous Galerkin method for solving the conservation law (2.1) is obtained by multiplying (2.1) with a test function $v(x)$, integrating over a cell $\Delta_j$, and integrating by parts:

$$\frac{d}{dt}\int_{\Delta_j} u(x,t)v(x)dx - \int_{\Delta_j} f(u)\cdot\nabla v\,dx + \int_{\partial\Delta_j} f(u)\cdot n v\,ds = 0 \tag{3.1}$$

where $n$ is the outward unit normal of the cell boundary $\partial\Delta_j$. We seek a piecewise trigonometric polynomial $u$ in $\mathbb{T}^k$ of degree at most $k$, such that (3.1) holds for any test function $v$ also in $\mathbb{T}^k$. We could actually change $k$ from cell to cell, but for simplicity we assume it is constant over the whole division. The boundary integral in (3.1) is typically discretized by a Gaussian quadrature of sufficiently high order of accuracy:

$$\int_{\partial\Delta_j} f\cdot n\,ds \approx |\partial\Delta_j|\sum_{G=1}^{q}\sigma_G f\big(u(x_G,t)\big)\cdot n$$

and $f(u(x_G,t))\cdot n$ is replaced by 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:

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

where $\bar{\alpha}$ 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. $u^-$ and $u^+$ are the respective values of $u$ inside and outside the cell $\Delta_j$ (i.e the latter refers to the value inside the neighboring cell) at the Gaussian point $x_G$. The idea of using such a numerical flux is borrowed from a finite volume methodology. The test function $v$ in the boundary integral in (3.1) is taken from inside the cell $\Delta_j$. The volume integral $\int_{\Delta_j} f(u)\cdot\nabla v\,dx$ in (3.1) is either computed exactly for simple $f(u)$ or by a numerical quadrature with sufficient accuracy, see [4, 7] for details. The semi-discrete scheme (3.1), written as $u_t = L(u)$ is then discretized in time by a TVB Runge-Kutta method (2.14).

To give the procedure using the TWENO reconstruction as a limiter to the RKDG method, we start with the description in the one dimensional case. For simplicity of presentation, we also assume that the mesh is uniformly distributed. The DG solution as well as the test function spaces are both given by $V_h^k = \{p : p|_{I_i} \in T^k(I_i)\}$, the trigonometric polynomial spaces of degree at most $k$ on the cell $I_i$. We adopt the following local orthogonal basis over $I_i$, $\{v_l^{(i)}(x), l = 0, \dots, k\}$:

$$\begin{aligned}
v_0^{(i)}(x) &= 1, \\
v_1^{(i)}(x) &= \sin\bigl(\alpha(x - x_i)\bigr), \\
v_2^{(i)}(x) &= \cos\bigl(\alpha(x - x_i)\bigr) - \frac{\sin(h\alpha/2)}{h\alpha/2}, \\
&\ \ \vdots\ .
\end{aligned}$$

Unless we specify otherwise, we set $\alpha = 1.0$ in all 1D examples of Sect. 4. Then, the numerical solution $u^h(x,t)$ in the space $V_h^k$ can be written as $u^h(x,t) = \sum_{l=0}^k u_i^{(l)}(t) v_l^{(i)}(x)$, for $x \in I_i$ and the degrees of freedom $u_i^{(l)}(t)$ are the moments defined by $u_i^{(l)}(t) = \frac{1}{\int_{I_i}(v_l^{(i)}(x))^2 dx} \int_{I_i} u^h(x,t) v_l^{(i)}(x) dx$, $l = 0, \dots, k$, due to the local orthogonal basis. In order to determine the approximate solution, we evolve the degrees of freedom $u_i^{(l)}(t)$ by using the following relation:

$$\begin{aligned}
\frac{d}{dt} u_i^{(l)}(t) = -\frac{1}{\int_{I_i}(v_l^{(i)}(x))^2 dx} \biggl( &-\int_{I_i} f\bigl(u^h(x,t)\bigr) \frac{d}{dx} v_l^{(i)}(x) dx \\
&+ \hat{f}\bigl(u_{i+1/2}^-, u_{i+1/2}^+\bigr) v_l^{(i)}(x_{i+1/2}) \\
&- \hat{f}\bigl(u_{i-1/2}^-, u_{i-1/2}^+\bigr) v_l^{(i)}(x_{i-1/2}) \biggr), \quad l = 0, \dots, k,
\end{aligned} \tag{3.2}$$

which is obtained by substituting the expansion for $u^h(x,t)$ and letting $v = v_m^{(i)}(x)$ in (3.1). $u_{i+1/2}^{\pm} = u^h(x_{i+1/2}^{\pm}, t)$ are the left and right limits of the discontinuous solution $u^h(x,t)$ at the cell interface $x_{i+1/2}$, $\hat{f}(u^-, u^+)$ is a monotone flux for the scalar case and an exact or approximate Riemann solver for the system case. The integral term in (3.2) can be computed either exactly for simple $f(u)$ or by a suitable numerical quadrature accurate to at least $O(h^{k+l+2})$. The semi-discrete scheme (3.2) is discretized in time by a nonlinearly stable Runge-Kutta time discretization, e.g. the fourth order version (2.14). The method described above can compute solutions to (2.1) 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, a limiter is needed.

In this paper, we will use the TVB limiter [7, 14, 19] to identify "troubled cells":

$$u_{i+1/2}^- = u_i^{(0)} + \tilde{u}_i = u_i^{(0)} + \sum_{l=1}^k u_i^{(l)} v_l^{(i)}(x_{i+1/2}), \tag{3.3}$$

$$u_{i-1/2}^+ = u_i^{(0)} - \tilde{\tilde{u}}_i = u_i^{(0)} - \biggl( -\sum_{l=1}^k u_i^{(l)} v_l^{(i)}(x_{i-1/2}) \biggr). \tag{3.4}$$

These are modified by the standard minmod limiter: $\tilde{u}_i^{(\mathrm{mod})} = m(\tilde{u}_i, \Delta_+ u_i^{(0)}, \Delta_- u_i^{(0)})$, $\tilde{\tilde{u}}_i^{(\mathrm{mod})} = m(\tilde{\tilde{u}}_i, \Delta_+ u_i^{(0)}, \Delta_- u_i^{(0)})$, where $m$ is given by:

$$m(a_1, \dots, a_n) = \begin{cases} s \cdot \min_{1\le j\le n} |a_j|, & \text{if } \operatorname{sign}(a_1) = \cdots = \operatorname{sign}(a_n) = s, \\ 0, & \text{otherwise}, \end{cases} \tag{3.5}$$

or by the TVB modified minmod function:

$$\tilde{m}(a_1, \dots, a_n) = \begin{cases} a_1, & \text{if } |a_1| \le M\Delta x^2, \\ m(a_1, \dots, a_n), & \text{otherwise}, \end{cases} \tag{3.6}$$

where the choice of the constant $M > 0$ depends on the solution of the problem. How to choose the optimal TVB constant $M$ for system case is still an open problem. If one of the minmod functions gets enacted (returns other than the first argument), this cell is declared as "troubled" and marked for further reconstruction. Since the TWENO reconstruction maintains the high order accuracy in the "troubled cells", it is less crucial to choose an accurate $M$. Basically, if $M$ is chosen too small, more good cells will be declared as "troubled cells" and will be subject to unnecessary TWENO reconstructions. This does increase the computational cost, but does not degrade the order of accuracy in the affected cells. For the "troubled cells", we would like to reconstruct the trigonometric polynomial solutions while retaining their cell averages. In other words, we will reconstruct the degrees of freedom, or the moments, $u_i^{(l)}$, $l = 1, \dots, k$ for the "troubled cells" $I_i$ and retain only the cell average $u_i^{(0)}$.

We will adopt a similar routine as in [14] to reconstruct the moments. First, we reconstruct point values of $u$ at the Gauss or Gauss-Lobatto quadrature points in the "troubled cell" $I_i$ by TWENO reconstruction, then use those reconstructed point values to reconstruct the moments in the cell.

For $(k+1)$-th order accuracy, we need a Gauss or Gauss-Lobatto quadrature rule accurate to at least $\mathrm{O}(h^{2k+2})$ and the order of accuracy for the TWENO 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. For example, for $k = 2$, we use the four-point Gauss-Lobatto quadrature points $x_G$: $x_{i-1/2}, x_{i-\sqrt{5}/10}, x_{i+\sqrt{5}/10}$ and $x_{i+1/2}$, respectively. The procedure of reconstructing point values of $u$ at $x_G$ is the same as in Steps 2.1–2.4. In particular, the expressions for the trigonometric polynomials $p_n(x)$, $n = 1, 2, 3$ at $x_G = x_{i\pm 1/2}$ are given by (2.7) and (2.8), and for the linear weights by (2.9) and (2.10). Then, we present the procedure of the other two Gauss-Lobatto quadrature points $x_G = x_{i\pm\sqrt{5}/10}$ in detail:

*Step 3.1.* The expressions for $p_n(x)$, $n = 1, 2, 3$ at $x_G = x_{i+\sqrt{5}/10}$ and $x_G = x_{i-\sqrt{5}/10}$ are:

$$\begin{aligned} p_1(x_{i+\sqrt{5}/10}) &= \chi\bigg(\big(u_{i-1}^{(0)} - u_{i-2}^{(0)}\big)\cos\bigg(\frac{(5+\sqrt{5})h\alpha}{10}\bigg) + \big(u_{i-1}^{(0)} - u_i^{(0)}\big)\cos\bigg(\frac{(15+\sqrt{5})h\alpha}{10}\bigg) \\ &\quad + \big(u_{i-2}^{(0)} + u_i^{(0)}\big)\frac{\sin(h\alpha)}{h\alpha} - u_{i-1}^{(0)}\frac{\sin(2h\alpha)}{h\alpha}\bigg), \\ p_2(x_{i+\sqrt{5}/10}) &= -\chi\bigg(\big(u_{i-1}^{(0)} - u_i^{(0)}\big)\cos\bigg(\frac{(-5+\sqrt{5})h\alpha}{10}\bigg) - \big(u_i^{(0)} - u_{i+1}^{(0)}\big)\cos\bigg(\frac{(5+\sqrt{5})h\alpha}{10}\bigg) \\ &\quad - \big(u_{i-1}^{(0)} + u_{i+1}^{(0)} - 2u_i^{(0)}\cos(h\alpha)\big)\frac{\sin(h\alpha)}{h\alpha}\bigg), \end{aligned} \tag{3.7}$$

$$
\begin{aligned}
p_3(x_{i+\sqrt{5}/10}) &= -\chi\biggl(\bigl(u_i^{(0)} - u_{i+1}^{(0)}\bigr)\cos\biggl(\frac{(-15+\sqrt{5})h\alpha}{10}\biggr) \\
&\quad - \bigl(u_{i+1}^{(0)} - u_{i+2}^{(0)}\bigr)\cos\biggl(\frac{(-5+\sqrt{5})h\alpha}{10}\biggr) \\
&\quad - \bigl(u_i^{(0)} + u_{i+2}^{(0)} - 2u_{i+1}^{(0)}\cos(h\alpha)\bigr)\frac{\sin(h\alpha)}{h\alpha}\biggr), \\
p_1(x_{i-\sqrt{5}/10}) &= \chi\biggl(\bigl(u_{i-1}^{(0)} - u_i^{(0)}\bigr)\cos\biggl(\frac{(-15+\sqrt{5})h\alpha}{10}\biggr) \\
&\quad + \bigl(u_{i-1}^{(0)} - u_{i-2}^{(0)}\bigr)\cos\biggl(\frac{(-5+\sqrt{5})h\alpha}{10}\biggr) \\
&\quad + \bigl(u_{i-2}^{(0)} + u_i^{(0)}\bigr)\frac{\sin(h\alpha)}{h\alpha} - u_{i-1}^{(0)}\frac{\sin(2h\alpha)}{h\alpha}\biggr), \\
p_2(x_{i-\sqrt{5}/10}) &= \chi\biggl(\bigl(u_i^{(0)} - u_{i+1}^{(0)}\bigr)\cos\biggl(\frac{(-5+\sqrt{5})h\alpha}{10}\biggr) \\
&\quad - \bigl(u_{i-1}^{(0)} - u_i^{(0)}\bigr)\cos\biggl(\frac{(5+\sqrt{5})h\alpha}{10}\biggr) \\
&\quad + \bigl(u_{i-1}^{(0)} + u_{i+1}^{(0)}\bigr)\frac{\sin(h\alpha)}{h\alpha} - u_i^{(0)}\frac{\sin(2h\alpha)}{h\alpha}\biggr), \\
p_3(x_{i-\sqrt{5}/10}) &= \chi\biggl(\bigl(u_{i+1}^{(0)} - u_{i+2}^{(0)}\bigr)\cos\biggl(\frac{(5+\sqrt{5})h\alpha}{10}\biggr) - \bigl(u_i^{(0)} - u_{i+1}^{(0)}\bigr)\cos\biggl(\frac{(15+\sqrt{5})h\alpha}{10}\biggr) \\
&\quad + \bigl(u_i^{(0)} + u_{i+2}^{(0)}\bigr)\frac{\sin(h\alpha)}{h\alpha} - u_{i+1}^{(0)}\frac{\sin(2h\alpha)}{h\alpha}\biggr),
\end{aligned}
\tag{3.8}
$$

where $\chi = \frac{h\alpha}{8\sin(h\alpha/2)^3\cos(h\alpha/2)}$.

*Step 3.2.* We find the linear weights, such that $q(x_{i+\sqrt{5}/10}) = \sum_{n=1}^{3}\gamma_n p_n(x_{i+\sqrt{5}/10})$. Specifically, we have that:

$$
\begin{aligned}
\gamma_1 &= \bigl(\cos\bigl(h(-1+\alpha)\bigr) - \cos\bigl(h(2+\alpha)\bigr) + \cos\bigl(h(2+3\alpha)\bigr) - \cos(h+3h\alpha) \\
&\quad + h\alpha\sin\bigl(\bigl(h(20+25\alpha-\sqrt{5}\alpha)\bigr)/10\bigr) + h\sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(-25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
&\quad + h\alpha\sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(-25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
&\quad - 2h\alpha\sin\bigl(h\bigl(1-\bigl((-15+\sqrt{5})\alpha\bigr)/10\bigr)\bigr) \\
&\quad - 2h\alpha\sin\bigl(\bigl(h\bigl(10+(5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) + 2h\sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
&\quad + 2h\alpha\sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) + h\alpha\sin\bigl(\bigl(h\bigl(20+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
&\quad - h\sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
&\quad - h\alpha\sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
&\quad + 2h\sin\bigl(h\alpha + \bigl(h(1+\alpha)\bigr)/2 - \bigl(h(1+\alpha)\bigr)/(2\sqrt{5})\bigr) + 2h\alpha\sin\bigl(h\alpha + \bigl(h(1+\alpha)\bigr)/2 \\
&\quad - \bigl(h(1+\alpha)\bigr)/(2\sqrt{5})\bigr)\bigr)/\bigl(\chi\bigl(-h\alpha\cos\bigl(\bigl((5+\sqrt{5})h\alpha\bigr)/10\bigr) + \sin(h\alpha)\bigr)\bigr), \\
\gamma_2 &= 1 - \gamma_1 - \gamma_3,
\end{aligned}
\tag{3.9}
$$

$$
\begin{aligned}
\gamma_3 = {} & \bigl(-\cos\bigl(h(-1+\alpha)\bigr) + \cos\bigl(h(2+\alpha)\bigr) - \cos\bigl(h(2+3\alpha)\bigr) + \cos(h+3h\alpha) \\
& + h\alpha \sin\bigl(\bigl(h\bigl(-20 + (-15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) - h\sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(-15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& - h\alpha \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(-15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& - 2h\alpha \sin\bigl(\bigl(h\bigl(-10+(-5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + 2h \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(-5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + 2h\alpha \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(-5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + 2h\alpha \sin\bigl(\bigl(h\bigl(10+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) - 2h\sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& - 2h\alpha \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) - h\alpha\sin\bigl(\bigl(h\bigl(20+(25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + h \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) + h\alpha\sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr)\bigr) \\
& /\bigl(\chi\bigl(h\alpha\cos\bigl(\bigl((-5+\sqrt{5})h\alpha\bigr)/10\bigr) - \sin(h\alpha)\bigr)\bigr),
\end{aligned}
$$

where $\chi = 32\sin((h(1+\alpha))/2)^3\cos((h(1+\alpha))/2)(\cos(h\alpha) - \cos(h(1+\alpha)))$.

