Journal of Computational Physics 228 (2009) 5787–5802

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Hierarchical reconstruction for spectral volume method on unstructured grids

Zhiliang Xu [a,*], Yingjie Liu [b,1], Chi-Wang Shu [c,2]

[a] Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556, United States
[b] School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332, United States
[c] Division of Applied Mathematics, Brown University, Providence, RI 02912, United States

ARTICLE INFO

Article history:
Received 30 November 2008
Received in revised form 9 April 2009
Accepted 1 May 2009
Available online 7 May 2009

Keywords:
Hierarchical reconstruction
Spectral volume method
Unstructured grids
Hyperbolic conservation laws

ABSTRACT

The hierarchical reconstruction (HR) [Y.-J. Liu, C.-W. Shu, E. Tadmor, M.-P. Zhang, Central discontinuous Galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction, SIAM J. Numer. Anal. 45 (2007) 2442–2467; Z.-L. Xu, Y.-J. Liu, C.-W. Shu, Hierarchical reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO type linear reconstruction and partial neighboring cells, J. Comput. Phys. 228 (2009) 2194–2212] is applied to a piecewise quadratic spectral volume method on two-dimensional unstructured grids as a limiting procedure to prevent spurious oscillations in numerical solutions. The key features of this HR are that the reconstruction on each control volume only uses adjacent control volumes, which forms a compact stencil set, and there is no truncation of higher degree terms of the polynomial. We explore a WENO-type linear reconstruction on each hierarchical level for the reconstruction of high degree polynomials. Numerical computations for scalar and system of nonlinear hyperbolic equations are performed. We demonstrate that the hierarchical reconstruction can generate essentially non-oscillatory solutions while keeping the resolution and desired order of accuracy for smooth solutions.

1. Introduction

In a series of papers [21–23], Wang et al. introduced a framework of the spectral volume method to solve nonlinear time dependent hyperbolic conservation laws

$$\begin{cases} \frac{\partial u_k}{\partial t} + \nabla \cdot \mathbf{F}_k(\mathbf{u}) = 0, \quad k = 1, \ldots, m, \text{ in } \Omega \times (0, T), \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \end{cases} \tag{1.1}$$

where $\Omega \subset R^d$, $\mathbf{x} = (x_1, \ldots, x_d)$, d is the dimension, $\mathbf{u} = (u_1, \ldots, u_m)^T$ and $\mathbf{F}_k(\mathbf{u}) = (F_{k,1}(\mathbf{u}), \ldots, F_{k,d}(\mathbf{u}))$ is the flux. The spectral volume method evolves the cell average of the solution in each control volume (CV) as does the traditional finite volume method. However, instead of performing a reconstruction procedure to obtain a high degree polynomial in each control volume, the spectral volume method groups certain numbers of control volumes to form a spectral volume (SV), so that a reconstruction procedure can be performed using all cell averages of the solution on control volumes within the spectral volume to obtain the polynomial. The reconstructed high degree polynomial in the spectral volume is utilized within all its control

[*] Corresponding author. Tel.: +1 574 631 3423.
E-mail addresses: zxu2@nd.edu (Z. Xu), yingjie@math.gatech.edu (Y. Liu), shu@dam.brown.edu (C.-W. Shu).
[1] Research supported in part by NSF Grant DMS-0810913.
[2] Research supported in part by DOE Grant DE-FG02-08ER25863 and NSF Grant DMS-0809086.

0021-9991/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2009.05.001

volumes. This technique greatly reduces the complexity of the reconstruction procedure for smooth solutions compared to traditional finite volume methods defined on control volumes.

Comparisons between the SV and discontinuous Galerkin (DG) methods can be found in papers by Sun and Wang [20] and Zhang and Shu [26]. Both methods are high-order accurate on unstructured grids. Also, they both use compact stencils, requiring only immediate cell (SV) neighbors to update the solution in a cell (SV).

Since the hyperbolic conservation laws may develop discontinuous solutions, in [22,23], the explicit, total variation diminishing (TVD) Runge–Kutta time discretizations [17] are used in time; exact or approximate Riemann solvers are used to compute the fluxes and the TVB projection limiters [22,23] are used to prevent or restrict the spurious oscillations near discontinuities. In particular, if the solution value on a quadrature point of a CV lies beyond the maximum and minimum of certain cell (CV) average values adjacent to the CV, the polynomial solution on the CV is truncated to a linear polynomial. This linear polynomial is reconstructed by the Super-bee or Minmod limiter using neighboring cell (CV) average values.

A goal of the present paper is to develop a new limiting technique involving only adjacent SV neighbors without the need to truncate the degree of polynomials in non-smooth regions. To achieve this, we explore a non-oscillatory hierarchical reconstruction (HR) method developed in [25] to reconstruct the polynomial computed by the spectral volume method. We consider the case of piecewise quadratic polynomial (P^2) in this paper.

The framework of the HR method was introduced by Liu et al. [12] for the staggered central DG method, in which cell averages of various orders of derivatives of a polynomial are calculated and used in the reconstruction of non-oscillatory linear polynomials on each hierarchical stage. The coefficients of reconstructed linear polynomials are used to update the corresponding ones of the original polynomial. In [25], the authors extended the HR method to DG on unstructured triangular meshes and developed a weighted linear reconstruction method for each hierarchical stage.

In this paper, we demonstrate that HR can generate essentially non-oscillatory solutions while keeping the resolution and desired order of accuracy for smooth solutions. We also notice that from the results of accuracy tests (see Section 3), HR seems to reduce the magnitude of the error as well.

This paper is organized as follows. Section 2 describes the spectral volume method and the limiting procedure. Numerical tests are presented in Section 3. Concluding remarks and a plan for future work are included in Section 4.

