Third Order WENO Scheme on Three Dimensional Tetrahedral Meshes

Yong-Tao Zhang[1,*] and Chi-Wang Shu[2]

[1] *Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-4618, USA.*
[2] *Division of Applied Mathematics, Brown University, Providence, RI 02912, USA.*

Abstract. We extend the weighted essentially non-oscillatory (WENO) schemes on two dimensional triangular meshes developed in [7] to three dimensions, and construct a third order finite volume WENO scheme on three dimensional tetrahedral meshes. We use the Lax-Friedrichs monotone flux as building blocks, third order reconstructions made from combinations of linear polynomials which are constructed on diversified small stencils of a tetrahedral mesh, and non-linear weights using smoothness indicators based on the derivatives of these linear polynomials. Numerical examples are given to demonstrate stability and accuracy of the scheme.

AMS subject classifications: 65M99

Key words: Weighted essentially non-oscillatory (WENO) schemes, finite volume schemes, high-order accuracy, tetrahedral meshes.

1 Introduction

The weighted essentially non-oscillatory (WENO) methodology adopted in this paper, for solving hyperbolic conservation laws with discontinuous solutions, was first developed in [9] for a third order finite volume version in one space dimension and in [8] for third and fifth order finite difference version in multi space dimensions with a general framework for the design of the smoothness indicators and non-linear weights. The main idea of the WENO scheme is to form a weighted combination of several local reconstructions based on different stencils (usually referred to as small stencils) and use it as the final WENO reconstruction. The combination coefficients (also called non-linear weights) depend on the linear weights, often chosen to increase the order of accuracy over that on each small stencil, and on the smoothness indicators which measure the smoothness of the reconstructed function in the relevant small stencils. WENO schemes have

---

*Corresponding author. *Email addresses:* `yzhang10@nd.edu` (Y.-T. Zhang), `shu@dam.brown.edu` (C.-W. Shu)

the advantage of attaining uniform high order accuracy in smooth regions of the solution while maintaining sharp and essentially monotone shock transitions. It is more difficult to design WENO schemes for unstructured meshes. First of all, the finite difference approach [8] requires mesh smoothness and cannot be used on unstructured meshes while maintaining local conservation, and we must use the more complicated and more costly finite volume approach [9, 12]. There are two types of WENO schemes designed in the literature on unstructured meshes. The first type consists of WENO schemes whose order of accuracy is not higher than that of the reconstruction on each small stencil. That is, for this type of WENO schemes, the non-linear weights are designed purely for the purpose of stability, or to avoid spurious oscillations, and they do not contribute towards the increase of the order of accuracy. Such WENO schemes are easier to construct, since the linear weights can be chosen as arbitrary positive numbers for better stability, for example the centered small stencil can be assigned a larger linear weight than the others. The WENO schemes in [4] for two dimensional triangulations and in [2, 3] for three dimensional triangulations belong to this class. The second type consists of WENO schemes whose order of accuracy is higher than that of the reconstruction on each small stencil. For example, the third order WENO scheme in [7] is based on second order linear polynomial reconstructions on small stencils, and the fourth order WENO scheme in [7] is based on third order quadratic polynomial reconstructions on small stencils. See also [15] for similar WENO schemes for solving Hamilton-Jacobi equations, which belong to the second type as well. This second type of WENO schemes are more difficult to construct, however they have a more compact stencil than the first type WENO schemes of the same accuracy, which is an advantage in some applications, such as when the WENO methodology is used as limiters for the discontinuous Galerkin methods [10, 11]. In this paper, we generalize the second type WENO schemes in [7] to three dimensions, and construct a third order finite volume WENO scheme on three dimensional tetrahedral meshes. We use the Lax-Friedrichs monotone flux as building blocks, third order reconstructions made from combinations of second order linear polynomials which are constructed on diversified small stencils of a tetrahedral mesh, and non-linear weights using smoothness indicators based on the derivatives of these linear polynomials. Numerical examples are given to demonstrate the stability and accuracy of the scheme.

   The organization of this paper is as follows. The algorithm is developed in Sections 2 and 3. Section 4 contains numerical examples verifying stability, convergence and accuracy of the algorithm. Concluding remarks are given in Section 5.