And for $q(x_{i-\sqrt{5}/10}) = \sum_{n=1}^{3}\gamma_n p_n(x_{i-\sqrt{5}/10})$. Specifically, we have that:

$$
\begin{aligned}
\gamma_1 = {} & \bigl(-\cos\bigl(h(-1+\alpha)\bigr) + \cos\bigl(h(2+\alpha)\bigr) - \cos\bigl(h(2+3\alpha)\bigr) + \cos(h+3h\alpha) \\
& + h\alpha \sin\bigl(\bigl(h\bigl(-20 + (-15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) - h\sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(-15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& - h\alpha \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(-15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& - 2h\alpha \sin\bigl(\bigl(h\bigl(-10+(-5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + 2h \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(-5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + 2h\alpha \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(-5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + 2h\alpha \sin\bigl(\bigl(h\bigl(10+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) - 2h\sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& - 2h\alpha \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) - h\alpha\sin\bigl(\bigl(h\bigl(20+(25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + h \sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) + h\alpha\sin\bigl(\bigl(h\bigl(5+\sqrt{5}+(25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr)\bigr) \\
& /\bigl(\chi\bigl(h\alpha\cos\bigl(\bigl((-5+\sqrt{5})h\alpha\bigr)/10\bigr) - \sin(h\alpha)\bigr)\bigr), \\
\gamma_2 = {} & 1 - \gamma_1 - \gamma_3, \qquad (3.10) \\
\gamma_3 = {} & \bigl(\cos\bigl(h(-1+\alpha)\bigr) - \cos\bigl(h(2+\alpha)\bigr) + \cos\bigl(h(2+3\alpha)\bigr) - \cos(h+3h\alpha) \\
& + h\alpha \sin\bigl(\bigl(h(20+25\alpha-\sqrt{5}\alpha)\bigr)/10\bigr) + h\sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(-25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + h\alpha \sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(-25+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& - 2h\alpha \sin\bigl(h\bigl(1 - \bigl((-15+\sqrt{5})\alpha\bigr)/10\bigr)\bigr) \\
& - 2h\alpha \sin\bigl(\bigl(h\bigl(10+(5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) + 2h\sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) \\
& + 2h\alpha \sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(5+\sqrt{5})\alpha\bigr)\bigr)/10\bigr) + h\alpha\sin\bigl(\bigl(h\bigl(20+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr)
\end{aligned}
$$

$$\begin{aligned}
&- h \sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr)\\
&- h\alpha \sin\bigl(\bigl(h\bigl(-5+\sqrt{5}+(15+\sqrt{5})\alpha\bigr)\bigr)/10\bigr)\\
&+ 2h \sin\bigl(h\alpha + \bigl(h(1+\alpha)\bigr)/2 - \bigl(h(1+\alpha)\bigr)/(2\sqrt{5})\bigr) + 2h\alpha \sin\bigl(h\alpha + \bigl(h(1+\alpha)\bigr)/2\\
&- \bigl(h(1+\alpha)\bigr)/(2\sqrt{5})\bigr)\bigr)\big/\bigl(\chi\bigl(-h\alpha\cos\bigl(\bigl((5+\sqrt{5})h\alpha\bigr)/10\bigr)+\sin(h\alpha)\bigr)\bigr),
\end{aligned}$$

where $\chi$ is same as above.

The latter steps are similar and omitted here for simplicity.

Finally, we obtain the reconstructed moments based on the reconstructed point values $u(x_G)$ at the Gauss-Lobatto quadrature points $x_G$ and a numerical integration:

$$u_i^{(l)}(t) \approx \frac{1}{\sum_G \sigma_G (v_l^{(i)}(x_G))^2} \sum_G \sigma_G u(x_G, t) v_l^{(i)}(x_G), \quad l = 1, 2, \tag{3.11}$$

where $\sigma_G$ is the Gauss-Lobatto quadrature weight for the point $x_G$ ($\frac{1}{12}$ for $x_{i\pm1/2}$ and $\frac{5}{12}$ for $x_{i\pm\sqrt{5}/10}$). The trigonometric polynomial solution in this cell $I_i$ is then obtained by $u^h(x,t) = \sum_{l=0}^{2} u_i^{(l)}(t) v_l^{(i)}(x)$, for $x \in I_i$ with these reconstructed moments $u_i^{(l)}(t), l = 1, 2$ and the original cell average $u_i^{(0)}(t)$.

**Remark:** *Instead of* using the Gauss-Lobatto *quadrature based on algebraic polynomials*, we also tested using the one based on *trigonometric polynomials*, which we define as follows: *Let* $\Phi = \frac{1}{h}\int_{x_i-h/2}^{x_i+h/2} \phi(x)dx$ *and* $\Phi_G = \sum_G \sigma_G \phi(x_G)$. *For* $k = 2$, *we define the four points Gauss-Lobatto quadrature such that* $\frac{1}{h}\int_{x_i-h/2}^{x_i+h/2} \phi(x)dx = \sum_G \sigma_G \phi(x_G)$, *for* $\phi(x)$ *to be the trigonometric polynomial functions* $1, \sin(\alpha(x-x_i)), \cos(\alpha(x-x_i)) - \frac{\sin(h\alpha/2)}{h\alpha/2}, \sin((\alpha+1)(x-x_i)), \cos((\alpha+1)(x-x_i)) - \frac{\sin(h(\alpha+1)/2)}{h(\alpha+1)/2}, \sin((\alpha+2)(x-x_i))$, *respectively*. We observed that numerical results of methods using the Gauss-Lobatto quadrature based on algebraic and trigonometric polynomials are similar to each other. Hence, for simplicity, we use in this paper the Gauss-Lobatto quadrature based on algebraic polynomials.

So, in this section, we have given the details of the minmod TVB limiters used to identify "troubled cells" for the one dimensional scalar cases (see also [7]).

For the one dimensional system case, we use the characteristic based limiter procedure [5], and all of the reconstructions are performed in the local characteristic directions to avoid oscillations.

For the two dimensional case, we adopt uniformly distributed cells $I_{ij} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$, with the cell sizes $x_{i+1/2} - x_{i-1/2} = \Delta x = h$, $y_{j+1/2} - y_{j-1/2} = \Delta y = h$ and cell centers $(x_i, y_j) = (\frac{1}{2}(x_{i+1/2}+x_{i-1/2}), \frac{1}{2}(y_{j+1/2}+y_{j-1/2}))$. The solution and test function space $V_h^k = \{p : p|_{I_{ij}} \in T^k(I_{ij})\}$ is the trigonometric polynomial spaces of degree at most $k$ on the cell $I_{ij}$, with a local orthogonal basis over $I_{ij}$, $\{v_l^{(ij)}(x,y), l = 0, 1, \dots, K; K = \frac{1}{2}(k+1)(k+2) - 1\}$, such as:

$$\begin{aligned}
v_0^{(ij)}(x,y) &= 1,\\
v_1^{(ij)}(x,y) &= \sin\bigl(\alpha(x-x_i)\bigr),\\
v_2^{(ij)}(x,y) &= \sin\bigl(\beta(y-y_j)\bigr),
\end{aligned}$$

$$
\begin{aligned}
v_3^{(ij)}(x,y) &= \cos\bigl(\alpha(x-x_i)\bigr) - \frac{\sin(h\alpha/2)}{h\alpha/2},\\
v_4^{(ij)}(x,y) &= \sin\bigl(\alpha(x-x_i)\bigr)\sin\bigl(\beta(y-y_j)\bigr),\\
v_5^{(ij)}(x,y) &= \cos\bigl(\beta(y-y_j)\bigr) - \frac{\sin(h\beta/2)}{h\beta/2},\\
&\vdots\,.
\end{aligned}
$$

Unless we specify otherwise, we set $\alpha = 1.0$ and $\beta = 1.0$ in all 2D examples of Sect. 4. To identify "troubled cells", we use the TVB limiter defined in [4] for the two dimensional scalar case and the characteristic based procedure described in [8] for the two dimensional system case. For the "troubled cells", we would like to reconstruct the trigonometric polynomial solutions while retaining their cell averages. In other words, we will reconstruct the degrees of freedom, or the moments, $u_{ij}^{(l)}$, $l = 1, \dots, K$ for the "troubled cells" $I_{ij}$ and retain only the cell averages $u_{ij}^{(0)}$. 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. For the actual TWENO reconstruction, we use a "dimension by dimension" procedure, which is the same as that in [17, 18] for WENO.

## 4 Numerical Tests

In this section, we present results of numerical tests using some of the schemes proposed in the previous sections. We will show the results obtained by the fifth order TWENO scheme (TWENO5), third order RKDG methods based on trigonometric polynomial space with or without TWENO limiter (TWENO5-RKDG-T and RKDG-T, respectively), and for comparison purposes by the corresponding WENO5, WENO5-RKDG-A or RKDG-A schemes, which are based on the algebraic polynomial space. We set $\alpha = 1.0$ and $\beta = 1.0$ in all the examples, unless we specify otherwise. We specifically employ the aforementioned schemes to solve the 1-D and 2-D Burger's equation, 1-D and 2-D System of Euler equations, as well as a particular 2-D Advection equation, all with various appropriate initial conditions. In the figures we provide, we use "plus" and "square" signs to indicate numerical solutions and a "solid line" for exact solutions.

*Example 4.1* We solve the following nonlinear scalar Burger's equation:

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

with the initial condition $u(x,0) = 0.5 + \sin(\pi x)$, $x \in [0,2]$ with periodic boundary conditions. We compute the solution up to $t = 0.5/\pi$. The errors and numerical orders of accuracy associated with the finite volume schemes and DG schemes with TVB constant $M = 0.01$ are shown in Table 1. And the numerical error against CPU time graphs are shown in Figs. 1, 2 and 3. We observe that all schemes achieve their designed orders of accuracy, as expected. Also, the numerical errors by the RKDG methods based on trigonometric polynomial spaces and algebraical polynomial spaces are very similar to one another at the same mesh level.

**Table 1** $u_t + (u^2/2)_x = 0$. $u(x,0) = 0.5 + \sin(\pi x)$. Periodic boundary conditions. TVB constant $M = 0.01$. $t = 0.5/\pi$. $L^1$ and $L^\infty$ errors and numerical orders of accuracy

| Cells | TWENO5, $\alpha = \pi$ | | | |
|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order |
| 10 | 4.23E−03 | | 1.68E−02 | |
| 20 | 6.59E−04 | 2.68 | 4.05E−03 | 2.05 |
| 40 | 6.91E−05 | 3.25 | 6.00E−04 | 2.76 |
| 80 | 3.65E−06 | 4.24 | 3.40E−05 | 4.14 |
| 160 | 1.51E−07 | 4.59 | 1.27E−06 | 4.74 |
| 320 | 4.71E−09 | 5.01 | 4.60E−08 | 4.79 |

| Cells | WENO5 | | | |
|---|---|---|---|---|
| | $L^1$ error | order | $L^\infty$ error | order |
| 10 | 5.85E−3 | | 1.71E−2 | |
| 20 | 9.88E−4 | 2.57 | 7.13E−3 | 1.26 |
| 40 | 8.71E−5 | 3.50 | 8.04E−4 | 3.15 |
| 80 | 4.11E−6 | 4.40 | 4.11E−5 | 4.29 |
| 160 | 1.64E−7 | 4.64 | 1.45E−6 | 4.82 |
| 320 | 5.17E−9 | 4.99 | 4.66E−8 | 4.96 |

| Cells | TWENO5-RKDG-T, $\alpha = \pi$ | | | | RKDG-T, $\alpha = \pi$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order | $L^1$ error | Order | $L^\infty$ error | Order |
| 10 | 6.52E−3 | | 7.05E−2 | | 1.64E−3 | | 2.49E−2 | |
| 20 | 6.31E−4 | 3.37 | 5.12E−3 | 3.78 | 1.94E−4 | 3.08 | 3.76E−3 | 2.73 |
| 40 | 4.67E−5 | 3.76 | 6.50E−4 | 2.98 | 2.51E−5 | 2.96 | 6.50E−4 | 2.53 |
| 80 | 4.64E−6 | 3.33 | 9.14E−5 | 2.83 | 3.21E−6 | 2.97 | 9.14E−5 | 2.83 |
| 160 | 5.27E−7 | 3.14 | 1.27E−5 | 2.85 | 4.08E−7 | 2.98 | 1.27E−5 | 2.85 |
| 320 | 6.24E−8 | 3.08 | 1.68E−6 | 2.92 | 5.15E−8 | 2.99 | 1.68E−6 | 2.92 |

| Cells | WENO5-RKDG-A | | | | RKDG-A | | | |
|---|---|---|---|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order | $L^1$ error | Order | $L^\infty$ error | Order |
| 10 | 6.23E−3 | | 7.69E−2 | | 1.76E−3 | | 2.92E−2 | |
| 20 | 5.53E−4 | 3.49 | 5.50E−3 | 3.80 | 2.07E−4 | 3.08 | 4.55E−3 | 2.68 |
| 40 | 4.04E−5 | 3.77 | 7.59E−4 | 2.86 | 2.67E−5 | 2.96 | 7.59E−4 | 2.58 |
| 80 | 4.31E−6 | 3.22 | 1.06E−4 | 2.84 | 3.38E−6 | 2.98 | 1.06E−4 | 2.84 |
| 160 | 5.20E−7 | 3.05 | 1.46E−5 | 2.86 | 4.27E−7 | 2.99 | 1.46E−5 | 2.86 |
| 320 | 6.44E−8 | 3.01 | 1.92E−6 | 2.93 | 5.37E−8 | 2.99 | 1.92E−6 | 2.93 |

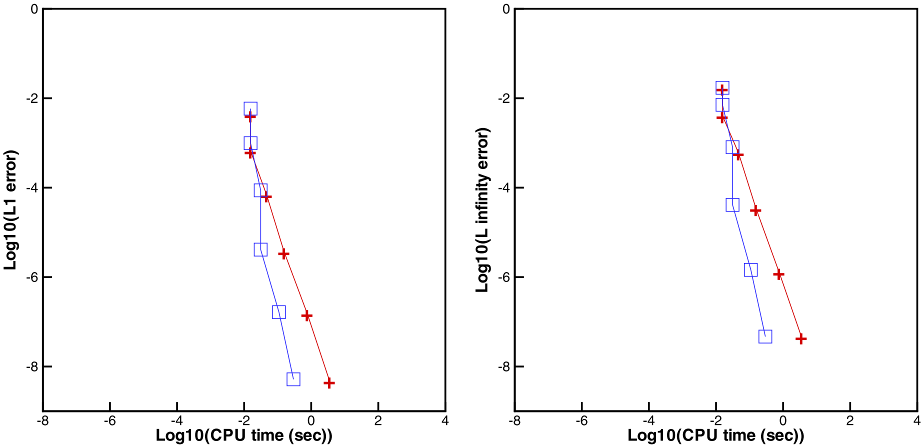

**Fig. 1** Computing time and error of $u_t + (u^2/2)_x = 0$. $u(x,0) = 0.5 + \sin(\pi x)$. Periodic boundary conditions. $t = 0.5/\pi$. *Plus signs* and a *solid line* denote the results of TWENO5, $\alpha = \pi$ scheme; *squares* and a *solid line* denote the results of WENO5 scheme

