Commun. Comput. Phys.
doi: 10.4208/cicp.300810.240511a

Vol. **11**, No. 3, pp. 985-1005
March 2012

Runge-Kutta Discontinuous Galerkin Method Using WENO-Type Limiters: Three-Dimensional Unstructured Meshes

Jun Zhu[1] and Jianxian Qiu[2,*]

[1] *College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, P.R. China.*
[2] *School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, P.R. China and Department of Mathematics, Nanjing University, Nanjing, Jiangsu 210093, P.R. China.*

Received 30 August 2010; Accepted (in revised version) 24 May 2011

Communicated by Chi-Wang Shu

Available online 18 November 2011

Abstract. This paper further considers weighted essentially non-oscillatory (WENO) and Hermite weighted essentially non-oscillatory (HWENO) finite volume methods as limiters for Runge-Kutta discontinuous Galerkin (RKDG) methods to solve problems involving nonlinear hyperbolic conservation laws. The application discussed here is the solution of 3-D problems on unstructured meshes. Our numerical tests again demonstrate this is a robust and high order limiting procedure, which simultaneously achieves high order accuracy and sharp non-oscillatory shock transitions.

AMS subject classifications: 65M06, 65M99, 35L65

Key words: Runge-Kutta discontinuous Galerkin method, limiter, WENO, HWENO, high order limiting procedure.

1 Introduction

Qiu *et al.* [16–18, 27, 28] have investigated weighted essentially non-oscillatory (WENO) and Hermite WENO (HWENO) finite volume methods as limiters for Runge-Kutta discontinuous Galerkin (RKDG) finite element methods [3–8], for the numerical solution of problems involving nonlinear hyperbolic conservation laws on structured and unstructured meshes. The goal is to construct a robust and high order limiting procedure that simultaneously achieves high order accuracy and sharp non-oscillatory shock transitions

[*]Corresponding author. *Email addresses:* zhujun@nuaa.edu.cn (J. Zhu), jxqiu@xmu.edu.cn, jxqiu@nju.edu.cn (J. Qiu)

for the RKDG method, and in this paper we consider the solution of problems involving 3-D nonlinear hyperbolic conservation laws of form

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

on 3-D unstructured meshes.

The WENO [9,11,12,14,25] and HWENO [16,18,26,27] schemes developed in recent years are a class of high order finite volume or finite difference schemes to numerically solve problems involving hyperbolic conservation laws, where both high order accuracy and essentially non-oscillatory shock transitions may be maintained. We have discussed third order finite volume WENO schemes in one space dimension [14], third and fifth order finite difference WENO schemes in various space dimensions with a general framework for the design of the smoothness indicators and nonlinear weights [12], and finite volume WENO schemes on structured and unstructured meshes [9, 11, 15, 21, 25]. The design of the WENO and also HWENO [16,18,26,27] schemes have been based on successful ENO schemes [10,23,24]. In both the ENO and WENO schemes, adaptive stencils were used in a reconstruction procedure based on local smoothness of the numerical solution, to automatically achieve high order accuracy and non-oscillatory behavior near discontinuities.

The first discontinuous Galerkin (DG) method was introduced in 1973 by Reed and Hill [19], for neutron transport described by steady state linear hyperbolic equations. A major development of the DG method was later carried out by Cockburn *et al.* in a series of papers [3–7]. They established a framework to readily solve problems involving *nonlinear* time-dependent hyperbolic conservation laws, via explicit nonlinearly stable high order Runge-Kutta time discretizations [23] and DG discretization in space, with exact or approximate Riemann solvers for interface fluxes and a total variation bounded (TVB) limiter [22] to achieve the non-oscillatory property for strong shocks. These schemes are now called RKDG methods.

To account for strong shocks in problems such as (1.1), an important component of a RKDG method is a nonlinear limiter to detect discontinuities and control any spurious oscillations that may arise nearby. Many such limiters have been used with RKDG methods. For example, the *minmod* TVB limiter [3–7] is a slope limiter using a technique borrowed from finite volume methodology, while a moment based limiter [1] and also an improved moment limiter [2] designed for discontinuous Galerkin methods use the moments of the numerical solution. However, these limiters tend to degrade accuracy when mistakenly used in smooth regions of the solution.

In [17], Qiu and Shu introduced the WENO methodology to provide limiters for the RKDG method on structured meshes, in the following way:

Step 1: First identify possible "troubled cells" – i.e. those cells that might need the limiting procedure.

Step 2: Replace the solution polynomials in these "troubled cells" by reconstructed polynomials, using WENO methodology that not only maintains the original cell averages (conservation) and the same orders of accuracy as before but is also less oscillatory.

This technique worked quite well for 1-D and 2-D test problems [17], and in our followup work where the more compact Hermite WENO method (HWENO method) was used in the ”troubled cells” [16,18,27].

In this paper, this approach is extended to 3-D problems on unstructured meshes, using both WENO [25] and HWENO [16,18,26] limiters involving cell averages or derivative cell averages of neighboring cells to reconstruct the moments directly. This has previously turned out to be a robust way to retain the original high order accuracy of the DG method. For the WENO limiter we adopt polynomials obtained by the finite volume WENO reconstruction procedure [25], and for the HWENO limiter we extend the finite volume Hermite WENO reconstruction [16,18,26,27], on 3-D tetrahedral meshes. The main differences and difficulty in constructing a WENO or HWENO limiter in 3-D, compared with lower dimensions, are as follows (cf. also [25]).

1. The methodology for choosing small stencils is not the same. Thus for non-overlapping tetrahedrons, we choose eight small stencils to do the reconstruction, and if necessary use least square methodology to solve for reconstructed polynomials other than for the optimal linear weights.
2. The numerical volume integral and area integral are involved in 3-D, whereas the numerical area integral and line integral apply in 1-D and 2-D, respectively [16,18,26,27].
3. Smoothness indicators are computed using numerical volume quadrature formulae in 3-D, whereas the numerical area volume quadrature formulae apply in 2-D.
4. Boundary numerical fluxes are defined on the facials (triangles) of the control volume (tetrahedrons) and the numerical area integral is required for the triangles in 3-D, whereas the boundary numerical fluxes are defined on the line segments of the control volume (triangle) and the numerical line integral is required for the segments in 2-D.

Details of our procedure for the second order DG method are discussed in Section 2, and extensive numerical results to verify accuracy and stability are presented in Section 3. Our concluding remarks are then made in Section 4.

2 WENO and HWENO reconstructions as limiters to the RKDG method on unstructured meshes

We now detail our procedure using WENO or HWENO reconstructions as limiters for the RKDG method.

Given the tetrahedral cell Δ_j, let $\mathbb{P}^k(\Delta_j)$ denote the set of polynomials of degree at most k defined on Δ_j. The k could change from cell to cell, but for simplicity we assume it is constant in this paper. In the DG method, both the solution and the test function space are in $V_h^k=\{v(x,y,z):v(x,y,z)|_{\Delta_j}\in\mathbb{P}^k(\Delta_j)\}$, and we emphasize the procedure does not depend on the specific basis chosen for the polynomials. We adopt a local orthogonal basis over the target tetrahedral cell, such as Δ_0: $\{v_l^{(0)}(x,y,z),l=0,\cdots,K;K=(k+1)(k+2)(k+3)/6-1\}$: thus

$$\begin{aligned}
&v_0^{(0)}(x,y,z)=1,\\
&v_1^{(0)}(x,y,z)=\frac{(x-x_0)}{|\Delta_0|^{1/3}},\\
&v_2^{(0)}(x,y,z)=a_{21}\frac{(x-x_0)}{|\Delta_0|^{1/3}}+\frac{(y-y_0)}{|\Delta_0|^{1/3}}+a_{22},\\
&v_3^{(0)}(x,y,z)=a_{31}\frac{(x-x_0)}{|\Delta_0|^{1/3}}+a_{32}\frac{(y-y_0)}{|\Delta_0|^{1/3}}+\frac{(z-z_0)}{|\Delta_0|^{1/3}}+a_{33},\\
&\quad\vdots
\end{aligned}$$

where (x_0,y_0,z_0) and $|\Delta_0|$ are the volume barycenter and the volume of the target tetrahedral cell Δ_0, respectively. We solve this linear system for the $a_{\ell m}$, by invoking the orthogonality property

$$\int_{\Delta_0} v_i^{(0)}(x,y,z)v_j^{(0)}(x,y,z)dxdydz=w_i\delta_{ij}, \tag{2.1}$$

where $w_i=\int_{\Delta_0}(v_i^{(0)}(x,y,z))^2dxdydz$. The numerical solution $u^h(x,y,z,t)$ in the space V_h^k can be written as

$$u^h(x,y,z,t)=\sum_{l=0}^{K}u_0^{(l)}(t)v_l^{(0)}(x,y,z)\quad \text{for}\,(x,y,z)\in\Delta_0,$$

and the degrees of freedom $u_0^{(l)}(t)$ are the moments defined by

$$u_0^{(l)}(t)=\frac{1}{w_l}\int_{\Delta_0}u^h(x,y,z,t)v_l^{(0)}(x,y,z)dxdydz,\quad l=0,\cdots,K.$$