2. Algorithm formulation

We use the method of lines approach to evolve the solution on the triangulated domain. The P^2 spectral volume method is used to compute the piecewise polynomial solution at each time level followed by the hierarchical reconstruction to remove spurious oscillations near discontinuities of the solution.

2.1. Spectral volume method

2.1.1. Spatial discretization

The physical domain $\Omega \subset R^2$ is partitioned into a collection of $\mathcal{N}$ triangular elements $\Omega = \cup_{i=1}^{\mathcal{N}} \mathcal{K}_i$ and

$$\mathcal{T}_h = \{\mathcal{K}_i : i = 1, \ldots, \mathcal{N}\}. \tag{2.1}$$

Each element $\mathcal{K}_i$ is called a spectral volume, which is further partitioned into sub-elements called control volumes, denoted by $\mathcal{C}_{ij}$. Fig. 1 shows a 2D SV and its partition used in all test cases for a third order accurate SV scheme, where the SV contains six CVs.

To represent the solution as a polynomial of degree q on a SV, we need $N = (q+1)(q+2)/2$ degrees of freedom (DOFs). The number of the DOFs matches the number of CVs in the SV.

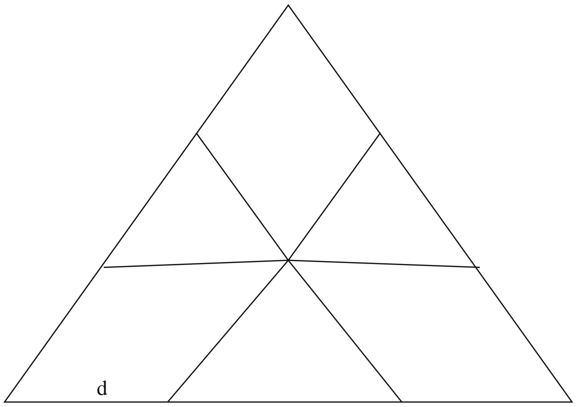


Fig. 1. The partition of a triangular spectral volume into six control volumes supporting a quadratic polynomial, $d = 1/3$ (assuming that the length of each triangle edge is 1).

We choose $\mathcal{K}_i$, the polynomial basis functions of degree q in a SV, to be monomials of the multidimensional Taylor expansion about its centroid. For a general triangular SV $\mathcal{K}_i$, the basis set in the (x, y) coordinate is

$$\mathcal{B} = \{g_m(x - x_i, y - y_i) : m = 1, \ldots, N\} = \{1, x - x_i, y - y_i, (x - x_i)^2, (x - x_i)(y - y_i), (y - y_i)^2, \ldots, (y - y_i)^q\}, \tag{2.2}$$

where $\mathbf{x}_i \equiv (x_i, y_i)$ is the centroid of $\mathcal{K}_i$, and $N = (q + 1)(q + 2)/2$.

Let the piecewise polynomial solution (polynomial in each SV) for the kth equation of (1.1) be u_h (neglecting its subscript k for convenience), expressed as

$$u_h(\mathbf{x}, t) = \sum_{m=1}^{N} u_{m,i}(t) g_m(x - x_i, y - y_i), \quad \mathbf{x} = (x, y) \in \mathcal{K}_i, \quad i = 1, \ldots, \mathcal{N}. \tag{2.3}$$

For convenience, we sometimes write $u_{m,i}(t)$ as $u_m(t)$ when there is no confusion.

Denote the CV-averaged state for $\mathcal{C}_{ij}$ as

$$\bar{u}_{ij} = (1/|\mathcal{C}_{ij}|) \int_{\mathcal{C}_{ij}} u_h \, d\mathbf{x}, \tag{2.4}$$

where $|\mathcal{C}_{ij}|$ is the area of $\mathcal{C}_{ij}$. The semi-discrete finite volume scheme on $\mathcal{C}_{ij}$ can be written as

$$\frac{d}{dt}\bar{u}_{ij} + \frac{1}{|\mathcal{C}_{ij}|} \int_{\partial \mathcal{C}_{ij}} \mathbf{F}_k(\mathbf{u}_h) \cdot \mathbf{n}_{ij} \, d\Gamma = 0, \quad j = 1, \ldots, N, \tag{2.5}$$

where $\mathbf{n}_{ij}$ is the outer unit normal vector of $\mathcal{C}_{ij}$, $i = 1, \ldots, \mathcal{N}$. At any time when $\{\bar{u}_{ij} : j = 1, \ldots, N\}$ have been solved, $\{u_{m,i}(t) : m = 1, \ldots, N\}$ can be recovered from Eqs. (2.4) and (2.3).

Since the approximated solution u_h could be discontinuous across CV edges after a limiting procedure is applied so that polynomials in some CVs are reconstructed, the flux function $\mathbf{F}_k(\mathbf{u}_h) \cdot \mathbf{n}_{ij}$ appearing in Eq. (2.5) is not uniquely determined and is replaced by a numerical flux function (the Lax–Friedrich flux, see e.g. [16]) defined by

$$h_k(\mathbf{x}, t) = h_k(\mathbf{u}_h^{in}, \mathbf{u}_h^{out}) = \frac{1}{2}\left(\mathbf{F}_k(\mathbf{u}_h^{in}) \cdot \mathbf{n}_{ij} + \mathbf{F}_k(\mathbf{u}_h^{out}) \cdot \mathbf{n}_{ij}\right) + \frac{\alpha}{2}\left(u_h^{in} - u_h^{out}\right),$$

where α is the largest characteristic speed,

$$\mathbf{u}_h^{in}(\mathbf{x}, t) = \lim_{\mathbf{y} \to \mathbf{x}, \mathbf{y} \in \mathcal{C}_{ij}^{\text{int}}} \mathbf{u}_h(\mathbf{y}, t),$$
$$\mathbf{u}_h^{out}(\mathbf{x}, t) = \lim_{\mathbf{y} \to \mathbf{x}, \mathbf{y} \notin \bar{\mathcal{C}}_{ij}} \mathbf{u}_h(\mathbf{y}, t).$$

The boundary integrals in Eq. (2.5) are computed with the $(q + 1)$th (or higher) order accurate Gaussian quadrature rule to preserve the $(q + 1)$th order of accuracy of the spectral volume discretization. For a third order accurate scheme, the two-point Gaussian quadrature rule is used for the boundary integral.