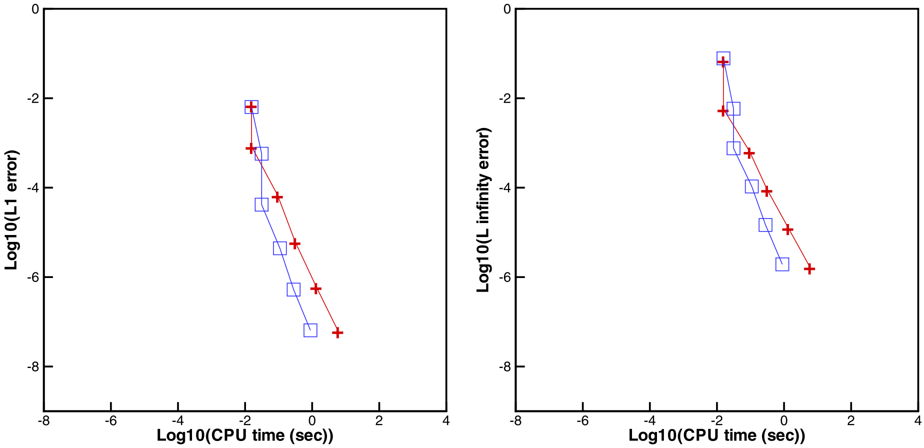

**Fig. 2** Computing time and error of $u_t + (u^2/2)_x = 0$. $u(x,0) = 0.5 + \sin(\pi x)$. Periodic boundary conditions. TVB constant $M = 0.01$. $t = 0.5/\pi$. *Plus signs* and a *solid line* denote the results of TWENO5-RKDG-T, $\alpha = \pi$ scheme; *squares* and a *solid line* denote the results of WENO5-RKDG-A scheme

*Example 4.2* We solve the one dimensional Euler equations:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2 + p\\ u(E+p)\end{pmatrix} = 0, \tag{4.2}$$

in which $\rho$ is density, $u$ is the velocity in the $x$-direction, $E$ is total energy, and $p$ is pressure. The initial conditions are: $\rho(x,0) = 1 + 0.2\sin(\pi x)$, $u(x,0) = 1$, $p(x,0) = 1$, $x \in [0,2]$ with periodic boundary conditions. We compute the density solution up to $t = 1$. Errors and numerical orders of accuracy achieved by the finite volume and DG schemes with TVB con-

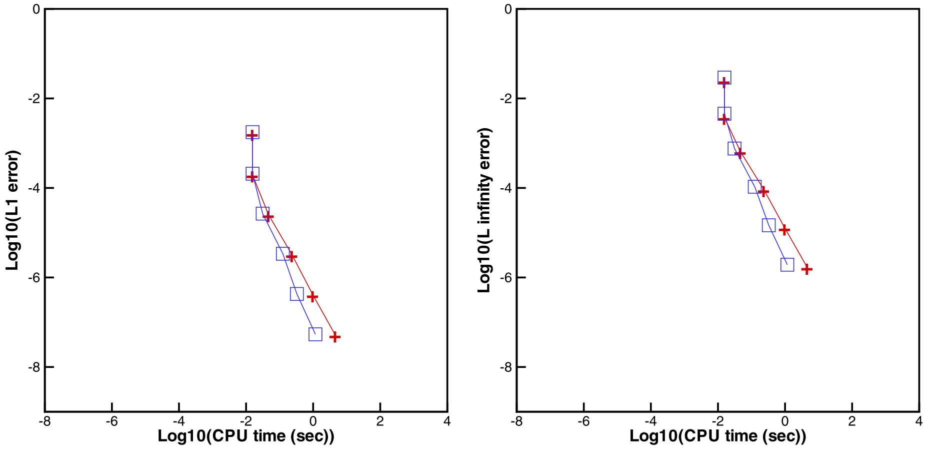

**Fig. 3** Computing time and error of $u_t + (u^2/2)_x = 0$. $u(x,0) = 0.5 + \sin(\pi x)$. Periodic boundary conditions. $t = 0.5/\pi$. *Plus signs* and a *solid line* denote the results of RKDG-T, $\alpha = \pi$ scheme; *squares* and a *solid line* denote the results of RKDG-A scheme

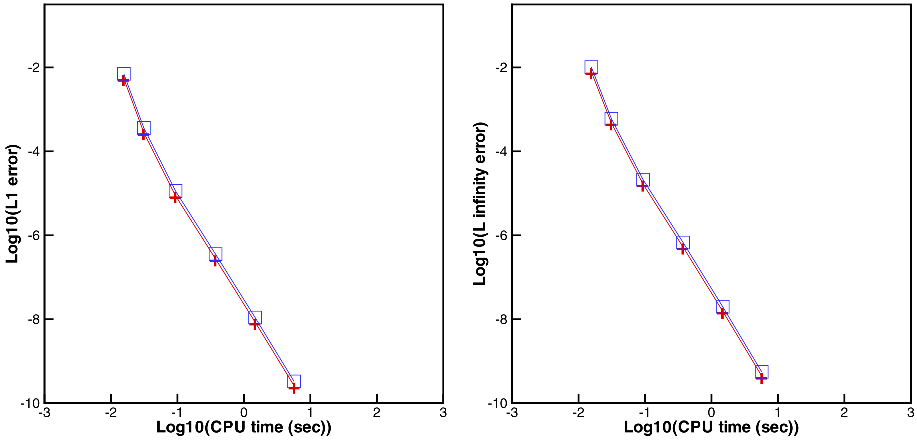

**Fig. 4** Computing time and error of 1D Euler equations. $\rho(x,0) = 1 + 0.2\sin(\pi x)$, $u(x,0) = 1$, $p(x,0) = 1$. Periodic boundary conditions. $t = 1$. *Plus signs* and a *solid line* denote the results of TWENO5, $\alpha = 1$ scheme; *squares* and a *solid line* denote the results of WENO5 scheme

stant $M = 0.01$ are shown in Table 2. And the numerical error against CPU time graphs are shown in Figs. 4, 5 and 6. When we set in this test case the parameter $\alpha$ to be optimal (i.e. $\alpha = \pi$), we obtain the exact solutions modulo round-off errors. That is because the approximation solution space of the RKDG methods is equal to the space of the exact solution of the problem. When we use the non-optimal $\alpha = 1.0$, instead, we observe that all schemes achieve their designed orders of accuracy, as expected. Moreover the numerical errors by the RKDG based on trigonometric polynomial spaces are smaller than those by RKDG based on algebraical polynomial spaces at the same mesh level.

**Table 2** 1D Euler equations. $\rho(x,0)=1+0.2\sin(\pi x)$, $u(x,0)=1$, $p(x,0)=1$. Periodic boundary conditions. TVB constant $M=0.01$. $t=1$. $L^1$ and $L^\infty$ errors and numerical orders of accuracy

| Cells | TWENO5, $\alpha=\pi$ | | TWENO5, $\alpha=1$ | | | |
|---|---|---|---|---|---|---|
| | $L^1$ error | $L^\infty$ error | $L^1$ error | Order | $L^\infty$ error | Order |
| 10 | 1.71E−13 | 2.68E−13 | 5.43E−3 | | 7.81E−3 | |
| 20 | 1.29E−14 | 2.42E−14 | 2.79E−4 | 4.28 | 4.75E−4 | 4.04 |
| 40 | 5.78E−15 | 1.22E−14 | 8.76E−6 | 5.00 | 1.65E−5 | 4.84 |
| 80 | 8.02E−15 | 2.64E−14 | 2.73E−7 | 5.00 | 5.22E−7 | 4.98 |
| 160 | 1.45E−14 | 7.91E−14 | 8.43E−9 | 5.02 | 1.54E−8 | 5.08 |
| 320 | 1.54E−13 | 4.37E−13 | 2.52E−10 | 5.06 | 4.32E−10 | 5.16 |

| Cells | WENO5 | | | |
|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order |
| 10 | 7.06E−3 | | 1.01E−2 | |
| 20 | 3.62E−4 | 4.28 | 5.99E−4 | 4.08 |
| 40 | 1.14E−5 | 4.99 | 2.11E−5 | 4.82 |
| 80 | 3.55E−7 | 5.00 | 6.73E−7 | 4.98 |
| 160 | 1.09E−8 | 5.02 | 1.99E−8 | 5.08 |
| 320 | 3.29E−10 | 5.06 | 5.60E−10 | 5.15 |

| Cells | TWENO5-RKDG-T, $\alpha=\pi$ | | RKDG-T, $\alpha=\pi$ | |
|---|---|---|---|---|
| | $L^1$ error | $L^\infty$ error | $L^1$ error | $L^\infty$ error |
| 10 | 3.45E−12 | 1.80E−11 | 3.34E−12 | 1.68E−11 |
| 20 | 4.20E−14 | 1.41E−13 | 1.31E−14 | 5.95E−14 |
| 40 | 1.33E−13 | 1.09E−12 | 7.69E−15 | 2.81E−14 |
| 80 | 5.52E−13 | 8.44E−12 | 1.08E−14 | 4.10E−14 |
| 160 | 1.15E−13 | 2.59E−12 | 2.75E−14 | 1.75E−13 |
| 320 | 1.21E−12 | 5.61E−11 | 1.71E−13 | 7.19E−13 |

| Cells | TWENO5-RKDG-T, $\alpha=1$ | | | | RKDG-T, $\alpha=1$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order | $L^1$ error | Order | $L^\infty$ error | Order |
| 10 | 5.88E−4 | | 1.57E−3 | | 1.14E−4 | | 7.14E−4 | |
| 20 | 2.79E−5 | 4.39 | 9.11E−5 | 4.11 | 1.44E−5 | 3.00 | 9.20E−5 | 2.96 |
| 40 | 2.13E−6 | 3.71 | 1.15E−5 | 2.98 | 1.79E−6 | 3.01 | 1.15E−5 | 2.99 |
| 80 | 2.34E−7 | 3.18 | 1.45E−6 | 3.00 | 2.23E−7 | 3.00 | 1.45E−6 | 3.00 |
| 160 | 2.84E−8 | 3.05 | 1.81E−7 | 3.00 | 2.79E−8 | 3.00 | 1.81E−7 | 3.00 |
| 320 | 3.52E−9 | 3.01 | 2.26E−8 | 3.00 | 3.49E−9 | 3.00 | 2.26E−8 | 3.00 |

| Cells | WENO5-RKDG-A | | | | RKDG-A | | | |
|---|---|---|---|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order | $L^1$ error | Order | $L^\infty$ error | Order |
| 10 | 5.99E−4 | | 1.60E−3 | | 1.27E−4 | | 7.94E−4 | |
| 20 | 3.09E−5 | 4.27 | 1.01E−4 | 3.98 | 1.60E−5 | 3.00 | 1.02E−4 | 2.96 |
| 40 | 2.37E−6 | 3.71 | 1.28E−5 | 2.98 | 1.99E−6 | 3.01 | 1.28E−5 | 3.00 |
| 80 | 2.60E−7 | 3.18 | 1.61E−6 | 3.00 | 2.48E−7 | 3.00 | 1.61E−6 | 3.00 |
| 160 | 3.15E−8 | 3.05 | 2.01E−7 | 3.00 | 3.10E−8 | 3.00 | 2.01E−7 | 3.00 |
| 320 | 3.90E−9 | 3.01 | 2.52E−8 | 3.00 | 3.88E−9 | 3.00 | 2.52E−8 | 3.00 |

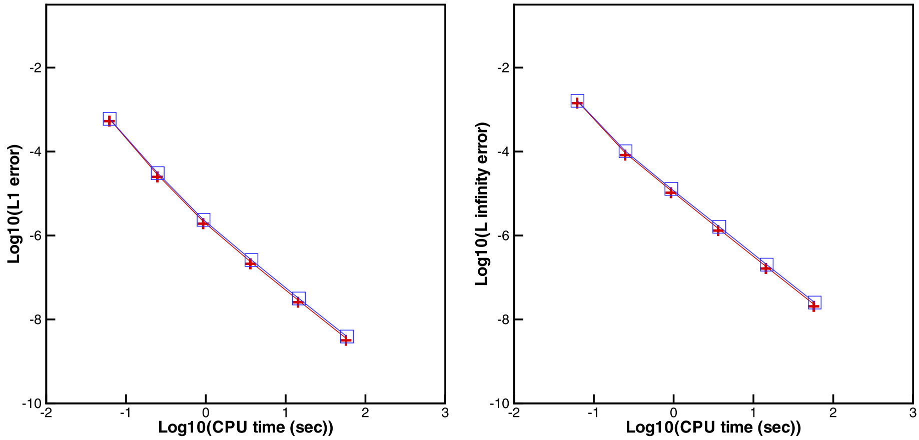

**Fig. 5** Computing time and error of 1D Euler equations. $\rho(x,0)=1+0.2\sin(\pi x)$, $u(x,0)=1$, $p(x,0)=1$. Periodic boundary conditions. TVB constant $M=0.01$. $t=1$. *Plus signs* and a *solid line* denote the results of TWENO5-RKDG-T, $\alpha=1$ scheme; *squares* and a *solid line* denote the results of WENO5-RKDG-A scheme

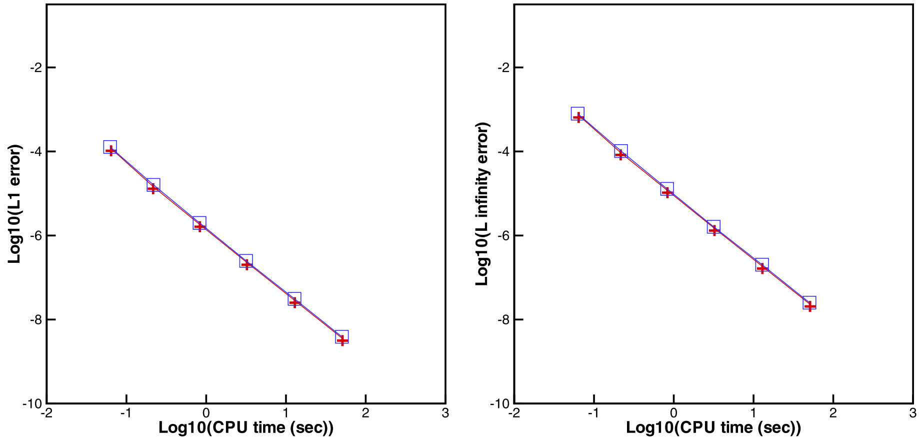

**Fig. 6** Computing time and error of 1D Euler equations. $\rho(x,0)=1+0.2\sin(\pi x)$, $u(x,0)=1$, $p(x,0)=1$. Periodic boundary conditions. $t=1$. *Plus signs* and a *solid line* denote the results of RKDG-T, $\alpha=1$ scheme; *squares* and a *solid line* denote the results of RKDG-A scheme

*Example 4.3* We solve the two dimensional nonlinear scalar Burger's equation:

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

with the initial condition $u(x,y,0)=0.5+\sin(\pi(x+y)/2)$, $(x,y)\in[0,4]\times[0,4]$ with periodic boundary conditions. Errors and numerical orders of accuracy achieved by the finite volume and DG schemes with TVB constant $M=0.01$ are shown in Table 3. And the numerical error against CPU time graphs are shown in Figs. 7, 8 and 9. As in Example 4.1, we also observe that all schemes achieve their designed orders of accuracy, as expected.