In order to obtain the approximate solution, we evolve the degrees of freedom $u_0^{(l)}(t)$ via

$$\begin{aligned}
\frac{d}{dt}u_0^{(l)}(t)=\frac{1}{w_l}\bigg(&\int_{\Delta_0}\bigg(f(u^h(x,y,z,t))\frac{\partial}{\partial x}v_l^{(0)}(x,y,z)+g(u^h(x,y,z,t))\frac{\partial}{\partial y}v_l^{(0)}(x,y,z)\\
&+r(u^h(x,y,z,t))\frac{\partial}{\partial z}v_l^{(0)}(x,y,z)\bigg)dxdydz\\
&-\int_{\partial\Delta_0}\Big(f(u^h(x,y,z,t)),g(u^h(x,y,z,t)),r(u^h(x,y,z,t))\Big)^T\cdot\mathbf{n}\,v_l^{(0)}(x,y,z)ds\bigg)
\end{aligned}\tag{2.2}$$

for $l=0,\cdots,K$, where $\mathbf{n}$ is the outward unit normal at the boundary $\partial\Delta_0$.

The integral terms in Eq. (2.2) can either be computed exactly or by suitable numerical quadratures. In this paper, we use A_G points ($A_G=5$ for $k=1$) for the volume quadrature and E_G points ($E_G=6$ for $k=1$) for the face quadrature such that

$$\begin{aligned}&\int_{\Delta_0}\Big(f(u^h(x,y,z,t))\frac{\partial}{\partial x}v_l^{(0)}(x,y,z)+g(u^h(x,y,z,t))\frac{\partial}{\partial y}v_l^{(0)}(x,y,z)\\&\qquad+r(u^h(x,y,z,t))\frac{\partial}{\partial z}v_l^{(0)}(x,y,z)\Big)dxdydz\\&\approx|\Delta_0|\sum_G\sigma_G\Big(f(u^h(x_G,y_G,z_G,t))\frac{\partial}{\partial x}v_l^{(0)}(x_G,y_G,z_G)+g(u^h(x_G,y_G,z_G,t))\frac{\partial}{\partial y}v_l^{(0)}(x_G,y_G,z_G)\\&\qquad+r(u^h(x_G,y_G,z_G,t))\frac{\partial}{\partial z}v_l^{(0)}(x_G,y_G,z_G)\Big),\end{aligned}\tag{2.3}$$

$$\begin{aligned}&\int_{\partial\Delta_0}\Big(f(u^h(x,y,z,t)),g(u^h(x,y,z,t)),r(u^h(x,y,z,t))\Big)^T\cdot\mathbf{n}\,v_l^{(0)}(x,y,z)ds\\&\approx\sum_{ll=1}^{4}|\partial\Delta_{0_{ll}}|\sum_G\bar\sigma_G\Big(f(u^h(\bar x_{ll_G},\bar y_{ll_G},\bar z_{ll_G},t)),g(u^h(\bar x_{ll_G},\bar y_{ll_G},\bar z_{ll_G},t)),r(u^h(\bar x_{ll_G},\bar y_{ll_G},\bar z_{ll_G},t))\Big)^T\\&\qquad\times n_{ll}v_l^{(0)}(\bar x_{ll_G},\bar y_{ll_G},\bar z_{ll_G}),\end{aligned}\tag{2.4}$$

where $(x_G,y_G,z_G)\in\Delta_0$ and $(\bar x_{ll_G},\bar y_{ll_G},\bar z_{ll_G})\in\partial\Delta_{0_{ll}}$ are the quadrature points, and σ_G and $\bar\sigma_G$ are the quadrature weights. Since the face integral is on boundaries where the numerical solution is discontinuous, the flux $(f(u^h(x,y,z,t)),g(u^h(x,y,z,t)),\, r(u^h(x,y,z,t)))^T\cdot\mathbf{n}$ is replaced by a monotone numerical flux. The simple Lax-Friedrichs flux is used in all of our numerical tests. The semi-discrete scheme (2.2) is discretized in time by a nonlinear stable Runge-Kutta time discretization – e.g. the third-order version [23]

$$\begin{cases}u^{(1)}=u^n+\Delta tL(u^n),\\u^{(2)}=\frac{3}{4}u^n+\frac{1}{4}u^{(1)}+\frac{1}{4}\Delta tL(u^{(1)}),\\u^{n+1}=\frac{1}{3}u^n+\frac{2}{3}u^{(2)}+\frac{2}{3}\Delta tL(u^{(2)}).\end{cases}\tag{2.5}$$

Without further modification, the method described above can compute solutions to Eq. (1.1) that are either smooth or have weak shocks and other discontinuities. However, if the discontinuities are strong, the scheme generates significant oscillations and even nonlinear instability. To avoid this, we borrow the technique of a slope limiter from the finite volume methodology, and use it after each Runge-Kutta inner stage or after the complete Runge-Kutta time step.

In this paper, we only use the limiter adopted in [7] to detect "troubled cells". The main procedure is as follows:

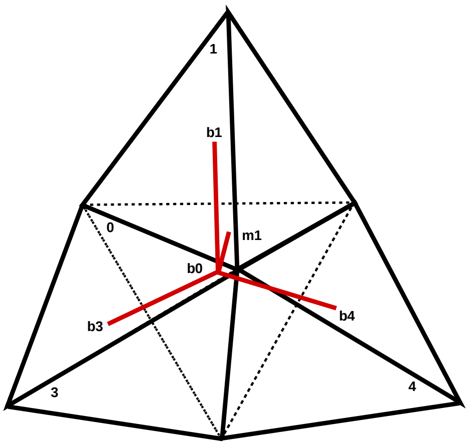

Figure 1: The limiting diagram.

- Use $(x_{m_\ell},y_{m_\ell},z_{m_\ell})$, $\ell=1, 2, 3, 4$, to denote the barycenters of the facial triangles on the boundaries of the target tetrahedral cell Δ_0 and $(x_{b_i},y_{b_i},z_{b_i})$, $i=1, 2, 3, 4$, to denote the barycenters of the neighboring tetrahedral cells Δ_i, $i=1, 2, 3, 4$, as shown in Fig. 1.

- Solve the four linear equations to get the nonnegative α_1, α_2, α_3, similar to [7] – i.e. solve

$$\begin{cases} x_{m_1}-x_{b_0}=\alpha_1(x_{b_1}-x_{b_0})+\alpha_2(x_{b_2}-x_{b_0})+\alpha_3(x_{b_3}-x_{b_0}), \\ y_{m_1}-y_{b_0}=\alpha_1(y_{b_1}-y_{b_0})+\alpha_2(y_{b_2}-y_{b_0})+\alpha_3(y_{b_3}-y_{b_0}), \\ z_{m_1}-z_{b_0}=\alpha_1(z_{b_1}-z_{b_0})+\alpha_2(z_{b_2}-z_{b_0})+\alpha_3(z_{b_3}-z_{b_0}), \end{cases} \tag{2.6}$$

$$\begin{cases} x_{m_1}-x_{b_0}=\alpha_1(x_{b_1}-x_{b_0})+\alpha_2(x_{b_2}-x_{b_0})+\alpha_3(x_{b_4}-x_{b_0}), \\ y_{m_1}-y_{b_0}=\alpha_1(y_{b_1}-y_{b_0})+\alpha_2(y_{b_2}-y_{b_0})+\alpha_3(y_{b_4}-y_{b_0}), \\ z_{m_1}-z_{b_0}=\alpha_1(z_{b_1}-z_{b_0})+\alpha_2(z_{b_2}-z_{b_0})+\alpha_3(z_{b_4}-z_{b_0}), \end{cases} \tag{2.7}$$

$$\begin{cases} x_{m_1}-x_{b_0}=\alpha_1(x_{b_1}-x_{b_0})+\alpha_2(x_{b_3}-x_{b_0})+\alpha_3(x_{b_4}-x_{b_0}), \\ y_{m_1}-y_{b_0}=\alpha_1(y_{b_1}-y_{b_0})+\alpha_2(y_{b_3}-y_{b_0})+\alpha_3(y_{b_4}-y_{b_0}), \\ z_{m_1}-z_{b_0}=\alpha_1(z_{b_1}-z_{b_0})+\alpha_2(z_{b_3}-z_{b_0})+\alpha_3(z_{b_4}-z_{b_0}), \end{cases} \tag{2.8}$$

$$\begin{cases} x_{m_1}-x_{b_0}=\alpha_1(x_{b_2}-x_{b_0})+\alpha_2(x_{b_3}-x_{b_0})+\alpha_3(x_{b_4}-x_{b_0}), \\ y_{m_1}-y_{b_0}=\alpha_1(y_{b_2}-y_{b_0})+\alpha_2(y_{b_3}-y_{b_0})+\alpha_3(y_{b_4}-y_{b_0}), \\ z_{m_1}-z_{b_0}=\alpha_1(z_{b_2}-z_{b_0})+\alpha_2(z_{b_3}-z_{b_0})+\alpha_3(z_{b_4}-z_{b_0}). \end{cases} \tag{2.9}$$