2   The finite volume formulation on 3D tetrahedral meshes

In this paper we solve the three-dimensional conservation law

$$\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} + \frac{\partial g(u)}{\partial y} + \frac{\partial h(u)}{\partial z} = 0 \tag{2.1}$$

using the finite volume formulation. Computational control volumes are tetrahedrons.

Taking the tetrahedron $\triangle_i$ as our control volume, we formulate the semi-discrete finite volume scheme of (2.1) as

$$\frac{d\bar{u}_i(t)}{dt}+\frac{1}{|\triangle_i|}\int_{\partial\triangle_i}F\cdot ndS=0, \tag{2.2}$$

where the cell average

$$\bar{u}_i(t)=\frac{1}{|\triangle_i|}\int_{\triangle_i}udxdydz, \quad F=(f,g,h)^T,$$

and n is the outward unit normal of the tetrahedron boundary $\partial\triangle_i$.

In (2.2), the integral on the four triangular surfaces of $\partial\triangle_i$ is discretized by a q-point Gaussian quadrature formula,

$$\int_{\partial\triangle_i}F\cdot nds=\sum_{k=1}^{4}S_k\sum_{j=1}^{q}w_jF(u(G_j^{(k)},t))\cdot n_k, \tag{2.3}$$

where S_k is the area of the k-th triangle of $\partial\triangle_i$, $G_j^{(k)}$ and w_j are the Gaussian quadrature points and weights respectively, and $F(u(G_j^{(k)},t))\cdot n_k$ is approximated by a numerical flux. We use the simple Lax-Friedrichs flux in this paper, which is given by

$$F(u(G_j^{(k)},t))\cdot n_k\approx\frac{1}{2}\Big[\Big(F(u^-(G_j^{(k)},t))+F(u^+(G_j^{(k)},t))\Big)\cdot n_k$$
$$-\alpha\Big(u^+(G_j^{(k)},t)-u^-(G_j^{(k)},t)\Big)\Big], \tag{2.4}$$

where α is taken as an upper bound for the magnitude of the eigenvalues of the Jacobian in the n direction, and u^- and u^+ are the values of u inside the tetrahedron and outside the tetrahedron (inside the neighboring tetrahedron) at the Gaussian point.

In this paper we only discuss the construction of a third order finite volume scheme, so the four-point Gaussian quadrature $q=4$ is used. We adopt the Gaussian quadrature in [6]. For the triangle with vertexes P_1, P_2 and P_3, the Gaussian quadrature points are

$$G_1=\lambda_1P_1+\lambda_2P_2+\lambda_3P_3, \quad G_2=\lambda_2P_1+\lambda_1P_2+\lambda_3P_3,$$
$$G_3=\beta_1P_1+\beta_2P_2+\beta_3P_3, \quad G_4=\beta_2P_1+\beta_1P_2+\beta_3P_3,$$

where

$$\lambda_1=\frac{6-\sqrt{6}}{10}\left(\frac{1}{2}-\frac{\sqrt{3}}{6}\right), \qquad \lambda_2=\frac{6-\sqrt{6}}{10}\left(\frac{1}{2}+\frac{\sqrt{3}}{6}\right), \qquad \lambda_3=\frac{4+\sqrt{6}}{10},$$

$$\beta_1=\frac{6+\sqrt{6}}{10}\left(\frac{1}{2}-\frac{\sqrt{3}}{6}\right), \qquad \beta_2=\frac{6+\sqrt{6}}{10}\left(\frac{1}{2}+\frac{\sqrt{3}}{6}\right), \qquad \beta_3=\frac{4-\sqrt{6}}{10};$$

and the Gaussian quadrature weights are

$$w_1=w_2=\frac{9-\sqrt{6}}{36}, \qquad w_3=w_4=\frac{9+\sqrt{6}}{36}.$$

3   Third order WENO reconstruction and WENO finite volume scheme

In this section we describe the third order WENO algorithm on three dimensional tetrahedral meshes.