2.1.2. Time integration

Eq. (2.5) is integrated in time using the widely used three stage (third order) TVD Runge–Kutta method [17].

2.2. Limiting by hierarchical reconstruction

Without an appropriate limiting procedure, the spectral volume method will produce non-physical oscillations in the vicinity of discontinuities. We use the hierarchical reconstruction introduced in [12] to process the spectral volume solution at each Runge–Kutta stage in order to eliminate such spurious oscillations. We refer to Refs. [12,13] for the implementation of HR for central and finite volume schemes, and Ref. [25] for the implementation of HR for the DG method on unstructured meshes.

2.2.1. General description of the hierarchical reconstruction

We briefly describe the general procedure of HR here. The goal of HR is to reconstruct the polynomial $u_j(\mathbf{x} - \mathbf{x}_j)$ of degree q supported on an element $\mathcal{V}_j$ with centroid $\mathbf{x}_j$ by using polynomials in elements adjacent to $\mathcal{V}_j$. HR recomputes the coefficients of $u_j(\mathbf{x} - \mathbf{x}_j)$ (in Taylor expansion around $\mathbf{x}_j$), denoted by (the new coefficients)

$$\frac{1}{m!}\tilde{u}_j^{(\mathbf{m})}(0), \quad |\mathbf{m}| = m, \quad m = q, q - 1, \ldots, 0,$$

iteratively from the highest to the lowest degree terms. We denote these adjacent elements (including element $\mathcal{V}_j$) as the set $\{\mathcal{V}_J\}$ and polynomials supported on these elements as the set $\{u_J(\mathbf{x} - \mathbf{x}_J)\}$. HR proceeds as follows:

Step 1. Suppose $q \geqslant 2$. For stage $m = q, q - 1, \ldots, 2$, do the following:

(a) Take a $(m - 1)$th order partial derivative for every $u_J(\mathbf{x} - \mathbf{x}_J)$ to obtain the polynomial $\partial^{m-1}u_J(\mathbf{x} - \mathbf{x}_J)$, respectively. In particular, denote $\partial^{m-1}u_j(\mathbf{x} - \mathbf{x}_j) = L_j(\mathbf{x} - \mathbf{x}_j) + R_j(\mathbf{x} - \mathbf{x}_j)$, where $L_j(\mathbf{x} - \mathbf{x}_j)$ is the linear part of $\partial^{m-1}u_j(\mathbf{x} - \mathbf{x}_j)$ and $R_j(\mathbf{x} - \mathbf{x}_j)$ is the remainder.

(b) Calculate the average value of every $\partial^{m-1}u_J(\mathbf{x}-\mathbf{x}_J)$ on element $\mathcal{K}_J$ to obtain $\overline{\partial^{m-1}u_J}$, respectively.

(c) Let $\widetilde{R}_j(\mathbf{x}-\mathbf{x}_j)$ be the $R_j(\mathbf{x}-\mathbf{x}_j)$ with its coefficients replaced by the recomputed new coefficients from previous $q,\dots,m+1$ stages. Compute the average values of $\widetilde{R}_j(\mathbf{x}-\mathbf{x}_j)$ on every element $\mathcal{K}_J$ to obtain $\overline{R}_J$, respectively. Note that at the qth stage, $R_j(\mathbf{x}-\mathbf{x}_j)=\widetilde{R}_j(\mathbf{x}-\mathbf{x}_j)=0$.

(d) Let $\overline{L}_J=\overline{\partial^{m-1}u_J}-\overline{R}_J$ for all J.

(e) Out of the new approximate cell average values $\{\overline{L}_J\}$, use a non-oscillatory reconstruction procedure to reconstruct the linear polynomial $L_j(\mathbf{x}-\mathbf{x}_j)$. The reconstructed coefficients in the first degree terms of $L_j(\mathbf{x}-\mathbf{x}_j)$ are candidates for the corresponding $u_j^{(\mathbf{m})}(0)$'s, $|\mathbf{m}|=m$.

(f) Repeat from (a) to (e) until all possible combinations of the $(m-1)$th order partial derivatives have been taken. Then the candidates for all coefficients in the mth degree terms of $u_j(\mathbf{x}-\mathbf{x}_j)$ have been computed. For each of these coefficients, say $\frac{1}{m!}u_j^{(\mathbf{m})}(0),\ |\mathbf{m}|=m$, let the new value $\tilde{u}_j^{(\mathbf{m})}(0)=F$ (candidates of $u_j^{(\mathbf{m})}(0)$) where F is a limiter function.

Step 2. In order to find new coefficients in the zeroth and first degree terms of $\{u_j(\mathbf{x}-\mathbf{x}_j)\}$, we perform the procedure of Step 1(a)–(f) with $m=1$. The new coefficient in the zeroth degree term is obtained by ensuring that the average value of $u_j(\mathbf{x}-\mathbf{x}_j)$ on element $\mathcal{K}_j$ is invariant with new coefficients.

2.2.2. Hierarchical reconstruction for the P^2 spectral volume solution

Since we use P^2 spectral volume solution in our computation, we describe the implementation of HR for piecewise quadratic polynomial on CVs and the related piecewise linear polynomial reconstruction procedure in this section.