At least one such set of linear equations may necessarily depend only on the position of $(x_{m_1},y_{m_1},z_{m_1})$ and the geometry of the tetrahedral meshes. We then define

$$\begin{aligned} &\tilde{u}^h(x_{m_1},y_{m_1},z_{m_1},t) \\ \equiv\, &u^h(x_{m_1},y_{m_1},z_{m_1},t)-u_0^{(0)}(t), \end{aligned} \tag{2.10}$$

$$\begin{aligned} &\Delta u(x_{m_1},y_{m_1},z_{m_1},t) \\ \equiv\, &\alpha_1(u_1^{(0)}(t)-u_0^{(0)}(t))+\alpha_2(u_3^{(0)}(t)-u_0^{(0)}(t))+\alpha_3(u_4^{(0)}(t)-u_0^{(0)}(t)). \end{aligned} \tag{2.11}$$

- Using the TVB modified *minmod* function [22] defined as

$$\tilde{m}(a_1,a_2)=\begin{cases} a_1, & \text{if } |a_1|\le M|\Delta_0|,\\ \begin{cases} s\min(|a_1|,|a_2|), & \text{if } s=sign(a_1)=sign(a_2),\\ 0, & \text{otherwise,}\end{cases} & \text{otherwise,}\end{cases} \tag{2.12}$$

where the choice of the TVB constant $M>0$ is problem dependent, compute the quantity

$$\tilde{u}^{mod}=\tilde{m}(\tilde{u}^h(x_{m_1},y_{m_1},z_{m_1},t),\gamma\Delta u(x_{m_1},y_{m_1},z_{m_1},t)) \tag{2.13}$$

with $\gamma>1$ (we take $\gamma=1.5$ in our numerical tests). If

$$\tilde{u}^{mod}\neq\tilde{u}^h(x_{m_1},y_{m_1},z_{m_1},t),$$

Δ_0 is marked as a "troubled cell" for further reconstruction. This procedure is then repeated for the other three faces of the tetrahedral cell Δ_0. Since the WENO-type reconstructions maintain high order accuracy in the "troubled cells", it is less crucial to choose an accurate M. Numerical tests for different choices of M are discussed in Section 3. For the "troubled cells", we reconstruct the polynomial solutions while retaining their cell averages. In other words, we reconstruct the degrees of freedom $u_0^{(l)}(t)$, $l=1,\cdots,K$ and retain only the cell average $u_0^{(0)}(t)$.

2.1 WENO reconstruction as a limiter to the RKDG method

For the k=1 case, let us now summarize the procedure for the first order moments $u_0^{(1)}(t)$, $u_0^{(2)}(t)$ and $u_0^{(3)}(t)$ in the "troubled cell" Δ_0 using the WENO reconstruction procedure [25]. For simplicity, we relabel the "troubled cell" and its neighboring cells, and write $u_*^{(*)}=u_*^{(*)}(t)$ wherever that will not cause confusion.

Step 1.1. Select the big stencil

$$S=\{\Delta_0,\Delta_1,\Delta_2,\Delta_3,\Delta_4,\Delta_{11},\Delta_{12},\Delta_{13},\Delta_{21},\Delta_{22},\Delta_{23},\Delta_{31},\Delta_{32},\Delta_{33},\Delta_{41},\Delta_{42},\Delta_{43}\}$$

that includes Δ_0, its four neighboring tetrahedrons Δ_1, Δ_2, Δ_3, Δ_4 and their neighboring tetrahedrons, where Δ_{j1}, Δ_{j2}, Δ_{j3} are adjacent to Δ_j but not Δ_0 for $j=1,2,3,4$.

Step 1.2. Divide S into sixteen smaller stencils and construct sixteen linear polynomials

$$q_i(x,y,z)\in span\left\{1,\frac{(x-x_0)}{|\Delta_0|^{1/3}},\frac{(y-y_0)}{|\Delta_0|^{1/3}},\frac{(z-z_0)}{|\Delta_0|^{1/3}}\right\},\quad i=1,\cdots,16,$$

which satisfy

$$\int_{\Delta_\ell} q_i(x,y,z)v_0^{(\ell)}(x,y,z)dxdydz=\int_{\Delta_\ell} u_\ell^{(0)}(v_0^{(\ell)}(x,y,z))^2dxdydz,\qquad \Delta_\ell\in S_i, \tag{2.14}$$

for

$i=1,\ \ell=0,1,2,3;$ $\quad i=2,\ \ell=0,2,3,4;$ $\quad i=3,\ \ell=0,3,4,1;$ $\quad i=4,\ \ell=0,4,1,2;$
$i=5,\ \ell=0,1,11,12;$ $\quad i=6,\ \ell=0,1,12,13;$ $\quad i=7,\ \ell=0,1,13,11;$ $\quad i=8,\ \ell=0,2,21,22;$
$i=9,\ \ell=0,2,22,23;$ $\quad i=10,\ \ell=0,2,23,21;$ $\quad i=11,\ \ell=0,3,31,32;$ $\quad i=12,\ \ell=0,3,32,33;$
$i=13,\ \ell=0,3,33,31;$ $\quad i=14,\ \ell=0,4,41,42;$ $\quad i=15,\ \ell=0,4,42,43;$ $\quad i=16,\ \ell=0,4,43,41.$

Step 1.3. Find the combination coefficients, also called linear weights, denoted by $\gamma_1^{(l)},\cdots,\gamma_{16}^{(l)}$ and satisfying

$$\begin{aligned}&\int_{\Delta_0} u(x,y,z)v_l^{(0)}(x,y,z)dxdydz\\ =&\int_{\Delta_0}\sum_{i=1}^{16}\gamma_i^{(l)}q_i(x,y,z)v_l^{(0)}(x,y,z)dxdydz, \quad l=1,\cdots,K,\end{aligned} \tag{2.15}$$

valid for any polynomial $u(z,y,z)$ of degree at most 2, when we can obtain a third order approximation to $u(x,y,z)$ at the volume quadrature point (x_G,y_G,z_G) for all sufficiently smooth functions $u(x,y,z)$. It is also notable that Eq. (2.15) holds for any polynomial $u(x,y,z)$ of degree at most 1 if $\sum_{i=1}^{16}\gamma_i^{(l)}=1$, because each individual $q_i(x,y,z)$ reconstructs linear polynomials exactly. There are six other constraints on the linear weights $\gamma_1^{(l)},\cdots,\gamma_{16}^{(l)}$, on requiring Eq. (2.15) to hold for

$$u(x,y,z)=\frac{(x-x_0)^2}{|\Delta_0|^{2/3}},\ \frac{(x-x_0)(y-y_0)}{|\Delta_0|^{2/3}},\ \frac{(x-x_0)(z-z_0)}{|\Delta_0|^{2/3}},\ \frac{(y-y_0)^2}{|\Delta_0|^{2/3}},\ \frac{(y-y_0)(z-z_0)}{|\Delta_0|^{2/3}},\ \frac{(z-z_0)^2}{|\Delta_0|^{2/3}},$$

respectively. This leaves 9 free parameters in determining the linear weights $\gamma_1^{(l)},\cdots,\gamma_{16}^{(l)}$. These free parameters are uniquely determined by the least square

$$\min\left(\sum_{i=1}^{16}\gamma_i^{(l)}\right)^2, \qquad l=1,\cdots,K,$$

subject to the constraints listed above. Thus we obtain the linear weights uniquely, but they may not always remain positive. However, we can use the methods in [11,21] and elsewhere to overcome this drawback. Thus in brief, the linear weights may be divided into two distinct groups

$$\bar{\gamma}_i^{(l)^+}=\frac{\gamma_i^{(l)}+3|\gamma_i^{(l)}|}{2}, \quad \bar{\gamma}_i^{(l)^-}=\frac{-\gamma_i^{(l)}+3|\gamma_i^{(l)}|}{2}, \quad i=1,\cdots,16; \quad l=1,\cdots,K. \tag{2.16}$$

such that

$$\sigma^{(l)^\pm}=\sum_{j=1}^{16}\bar{\gamma}_j^{(l)^\pm}, \qquad \gamma_i^{(l)^\pm}=\frac{\bar{\gamma}_i^{(l)^\pm}}{\sigma^{(l)^\pm}}, \qquad i=1,\cdots,16; \quad l=1,\cdots,K. \tag{2.17}$$