3.1   The big stencil

To build a third-order reconstruction for the point values at the Gaussian quadrature points, we would like to construct a quadratic polynomial $p(x,y,z)$ that has the same cell average as u on the target tetrahedron $\triangle_0$:

$$\bar{u}_0 \equiv \frac{1}{|\triangle_0|} \int_{|\triangle_0|} u(x,y,z)dxdydz. \tag{3.1}$$

Here we have suppressed the time variable t as the reconstruction is performed at a fixed time level. Assuming that $\triangle_0$ has the barycenter (x_0,y_0,z_0), we define the local variables $\xi = (x-x_0)/h$, $\eta = (y-y_0)/h$ and $\zeta = (z-z_0)/h$, where $h = |\triangle_0|^{1/3}$. The quadratic reconstruction polynomial is presented as

$$p(x,y,z) = a_0 + a_1\xi + a_2\eta + a_3\zeta + a_4\xi^2 + a_5\eta^2 + a_6\zeta^2 + a_7\xi\eta + a_8\xi\zeta + a_9\eta\zeta.$$

A typical big stencil S includes $\triangle_0$, its four neighbors $\triangle_1,\triangle_2,\triangle_3,\triangle_4$ and the neighbors of these four neighbors (two layers of neighbors):

$$S = \{\triangle_0;\triangle_1,\triangle_2,\triangle_3,\triangle_4;\triangle_{11},\triangle_{12},\triangle_{13};\triangle_{21},\triangle_{22},\triangle_{23};\triangle_{31},\triangle_{32},\triangle_{33};\triangle_{41},\triangle_{42},\triangle_{43}\},$$

where the subscripts are self-evident, for example, $\triangle_{11},\triangle_{12},\triangle_{13}$ are the three neighboring tetrahedrons of $\triangle_1$ other than $\triangle_0$.

We determine the quadratic polynomial $p(x,y,z)$ by requiring that it has the same cell average as u on $\triangle_0$, and also it matches the cell averages of u on the tetrahedrons of the big stencil S except $\triangle_0$ in a least-square sense [1]. Notice that some of the neighbors of the four neighbors may coincide, but this does not affect the least-square procedure to determine $p(x,y,z)$. Let m denote the total number of cells in the big stencil S. If $m < 10$, we must go to the next neighboring layer and include more cells into the big stencil S to provide enough information for constructing a quadratic polynomial.

For every quadrature point (x^G,y^G,z^G), we compute a series of constants $\{c_l\}_{l=1}^m$ which depend on the local geometry only, such that

$$p(x^G,y^G,z^G) = \sum_{l=1}^m c_l\bar{u}_l, \tag{3.2}$$

where every constant c_l corresponds to one cell in the big stencil S, and $\bar{u}_l$ is the cell average of u on that cell.

Table 1: Small stencil candidates.

polynomial	stencil members	polynomial	stencil members
p_1	$\triangle_0, \triangle_1, \triangle_2, \triangle_3$	p_2	$\triangle_0, \triangle_1, \triangle_2, \triangle_4$
p_3	$\triangle_0, \triangle_1, \triangle_3, \triangle_4$	p_4	$\triangle_0, \triangle_2, \triangle_3, \triangle_4$
p_5	$\triangle_0, \triangle_1, \triangle_{11}, \triangle_{12}$	p_6	$\triangle_0, \triangle_1, \triangle_{11}, \triangle_{13}$
p_7	$\triangle_0, \triangle_1, \triangle_{12}, \triangle_{13}$	p_8	$\triangle_0, \triangle_2, \triangle_{21}, \triangle_{22}$
p_9	$\triangle_0, \triangle_2, \triangle_{21}, \triangle_{23}$	p_{10}	$\triangle_0, \triangle_2, \triangle_{22}, \triangle_{23}$
p_{11}	$\triangle_0, \triangle_3, \triangle_{31}, \triangle_{32}$	p_{12}	$\triangle_0, \triangle_3, \triangle_{31}, \triangle_{33}$
p_{13}	$\triangle_0, \triangle_3, \triangle_{32}, \triangle_{33}$	p_{14}	$\triangle_0, \triangle_4, \triangle_{41}, \triangle_{42}$
p_{15}	$\triangle_0, \triangle_4, \triangle_{41}, \triangle_{43}$	p_{16}	$\triangle_0, \triangle_4, \triangle_{42}, \triangle_{43}$

3.2   The small stencils

The key step in building a high order WENO scheme (the second type as explained in
section 1) is to construct lower order polynomials whose weighted average will give the
same result as the high order reconstruction at each quadrature point (the weights are
different for different quadrature points). We will build several linear polynomials

$$p_s(x,y,z) = a_0^{(s)} + a_1^{(s)}\xi + a_2^{(s)}\eta + a_3^{(s)}\zeta$$

to give the lower order reconstructions.