**Table 3** $u_t + (u^2/2)_x + (u^2/2)_y = 0$. $u(x,0) = 0.5 + \sin(\pi(x+y)/2)$. Periodic boundary conditions. TVB constant $M = 0.01$. $t = 0.5/\pi$. $L^1$ and $L^\infty$ errors and numerical orders of accuracy

| Cells | TWENO5, $\alpha = \pi/2, \beta = \pi/2$ | | | |
|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order |
| 10×10 | 3.90E−03 | | 1.13E−02 | |
| 20×20 | 4.72E−04 | 3.05 | 2.41E−03 | 2.23 |
| 40×40 | 6.50E−05 | 2.86 | 5.13E−04 | 2.23 |
| 80×80 | 3.57E−06 | 4.19 | 3.21E−05 | 4.00 |
| 160×160 | 1.50E−07 | 4.57 | 1.22E−06 | 4.71 |

| Cells | WENO5 | | | |
|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order |
| 10×10 | 6.26E−3 | | 1.90E−2 | |
| 20×20 | 8.76E−4 | 2.83 | 5.77E−3 | 1.72 |
| 40×40 | 8.08E−5 | 3.43 | 6.73E−4 | 3.10 |
| 80×80 | 4.01E−6 | 4.33 | 3.90E−5 | 4.11 |
| 160×160 | 1.63E−7 | 4.62 | 1.41E−6 | 4.78 |

| Cells | TWENO5-RKDG-T, $\alpha = \pi/2, \beta = \pi/2$ | | | | RKDG-T, $\alpha = \pi/2, \beta = \pi/2$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order | $L^1$ error | Order | $L^\infty$ error | Order |
| 10×10 | 1.19E−2 | | 1.90E−1 | | 5.10E−3 | | 1.77E−1 | |
| 20×20 | 1.45E−3 | 3.04 | 3.98E−2 | 2.26 | 8.25E−4 | 2.63 | 4.01E−2 | 2.14 |
| 40×40 | 1.43E−4 | 3.35 | 5.83E−3 | 2.77 | 1.12E−4 | 2.88 | 5.83E−3 | 2.78 |
| 80×80 | 1.69E−5 | 3.07 | 9.72E−4 | 2.58 | 1.44E−5 | 2.96 | 9.72E−4 | 2.58 |
| 160×160 | 2.00E−6 | 3.08 | 1.33E−4 | 2.86 | 1.83E−6 | 2.98 | 1.33E−4 | 2.86 |

| Cells | WENO5-RKDG-A | | | | RKDG-A | | | |
|---|---|---|---|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order | $L^1$ error | Order | $L^\infty$ error | Order |
| 10×10 | 1.17E−2 | | 1.84E−1 | | 5.19E−3 | | 1.82E−1 | |
| 20×20 | 1.05E−3 | 3.47 | 4.14E−2 | 2.15 | 8.29E−4 | 2.65 | 4.15E−2 | 2.14 |
| 40×40 | 1.22E−4 | 3.11 | 6.03E−3 | 2.78 | 1.12E−4 | 2.88 | 6.03E−3 | 2.78 |
| 80×80 | 1.52E−5 | 3.01 | 1.00E−3 | 2.59 | 1.44E−5 | 2.96 | 1.00E−3 | 2.59 |
| 160×160 | 1.91E−6 | 2.99 | 1.37E−4 | 2.86 | 1.83E−6 | 2.98 | 1.37E−4 | 2.86 |

Moreover the numerical errors and computing times by the RKDG based on trigonometric polynomial spaces and algebraical polynomial spaces are very similar at the same mesh level.

*Example 4.4* We solve the two dimensional Euler equations:

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ E\end{pmatrix} + \frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2 + p\\ \rho u v\\ u(E+p)\end{pmatrix} + \frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho u v\\ \rho v^2 + p\\ v(E+p)\end{pmatrix} = 0, \tag{4.4}$$

in which $\rho$ is density; $u$ and $v$ are the velocities in the $x$ and $y$-directions, respectively; $E$ is total energy; and $p$ is pressure. The initial conditions are: $\rho(x,y,0) = 1 + 0.2(\sin(\pi x) +$

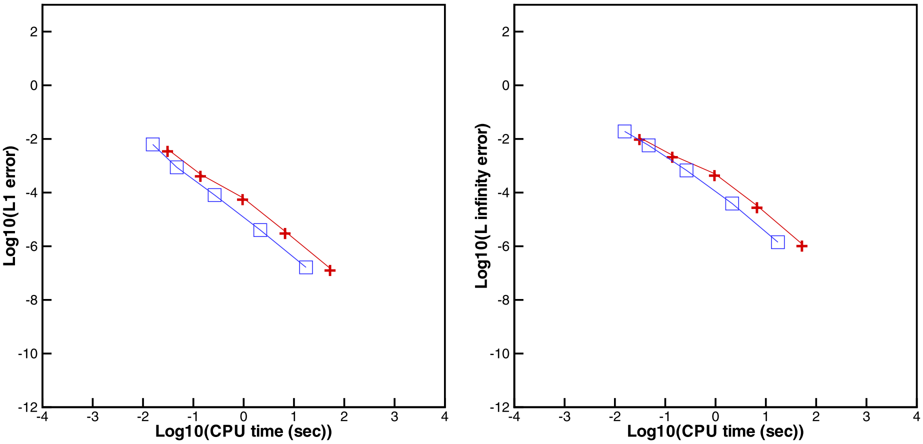

**Fig. 7** Computing time and error of $u_t + (u^2/2)_x + (u^2/2)_y = 0$. $u(x,0) = 0.5 + \sin(\pi(x+y)/2)$. Periodic boundary conditions. $t = 0.5/\pi$. *Plus signs* and a *solid line* denote the results of TWENO5, $\alpha = \pi/2$, $\beta = \pi/2$ scheme; *squares* and a *solid line* denote the results of WENO5 scheme

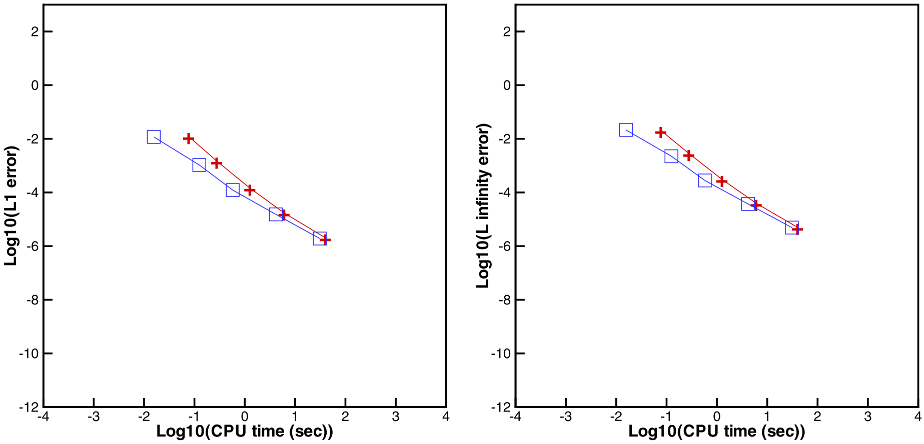

**Fig. 8** Computing time and error of $u_t + (u^2/2)_x + (u^2/2)_y = 0$. $u(x,0) = 0.5 + \sin(\pi(x+y)/2)$. Periodic boundary conditions. TVB constant $M = 0.01$. $t = 0.5/\pi$. *Plus signs* and a *solid line* denote the results of TWENO5-RKDG-T, $\alpha = \pi/2, \beta = \pi/2$ scheme; *squares* and a *solid line* denote the results of WENO5-RKDG-A scheme

$\sin(\pi y))$, $u(x,y,0) = 0.7$, $v(x,y,0) = 0.3$, $p(x,y,0) = 1$, $(x,y) \in [0,2] \times [0,2]$ with periodic boundary conditions in both directions. We compute the density solution up to $t = 1$. Errors and numerical orders of accuracy achieved by the finite volume and DG schemes with TVB constant $M = 0.01$ are shown in Table 4. And the numerical error against CPU time graphs are shown in Figs. 10, 11 and 12. As in Example 4.2, the numerical results show that the right trigonometric space functions with optimal parameter $\alpha$ may lead to modulo round off errors, with non-optimal parameter $\alpha$ leading to proper order of accuracy. And at the same time, the computing costs are bigger than those proposed by the WENO5, WENO5-RKDG-A and RKDG-A schemes.

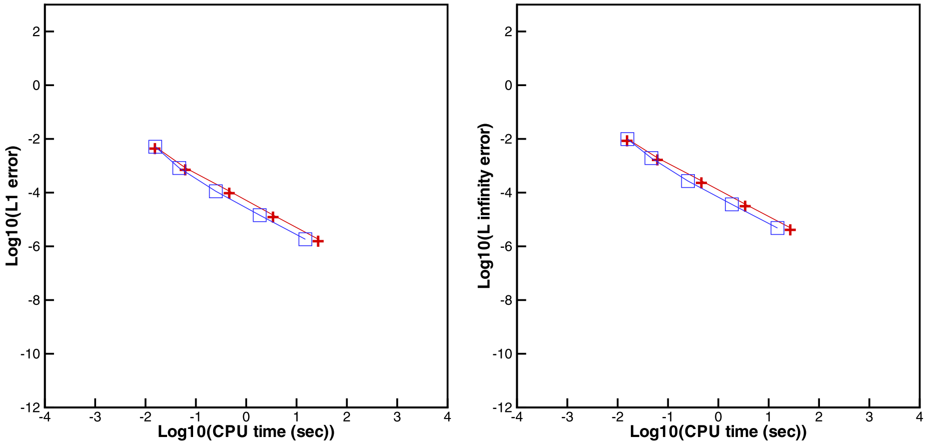

**Fig. 9** Computing time and error of $u_t + (u^2/2)_x + (u^2/2)_y = 0$. $u(x,0) = 0.5 + \sin(\pi(x+y)/2)$. Periodic boundary conditions. $t = 0.5/\pi$. *Plus signs* and a *solid line* denote the results of RKDG-T, $\alpha = \pi/2$, $\beta = \pi/2$ scheme; *squares* and a *solid line* denote the results of RKDG-A scheme

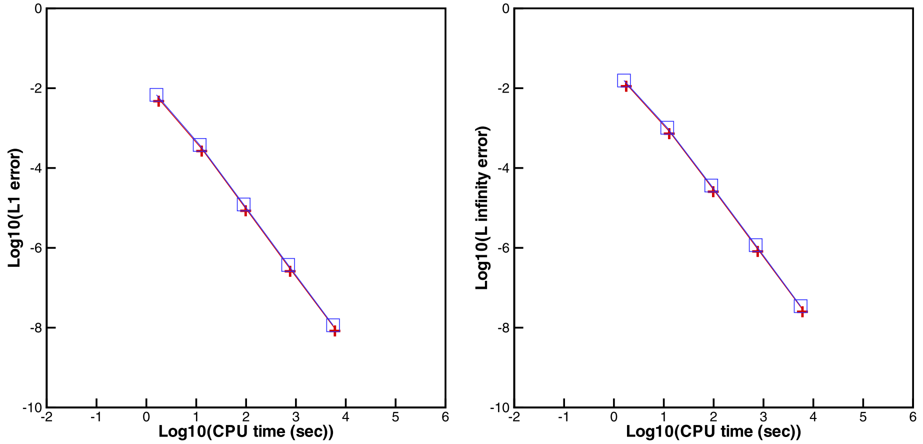

**Fig. 10** Computing time and error of 2D Euler equations. $\rho(x,y,0) = 1 + 0.2(\sin(\pi x) + \sin(\pi y))$, $u(x,y,0) = 0.7$, $v(x,y,0) = 0.3$, $p(x,y,0) = 1$. Periodic boundary conditions. $t = 1$. *Plus signs* and a *solid line* denote the results of TWENO5, $\alpha = 1, \beta = 1$ scheme; *squares* and a *solid line* denote the results of WENO5 scheme

*Example 4.5* We solve the one dimensional Euler equations (4.2) with a moving Mach = 3 shock interacting with sine waves in density:

$$(\rho, u, p)^T = \begin{cases} (3.857143, 2.629369, 10.333333)^T, & x < -4, \\ (1 + 0.2\sin(5x), 0, 1)^T, & x \geq -4. \end{cases} \tag{4.5}$$

The reference solution is a converged solution computed by the WENO5 scheme with 2000 cells. The computed densities $\rho$, obtained by the TWENO5 (plusses), WENO5 (squares), TWENO5-RKDG-T (plusses) and WENO5-RKDG-A (squares) schemes, are plotted in

**Table 4** 2D Euler equations. $\rho(x,y,0)=1+0.2(\sin(\pi x)+\sin(\pi y))$, $u(x,y,0)=0.7$, $v(x,y,0)=0.3$, $p(x,y,0)=1$. Periodic boundary conditions. TVB constant $M=0.01$. $t=1$. $L^1$ and $L^\infty$ errors and numerical orders of accuracy

| Cells | TWENO5, $\alpha=\pi,\beta=\pi$ | | TWENO5, $\alpha=1,\beta=1$ | | | |
|---|---|---|---|---|---|---|
| | $L^1$ error | $L^\infty$ error | $L^1$ error | Order | $L^\infty$ error | Order |
| $10\times10$ | 3.24E−15 | 9.32E−15 | 5.26E−3 | | 1.24E−2 | |
| $20\times20$ | 3.57E−15 | 1.42E−14 | 2.96E−4 | 4.15 | 8.09E−4 | 3.95 |
| $40\times40$ | 4.18E−15 | 1.95E−14 | 9.46E−6 | 4.97 | 2.84E−5 | 4.83 |
| $80\times80$ | 1.37E−14 | 1.15E−13 | 2.88E−7 | 5.03 | 9.07E−7 | 4.97 |
| $160\times160$ | 3.11E−14 | 1.58E−13 | 9.25E−9 | 4.96 | 2.79E−8 | 5.02 |

| Cells | WENO5 | | | |
|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order |
| $10\times10$ | 6.81E−3 | | 1.55E−2 | |
| $20\times20$ | 3.78E−4 | 4.17 | 1.02E−3 | 3.93 |
| $40\times40$ | 1.22E−5 | 4.95 | 3.64E−5 | 4.81 |
| $80\times80$ | 3.74E−7 | 5.03 | 1.16E−6 | 4.96 |
| $160\times160$ | 1.14E−8 | 5.02 | 3.46E−8 | 5.08 |

| Cells | TWENO5-RKDG-T, $\alpha=\pi,\beta=\pi$ | | RKDG-T, $\alpha=\pi,\beta=\pi$ | |
|---|---|---|---|---|
| | $L^1$ error | $L^\infty$ error | $L^1$ error | $L^\infty$ error |
| $10\times10$ | 7.93E−12 | 4.63E−11 | 7.89E−12 | 4.58E−11 |
| $20\times20$ | 3.89E−12 | 8.82E−12 | 2.47E−14 | 1.62E−13 |
| $40\times40$ | 1.29E−11 | 4.41E−11 | 7.75E−15 | 4.60E−14 |
| $80\times80$ | 6.51E−12 | 3.72E−11 | 4.45E−14 | 2.12E−13 |
| $160\times160$ | 3.74E−10 | 3.23E−9 | 1.69E−13 | 7.09E−13 |

| Cells | TWENO5-RKDG-T, $\alpha=1,\beta=1$ | | | | RKDG-T, $\alpha=1,\beta=1$ | | | |
|---|---|---|---|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order | $L^1$ error | Order | $L^\infty$ error | Order |
| $10\times10$ | 5.67E−4 | | 2.10E−3 | | 3.33E−4 | | 2.02E−3 | |
| $20\times20$ | 9.19E−5 | 2.63 | 5.04E−4 | 2.06 | 6.28E−5 | 2.41 | 3.84E−4 | 2.40 |
| $40\times40$ | 1.22E−5 | 2.91 | 7.58E−5 | 2.73 | 9.79E−6 | 2.68 | 5.91E−5 | 2.70 |
| $80\times80$ | 1.49E−6 | 3.04 | 8.59E−6 | 3.14 | 1.32E−6 | 2.88 | 7.95E−6 | 2.89 |
| $160\times160$ | 1.79E−7 | 3.06 | 1.01E−6 | 3.08 | 1.69E−7 | 2.97 | 1.01E−6 | 2.97 |

| Cells | WENO5-RKDG-A | | | | RKDG-A | | | |
|---|---|---|---|---|---|---|---|---|
| | $L^1$ error | Order | $L^\infty$ error | Order | $L^1$ error | Order | $L^\infty$ error | Order |
| $10\times10$ | 3.60E−3 | | 1.05E−2 | | 3.71E−4 | | 2.25E−3 | |
| $20\times20$ | 1.40E−4 | 4.68 | 6.37E−4 | 4.05 | 6.99E−5 | 2.41 | 4.27E−4 | 2.40 |
| $40\times40$ | 1.28E−5 | 3.45 | 6.77E−5 | 3.23 | 1.09E−5 | 2.68 | 6.57E−5 | 2.70 |
| $80\times80$ | 1.64E−6 | 2.97 | 8.85E−6 | 2.94 | 1.47E−6 | 2.88 | 8.84E−6 | 2.89 |
| $160\times160$ | 1.98E−7 | 3.05 | 1.12E−6 | 2.97 | 1.88E−7 | 2.97 | 1.12E−6 | 2.97 |