Step 1.4. Compute the smoothness indicators denoted by β_i, $i=1,\cdots,16$, which measure how smooth the functions $q_i(x,y,z)$, $i=1,\cdots,16$ are in the target tetrahedral cell Δ_0. The smaller these smoothness indicators, the smoother the functions in the target cell. We use the recipe for the smoothness indicators in [12] – viz.

$$\beta_i=\sum_{|\ell|=1}^{k}|\Delta_0|^{\frac{2|\ell|}{3}-1}\int_{\Delta_0}\left(\frac{\partial^{|\ell|}}{\partial x^{\ell_1}\partial y^{\ell_2}\partial z^{\ell_3}}q_i(x,y,z)\right)^2 dxdydz, \tag{2.18}$$

where $\ell=(\ell_1,\ell_2,\ell_3)$.

Step 1.5. Compute the nonlinear weights based on the smoothness indicators

$$\omega_i^{(l)\pm}=\frac{\bar{\omega}_i^{(l)\pm}}{\sum_{\ell=1}^{16}\bar{\omega}_\ell^{(l)\pm}},\qquad \bar{\omega}_\ell^{(l)\pm}=\frac{\gamma_\ell^{(l)\pm}}{(\varepsilon+\beta_\ell)^2},\qquad l=1,\cdots,K, \tag{2.19}$$

where ε is a small positive number to prevent the denominator becoming zero. We found that the computations for the 3-D test cases are not sensitive if ε varies from 10^{-3} to 10^{-6}, and we chose to set $\varepsilon=10^{-3}$ as in [25].

For $l=1,\cdots,K$, the moments of the reconstructed polynomial are then

$$u_0^{(l)}(t)=\frac{\displaystyle\int_{\Delta_0}\Big(\sigma^{(l)^+}\sum_{i=1}^{16}\omega_i^{(l)^+}q_i(x,y,z)-\sigma^{(l)^-}\sum_{i=1}^{16}\omega_i^{(l)^-}q_i(x,y,z)\Big)v_l^{(0)}(x,y,z)dxdydz}{\displaystyle\int_{\Delta_0}(v_l^{(0)}(x,y,z))^2dxdydz}. \tag{2.20}$$

Remark 2.1. The above WENO reconstruction assumes that none of the tetrahedral meshes overlap and the sixteen small stencils are all workable. However, the reconstruction procedure is still practicable if at least seven small stencils are available, even when some tetrahedrons overlap. On the other hand, the WENO reconstruction procedure is inapplicable if the small stencil number is less than seven – although we can then proceed to scan the next neighboring tetrahedral layers, to see if they include enough small stencil candidates to render the procedure workable.

2.2 HWENO reconstruction as a limiter to the RKDG method

For the $k=1$ case, let us now summarize how to reconstruct the first order moments $u_0^{(1)}(t)$, $u_0^{(2)}(t)$ and $u_0^{(3)}(t)$ in the "troubled cell" Δ_0 using the HWENO reconstruction procedure. For simplicity, we relabel the "troubled cell" and its neighboring cells.

Step 2.1. Select the big stencil $S=\{\Delta_0,\Delta_1,\Delta_2,\Delta_3,\Delta_4\}$.

Step 2.2. Divide S into eight smaller stencils and construct eight linear polynomials

$$q_i(x,y,z)\in span\left\{1,\frac{(x-x_0)}{|\Delta_0|^{1/3}},\frac{(y-y_0)}{|\Delta_0|^{1/3}},\frac{(z-z_0)}{|\Delta_0|^{1/3}}\right\},\quad i=1,\cdots,8.$$

The first four polynomials should satisfy the conditions

$$\int_{\Delta_\ell} q_i(x,y,z)v_0^{(\ell)}(x,y,z)dxdydz = \int_{\Delta_\ell} u_\ell^{(0)}(v_0^{(\ell)}(x,y,z))^2 dxdydz, \quad \Delta_\ell \in S_i, \tag{2.21}$$

for

$$i=1,\ \ell=0,1,2,3; \quad i=2,\ \ell=0,2,3,4; \quad i=3,\ \ell=0,3,4,1; \quad i=4,\ \ell=0,4,1,2;$$

and the next four polynomials should satisfy the conditions

$$\int_{\Delta_0} q_i(x,y,z)v_0^{(0)}(x,y,z)dxdydz = \int_{\Delta_0} u_0^{(0)}(v_0^{(0)}(x,y,z))^2 dxdydz, \quad \Delta_0 \in S_i, \tag{2.22}$$

$$\begin{aligned}\min\Bigg(&\Big(\int_{\Delta_\ell} q_i(x,y,z)v_0^{(\ell)}(x,y,z) - u_\ell^{(0)}(v_0^{(\ell)}(x,y,z))^2 dxdydz\Big)^2 \\ &+\Big(\int_{\Delta_{\ell_x}} q_i(x,y,z)v_1^{(\ell_x)}(x,y,z) - u_{\ell_x}^{(1)}(v_1^{(\ell_x)}(x,y,z))^2 dxdydz\Big)^2 \\ &+\Big(\int_{\Delta_{\ell_y}} q_i(x,y,z)v_2^{(\ell_y)}(x,y,z) - u_{\ell_y}^{(2)}(v_2^{(\ell_y)}(x,y,z))^2 dxdydz\Big)^2 \\ &+\Big(\int_{\Delta_{\ell_z}} q_i(x,y,z)v_3^{(\ell_z)}(x,y,z) - u_{\ell_z}^{(3)}(v_3^{(\ell_z)}(x,y,z))^2 dxdydz\Big)^2\Bigg), \\ &\qquad \Delta_\ell|_{\ell\neq 0}, \Delta_{\ell_x}, \Delta_{\ell_y}, \Delta_{\ell_z} \in S_i,\end{aligned} \tag{2.23}$$

for

$$\begin{aligned} &i=5,\ \ell=0,1,\ \ell_x=1,\ \ell_y=1,\ \ell_z=1; \qquad i=6,\ \ell=0,2,\ \ell_x=2,\ \ell_y=2,\ \ell_z=2; \\ &i=7,\ \ell=0,3,\ \ell_x=3,\ \ell_y=3,\ \ell_z=3; \qquad i=8,\ \ell=0,4,\ \ell_x=4,\ \ell_y=4,\ \ell_z=4.\end{aligned}$$

Step 2.3. Find the combination coefficients, also called linear weights, denoted by $\gamma_1^{(l)},\cdots,\gamma_8^{(l)}$ that satisfy

$$\int_{\Delta_0} u(x,y,z)v_l^{(0)}(x,y,z)dxdydz = \int_{\Delta_0}\sum_{i=1}^{8}\gamma_i^{(l)} q_i(x,y,z)v_l^{(0)}(x,y,z)dxdydz, \quad l=1,\cdots,K, \tag{2.24}$$

valid for any polynomial $u(z,y,z)$ of degree at most 2, when we can obtain a third order approximation to $u(x,y,z)$ at the volume quadrature point (x_G,y_G,z_G) for all sufficiently smooth functions $u(x,y,z)$. It is again notable that (2.24) also holds for any polynomial $u(x,y,z)$ of degree at most 1 if $\sum_{i=1}^{8}\gamma_i^{(l)}=1$, because each individual $q_i(x,y,z)$ reconstructs linear polynomials exactly. There are also six other constraints on the linear weights $\gamma_1^{(l)},\cdots,\gamma_8^{(l)}$ as before, but now on requiring Eq. (2.24) to hold for

$$u(x,y,z) = \frac{(x-x_0)^2}{|\Delta_0|^{2/3}},\ \frac{(x-x_0)(y-y_0)}{|\Delta_0|^{2/3}},\ \frac{(x-x_0)(z-z_0)}{|\Delta_0|^{2/3}},\ \frac{(y-y_0)^2}{|\Delta_0|^{2/3}},\ \frac{(y-y_0)(z-z_0)}{|\Delta_0|^{2/3}},\ \frac{(z-z_0)^2}{|\Delta_0|^{2/3}},$$

respectively. In determining the linear weights $\gamma_1^{(l)},\cdots,\gamma_8^{(l)}$, this leaves one free parameter, which is uniquely determined by the least square

$$\min\left(\sum_{i=1}^{8}\gamma_i^{(l)}\right)^2, \qquad l=1,\cdots,K$$

subject to the constraints listed above. Thus we can get the linear weights uniquely, but again they may always not be positive, so we use the methods as before to get $\gamma_i^{(l)^{\pm}}$, $\sigma^{(l)^{\pm}}$, etc..