Using the big stencil S, we have 16 small stencil candidates and the corresponding 16
linear polynomials by agreeing with the cell averages of u on these small stencils. They
are listed in Table 1.

Notice that some candidates may coincide, so we check the small stencil candidates
and merge the same ones. Let q denote the total number of small stencils for the target
cell $\triangle_0$, and $\{S^{(i)}\}_{i=1}^q$ the small stencils.

For every quadrature point (x^G, y^G, z^G), on every small stencil $S^{(i)}$ we compute four
constants $\{c_l^{(i)}\}_{l=1}^4$ which depend on the local geometry only, such that

$$p_i(x^G, y^G, z^G) = \sum_{l=1}^4 c_l^{(i)} \bar{u}_l, \tag{3.3}$$

where every constant $c_l^{(i)}$ corresponds to one cell in the small stencil $S^{(i)}$, and $\bar{u}_l$ is the cell
average of u on that cell.

The quadratic polynomial $p(x,y,z)$ has six more degrees of freedom than each lin-
ear polynomial $p_s(x,y,z)$, namely $\xi^2, \eta^2, \zeta^2, \xi\eta, \xi\zeta, \eta\zeta$. For the degrees of freedom $1, \xi, \eta, \zeta$,
both the quadratic polynomial reconstruction and linear polynomial reconstruction can
reproduce them exactly. According to the argument in [7], we need the number of small
stencils $q \geq 7$. This can be easily achieved for most triangulations by just including the first
two layers of neighbors of $\triangle_0$. For the triangulations in our computation, we have not

met the situation that $q < 7$. We will assume that $q \geq 7$ in the following discussion. Even if $q < 7$, for example for a very distorted mesh, we can always go to the next neighboring layer and include more candidates of small stencils.

To obtain the linear weights $\{\gamma_i\}_{i=1}^q$, we form the linear system at every Gaussian quadrature point (x^G, y^G, z^G): take $u = \xi^2, \eta^2, \zeta^2, \xi\eta, \xi\zeta, \eta\zeta$ respectively, the equalities are:

$$\sum_{i=1}^q \gamma_i p_i(x^G, y^G, z^G) = u(x^G, y^G, z^G), \tag{3.4}$$

where p_i is the linear reconstruction polynomial for u, using the small stencil $S^{(i)}$. Together with the requirement

$$\sum_{i=1}^q \gamma_i = 1, \tag{3.5}$$

we obtain a $7 \times q$ linear system

$$A\gamma = b, \tag{3.6}$$

where $A \in R^{7 \times q}$, and $b \in R^7$. For $q > 7$, this is a under-determined system and there are infinitely many solutions. We define the optimal linear weights $\{\gamma_i\}_{i=1}^q$ as following. The third order reconstruction by the linear combination of the second order reconstructions using the optimal linear weights $\{\gamma_i\}_{i=1}^q$ is the "closest" one from the third order reconstruction by the big stencil S, in the least square sense. We form the linear system

$$M\gamma \overset{1}{=} c, \tag{3.7}$$

where $\overset{1}{=}$ means that the equality holds in the least square sense, and $M \in R^{m \times q}$, $c \in R^m$. The vector $c = (c_1, c_2, \cdots, c_m)^T$, and $\{c_l\}_{l=1}^m$ are the approximation constants in (3.2) for the big stencil. Each column of the matrix M corresponds to the approximation constants in (3.3) for one of the small stencils. The systems (3.6) and (3.7) are solved together to give the optimal linear weights $\{\gamma_i\}_{i=1}^q$.

3.3 Non-linear weights and the WENO scheme

In this section, we construct the WENO scheme based on non-linear weights. In order to compute the non-linear weights, we need to compute the smoothness indicators first.