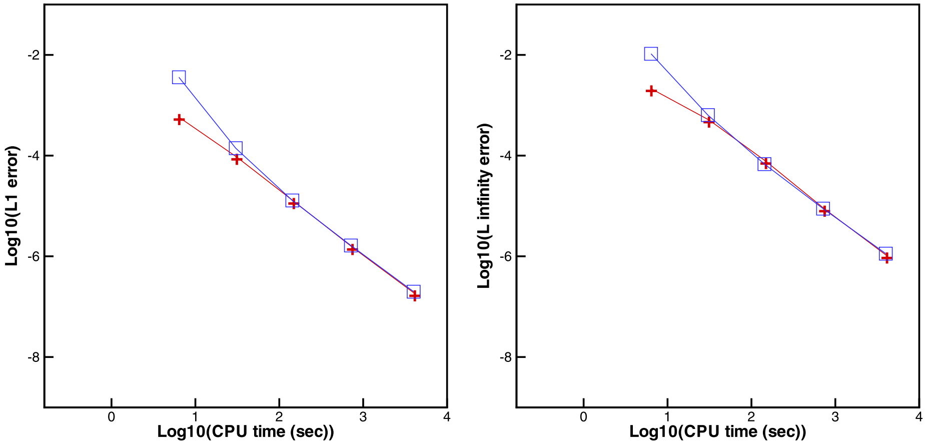

**Fig. 11** Computing time and error of 2D Euler equations. $\rho(x, y, 0) = 1 + 0.2(\sin(\pi x) + \sin(\pi y))$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$, $p(x, y, 0) = 1$. Periodic boundary conditions. TVB constant $M = 0.01$. $t = 1$. *Plus signs* and a *solid line* denote the results of TWENO5-RKDG-T, $\alpha = 1, \beta = 1$ scheme; *squares* and a *solid line* denote the results of WENO5-RKDG-A scheme

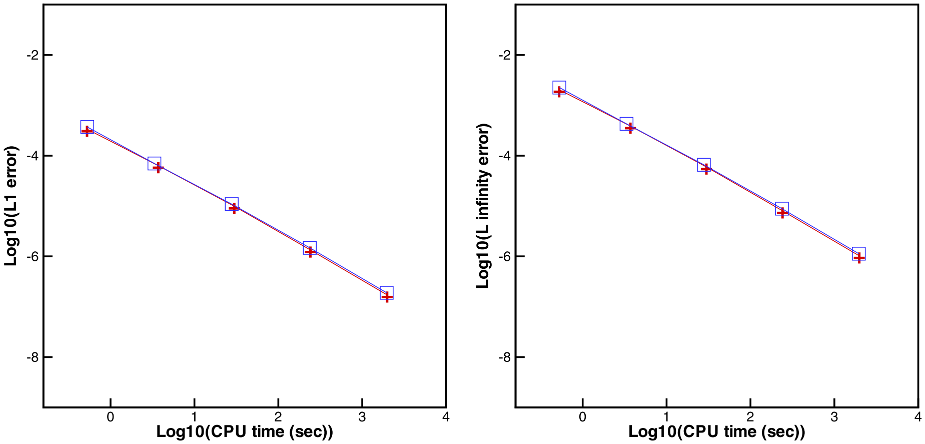

**Fig. 12** Computing time and error of 2D Euler equations. $\rho(x, y, 0) = 1 + 0.2(\sin(\pi x) + \sin(\pi y))$, $u(x, y, 0) = 0.7$, $v(x, y, 0) = 0.3$, $p(x, y, 0) = 1$. Periodic boundary conditions. $t = 1$. *Plus signs* and a *solid line* denote the results of RKDG-T, $\alpha = 1, \beta = 1$ scheme; *squares* and a *solid line* denote the results of RKDG-A scheme

Figs. 13 and 14 at $t = 1.8$ along with the reference solution (solid line). In all the numerical schemes for this example, we use 200 cells and TVB limiter with $M = 1.0$ as trouble cell indicator. We observe that the results of TWENO5 are similar to those of WENO5 and the results of the TWENO5-RKDG-T scheme are better than those of WENO5-RKDG-A with the same TVB constant in the smooth region while maintaining the essentially non-oscillatory property nearby the vicinity of the discontinuity.

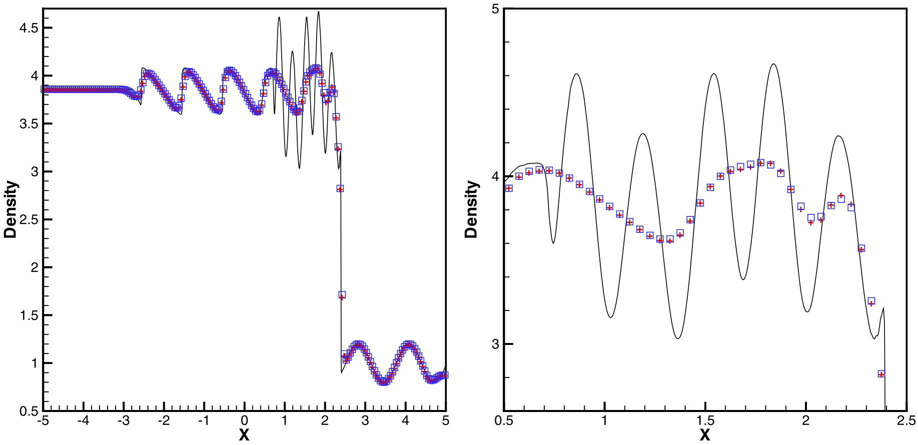

**Fig. 13** The shock density wave interaction problem. Computed densities at $t = 1.8$ with 200 cells. TWENO5 (*plusses*) and WENO5 (*squares*) against the reference solution (*solid line*) (*left*) and zoom in (*right*)

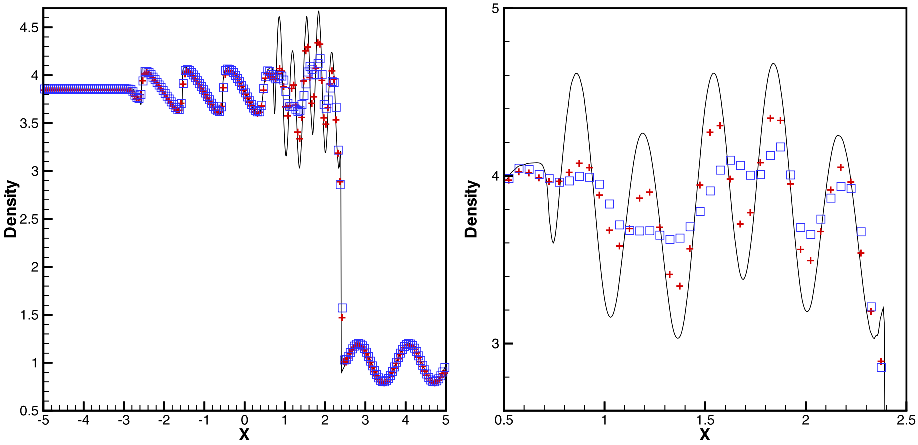

**Fig. 14** The shock density wave interaction problem. Computed densities at $t = 1.8$ with 200 cells. TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*) against the reference solution (*solid line*) (*left*) and zoom in (*right*), $M = 1.0$

*Example 4.6* Shock entropy wave interactions [18]. This is a famous test case for high order accuracy and shock capturing schemes and the governing equation is the one dimensional Euler equations (4.2). The initial conditions, which represent a Mach 3 shock, are:

$$(\rho, u, p)^T = \begin{cases} (3.85714, 2.629369, 10.33333)^T, & 0 \le x < 0.5, \\ (e^{-\varepsilon \sin(\kappa x)}, 0, 1)^T, & 0.5 < x \le 5. \end{cases} \tag{4.6}$$

To further test the resolutions of the TWENO5, WENO5, TWENO5-RKDG-T and WENO5-RKDG-A schemes in the presence of shock waves, the interaction of shock and entropy waves is employed. In this test case, small amplitude, low-frequency entropy waves are set

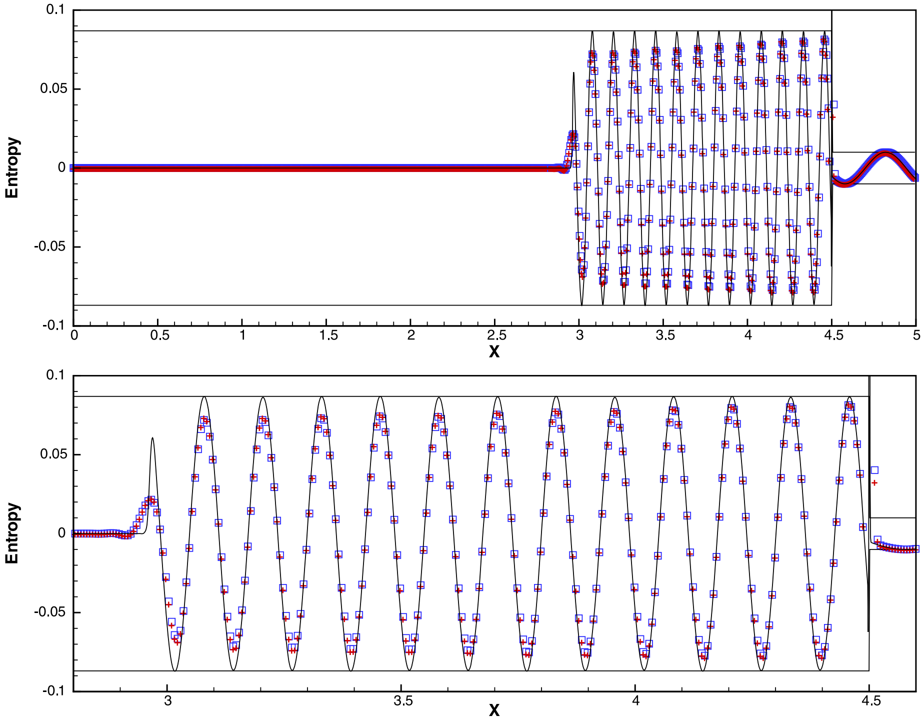

**Fig. 15** The shock entropy wave interactions problem, $\kappa = 13$. Computed entropies with 800 cells. TWENO5 (*plusses*) and WENO5 (*squares*) against the reference solution (*solid line*) (*top*) and zoom in (*bottom*)

in the upstream of a shock. After the interaction with the shock, both the amplitude and frequency of the waves increases. We use a linear analysis to estimate the wave amplitude after the shock, which depends only on the shock strength and the amplitude before the shock. We evaluate the performance of the proposed schemes by checking against the linear analysis. We take $\varepsilon = 0.01$ and $\kappa = 13, 26, 39$, and 52 to be the amplitude and wave-number of the entropy wave before the shock. Accordingly, the amplitude after the shock is determined to be 0.08690716 by the linear analysis. The reference entropy solution is a convergent solution computed by the WENO5 scheme with 5000 cells (for $\kappa = 13$ and 26) and 8000 cells (for $\kappa = 39$ and 52). The computed solutions, obtained by the TWENO5 (plusses), WENO5 (squares), TWENO5-RKDG-T (plusses) and WENO5-RKDG-A (squares) schemes with the same TVB constant $M = 10.0$, are plotted in Figs. 15–22 for each $\kappa$ along with the reference solution (solid line). We observe that the results of TWENO5 are similar to those of WENO5. Although the trigonometric polynomial space does not include the exponential polynomial space specified in this problem, we observe that the results of the TWENO5-RKDG-T schemes are better than those of WENO5-RKDG-A. In particular: (i) there is less amplitude decrease to the right of the position $x = 2.95$, and (ii) the trigonometric polynomial basis used in the TWENO5-RKDG-T scheme can retain the high frequency nature of this case better than the algebraic polynomial basis used in WENO5-RKDG-A.