Step 2.4. Compute the smoothness indicators denoted by β_i, $i=1,\cdots,8$ that measure how smooth the functions $q_i(x,y,z)$, $i=1,\cdots,8$ are in the target tetrahedral cell Δ_0. Once again, the smaller these smoothness indicators the smoother the functions in the target cell, where we use Eq. (2.18).

Step 2.5. Compute the nonlinear weights based on the smoothness indicators:

$$\omega_i^{(l)^{\pm}}=\frac{\bar{\omega}_i^{(l)^{\pm}}}{\sum_{\ell=1}^{8}\bar{\omega}_\ell^{(l)^{\pm}}}, \qquad \bar{\omega}_\ell^{(l)^{\pm}}=\frac{\gamma_\ell^{(l)^{\pm}}}{(\varepsilon+\beta_\ell)^2}, \qquad l=1,\cdots,K. \tag{2.25}$$

The 3-D test cases were again found to be insensitive to ε varying from 10^{-3} to 10^{-6}, and we chose $\varepsilon=10^{-3}$ in our computations [25].

For $l=1,\cdots,K$,the moments of the reconstructed polynomial are then

$$u_0^{(l)}(t)=\frac{\int_{\Delta_0}(\sigma^{(l)^+}\sum_{i=1}^{8}\omega_i^{(l)^+}q_i(x,y,z)-\sigma^{(l)^-}\sum_{i=1}^{8}\omega_i^{(l)^-}q_i(x,y,z))v_l^{(0)}(x,y,z)dxdydz}{\int_{\Delta_0}(v_l^{(0)}(x,y,z))^2dxdydz}. \tag{2.26}$$

3 Numerical results

In this Section, we provide numerical results demonstrating the performance of the WENO and HWENO reconstructions as limiters for the RKDG method on unstructured meshes (cf. Section 2). The CFL number used is 0.3 for all of the numerical tests. In order to magnify the possible effect of the WENO and HWENO limiters on accuracy, we often used a small M value near zero (viz. $M=0.01$) for the constant in the TVB minmod limiter to identify "troubled cells", such that many good cells are also identified as "troubled cells".

Example 3.1. We solved the linear scalar equation

$$u_t+u_x+u_y+u_z=0 \tag{3.1}$$

on a uniform tetrahedral mesh over the domain $[-2,2]\times[-2,2]\times[-2,2]$, with initial condition $u(x,y,z,0)=\sin(\pi(x+y+z)/2)$ and periodic boundary conditions in each direction.

Table 1: $u_t+u_x+u_y+u_z=0$. $u(x,y,z,0)=\sin(\pi(x+y+z)/2)$. Periodic boundary conditions in each direction. $t=1$. L^1 and L^∞ errors. RKDG with the WENO and HWENO limiters ($M=0.01$) compared to RKDG without limiter. Uniform tetrahedral mesh.

	DG with WENO limiter				DG without limiter			
tetrahedrons	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
750	5.11E-1		8.20E-1		9.76E-2		2.98E-1	
6000	2.18E-1	1.23	4.04E-1	1.02	1.55E-2	2.65	6.82E-2	2.13
48000	6.67E-2	1.71	1.41E-1	1.51	3.15E-3	2.30	1.60E-2	2.09
384000	1.31E-2	2.34	3.29E-2	2.11	7.34E-4	2.10	3.84E-3	2.06
	DG with HWENO limiter				DG without limiter			
tetrahedrons	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
750	5.84E-1		9.23E-1		9.76E-2		2.98E-1	
6000	3.65E-1	0.68	6.00E-1	0.62	1.55E-2	2.65	6.82E-2	2.13
48000	1.17E-1	1.63	2.43E-1	1.30	3.15E-3	2.30	1.60E-2	2.09
384000	2.66E-2	2.15	6.13E-2	1.99	7.34E-4	2.10	3.84E-3	2.06

Table 2: $u_t+u_x+u_y+u_z=0$. $u(x,y,z,0)=\sin(\pi(x+y+z)/2)$. Periodic boundary conditions in each direction. $t=1$. CPU time (second). RKDG with the WENO and HWENO limiters ($M=0.01$) compared to RKDG without limiter. Uniform tetrahedral mesh.

	DG with WENO limiter	DG with HWENO limiter	DG without limiter
tetrahedrons	CPU time (second)		
750	2.573	0.993	0.232
6000	48.90	19.09	6.650
48000	813.5	300.4	158.6
384000	9312	4358	2407

We computed the solution up to $t=1$. The errors and numerical orders of accuracy for the RKDG method with the WENO and HWENO limiters, compared with the original RKDG method without any limiter, are shown in Table 1. The computational costs of the RKDG method with and without the WENO and HWENO limiters are shown in Table 2. It can be seen that the WENO and HWENO limiters retain the designed order of accuracy, but the error magnitudes are larger than for the original RKDG method on the same mesh.

Example 3.2. We solved the nonlinear scalar Burgers equation

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

on a uniform tetrahedral mesh over the computing domain $[-3,3]\times[-3,3]\times[-3,3]$, with the initial condition $u(x,y,z,0)=0.5+\sin(\pi(x+y+z)/3)$ and periodic boundary conditions in each direction. We computed the solution up to $t=0.5/\pi^2$, where the solution is still smooth. The errors and numerical order of accuracy for the RKDG method with the WENO and HWENO limiters compared with the original RKDG method without limiter

Table 3: $u_t+(u^2/2)_x+(u^2/2)_y+(u^2/2)_z=0$. $u(x,y,z,0)=0.5+\sin(\pi(x+y+z)/3)$. Periodic boundary conditions in each direction. $t=0.5/\pi^2$. L^1 and L^∞ errors. RKDG with the WENO and HWENO limiters ($M=0.01$) compared to RKDG without limiter. Uniform tetrahedral mesh.

	DG with WENO limiter				DG without limiter			
tetrahedrons	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
750	7.11E-2		2.08E-1		3.32E-2		1.40E-1	
6000	2.34E-2	1.60	8.97E-2	1.22	1.08E-2	1.61	5.07E-2	1.47
48000	5.87E-3	2.00	2.61E-2	1.78	3.23E-3	1.75	1.49E-2	1.77
384000	8.79E-4	2.74	4.34E-3	2.59	8.50E-4	1.93	3.94E-3	1.92
	DG with HWENO limiter				DG without limiter			
tetrahedrons	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
750	1.13E-1		4.03E-1		3.32E-2		1.40E-1	
6000	4.92E-2	1.20	1.80E-1	1.15	1.08E-2	1.61	5.07E-2	1.47
48000	1.32E-2	1.89	6.12E-2	1.55	3.23E-3	1.75	1.49E-2	1.77
384000	2.04E-3	2.69	1.48E-2	2.04	8.50E-4	1.93	3.94E-3	1.92

Table 4: $u_t+(u^2/2)_x+(u^2/2)_y+(u^2/2)_z=0$. $u(x,y,z,0)=0.5+\sin(\pi(x+y+z)/3)$. Periodic boundary conditions in each direction. $t=0.5/\pi^2$. CPU time (second). RKDG with the WENO and HWENO limiters ($M=0.01$) compared to RKDG without limiter. Uniform tetrahedral mesh.