For every reconstruction polynomial $p_i(x,y,z)$ defined on the target cell $\triangle_0$ with degree up to k, we take the smoothness indicator β_i as:

$$\beta_i = \sum_{1 \leq |\alpha| \leq k} \int_{\triangle_0} |\triangle_0|^{\frac{2}{3}|\alpha|-1} (D^\alpha p_i(x,y,z))^2 dxdydz, \tag{3.8}$$

where α is a multi-index and D is the derivative operator. The smoothness indicator measures how smooth the function p_i is on the triangle $\triangle_0$: the smaller the smoothness

indicator, the smoother the function p_i is on $\triangle_0$. The scaling factor in front of the derivatives renders the smoothness indicator self-similar and invariant under uniform scaling of the mesh in all directions. Since we are constructing third order WENO schemes and $\{p_i\}_{i=1}^q$ are linear polynomials, $k=1$ in our scheme.

Now we define the non-linear weights as:

$$\omega_i = \frac{\widetilde{\omega}_i}{\sum_j \widetilde{\omega}_j}, \qquad \widetilde{\omega}_i = \frac{\gamma_i}{(\varepsilon + \beta_i)^2}, \tag{3.9}$$

where γ_i is the i-th linear weight determined in section 3.2, β_i is the smoothness indicator for the i-th reconstruction polynomial $p_i(x,y,z)$ associated with the i-th small stencil, and ε is a small positive number to avoid the denominator to become 0. We take $\varepsilon = 10^{-3}$ for all the computations in this paper.

Finally, we form the finite volume WENO schemes using the WENO reconstructions of the numerical values at quadrature points in (2.4):

$$u(x^G, y^G, z^G) = \sum_{i=1}^q w_i p_i(x^G, y^G, z^G). \tag{3.10}$$

If we use the linear weights in the reconstruction,

$$u(x^G, y^G, z^G) = \sum_{i=1}^q \gamma_i p_i(x^G, y^G, z^G), \tag{3.11}$$

we would obtain the linear scheme.

The linear weights $\{\gamma_i\}_{i=1}^q$ depend on the local geometry of the mesh and can be negative. If $\min(\gamma_1, \cdots, \gamma_q) < 0$, we adopt the splitting technique of treating negative weights in WENO schemes developed by Shi, Hu and Shu [12] as following. First we split the linear weights into two groups:

$$\widetilde{\gamma}_i^+ = \frac{1}{2}(\gamma_i + 3|\gamma_i|), \qquad \widetilde{\gamma}_i^- = \widetilde{\gamma}_i^+ - \gamma_i, \qquad i=1,\cdots,q, \tag{3.12}$$

then we scale them by

$$\sigma^\pm = \sum_{l=1}^q \widetilde{\gamma}_l^\pm; \qquad \gamma_i^\pm = \widetilde{\gamma}_i^\pm / \sigma^\pm, \qquad i=1,\cdots,q. \tag{3.13}$$

Now we compute the nonlinear weights (3.9) for the positive and negative groups $\gamma_i^\pm$ separately, denoted by $\omega_i^\pm$, based on the same smoothness indicator β_i. Then we compute the WENO reconstructions $u^\pm(x^G, y^G, z^G)$ separately by (3.10), using $\omega_i^\pm$, and form the final WENO reconstruction by

$$u(x^G, y^G, z^G) = \sigma^+ u^+(x^G, y^G, z^G) - \sigma^- u^-(x^G, y^G, z^G). \tag{3.14}$$
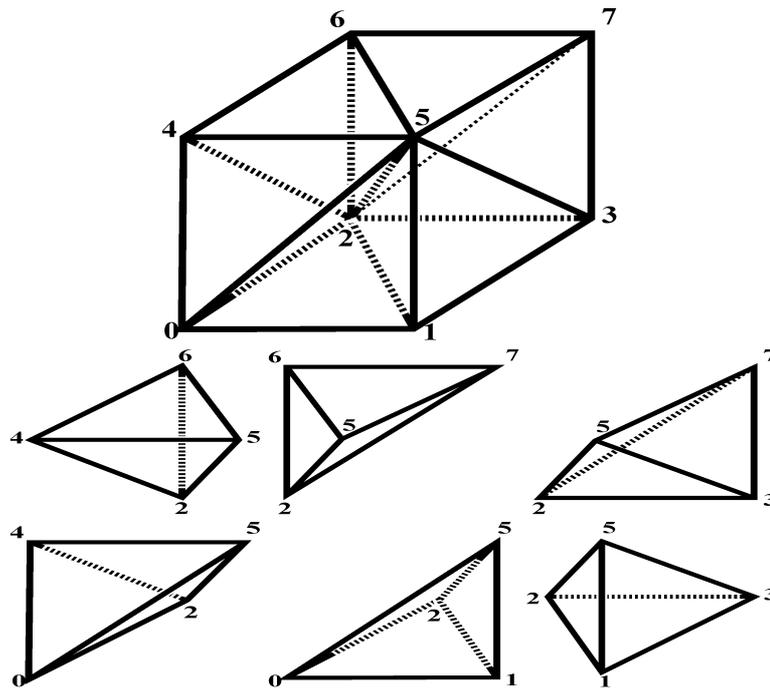
Figure 1: Construction of a uniform tetrahedron mesh.