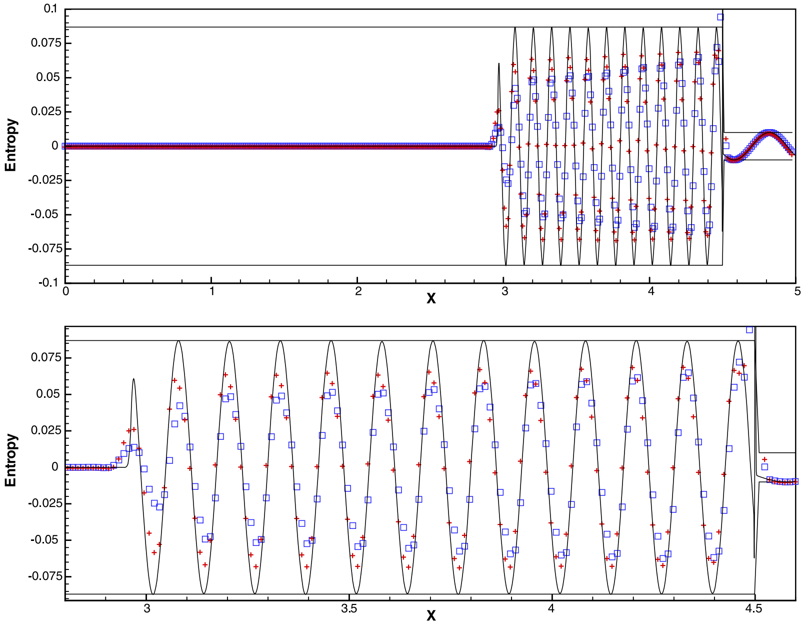

**Fig. 16** The shock entropy wave interactions problem, $\kappa = 13$. Computed entropies with 400 cells. TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*) against the reference solution (*solid line*) (*top*) and zoom in (*bottom*), $M = 10.0$

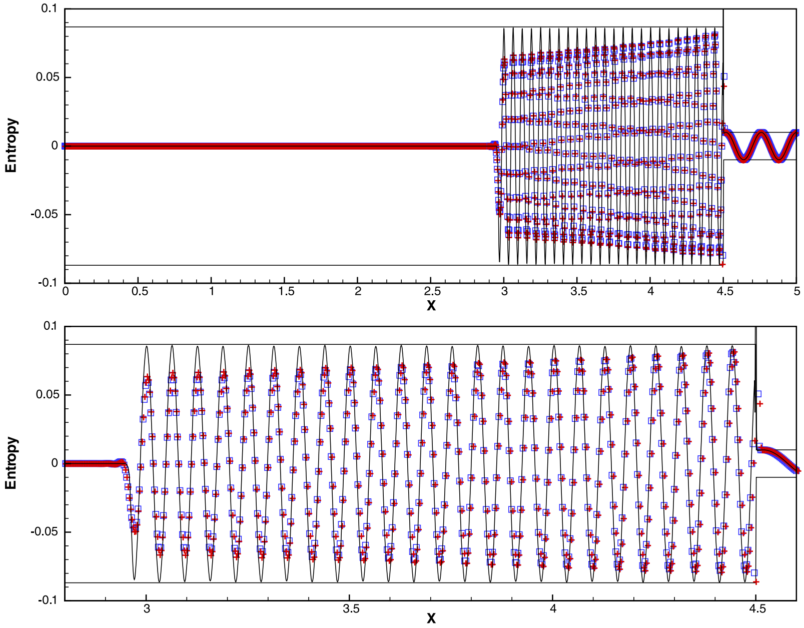

**Fig. 17** The shock entropy wave interactions problem, $\kappa = 26$. Computed entropies with 1800 cells. TWENO5 (*plusses*) and WENO5 (*squares*) against the reference solution (*solid line*) (*top*) and zoom in (*bottom*)

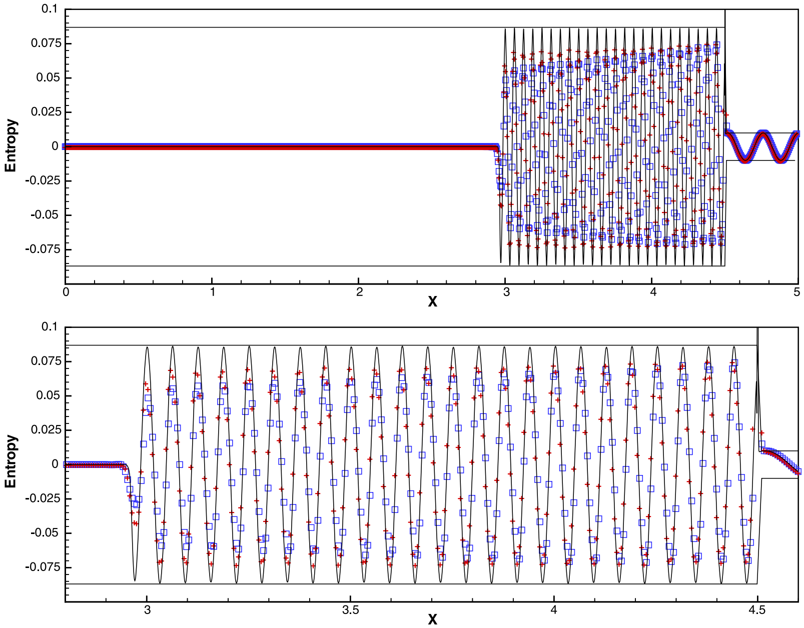

**Fig. 18** The shock entropy wave interactions problem, $\kappa = 26$. Computed entropies with 900 cells. TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*) against the reference solution (*solid line*) (*top*) and zoom in (*bottom*), $M = 10.0$

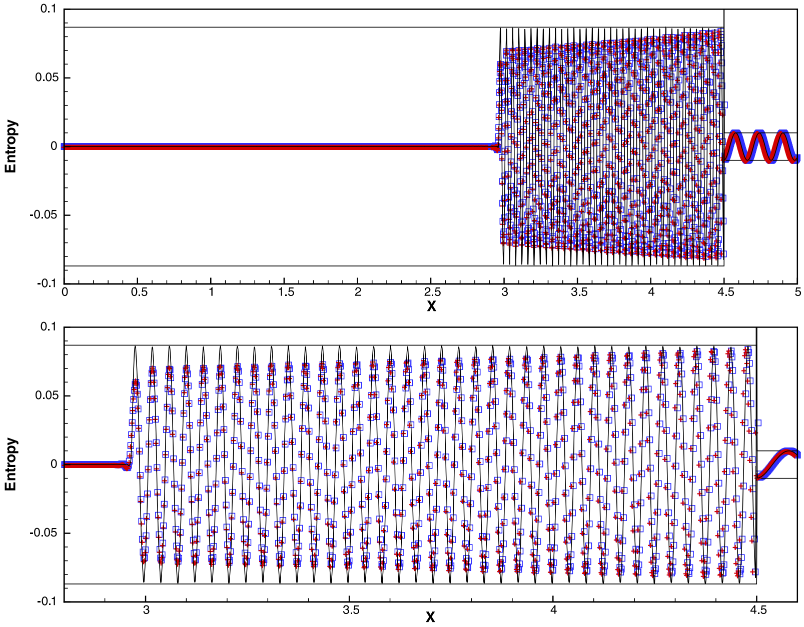

**Fig. 19** The shock entropy wave interactions problem, $\kappa = 39$. Computed entropies with 2800 cells. TWENO5 (*plusses*) and WENO5 (*squares*) against the reference solution (*solid line*) (*top*) and zoom in (*bottom*)

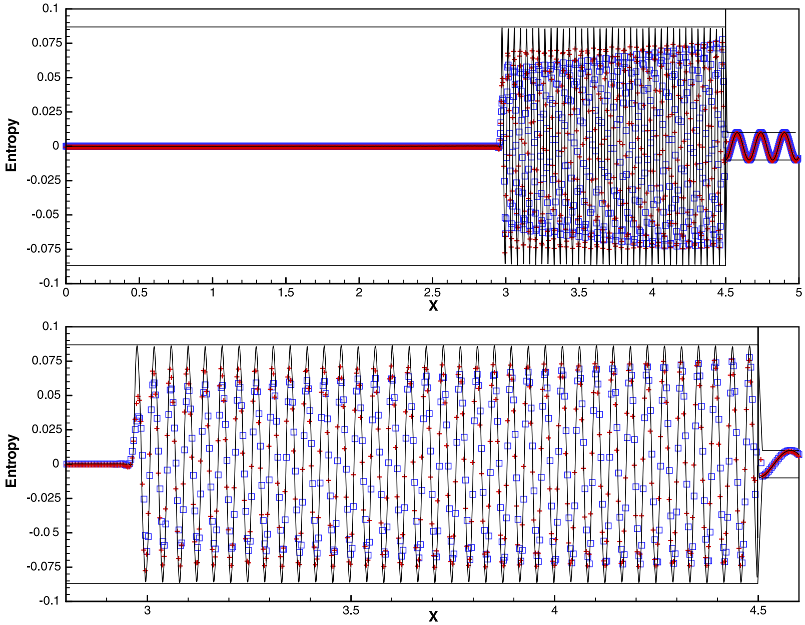

**Fig. 20** The shock entropy wave interactions problem, $\kappa = 39$. Computed entropies with 1400 cells. TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*) against the reference solution (*solid line*) (*top*) and zoom in (*bottom*), $M = 10.0$

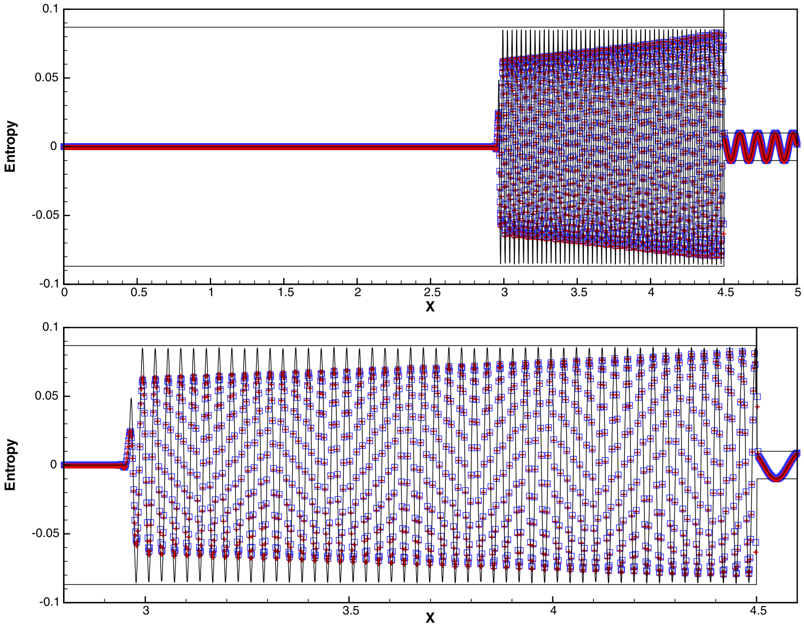

**Fig. 21** The shock entropy wave interactions problem, $\kappa = 52$. Computed entropies with 3600 cells. TWENO5 (*plusses*) and WENO5 (*squares*) against the reference solution (*solid line*) (*top*) and zoom in (*bottom*)

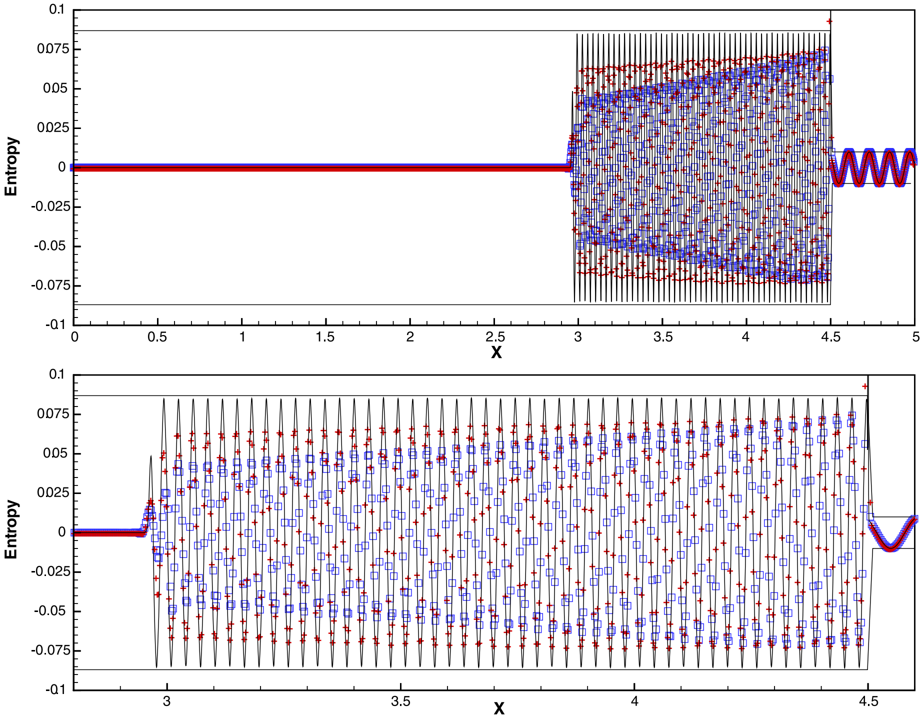

**Fig. 22** The shock entropy wave interactions problem, $\kappa = 52$. Computed entropies with 1800 cells. TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*) against the reference solution (*solid line*) (*top*) and zoom in (*bottom*), $M = 10.0$

*Example 4.7* We solve the one dimensional Euler equations (4.2) with a moving sine wave in density:

$$(\rho, u, p)^T = \begin{cases} (1,1,1)^T, & -1 \le x < 0, \\ (1 + 0.2|\sin(\kappa\pi x)|, 1, 1)^T, & 0 \le x < 1, \\ (1,1,1)^T, & 1 \le x \le 2. \end{cases} \tag{4.7}$$

The computed densities $\rho$, obtained by the TWENO5 (plusses), WENO5 (squares), TWENO5-RKDG-T (plusses) and WENO5-RKDG-A (squares) schemes with the same TVB constant $M = 1.0$, are plotted in Figs. 23 and 24 at $t = 6$ and for each $\kappa$ along with the exact solution (solid line). We observe that the results of TWENO5 are better than those of WENO5. The exact solution space of this problem is not compatible to the DG spatial trigonometric polynomial space used in the TWENO5-RKDG-T scheme. Despite this, we still observe that the results of TWENO5-RKDG-T are better than those of WENO5-RKDG-A. This is especially true at the local smooth extremum, where the amplitude decreases much less.

*Example 4.8* We solve the one dimensional Euler equations (4.2) with a moving exponential wave in density:

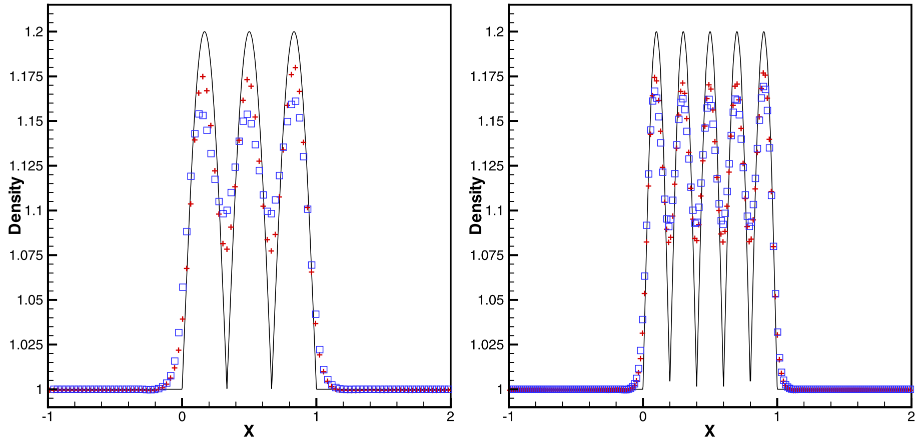

**Fig. 23** The moving sine wave problem. Computed densities. *Left*: $\kappa = 3$ with 100 cells; *right*: $\kappa = 5$ with 200 cells. $t = 6$. TWENO5 (*plusses*) and WENO5 (*squares*) against the exact solution (*solid line*)

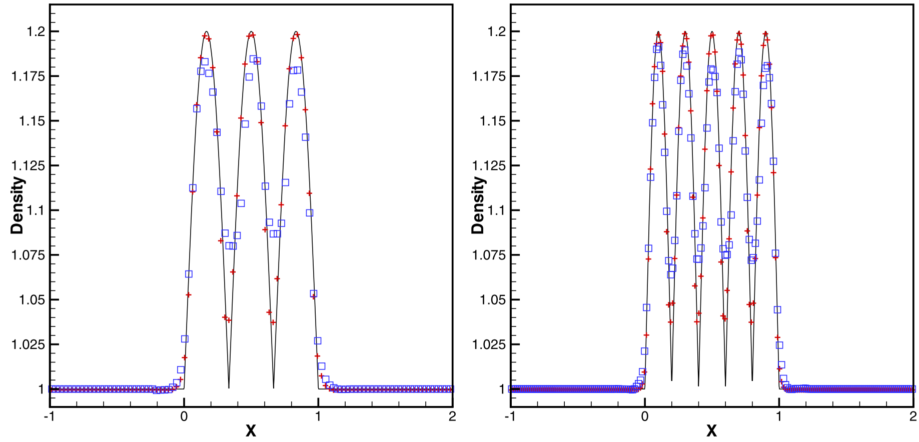

**Fig. 24** The moving sine wave problem. Computed densities. *Left*: $\kappa = 3$ with 100 cells; *right*: $\kappa = 5$ with 200 cells. $t = 6$. TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*) against the exact solution (*solid line*), $M = 1.0$

$$(\rho, u, p)^T = \begin{cases} (1, 1, 1)^T, & -1 \le x < 0, \\ (e^{|\sin(\kappa\pi x)|}, 1, 1)^T, & 0 \le x < 1, \\ (1, 1, 1)^T, & 1 \le x \le 2, \end{cases} \tag{4.8}$$

with boundary conditions and $\kappa = 3$ and 5. The computed densities $\rho$, obtained by the TWENO5 (plusses), WENO5 (squares), TWENO5-RKDG-T (plusses) and WENO5-RKDG-A (squares) schemes with the same TVB constant $M = 1.0$, are plotted in Figs. 25 and 26 at $t = 6$ and for each $\kappa$ along with the exact solution (solid line). We observe that the results of TWENO5 are better than those of WENO5. The exact solution of this problem consists of an exponential basis in the density field and is not compatible with the DG