Suppose on each SV $\mathcal{K}_i\in\{\mathcal{K}_0,\mathcal{K}_1,\mathcal{K}_2,\mathcal{K}_3\}$ of Fig. 2, a quadratic polynomial is given in the form of a two-dimensional Taylor expansion

$$u_i(x-x_i,y-y_i)=u_i(0,0)+\partial_x u_i(0,0)(x-x_i)+\partial_y u_i(0,0)(y-y_i)+\frac{1}{2}\partial_{xx}u_i(0,0)(x-x_i)^2+\partial_{xy}u_i(0,0)(x-x_i)(y-y_i)+\frac{1}{2}\partial_{yy}u_i(0,0)(y-y_i)^2, \tag{2.6}$$

where (x_i,y_i) is the centroid of $\mathcal{K}_i$. We will reconstruct a new quadratic polynomial on each CV $C_{0j},\ j=0,\dots,5$, of $\mathcal{K}_0$ with a point-wise error $\mathcal{O}(\Delta x^3)$.

We first rewrite the polynomial (2.6) on every CV of $\mathcal{K}_i,\ i=0,\dots,3$ as

$$u_{ij}(x-x_{ij},y-y_{ij})=u_{ij}(0,0)+\partial_x u_{ij}(0,0)(x-x_{ij})+\partial_y u_{ij}(0,0)(y-y_{ij})+\frac{1}{2}\partial_{xx}u_{ij}(0,0)(x-x_{ij})^2+\partial_{xy}u_{ij}(0,0)(x-x_{ij})(y-y_{ij})+\frac{1}{2}\partial_{yy}u_{ij}(0,0)(y-y_{ij})^2, \tag{2.7}$$

where (x_{ij},y_{ij}) is the centroid of CV $C_{ij},\ i=0,\dots,3$ and $j=0,\dots,5,\ \partial_x u_{ij}(0,0)=\partial_x u_i(x_{ij},y_{ij})$, similarly for other coefficients.

The difference between applying HR on a triangular CV and on a quadrilateral CV is that different adjacent CVs are used. To perform HR on a quadrilateral CV, the four edge-adjacent CVs are used. To perform HR on a triangular CV, both the edge-adjacent CVs and some neighbors of these adjacent CVs are used. For example, to perform HR on the triangular CV C_{01}, the stencil set consists of $C_{01},\ C_{00},\ C_{05},\ C_{03},\ C_{02},\ C_{10},\ C_{15}$ and C_{14}; to perform HR on CV C_{00}, the stencil set consists of $C_{00},\ C_{20},\ C_{01},\ C_{10}$ and C_{05}. In general, for a triangular CV, the principle of the selection of elements to build a stencil is to use edge adjacent elements and vertex adjacent elements. while for a quadrilateral CV, the principle of the selection of elements to build a stencil is to use edge adjacent elements. The choice of this stencil is motivated by numerical test.

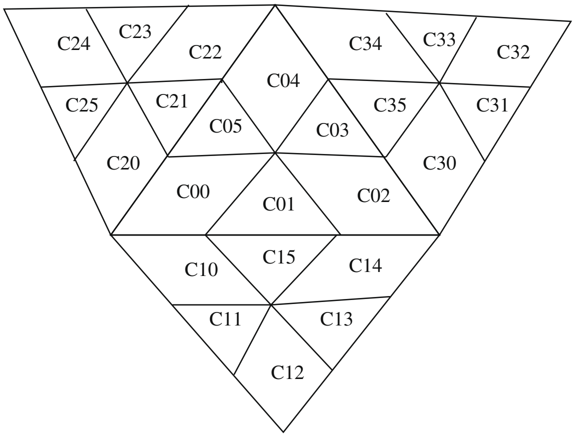


Fig. 2. Schematic of 2D HR for CVs of $\mathcal{K}_0$ (the center SV). Adjacent SVs $\mathcal{K}_1,\ \mathcal{K}_2$ and $\mathcal{K}_3$ (clockwisely from the bottom) are used. $\mathcal{K}_i$ consists of CVs $C_{ij},\ j=0,\dots,5$.

We now describe the HR algorithm. The algorithm applies to both triangular CVs and quadrilateral CVs. For convenience, we relabel the stencil set as $\{C_0, C_1, \ldots, C_k\}$, where C_0 is the CV on which the polynomial defined is to be reconstructed, $k = 7$ if C_0 is a triangle and $k = 4$ if C_0 is a rectangle. The adjacent CVs $\{C_1, \ldots, C_k\}$ are labeled clockwisely surrounding CV C_0. We also relabel the centroid of C_l as (x_l, y_l), and the polynomial defined on it as $u_l(x - x_l, y - y_l),\ l = 0, \ldots, k$, similarly for their coefficients.

According to the algorithm, we first take the first partial derivative with respect to x for $u_l(x - x_l, y - y_l)$ to obtain

$$L_l(x - x_l, y - y_l) = \partial_x u_l(0,0) + \partial_{xx} u_l(0,0)(x - x_l) + \partial_{xy} u_l(0,0)(y - y_l), \quad l = 0, \ldots, k. \tag{2.8}$$