The key idea of this decomposition is to make sure that every stencil has a significant representation in both the positive and the negative weight groups. Within each group, the WENO idea of redistributing the weights subject to a fixed sum according to the smoothness of the approximation is still followed as before. See [12] for more details.

To improve the accuracy of the WENO scheme, we adopt a mapped weights technique introduced by Henrick, Aslam and Powers [5], see also its application for solving steady state problems in [14]. For every linear weight γ_i, $i=1,\cdots,q$, the mapping function is defined as

$$g_i(\omega;\gamma_i) = \frac{\omega(\gamma_i + \gamma_i^2 - 3\gamma_i\omega + \omega^2)}{\gamma_i^2 + (1-2\gamma_i)\omega}, \tag{3.15}$$

where $\omega \in [0,1]$. The non-linear weights ω_i computed by (3.9) are replaced by the mapped weights

$$\omega_i^M = g_i(\omega_i;\gamma_i). \tag{3.16}$$

Likewise, the non-linear weights $\omega_i^{\pm}$ are replaced by the mapped weights

$$\omega_i^{\pm,M} = g_i(\omega_i^{\pm};\gamma_i^{\pm}). \tag{3.17}$$

Table 2: Accuracy for 3D linear equation. Third order linear and WENO schemes. Uniform tetrahedron mesh. $T=1$.

# of cells	Linear scheme				WENO scheme			
	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
6000	2.27E-02	—	4.36E-02	—	1.03E-01	—	2.73E-01	—
48000	2.84E-03	2.99	5.28E-03	3.05	1.50E-02	2.78	4.74E-02	2.53
384000	3.54E-04	3.01	6.40E-04	3.04	4.68E-04	5.00	2.35E-03	4.34

Table 3: Accuracy for 3D linear equation. Third order linear and WENO schemes. Computational mesh is obtained by randomly perturbing the uniform tetrahedron mesh within 4% in the 3D space. $T=1$.

# of cells	Linear scheme				WENO scheme			
	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
6000	2.29E-02	—	4.63E-02	—	1.03E-01	—	2.80E-01	—
48000	2.87E-03	3.00	5.52E-03	3.07	1.48E-02	2.80	4.78E-02	2.55
384000	3.59E-04	3.00	6.81E-04	3.02	5.03E-04	4.88	2.63E-03	4.19

4 Numerical examples

In this section, we apply the WENO scheme developed in the previous section to both linear and non-linear three dimensional problems. The CFL number is taken as 0.6 in all the cases. For the temporal discretization, we use the third-order TVD Runge-Kutta scheme of Shu and Osher in [13].

We use the uniform tetrahedral meshes and random perturbations of the uniform meshes in the computations of this paper. The uniform meshes are generated by cutting each cube of a rectangular mesh into six tetrahedrons, as shown in Fig. 1.

Example 4.1. Linear equation:

$$\begin{cases} u_t+u_x+u_y+u_z=0, & -2\leq x\leq 2, -2\leq y\leq 2, -2\leq z\leq 2; \\ u(x,y,z,0)=\sin(\frac{\pi}{2}(x+y+z)), \end{cases} \quad (4.1)$$

with periodic boundary condition. We use both the third order linear and WENO schemes to solve the PDE (4.1) to $T=1$, on both the uniform meshes and the perturbed meshes. From Table 2 and Table 3, we can observe third order accuracy in L^1 and L^∞ errors for both the linear and WENO schemes. Notice that the numerical order of accuracy for the WENO scheme has not settled down to the expected order yet for the finest meshes in Table 2 and Table 3. It is typical for WENO schemes to "catch up" with the error magnitude of linear schemes with refined meshes. In this process the numerical order of accuracy for the WENO schemes can be higher than expected, due to the larger gap in the magnitude of the errors between the WENO schemes and the corresponding linear schemes on coarser meshes. We have not computed the results in Tables 2 and 3 using more refined

Table 4: Accuracy for 3D Burgers equation. Third order linear and WENO schemes. Uniform tetrahedron mesh. $T = 0.5/\pi^2$.