	DG with WENO limiter	DG with HWENO limiter	DG without limiter
tetrahedrons	CPU time (second)		
750	1.186	0.484	0.340
6000	21.73	9.473	6.215
48000	326.4	144.8	111.6
384000	4593	2254	1504

are shown in Table 3. The computational costs of the RKDG method with and without the WENO and HWENO limiters are shown in Table 4. It can again be seen that the WENO and HWENO limiters retain the designed order of accuracy, but the error magnitudes are larger than for the original RKDG method on the same mesh.

Example 3.3. We solved the system of Euler equations

$$\frac{\partial}{\partial t}\begin{pmatrix}\rho\\ \rho u\\ \rho v\\ \rho w\\ E\end{pmatrix}+\frac{\partial}{\partial x}\begin{pmatrix}\rho u\\ \rho u^2+p\\ \rho vu\\ \rho wu\\ u(E+p)\end{pmatrix}+\frac{\partial}{\partial y}\begin{pmatrix}\rho v\\ \rho uv\\ \rho v^2+p\\ \rho wv\\ v(E+p)\end{pmatrix}+\frac{\partial}{\partial z}\begin{pmatrix}\rho w\\ \rho uw\\ \rho vw\\ \rho w^2+p\\ w(E+p)\end{pmatrix}=0 \tag{3.3}$$

where ρ is the density, u is the x-component of the velocity, v its y-component and w its z-component, E the total energy and p the pressure. The initial conditions were $\rho(x,y,z,0)=1+0.2\sin(\pi(x+y+z)/3)$, $u(x,y,z,0)=1$, $v(x,y,z,0)=1$, $w(x,y,z,0)=1$, $p(x,y,0)=1$ and the computing domain was $[-3,3]\times[-3,3]\times[-3,3]$ with uniform tetrahedral mesh, and periodic boundary conditions were applied in each direction. We computed the solution up

Table 5: 3D-Euler equations: initial data $\rho(x,y,z,0)=1+0.2\sin(\pi(x+y+z)/3)$, $u(x,y,z,0)=1$, $v(x,y,z,0)=1$, $w(x,y,z,0)=1$ and $p(x,y,z,0)=1$. Periodic boundary conditions in each direction. $t=1$. L^1 and L^∞ errors. RKDG with the WENO and HWENO limiters ($M=0.01$) compared to RKDG without limiter. Uniform tetrahedral mesh.

	DG with WENO limiter				DG without limiter			
tetrahedrons	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
750	8.48E-2		1.34E-1		1.80E-2		6.07E-2	
6000	4.16E-2	1.03	7.32E-2	0.88	3.12E-3	2.52	1.22E-2	2.30
48000	6.38E-3	2.71	1.47E-2	2.32	7.26E-4	2.10	2.91E-3	2.07
	DG with HWENO limiter				DG without limiter			
tetrahedrons	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
750	9.58E-2		1.55E-1		1.80E-2		6.07E-2	
6000	6.31E-2	0.60	1.04E-1	0.57	3.12E-3	2.52	1.22E-2	2.30
48000	8.29E-3	2.92	1.91E-2	2.44	7.26E-4	2.10	2.91E-3	2.07

to $t=1$. The errors and numerical orders of accuracy of the density for the RKDG method with the WENO and HWENO limiters compared with the original RKDG method without a limiter are shown in Table 5. As in the previous example, it can be seen that the WENO and HWENO limiters again retain the designed order of accuracy, and the error magnitudes are larger than for the original RKDG method on the same mesh.

We then tested the performance of the RKDG method with the WENO and HWENO limiters for problems containing shocks. For a direct comparison with the RKDG method using the original *minmod* TVB limiter, we refer to the results in [3–5, 7]. In general, they are comparable when M is chosen adequately. The RKDG method with the WENO and HWENO limiters produced much better results than the original *minmod* TVB limiter.

Example 3.4. We solved the same nonlinear Burgers equation (3.2) with the same initial condition $u(x,y,z,0)=0.5+\sin(\pi(x+y+z)/3)$, except that the results plotted for $t=5/\pi^2$ are after a shock has appeared. A uniform tetrahedral mesh with 384000 tetrahedrons was used in the computation. In Fig. 2, we show the contours on the surface and one dimensional cutting-plot along $x=y$, $z=0$ of the solutions by the RKDG method with the WENO and HWENO limiters. It can be seen that the scheme gives non-oscillatory shock transitions for this problem.

Example 3.5. Transonic flow over the Onera M6 wing [20] is a classic CFD validation case for external flows, because of its simple geometry combined with complexities in the transonic flow. We assumed the Mach number $M_\infty=0.84$ and angle of attack $\alpha=3.06°$. The computational domain is $\sqrt{x^2+y^2+z^2}\le 16$ and $z\ge 0$, consisting of 143645 tetrahedrons and 24382 points with 1311 triangles over the surface (the surface mesh used is shown in Fig. 3). In this case, the second order RKDG scheme with the WENO and HWENO limiters and TVB constants $M=1,10$ and 100 were adopted in the numerical tests. In Table 6, we document the maximal percentage and the average percentage of cells declared to be "troubled cells", for different TVB constants in the *minmod* limiter to identify "troubled

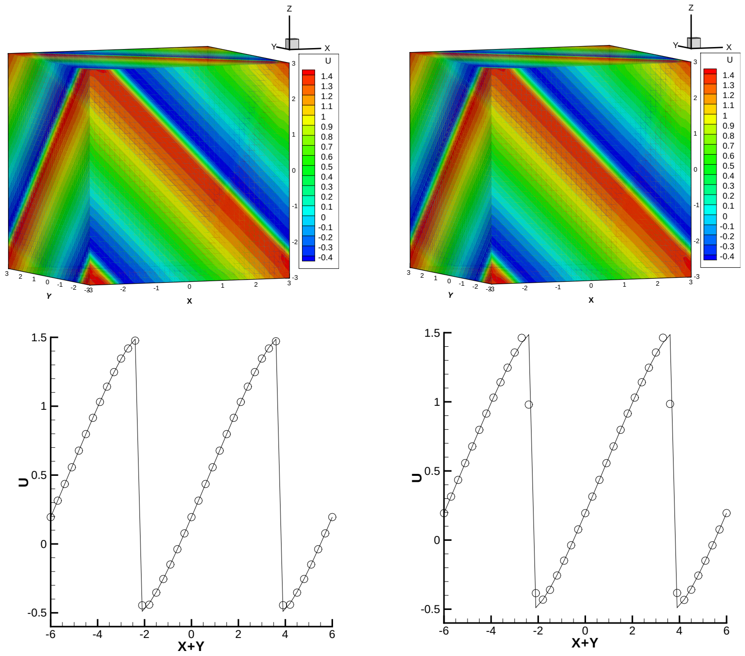

Figure 2: Burgers equation. $t=5/\pi^2$. Contour plot on the surface (top) and 1D cutting-plot along $x=y$, $z=0$ with circles representing the numerical solution and the line representing the exact solution (bottom) by WENO-RKDG (left) and HWENO-RKDG (right).

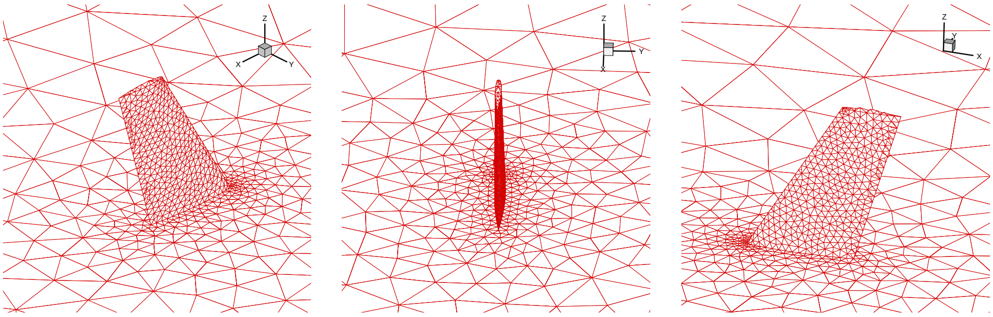

Figure 3: Zoom in on the Onera M6 wing surface mesh.

Table 6: Onera M6 wing problem. The maximal percentage and the average percentage of cells declared to be "troubled cells" under the WENO and HWENO limitings.