Calculate the average of $L_l(x - x_l, y - y_l)$ on element C_l to obtain $\overline{L}_l = \partial_x u_l(0,0),\ l = 0, \ldots, k$. We then apply a non-oscillatory reconstruction procedure to the cell averages $\overline{L}_l$, which will be described at the end of this section, to obtain a new linear polynomial on element C_0,

$$\widetilde{L}_0(x - x_0, y - y_0) = \partial_x \tilde{u}_0(0,0) + \partial_{xx} \tilde{u}_0(0,0)(x - x_0) + \partial_{xy} \tilde{u}_0(0,0)(y - y_0), \tag{2.9}$$

where $\partial_x \tilde{u}_0(0,0) = \overline{L}_0$. Then we take the first partial derivative with respect to y for $u_l(x - x_l, y - y_l)$ to redefine $L_l(x - x_l, y - y_l) = \partial_y u_l(0,0) + \partial_{xy} u_l(0,0)(x - x_l) + \partial_{yy} u_l(0,0)(y - y_l),\ l = 0, \ldots, k$, and perform the same reconstruction procedure to obtain another polynomial on C_0,

$$\widetilde{L}_0(x - x_0, y - y_0) = \partial_y \tilde{u}_0(0,0) + \partial_{xy} \tilde{u}_0(0,0)(x - x_0) + \partial_{yy} \tilde{u}_0(0,0)(y - y_0). \tag{2.10}$$

$\partial_{xx} \tilde{u}_0(0,0)$ and $\partial_{yy} \tilde{u}_0(0,0)$ will be the corresponding new coefficients of the reconstructed quadratic polynomial on element C_0. $\partial_{xy} \tilde{u}_0(0,0)$ appears twice in the above procedures and serves as candidates for the corresponding coefficient. A limiter function will be used to determine the new value (still denoted by $\partial_{xy} \tilde{u}_0(0,0)$) for coefficient $\partial_{xy} u_0(0,0)$ out of its candidates, which will be described later.

Now we perform Step 2 of the algorithm. We compute the average of $u_l(x - x_l, y - y_l)$ on element C_l to obtain $\bar{u}_l,\ l = 0, \ldots, k$. And compute averages of the polynomial (note that its coefficients are updated by corresponding new values)

$$\widetilde{R}_0(x - x_0, y - y_0) = \frac{1}{2} \partial_{xx} \tilde{u}_0(0,0)(x - x_0)^2 + \partial_{xy} \tilde{u}_0(0,0)(x - x_0)(y - y_0) + \frac{1}{2} \partial_{yy} \tilde{u}_0(0,0)(y - y_0)^2 \tag{2.11}$$

on elements $C_0, \ldots, C_k$ to obtain $\overline{R}_0, \ldots, \overline{R}_k$, respectively. Redefine $\overline{L}_l = \bar{u}_l - \overline{R}_l,\ l = 0, \ldots, k$. The same reconstruction procedure is applied to the cell averages $\{\overline{L}_l\}$ to obtain new coefficients $\partial_x \tilde{u}_0(0,0)$ and $\partial_y \tilde{u}_0(0,0)$ for the polynomial on element C_0. Finally let the new coefficient $\tilde{u}_0(0,0) = \overline{L}_0$ to ensure conservation of the average of $u_0(x - x_0, y - y_0)$ on element C_0.

Next we describe the non-oscillatory reconstruction procedure. The following limiter functions are needed. The *minmod* limiter function [4]

$$m(c_1, c_2, \ldots, c_r) = \begin{cases} \min\{c_1, c_2, \ldots, c_r\}, & \text{if } c_1, c_2, \ldots, c_r > 0, \\ \max\{c_1, c_2, \ldots, c_r\}, & \text{if } c_1, c_2, \ldots, c_r < 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.12}$$

gives a MUSCL reconstruction [9,11]; the limiter function

$$m_2(c_1, c_2, \ldots, c_r) = c_j, \quad \text{where } |c_j| = \min\{|c_1|, |c_2|, \ldots, |c_r|\}, \tag{2.13}$$

gives the second order ENO [5] reconstruction; and the center biased *minmod* limiter m_b and ENO limiter m_{2b} can be formulated as

$$\begin{aligned} m_b(c_1, c_2, \ldots, c_r) &= m\left((1 + \varepsilon) m(c_1, c_2, \ldots, c_r), \frac{1}{r} \sum_{i=1}^{r} c_i\right), \\ m_{2b}(c_1, c_2, \ldots, c_r) &= m_2\left((1 + \varepsilon) m_2(c_1, c_2, \ldots, c_r), \frac{1}{r} \sum_{i=1}^{r} c_i\right), \end{aligned} \tag{2.14}$$

where $\varepsilon > 0$ is a small perturbation number. In [13], to construct central and finite volume schemes on structured meshes, a value of $\varepsilon = 0.01$ is used.

We employ the weighted combination of functions used in [25] which follows the line of Refs. [9,10,15,8,6]. The reconstruction procedure proceeds as follows.

Take the reconstruction of polynomial (2.9) as an example. We form k stencils $\{C_0, C_l, C_{l+1}\}$, where $l = 1, \ldots, k$ and $C_{k+1} = C_1$. On the first stencil, we solve the following equations for $\partial_{xx} u_{0,1}(0,0)$ and $\partial_{xy} u_{0,1}(0,0)$

$$\frac{1}{|C_l|} \int_{C_l} L_{0,1}(x - x_0, y - y_0)\, dx\, dy \equiv \overline{L}_0 + \partial_{xx} u_{0,1}(0,0)(x_l - x_0) + \partial_{xy} u_{0,1}(0,0)(y_l - y_0) = \overline{L}_l, \tag{2.15}$$

$l = 1, 2$, similarly for other stencils.