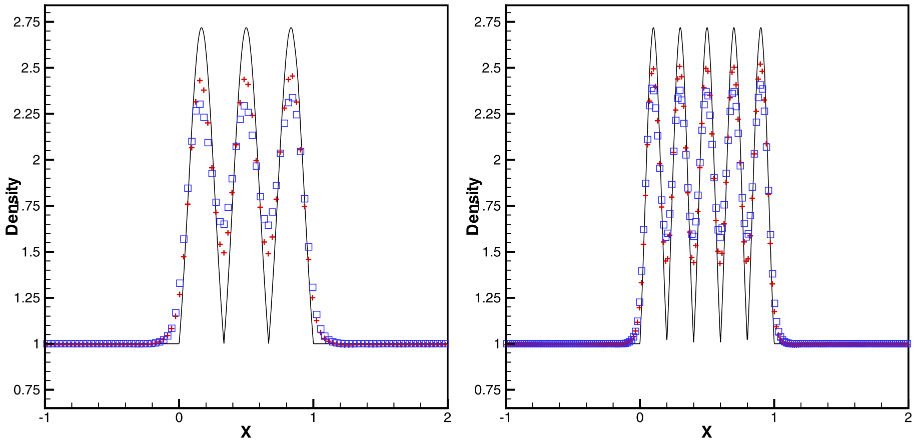

**Fig. 25** The moving exponential wave problem. Computed densities. *Left*: $\kappa = 3$ with 100 cells; *right*: $\kappa = 5$ with 200 cells. $t = 6$. TWENO5 (*plusses*) and WENO5 (*squares*) against the exact solution (*solid line*)

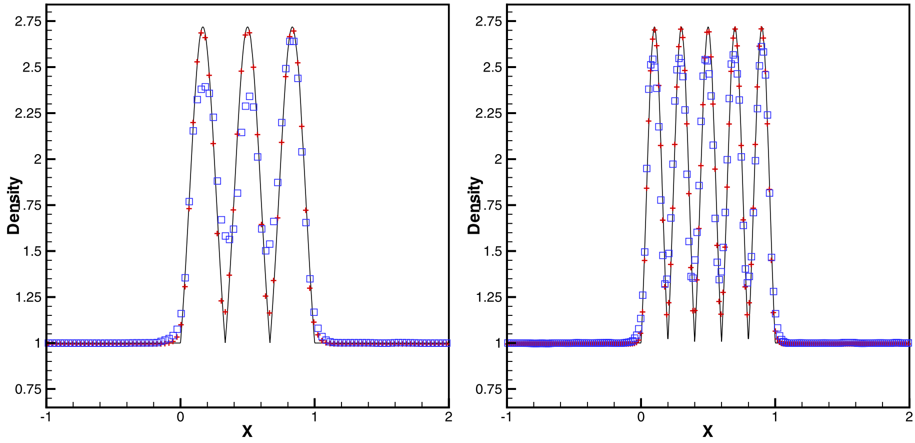

**Fig. 26** The moving exponential wave problem. Computed densities. *Left*: $\kappa = 3$ with 100 cells; *right*: $\kappa = 5$ with 200 cells. $t = 6$. TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*) against the exact solution (*solid line*), $M = 1.0$

spatial orthogonal trigonometric polynomial space used in the TWENO5-RKDG-T scheme. Despite this, we still observe that the results of TWENO5-RKDG-T are better than those of WENO5-RKDG-A, which uses algebraic polynomials.

*Example 4.9* We solve the one dimensional Euler equations (4.2) with two different initial data and periodic boundary conditions.

(1) Divided the interval [0, 1] into 100 cells of equal size. The initial condition for the linear semi-circle wave problem is:

$$(\rho, u, p)^T = \begin{cases} (1 + \sqrt{1 - (\frac{i-50}{15})^2}, 1, 1)^T, & 35 \le i \le 65, \\ (1, 1, 1)^T, & \text{otherwise.} \end{cases} \tag{4.9}$$

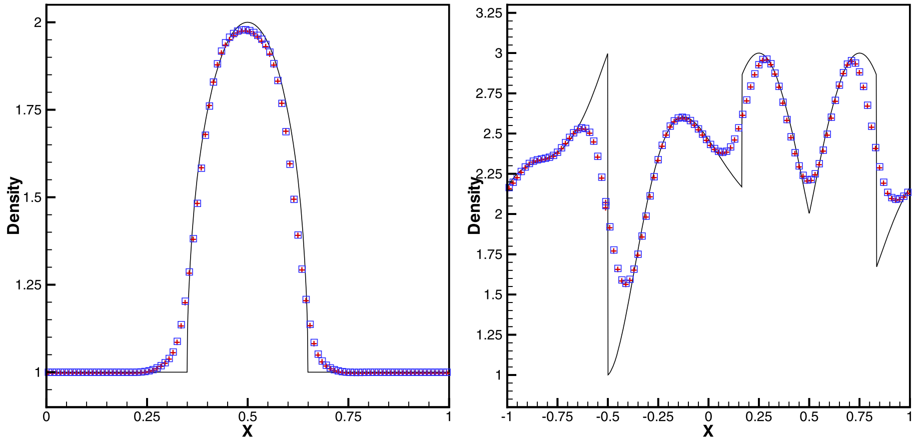

**Fig. 27** *Left*: (1) the linear semi-circle wave problem; *right*: (2) the highly discontinuous initial data problem. $t = 16$. Computed densities with 100 cells. TWENO5 (*plusses*) and WENO5 (*squares*) against the exact solution (*solid line*)

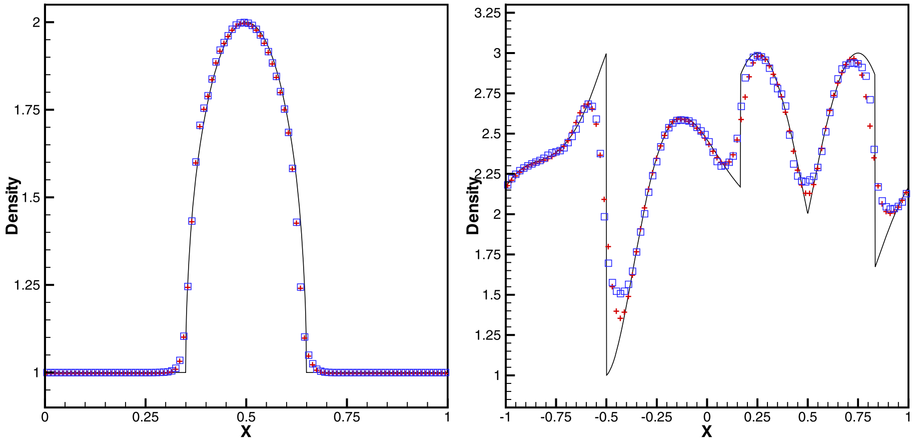

**Fig. 28** *Left*: (1) the linear semi-circle wave problem; *right*: (2) the highly discontinuous initial data problem. $t = 16$. Computed densities with 100 cells. TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*) against the exact solution (*solid line*), $M = 1.0$

(2) The highly discontinuous initial data is:

$$(\rho, u, p)^T|_{x+0.5} = \begin{cases} (2 - x\sin(1.5\pi x^2), 1, 1)^T, & -1 < x < -\frac{1}{3}, \\ (2 + |\sin(2\pi x)|, 1, 1)^T, & |x| < \frac{1}{3}, \\ (2 + 2x - 1 - \sin(3\pi x)/6, 1, 1)^T, & \frac{1}{3} < x < 1. \end{cases} \tag{4.10}$$

The computed densities $\rho$, obtained by the TWENO5 (plusses), WENO5 (squares), TWENO5-RKDG-T (plusses) and WENO5-RKDG-A (squares) schemes with the same TVB constant $M = 1.0$, are plotted in Figs. 27 and 28 at $t$ =16 along with the exact solution (solid line). We observe that the results of TWENO5 are similar to those of WENO5. Once again, we observe that the results of TWENO5-RKDG-T are better than or similar to those of WENO5-RKDG-A. Although we could not use in this example an optimal parameter $\alpha$ in the new scheme, we still obtained better resolutions, for example in the smooth extrema and nearby the contact discontinuity. Given these results, we anticipate that if we were to apply the new trigonometric polynomial space without choosing a proper parameter $\alpha$ in other 1D test cases, we could still get better results than with the previous RKDG schemes. Nevertheless, it is harder to calculate the linear weights, smoothness indicators, and nonlinear weights associated with the new schemes.

*Example 4.10* We solve the two dimensional advection equation:

$$u_t + \big(-(y-y_0)\omega u\big)_x + \big((x-x_0)\omega u\big)_y = 0. \tag{4.11}$$

The exact solution consists of a rotation of the initial values around $(x_0, y_0) = (0, 0)$. We set the angular velocity at $\omega = 1$ and the initial condition as:

$$u(x,y,0) = \begin{cases} 10 - \frac{10}{0.3}\sqrt{(x-0.5)^2+y^2}, & \{(x,y)|\sqrt{(x-0.5)^2+y^2} \le 0.3\}/ \\ & \{(x,y)|y \le 0.1 \cap |x-0.5| \le 0.06\}, \\ 0, & \text{otherwise } (x,y) \in [-1,1]\times[-1,1]. \end{cases} \tag{4.12}$$

This is a computationally challenging test case as the rotating angular velocity is large and the initial condition contains a smooth region, local extrema, and sharp discontinuity. The computed variable $u$ is plotted at $t = 10\pi$, which corresponds to 5 full rotations of the initial values. The surface of the solutions, obtained by the TWENO5, WENO5, TWENO5-RKDG-T and WENO5-RKDG-A schemes with the same TVB constant $M = 10$, as well as the exact solution are shown in Fig. 29 as different graphs. The 6-th and 7-th graphs in Fig. 29 are two 1D cutting plot along the line $y = 0$, which demonstrate that the TWENO5 scheme can get similar results as WENO5 scheme and TWENO5-RKDG-T scheme can obtain better results than WENO5-RKDG-A scheme in this two dimensional problem, as well.

*Example 4.11* Double Mach reflection problem. We solve the two dimensional Euler equations (4.4) in a computational domain of $[0,4] \times [0,1]$. A reflection wall lies at the bottom of the domain starting from $x = \frac{1}{6}$, $y = 0$, making a 60° angle with the $x$-axis. The reflection boundary condition is used at the wall, which for the rest of the bottom boundary (the part from $x = 0$ to $x = \frac{1}{6}$), the exact post-shock condition is imposed. At the top boundary is the exact motion of the Mach 10 shock. The results are shown at $t = 0.2$. We present the pictures of region $[0,3] \times [0,1]$, the blow-up region around the double Mach stems in Figs. 30 and 31. The schemes proposed here can obtain high order of accuracy in smooth region and retain essentially non-oscillatory property nearby the vicinity of significant shock or contact discontinuities robustly by using the trigonometric polynomial basis during the TWENO reconstruction and the DG spatial discretization procedures.

*Example 4.12* A Mach 3 wind tunnel with a step. 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 end of the tunnel. Initially, a right going

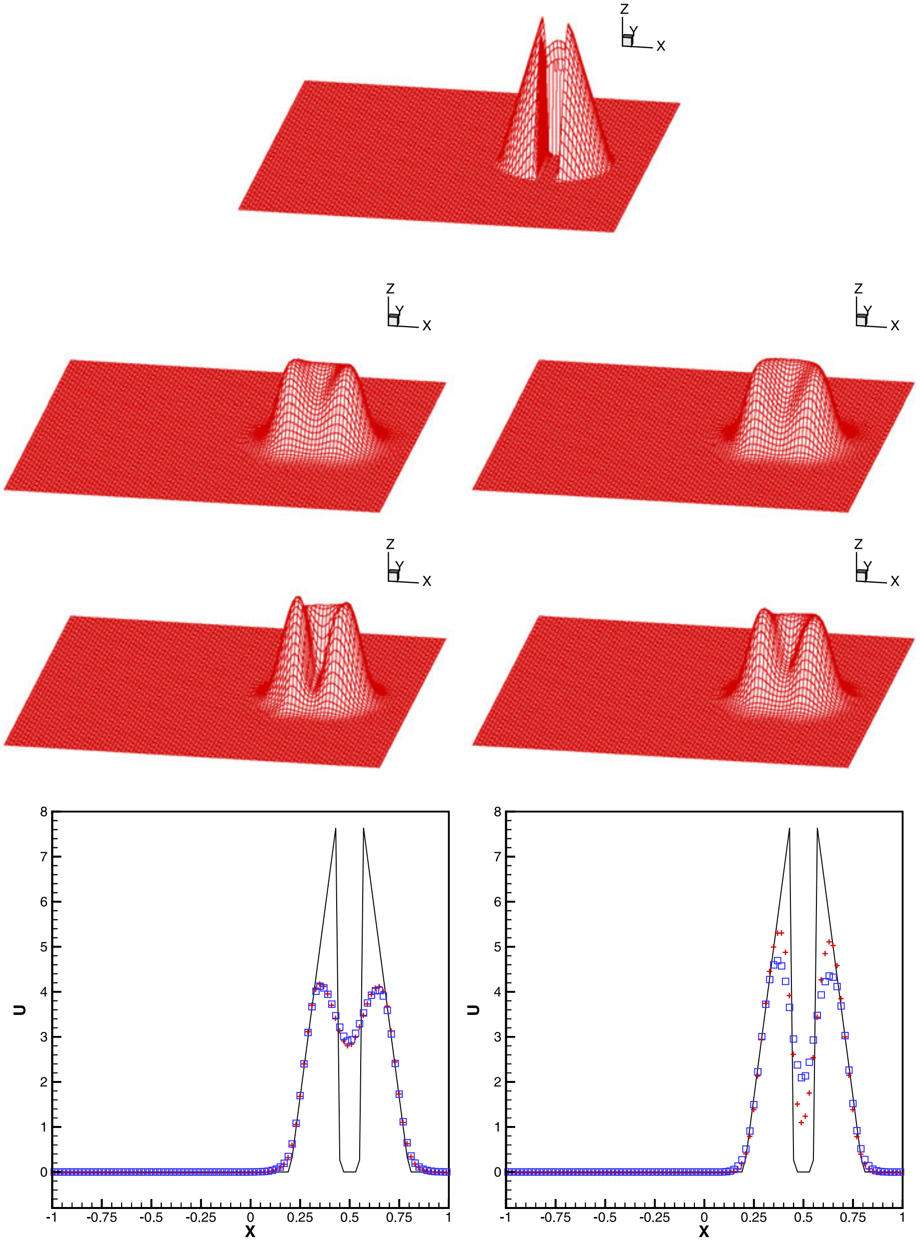

**Fig. 29** The advection problem. $t = 10\pi$. From left to right and top to bottom: (1) surface of the exact solution; (2) surface of TWENO5 solution; (3) surface of WENO5 solution; (4) surface of TWENO5-RKDG-T solution; (5) surface of WENO5-RKDG-A solution; (6) 1D cutting plot along the line $y = 0$ with TWENO5 (*plusses*) and WENO5 (*squares*) against the exact solution (*solid line*); (7) 1D cutting plot along the line $y = 0$ with TWENO5-RKDG-T (*plusses*) and WENO5-RKDG-A (*squares*). $M = 10.0$. $100 \times 100$ cells