# of cells	Linear scheme				WENO scheme			
	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order
6000	3.91E-03	—	1.87E-02	—	1.20E-02	—	8.94E-02	—
48000	4.81E-04	3.02	2.60E-03	2.85	1.30E-03	3.20	1.32E-02	2.75
384000	5.96E-05	3.01	3.31E-04	2.97	7.11E-05	4.20	3.31E-04	5.32

Table 5: Accuracy for 3D Burgers equation. Third order linear and WENO schemes. Computational mesh is obtained by randomly perturbing the uniform tetrahedron mesh within 4% in the 3D space. $T = 0.5/\pi^2$.

# of cells	Linear scheme				WENO scheme				
	L^1 error	order	L^∞ error	order	L^1 error	order	L^∞ error	order	CPU(s)
6000	3.92E-03	—	1.99E-02	—	1.20E-02	—	9.54E-02	—	1.18
48000	4.82E-04	3.02	2.79E-03	2.84	1.31E-03	3.20	1.61E-02	2.57	14.51
384000	5.99E-05	3.01	3.80E-04	2.88	7.13E-05	4.20	4.00E-04	5.33	188.69

meshes because of a limitation of the computer resource. We plan to implement the code on parallel machines and run it on more refined meshes in the future.

Example 4.2. Nonlinear Burgers equation:

$$\begin{cases} u_t + (\frac{u^2}{2})_x + (\frac{u^2}{2})_y + (\frac{u^2}{2})_z = 0, & -3 \le x \le 3, -3 \le y \le 3, -3 \le z \le 3; \\ u(x,y,z,0) = 0.3 + 0.7\sin(\frac{\pi}{3}(x+y+z)), \end{cases} \qquad (4.2)$$

with periodic boundary condition. First we use both the third order linear and WENO schemes to solve the PDE (4.2) to $T = 0.5/\pi^2$, when the solution is still smooth. From Table 4 and Table 5, we can observe third order accuracy for both the linear and WENO schemes, on the uniform meshes and the perturbed meshes respectively. Again we note that the numerical order of accuracy for the WENO scheme has not settled down to the expected order yet for the finest meshes in Tables 4 and 5, while the gap of the magnitude of errors between the WENO scheme and the corresponding linear scheme narrows during mesh refinement, as expected. The total computational time needed for the time evolution (excluding pre-processing stages for preparing geometry constants) in our implementation on a single processor PC is listed for the WENO scheme in Table 5. Then we solve the PDE to $T = 5/\pi^2$ by the third order WENO scheme, using 384000 tetrahedrons, and present the contour plots on the whole 3D domain surface (top), the 2D plane $z = 0$ (middle), and the 1D cutting-plot along the line $x = y$, $z = 0$ (bottom) in Fig. 2. We can observe that the solution is non-oscillatory and the shock is resolved sharply.
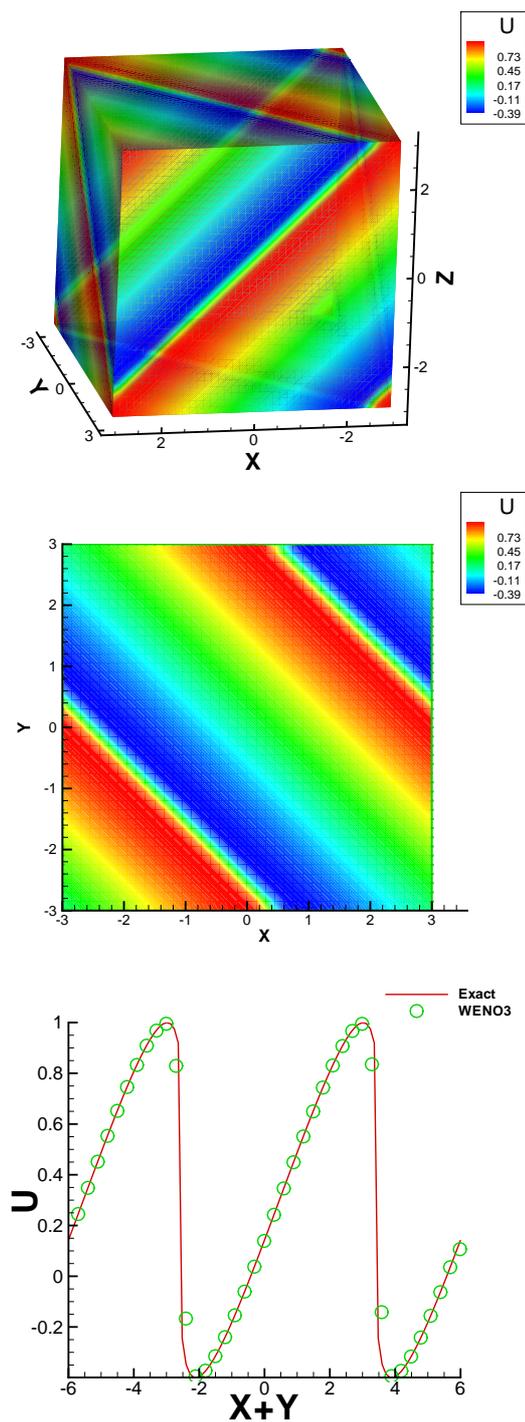
Figure 2: Solution of three-dimensional Burgers equation by the third order WENO scheme, $t=5/\pi^2$. Top: contour plot on the surface; middle: contour plot on the cut $z=0$ plane; bottom: 1D cutting-plot along $x=y$, $z=0$ with circles representing the numerical solution and the line representing the exact solution.