The linear polynomials computed from these stencils are denoted by $L_{0,l}(x - x_0, y - y_0)$ with first degree coefficients $\partial_{xx} u_{0,l}(0,0),\ \partial_{xy} u_{0,l}(0,0),\ l = 1, \ldots, k$. The reconstructed linear polynomial (2.9) is a convex combination of these computed polynomials, *i.e.*,

$$\widetilde{L}_0(x - x_0, y - y_0) = \sum_{l=1}^{k} w_l L_{0,l}(x - x_0, y - y_0). \tag{2.16}$$

The weight w_l depends on $L_{0,l}$ and satisfies

$$w_l \geqslant 0, \quad \sum_{l=1}^{k} w_l = 1. \tag{2.17}$$

The weights are smooth functions, and when a stencil contains a discontinuity of the solution, the corresponding weight should be essentially 0. We define the weights as follows:

$$w_l = \frac{\alpha_l}{\sum_{s=1}^{k}\alpha_s}, \quad l = 1, \ldots, k, \tag{2.18}$$

where α_s will be defined later. Let

$$d_l = \frac{1/\theta_l}{\sum_{s=1}^{k} 1/\theta_s}, \tag{2.19}$$

where θ_l is the condition number of the corresponding stencil, $\|A\|\|A^{-1}\|$, where A is the 2×2 coefficient matrix of (2.15) for the lth stencil, $\|\cdot\|$ denotes the 1-norm. This puts the condition number of a stencil into consideration so that an ill-conditioned stencil has a smaller weight. Let

$$\alpha_l = \frac{d_l}{1 + h\beta_l}, \tag{2.20}$$

where

$$\beta_l = (\partial_{xx} u_{0,l}(0,0))^2 + (\partial_{xy} u_{0,l}(0,0))^2, \tag{2.21}$$

h is the local mesh size. The constant 1 in (2.20) is used because it is large with respect to h. In principle the constant should be chosen proportional to the size of the computational domain.

After all weights $\{w_l\}$ have been computed, the new coefficient $\partial_{xx}\tilde{u}_0(0,0)$ is defined to be

$$\partial_{xx}\tilde{u}_0 = \begin{cases} \sum_{l=1}^{k} w_l \partial_{xx} u_{0,l}, & \text{if } \bar{L}_{\min} < \bar{L}_0 < \bar{L}_{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.22}$$

where $\bar{L}_{\min} = \min\{\bar{L}_j : j = 0, \ldots, k\}$ and $\bar{L}_{\max} = \max\{\bar{L}_j : j = 0, \ldots, k\}$. Violation of $\bar{L}_{\min} < \bar{L}_0 < \bar{L}_{\max}$ detects an extreme value. This option is used to further reduce oscillations. The candidate coefficient $\partial_{xy}\tilde{u}_0(0,0)$ is determined similarly.

The reconstruction of function (2.10) follows the above procedure. After the reconstruction of functions (2.9) and (2.10), $\partial_{xx}\tilde{u}_0(0,0)$ and $\partial_{yy}\tilde{u}_0(0,0)$ are new values for corresponding terms of coefficients of $u_0(x - x_0, y - y_0)$ as in (2.7). However, the reconstruction of functions (2.9) and (2.10) leaves us two choices for the coefficient $\partial_{xy}u_0(0,0)$, from (2.9) and (2.10). We use the center-biased ENO limiter function m_{2b} to determine the new coefficient $\partial_{xy}\tilde{u}_0(0,0)$ from them, in which ε is set to be 0.01.

In Step 2 of HR, a similar reconstruction procedure out of cell averages $\{\bar{L}_l\}$ uses the following weight [16]:

$$\alpha_l = \frac{d_l}{(\epsilon_1 + \beta_l)^2}, \tag{2.23}$$

where β_l is the "smoothness indicator" of the lth stencil similar to those used in the WENO scheme,

$$\beta_l = (\partial_x u_{0,l}(0,0))^2 + (\partial_y u_{0,l}(0,0))^2, \tag{2.24}$$

$\partial_x u_{0,l}(0,0)$ and $\partial_y u_{0,l}(0,0)$ are the first degree coefficients determined in the lth stencil by an equation similar to (2.15). $\epsilon_1 > 0$ is a small number introduced to keep the denominator from being 0. For all numerical experiments conducted in the paper, we use a value of $\varepsilon_1 = 1.0 \times 10^{-6}$ and find it does not increase the overshoots or undershoots at discontinuities. Note that in Step 2 one can also adopt the weight

$$\alpha_l = \frac{d_l}{1 + (\beta_l)^2} \tag{2.25}$$

similar to the form of Eq. (2.20). We found that (2.25) gives slightly bigger overshoots or undershoots.

A function similar to Eq. (2.22) is used to determine the new coefficients $\partial_x\tilde{u}_0(0,0)$ and $\partial_y\tilde{u}_0(0,0)$ for the function $u_0(x - x_0, y - y_0)$ as in (2.7). The extreme value detector (*i.e.*, the "0" case in (2.22)) is also applied here.

For systems, we perform the reconstruction component-wisely on conservative variables.

2.3. Local limiting procedure

Since shock waves or contact discontinuities are all local phenomena, in principle the limiting procedure only needs to be applied to a small region covering the discontinuities. To speed up the computation, we modify the local limiting procedure in [25] to speed up the computation. This local limiting procedure adopts the limiter in [2] to identify "bad cells", *i.e.*, CVs which may contain oscillatory solutions.

Denote the polynomial solution in the interior of a CV as

$$u_h^{in}(\mathbf{x}) = \bar{u} + \tilde{u},$$

where $\mathbf{x}$ is a quadrature point on an edge of the CV, $\bar{u}$ is the average value of the solution on the CV, and $\tilde{u}$ is the variation. We first compute

$$m(\tilde{u}, \mu\Delta\bar{u}),$$

where m is the minmod function, $\Delta\bar{u} = \bar{u}_1 - \bar{u}$, $\bar{u}_1$ is the average value of the solution of the adjacent CV sharing the edge and $\mu > 1$. We take $\mu = 1.2$ in our numerical runs. If the minmod function returns other than the first argument, this CV is identified as a "bad cell", and the polynomial solution on it is regarded to be oscillatory and marked for reconstructions. The limiting process is applied to SVs which contain "bad CVs" while keeping the computed spectral volume solution unchanged for other elements.