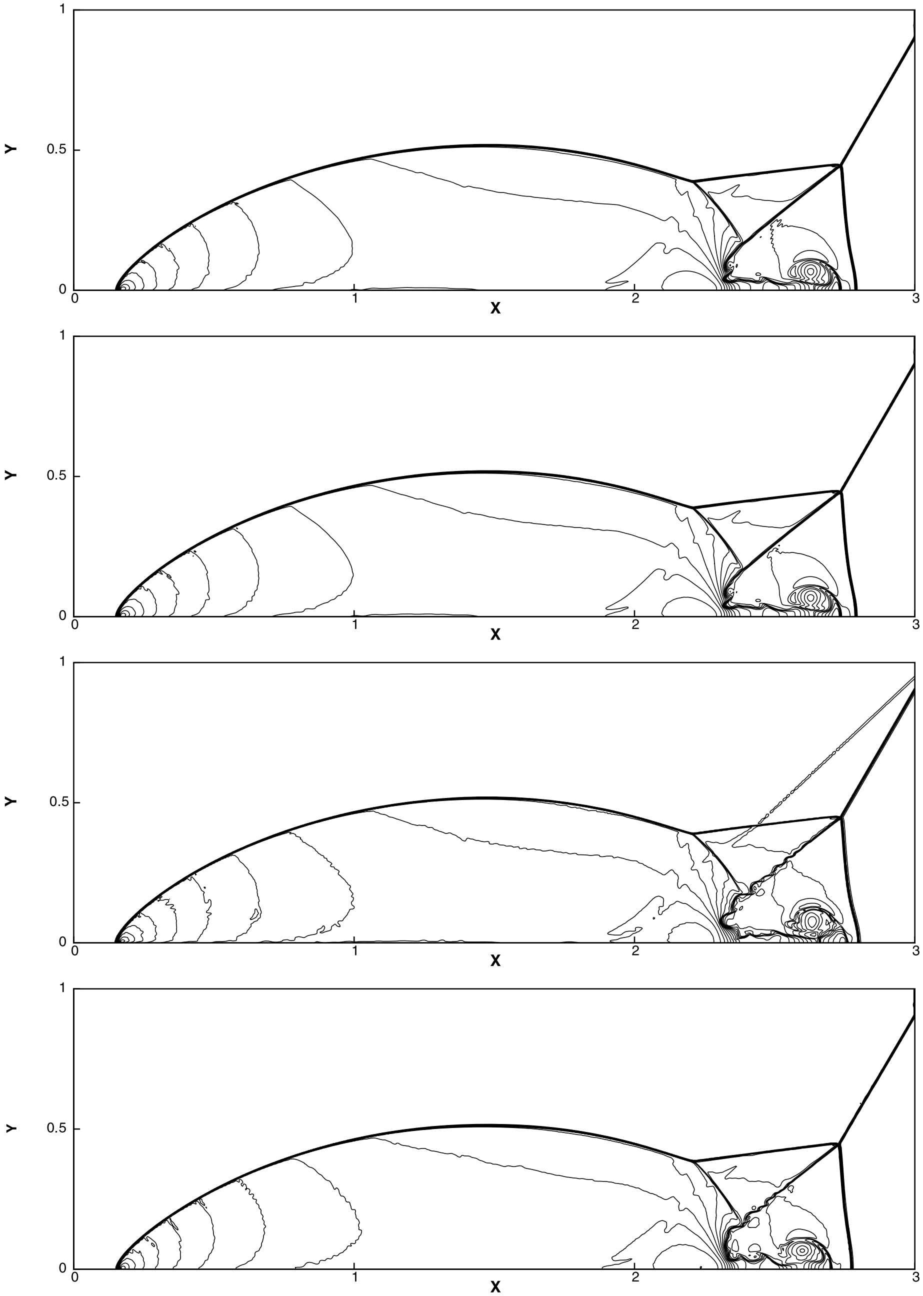

**Fig. 30** Double Mach reflection problem. Top to bottom: WENO5; TWENO5; WENO5-RKDG-A; TWENO5-RKDG-T. $t = 0.2$. 30 equally spaced density contours from 1.5 to 21.5. $M = 100.0$. $1600 \times 400$ cells

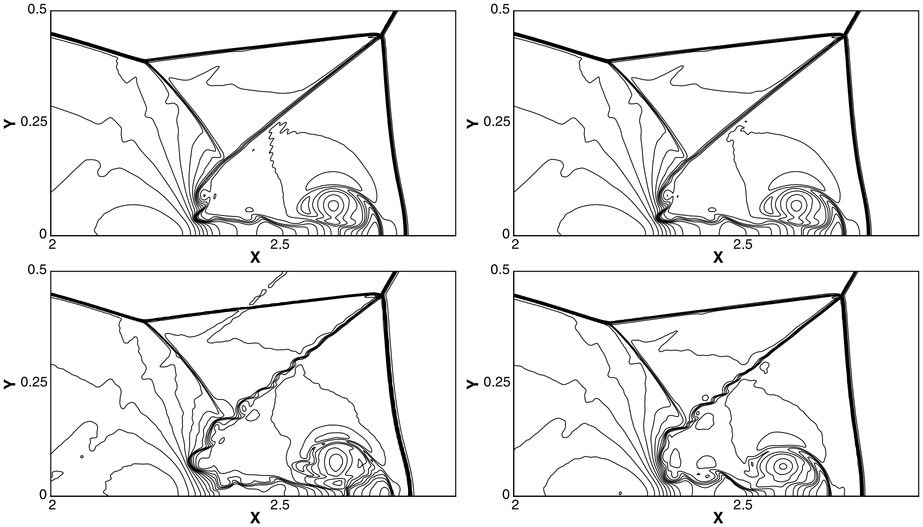

**Fig. 31** Double Mach reflection problem, zoom in. Left to right and top to bottom: WENO5; TWENO5; WENO5-RKDG-A; TWENO5-RKDG-T. $t = 0.2$. 30 equally spaced density contours from 1.5 to 21.5. $M = 100.0$. $1600 \times 400$ cells

Mach 3 flow is used. Reflective boundary conditions are applied along the walls of the tunnel and in flow and out flow boundary conditions are applied at the entrance and the exit. The results are shown at $t = 4$. We present the pictures of whole region $[0, 3] \times [0, 1]$ in Fig. 32. In this case, the TWENO5 and TWENO5-RKDG-T schemes using the trigonometric polynomial basis can retain the same features as those of the WENO5 and WENO5-RKDG-A schemes.

## 5 Concluding Remarks

We have developed new trigonometric WENO schemes and applied them as limiters for the RKDG methods on orthogonal trigonometric polynomial spaces to solve hyperbolic conservation laws by using finite volume high order TWENO reconstructions. The idea is to first identify "troubled cells" subject to the TWENO limiting using a TVB minmod-type limiter, then reconstruct the trigonometric polynomial solution inside the "troubled cells" by TWENO reconstruction using the cell averages of neighboring cells, while maintaining the original cell averages of the "troubled cells". We provide numerical results to show that the methods are stable, accurate in smooth regions, robust in maintaining accuracy, and sharp, non-oscillatory shock transition for RKDG methods. Numerical results demonstrate advantage of schemes based on trigonometric polynomial spaces over the schemes based on algebraic polynomial spaces when they are used to simulate the wave-like and highly oscillatory cases. We also observed that numerical results by the schemes based on trigonometric polynomial spaces were better than or similar to those by the schemes based on algebraic polynomial spaces even though we do not take optimal parameter for trigonometric polynomial spaces.

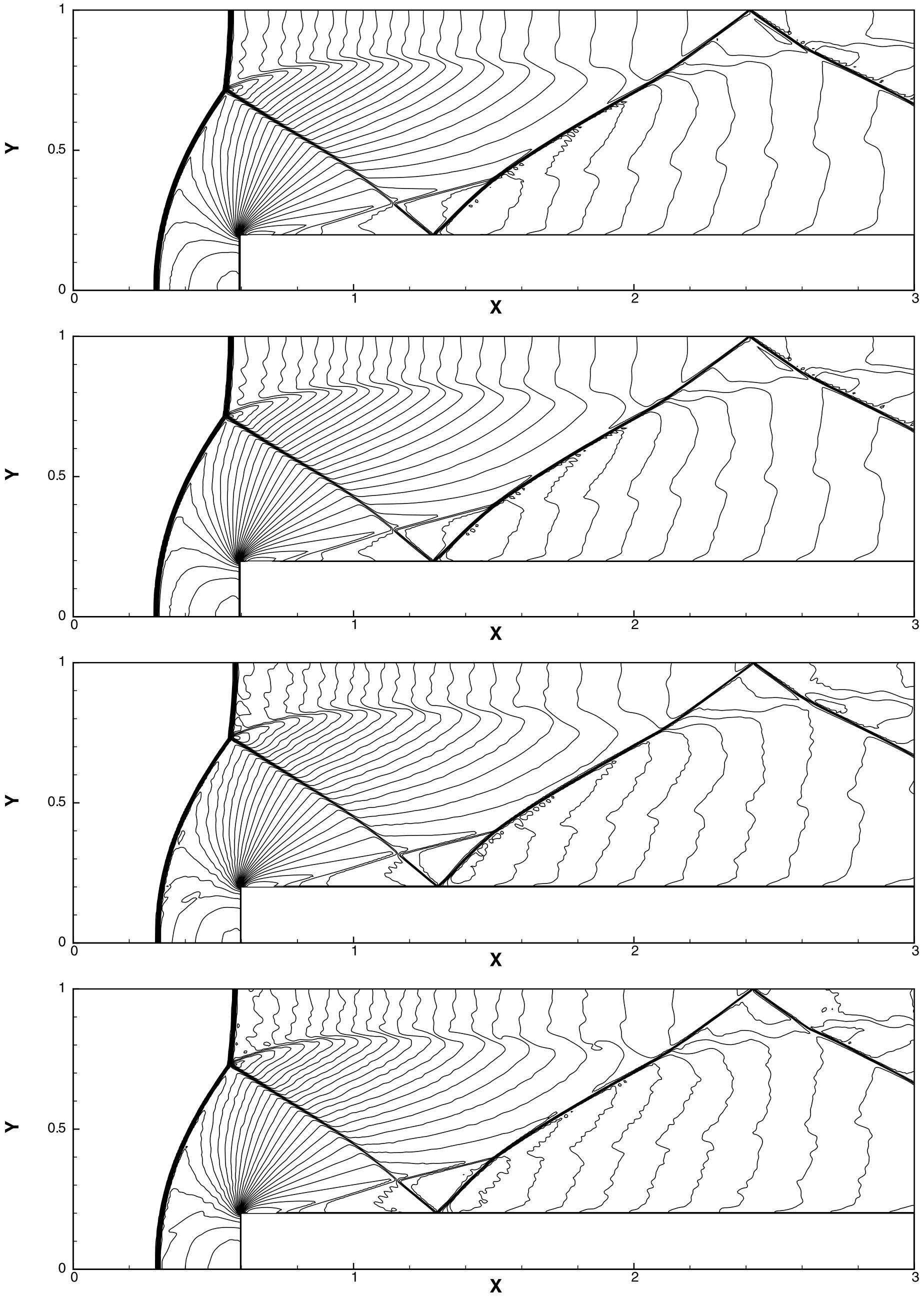

**Fig. 32** Forward step problem. TWENO5-RKDG-T. $t = 4.0$. 30 equally spaced density contours from 0.32 to 6.15. Top to bottom: WENO5; TWENO5; WENO5-RKDG-A; TWENO5-RKDG-T. $t = 0.2$. 30 equally spaced density contours from 1.5 to 21.5. $M = 100.0$. $600 \times 200$ cells

## References

1. Biswas, R., Devine, K.D., Flaherty, J.: Parallel, adaptive finite element methods for conservation laws. Appl. Numer. Math. **14**, 255–283 (1994)
2. Burbeau, A., Sagaut, P., Bruneau, C.H.: A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods. J. Comput. Phys. **169**, 111–150 (2001)
3. Christofi, S.: The study of building blocks for essentially non-oscillatory (ENO) schemes. Ph.D. thesis, Division of Applied Mathematics, Brown University (1996)
4. 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. **54**, 545–581 (1990)
5. 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. **84**, 90–113 (1989)
6. Cockburn, B., Shu, C.W.: The Runge-Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws. Modél. Math. Anal. Numér. **2AN**(25), 337–361 (1991)
7. Cockburn, B., Shu, C.W.: TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math. Comput. **52**, 411–435 (1989)
8. Cockburn, B., Shu, C.W.: The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J. Comput. Phys. **141**, 199–224 (1998)
9. Harten, A.: High resolution schemes for hyperbolic conservation laws. J. Comput. Phys. **49**, 357–393 (1983)
10. Harten, A., Osher, S.: Uniformly high-order accurate non-oscillatory schemes. IMRC Technical Summary Report 2823, Univ. of Wisconsin, Madison, WI (1985)
11. Jiang, G.S., Shu, C.W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**, 202–228 (1996)
12. Liu, X.D., Osher, S., Chan, T.: Weighted essentially non-oscillatory schemes. J. Comput. Phys. **115**, 200–212 (1994)
13. 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. **193**, 115–135 (2004)
14. Qiu, J., Shu, C.W.: Runge-Kutta discontinuous Galerkin method using WENO limiters. SIAM J. Sci. Comput. **26**, 907–929 (2005)
15. Qiu, J., Shu, C.W.: Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two-dimensional case. Comput. Fluids **34**, 642–663 (2005)
16. Reed, W.H., Hill, T.R.: Triangular mesh methods for neutron transport equation. Tech. Report LA-UR-73-479, Los Alamos Scientific Laboratory (1973)
17. Shi, J., Hu, C., Shu, C.W.: A technique of treating negative weights in WENO schemes. J. Comput. Phys. **175**, 108–127 (2002)
18. Shu, C.W.: Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. ICASE Report 97-65
19. Shu, C.W.: TVB uniformly high-order schemes for conservation laws. Math. Comput. **49**, 105–121 (1987)
20. Shu, C.W., Osher, S.: Efficient implementation of essentially non-oscillatory shock capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
21. Yuan, L., Shu, C.W.: Discontinuous Galerkin method based on non-polynomial approximation spaces. J. Comput. Phys. **218**, 295–323 (2006)
22. Yuan, L., Shu, C.W.: Discontinuous Galerkin method for a class of elliptic multi-scale problems. Int. J. Numer. Methods Fluids **56**, 1017–1032 (2008)
23. Zhu, J., Qiu, J.: A class of the fourth order finite volume Hermite weighted essentially non-oscillatory schemes. Sci. China Ser. A **51**, 1549–1560 (2008)
24. Zhu, J., Qiu, J.: Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method III: unstructured meshes. J. Sci. Comput. **39**, 293–321 (2009)
25. Zhu, J., Qiu, J.: Trigonometric WENO schemes for hyperbolic conservation laws and highly oscillatory problems. Commun. Comput. Phys. **8**, 1242–1263 (2010)
26. Zhu, J., Qiu, J., Shu, C.W., Dumbser, M.: Runge-Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes. J. Comput. Phys. **227**, 4330–4353 (2008)