5   Concluding remarks

We have constructed a third order finite volume WENO scheme on three dimensional tetrahedral meshes. The scheme relies on writing a third order quadratic polynomial reconstruction as a linear combination of several second order linear polynomial reconstructions based on various small stencils, with the combination coefficients obtained from linear weights for accuracy and nonlinear weights determined by smoothness indicators. Numerical tests are performed on a uniform tetrahedral mesh as well as on meshes obtained by random perturbations of the uniform mesh. Further numerical experiments on more general tetrahedral meshes, for three dimensional hyperbolic systems, and for higher order finite volume WENO schemes, will be performed in the future.

Acknowledgments

References

[1] T. Barth and P. Frederickson, High order solution of the Euler equations on unstructured grids using quadratic reconstruction, AIAA Paper No. 90-0013.
[2] M. Dumbser and M. Käser, Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems, J. Comput. Phys., 221 (2007), 693-723.
[3] M. Dumbser, M. Käser, V.A. Titarev and E.F. Toro, Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems, J. Comput. Phys., 226 (2007), 204-243.
[4] O. Friedrichs, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, J. Comput. Phys., 144 (1998), 194-212.
[5] A.K. Henrick, T.D. Aslam and J.M. Powers, Mapped weighted essentially non-oscillatory schemes: achieving optimal order near critical points, J. Comput. Phys., 207 (2005), 542-567.
[6] P. Hillion, Numerical integration on a triangle, Int. J. Numer. Met. Eng., 11 (1977), 797-815.
[7] C. Hu and C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys., 150 (1999), 97-127.
[8] G. Jiang and C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys., 126 (1996), 202-228.
[9] X.-D. Liu, S. Osher and T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys., 115 (1994), 200-212.
[10] J. Qiu and C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, SIAM J. Sci. Comput., 26 (2005), 907-929.
[11] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two dimensional case, Computers and Fluids, 34 (2005), 642-663.
[12] J. Shi, C. Hu and C.-W. Shu, A technique of treating negative weights in WENO schemes, J. Comput. Phys., 175 (2002), 108-127.

[13] C.-W. Shu and S. Osher, Efficient implementation of essentially non-oscillatory shock cap-
     turing schemes, J..Comput. Phys., 77 (1988), 439-471.
[14] S. Zhang and C.-W. Shu, A new smoothness indicator for the WENO schemes and its effect
     on the convergence to steady state solutions, J. Sci. Comput., 31 (2007), 273-305.
[15] Y.-T. Zhang and C.-W. Shu, High order WENO schemes for Hamilton-Jacobi equations on
     triangular meshes, SIAM J. Sci. Comput., 24 (2003), 1005-1030.