3. Numerical examples

We first study the limiter functions and test the capability of the method to achieve the desired third order accuracy, using the scalar Burgers equation and the Euler equation for gas dynamics. In this case, if HR is used, it is applied to all CVs. In the two-dimensional space, the Euler equation can be expressed in conservation form

$$\mathbf{u}_t + f(\mathbf{u})_x + g(\mathbf{u})_y = 0, \tag{3.1}$$

where $\mathbf{u} = (\rho, \rho u, \rho v, E)$, $f(\mathbf{u}) = (\rho u, \rho u^2 + p, \rho u v, u(E+p))$ and $g(\mathbf{u}) = (\rho v, \rho u v, \rho v^2 + p, v(E+p))$. Here ρ is the density, (u, v) is the velocity, E is the total energy, p is the pressure and $E = \frac{p}{\gamma-1} + \frac{1}{2}\rho(u^2 + v^2)$. γ is equal to 1.4 for all test cases. We then test problems with discontinuities to assess the non-oscillatory property of the scheme, again using the Euler equation for gas dynamics. The CFL numbers showed in the test problems are ones with respect to control volumes.

3.1. Numerical errors for smooth solutions of Burgers' equation

The two-dimensional Burgers' equation with a periodic boundary condition is employed:

$$\begin{aligned} &\partial_t u + \partial_x\left(\frac{u^2}{2}\right) + \partial_y\left(\frac{u^2}{2}\right) = 0, \quad \text{in } (0,T)\times\Omega, \\ &u(t=0,x,y) = \frac{1}{4} + \frac{1}{2}\sin(\pi(x+y)), \quad (x,y)\in\Omega, \end{aligned} \tag{3.2}$$

where the domain Ω is the square $[-1,1]\times[-1,1]$. At $T = 0.1$ the exact solution is smooth. To demonstrate the robustness of the WENO-type reconstruction, this reconstruction was performed on the irregular triangular meshes. The structure of the mesh is shown in Fig. 3(a). The errors presented are those of the cell averages of u.

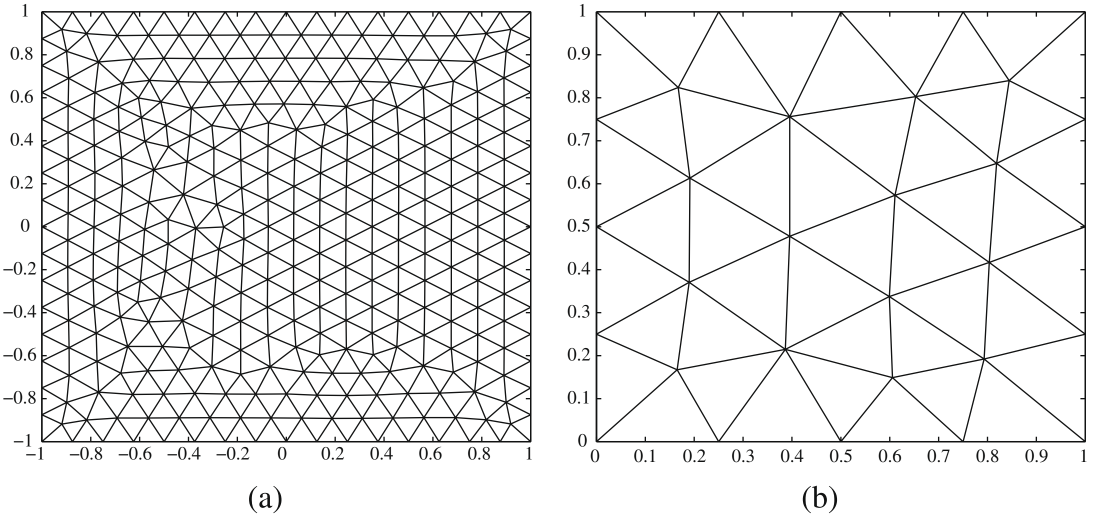


Fig. 3. Meshes for accuracy tests.

Table 1
Accuracy for 2D Burgers equation with smooth solution, using HR, CFL = 0.4.

h	L_1 error	Order	L_∞ error	Order
1/4	2.89E−3	–	1.84E−2	–
1/8	3.57E−4	3.02	1.58E−3	3.54
1/16	4.71E−5	2.92	1.11E−3	0.51
1/32	6.22E−6	2.92	1.87E−4	2.57
1/64	7.95E−7	2.97	1.03E−5	4.18
1/128	1.01E−7	2.98	5.33E−7	4.27

Table 2
Accuracy for 2D Burgers equation with smooth solution, without using HR, CFL = 0.4.

h	L_1 error	Order	L_∞ error	Order
1/4	5.49E−3	–	9.96E−3	–
1/8	9.62E−4	2.51	5.51E−3	0.85
1/16	1.53E−4	2.65	1.11E−3	2.31
1/32	2.32E−5	2.72	9.05E−5	3.61
1/64	3.21E−6	2.85	4.10E−5	1.14
1/128	4.31E−7	2.90	1.40E−5	1.55

Table 3
Accuracy for 2D Euler equation with smooth solution, on triangular meshes, using HR, CFL = 0.4.

h	L_1 error	Order	L_∞ error	Order
1/4	1.23E−5	–	6.23E−5	–
1/8	2.35E−6	2.39	1.01E−5	2.62
1/16	2.94E−7	3.00	1.92E−6	2.40
1/32	3.33E−8	3.14	1.93E−7	3.31
1/64	4.43E−9	2.91	2.90E−8	2.73
1/128	5.35E−10	3.05	4.20E−9	2.79

The accuracy results are shown in Table 1. This test problem shows that HR reconstruction of piecewise quadratic polynomial is approximately third order accurate, and stable. The accuracy results of the numerical solution without the use of HR are presented in Table 2 for comparison.