$M_\infty=0.84$, angle of attack $\alpha=3.06°$							
WENO-RKDG				HWENO-RKDG			
TVB constant M	1	10	100	TVB constant M	1	10	100
maximum percentage	8.20	5.32	1.81	maximum percentage	9.07	6.03	2.12
average percentage	7.02	4.29	1.26	average percentage	8.10	5.08	1.55

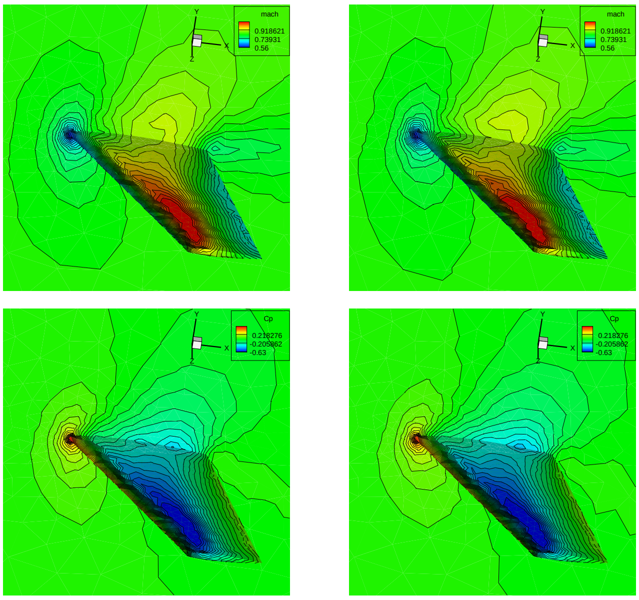

Figure 4: Onera M6 wing problem. $M_\infty = 0.84$, angle of attack $\alpha = 3.06°$. Mach number with contour plot on the surface (top) and Pressure coefficient number with contour plot on the surface (bottom) by WENO-RKDG (left) and HWENO-RKDG (right).

cells". The results for the different TVB constants M do not appear to differ much – and to save space, we only show the 30 equally spaced Mach number contours from 0.56 to 1.08, and 30 equally spaced pressure coefficients $C_p = (p - p_\infty)/0.5\rho_\infty(u_\infty^2 + v_\infty^2)$ for $M = 100$m where p is the local pressure (and p_∞, ρ_∞, u_∞ and v_∞ are the pressure, density, and x and y velocity components in the faraway free fluid region), with numbered contours from -0.63 to 0.6 in Fig. 4. It is seen that the schemes perform well with good resolution, with both the shock and contact discontinuities well captured.

Example 3.6. We used INRIA's 3D tetrahedral elements for the BTC0 (streamlined body, laminar) test case in project ADIGMA with the Mach number M_∞=0.5 and angle of attack $\alpha = 0°$ [13]. The computational domain used was $\sqrt{x^2 + y^2 + z^2} \leq 10$, consisting of 191753 tetrahedrons and 33708 points with 8244 triangles over the surface. The surface mesh used in the computation is shown in Fig. 5. The second order RKDG scheme with the WENO and HWENO limiters and the TVB constant values $M = 1$,10 and 100 were again used here in the numerical tests. In Table 7, we document the maximal percentage and the average percentage of cells declared to be "troubled cells" for different TVB constant M in the *minmod* limiter to identify "troubled cells", and for large M we see that only a small percentage are declared "troubled cells". There is again little perceptible difference for the different TVB constants M, so to save space we show only 80 equally spaced Mach numbers (from 0.15 to 1.44) and 80 equally spaced pressure coefficient numbers (from -0.12 to 1.11) for $M = 100$ in Fig. 6. The schemes again perform well with good resolution.

Table 7: BTC0 problem. The maximal percentage and the average percentage of cells declared to be "troubled cells" under the WENO and HWENO limitings.

$M_\infty = 0.5$, angle of attack $\alpha = 0°$							
WENO-RKDG				HWENO-RKDG			
TVB constant M	1	10	100	TVB constant M	1	10	100
maximum percentage	5.68	2.09	0.18	maximum percentage	7.31	3.43	0.46
average percentage	3.06	0.47	0.00	average percentage	5.60	1.91	0.09

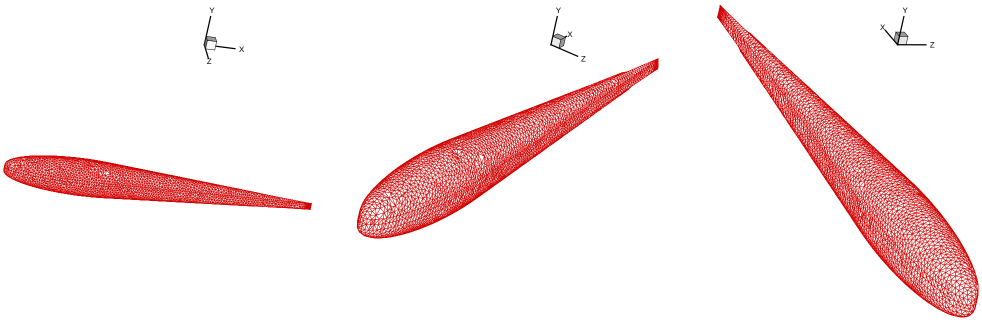

Figure 5: Zoom in on the BTC0 surface mesh.

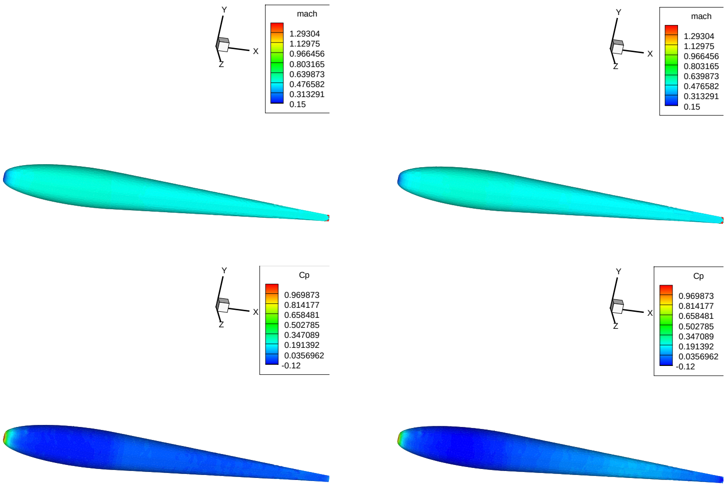

Figure 6: BTC0 problem. $M_\infty = 0.5$, angle of attack $\alpha = 0°$. Mach number with contour plot on the surface (top) and Pressure coefficient number with contour plot on the surface (bottom) by WENO-RKDG (left) and HWENO-RKDG (right).

Example 3.7. We considered inviscid Euler transonic flow past a single Y3815-pb11 plane (the repository of this free 3D model is available at INRIA's Free 3D Mesh Download `http://www-rocq1.inria.fr/gamma`), with the Mach number $M_\infty = 0.8$ and angle of attack $\alpha = 1.25°$, and for $M_\infty = 0.85$ and $\alpha = 1°$. The computational domain was $\sqrt{x^2+y^2+z^2} \leq 100$, consisting of 180855 tetrahedrons and 50588 points with 24640 trian-

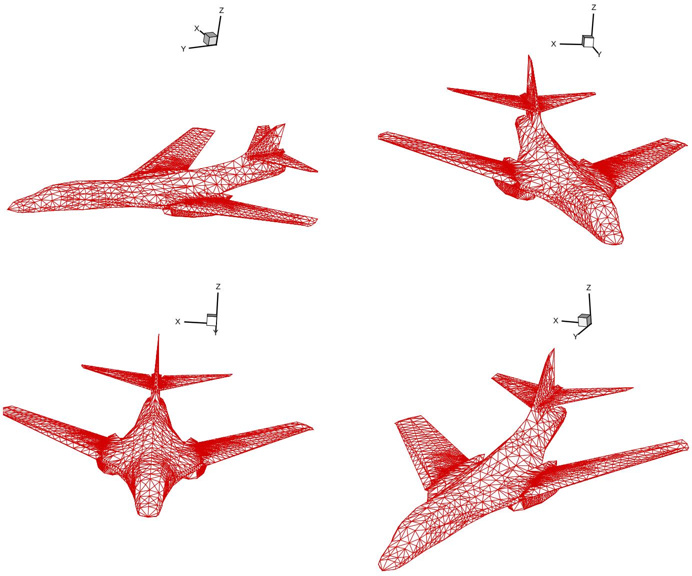

Figure 7: Zoom in on the Y3815-pb1l plane surface mesh.

gles over the surface. The surface mesh used in the computation is shown in Fig. 7. The second order RKDG scheme with the WENO and HWENO limiters and the TVB constants $M = 1,10$ and 100 were used in the numerical tests. In Table 8, we document the maximal percentage and the average percentage of cells declared to be "troubled cells" for different TVB constants M in the *minmod* limiter to identify "troubled cells". For large M, only a small percentage are again declared "troubled cells". To save space, only the results for $M = 100$ are shown as before. Mach number contours plotted on the surface with 80 equally spaced contours from 0.11 to 1.86, and pressure coefficient number contours plotted on the surface with 80 equally spaced contours from -2.09 to 1.28 for the Mach number $M_\infty = 0.8$ and angle of attack $\alpha = 1.25°$, are shown in Fig. 8. Mach number

Table 8: Y3815-pb1l plane problem. The maximal percentage and the average percentage of cells declared to be "troubled cells" under the WENO and HWENO limitings.

$M_\infty = 0.8$, angle of attack $\alpha = 1.25°$							
WENO-RKDG				HWENO-RKDG			
TVB constant M	1	10	100	TVB constant M	1	10	100
maximum percentage	5.64	1.79	0.41	maximum percentage	6.16	1.92	0.43
average percentage	4.16	1.16	0.22	average percentage	5.13	1.45	0.29
$M_\infty = 0.85$, angle of attack $\alpha = 1°$							
WENO-RKDG				HWENO-RKDG			
TVB constant M	1	10	100	TVB constant M	1	10	100
maximum percentage	5.78	1.85	0.42	maximum percentage	6.34	1.97	0.45
average percentage	4.45	1.25	0.24	average percentage	5.40	1.54	0.30

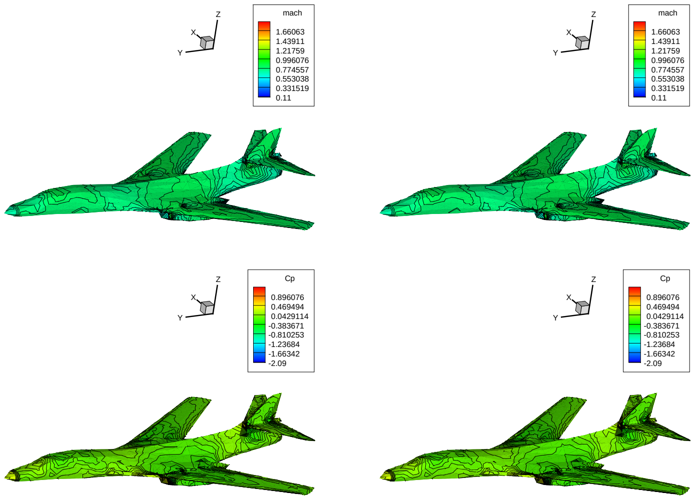

Figure 8: Y3815-pb1l plane problem. $M_\infty = 0.8$, angle of attack $\alpha = 1.25°$. Mach number with contour plot on the surface (top) and pressure coefficient number with contour plot on the surface (bottom) by WENO-RKDG (left) and HWENO-RKDG (right).

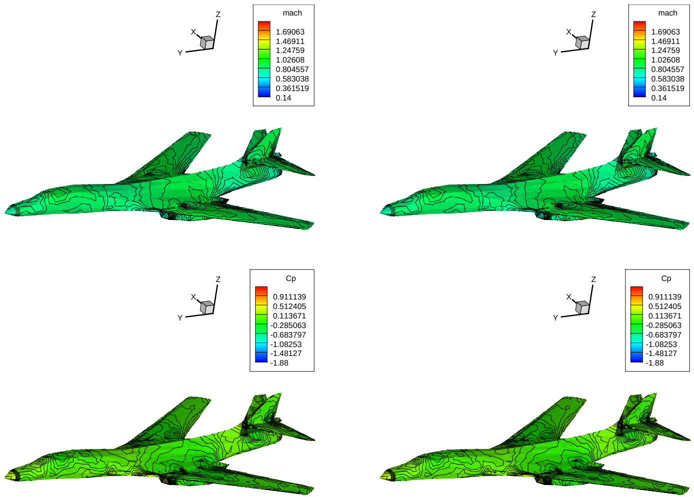

Figure 9: Y3815-pb1l plane problem. $M_\infty = 0.85$, angle of attack $\alpha = 1°$. Mach number with contour plot on the surface (top) and Pressure coefficient number with contour plot on the surface (bottom) by WENO-RKDG (left) and HWENO-RKDG (right).

contours plotted on the surface with 80 equally spaced contours from 0.14 to 1.89, and pressure coefficient number contours plotted on the surface with 80 equally spaced contours from -1.88 to 1.27 for the Mach number $M_\infty = 0.85$ and angle of attack $\alpha = 1°$, are shown in Fig. 9. It can be seen that the schemes perform well with good resolution, with both the shock and contact discontinuities well captured.

4 Concluding remarks

We have developed limiters for the RKDG method for the numerical solution of problems involving hyperbolic conservation laws, using finite volume high order WENO and HWENO reconstructions on 3-D unstructured meshes. Thus ”troubled cells” are first identified under a WENO-type limiting, using a TVB *minmod*-type limiter. The polynomial solution inside the ”troubled cells” is then obtained by WENO or HWENO reconstructions, using cell averages or derivative averages of neighboring tetrahedrons while retaining the original cell averages of the ”troubled cells”. Numerical results show that the method is stable, accurate, and robust in maintaining accuracy.

Acknowledgments

The research was partially supported by NSFC grant 10931004, 10871093, 11002071 and the European project ADIGMA on the development of innovative solution algorithms for aerodynamic simulations.

References

[1] R. Biswas, K.D. Devine and J. Flaherty, Parallel, adaptive finite element methods for conservation laws, Applied Numerical Mathematics, 14 (1994), 255-283.

[2] A. Burbeau, P. Sagaut and C.H. Bruneau, A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods, Journal of Computational Physics, 169 (2001), 111-150.

[3] B. Cockburn, S. Hou and C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: The multidimensional case, Mathematics of Computation, 54 (1990), 545-581.

[4] B. Cockburn, S.-Y. Lin and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: One dimensional systems, Journal of Computational Physics, 84 (1989), 90-113.

[5] B. Cockburn and C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Mathematics of Computation, 52 (1989), 411-435.

[6] B. Cockburn and C.-W. Shu, The Runge-Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws, Mathematical Modelling and Numerical Analysis (M^2AN), 25 (1991), 337-361.

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

[8] B. Cockburn and C.-W. Shu, Runge-Kutta discontinuous Galerkin method for convection-dominated problems, Journal of Scientific Computing, 16 (2001), 173-261.

[9] O. Friedrichs, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, Journal of Computational Physics, 144 (1998), 194-212.

[10] A. Harten, B. Engquist, S. Osher and S. Chakravathy, Uniformly high order accurate essentially non-oscillatory schemes, III, Journal of Computational Physics, 71 (1987), 231-303.

[11] C. Hu and C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, Journal of Computational Physics, 150 (1999), 97-127.

[12] G. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, Journal of Computational Physics, 126 (1996), 202-228.

[13] N. Kroll *et al.* (Editors), ADIGMA - A European Initiative on the Development of Adaptive Higher-Order Variational Methods for Aerospace, Applications Notes on Numerical Fluid Mechanics and Multidisciplinary Design, V113, 2010, Springer.

[14] X. Liu, S. Osher and T. Chan, Weighted essentially non-oscillatory schemes, Journal of Computational Physics, 115 (1994), 200-212.

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

[16] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method: One dimensional case, Journal of Computational Physics, 193 (2003), 115-135.

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

[18] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: Two dimensional case, Computers & Fluids, 34 (2005), 642-663.

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

[20] V. Schmitt and F. Charpin, Pressure Distributions on the ONERA-M6-Wing at Transonic Mach Numbers, Experimental Data Base for Computer Program Assessment. Report of the Fluid Dynamics Panel Working Group 04, AGARD AR 138, May 1979.

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

[22] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Mathematics of Computation, 49 (1987), 105-121.

[23] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes, Journal of Computational Physics, 77 (1988), 439-471.

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

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

[26] J. Zhu and J. Qiu, A class of the fourth order finite volume Hermite weighted essentially non-oscillatory schemes, Science in China, Series A–Mathematics, 51 (2008), 1549-1560.

[27] J. Zhu and J. Qiu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method III: Unstructured meshes, Journal of Scientific Computing, 39 (2009), 293-321.

[28] J. Zhu, J. Qiu, C.-W. Shu and M. Dumbser, Runge-Kutta discontinuous Galerkin method using WENO limiters II: Unstructured meshes, Journal of Computational Physics, 227 (2008), 4330-4353.