3.2. Accuracy test for smooth inviscid compressible flow

A two-dimensional test problem [16] for the Euler equations is used, for ideal gas with $\gamma = 1.4$. The exact solution is given by $\rho = 1 + 0.5\sin(x + y - (u + v)t)$, $u = 1.0$, $v = -0.7$ and $p = 1$. The convergence test is conducted on irregular triangular meshes on the spatial domain $[0,1] \times [0,1]$ from the time $T = 0$ to $T = 0.1$, see Fig. 3(b) for a typical mesh. The triangle size is roughly equal to a rectangular element case of $\Delta x = \Delta y = h$, as indicated in the first column of Table 3. The accuracy results of numerical solutions with the use of HR are shown in Table 3. The errors presented are those of the cell averages of density. It is clear that HR is third order accurate. Table 4 shows the accuracy results without the use of HR for the purpose of comparison.

3.3. Accuracy test for vortex evolution problem

This test problem is taken from [16] to investigate the accuracy of the scheme for the nonlinear problem with a smooth solution. The computational domain is $[0,10] \times [0,10]$. The vortex is described by a perturbation to the velocity (u, v), and the temperature $\left(T = \frac{p}{\rho}\right)$. There is no perturbation in the entropy $\left(S = \frac{p}{\rho^\gamma}\right)$. The perturbation is described by

$$\begin{aligned}
\delta u &= \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(5.0 - y),\\
\delta v &= \frac{\epsilon}{2\pi} e^{0.5(1-r^2)}(x - 5.0),\\
\delta T &= -\frac{(\gamma-1)\epsilon^2 e^{2\alpha(1-r^2)}}{8\gamma\pi^2},
\end{aligned} \tag{3.3}$$

where $r = \sqrt{(x-5.0)^2 + (y-5.0)^2}$. and the strength of the vortex ϵ is equal to 5.0. We see from Table 5 that the desired third order accuracy is achieved in L_1 norm. Table 6 shows the accuracy results without the use of HR for the purpose of comparison.

Table 4
Accuracy for 2D Euler equation with smooth solution, on triangular meshes, without using HR, CFL = 0.4.

h	L_1 error	Order	L_∞ error	Order
1/4	2.52E−5	–	6.57E−5	–
1/8	4.13E−6	2.61	1.51E−5	2.12
1/16	6.47E−7	2.67	3.00E−6	2.33
1/32	1.05E−7	2.62	5.10E−7	2.56
1/64	9.34E−9	3.49	5.20E−8	3.29
1/128	1.09E−9	3.10	7.55E−9	2.78

Table 5
Accuracy for 2D vortex evolution problem, using HR, CFL = 0.4.

h	L_1 density error	Order	L_∞ density error	Order	L_1 energy error	Order	L_∞ energy error	Order
1/2	5.35E−3	–	6.33E−3	–	2.26E−2	–	2.42E−2	–
1/4	6.81E−4	2.97	9.49E−4	2.74	2.73E−3	3.05	4.12E−3	2.55
1/8	8.46E−5	3.01	1.26E−4	2.91	3.36E−4	3.02	5.52E−4	2.90
1/16	1.05E−5	3.01	1.52E−5	3.05	4.31E−5	2.96	6.96E−5	2.99
1/32	1.32E−6	2.99	1.80E−6	3.08	6.32E−6	2.77	2.39E−5	1.54

Table 6
Accuracy for 2D vortex evolution problem, without using HR, CFL = 0.4.

h	L_1 density error	Order	L_∞ density error	Order	L_1 energy error	Order	L_∞energy error	Order
1/2	7.72E−3	–	1.24E−2	–	3.24E−2	–	6.49E−2	–
1/4	1.26E−3	2.62	1.92E−3	2.69	5.14E−3	2.66	1.02E−2	2.67
1/8	2.18E−4	2.53	2.82E−4	2.77	8.54E−4	2.59	1.72E−3	2.57
1/16	3.78E−5	2.53	5.26E−5	2.42	1.50E−4	2.51	2.95E−4	2.54
1/32	6.53E−6	2.53	1.17E−5	2.17	2.92E−5	2.36	5.56E−5	2.41

Remark. Recently a weak instability was found in some partitions [1], which include the partition used in the present paper. From the above accuracy tests, it seems that HR can stabilize the SV method on this partition and as a result improve the accuracy. We have observed similar phenomena when applying HR to central DG and DG methods. For example, in [12], convergence tests verify the expected third order accuracy for central DG with HR at the CFL number 0.4, while linear stability analysis shows that the central DG (without HR) should be stable at a CFL number no more than 0.33 [14]. In preliminary convergence tests conducted along with [25] (which will be further studied in the future), we have observed test results verifying the expected third order accuracy for DG with HR at the CFL number 0.5 (0.4 when using partial neighboring cells), while linear stability analysis shows that the DG (without HR) should be stable at a CFL number no more than 0.2 [3].

3.4. Shu–Osher problem

The initial data of the Shu–Osher problem [18] are:

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if } x \leqslant -4, \\ (1 + 0.2\sin(5x), 0, 1) & \text{if } x \geqslant -4. \end{cases} \tag{3.4}$$

We consider the solution of the Euler equations in a rectangular domain of $[-5, 5] \times [0, 0.1]$ with a triangulation of 301 vertices in the x-direction and 4 vertices in the y-direction. The triangular mesh is obtained by adding a diagonal line in each rectangle on a rectangular grid. The CFL number is 0.4. The initial velocity along the y-direction is zero. The density at $t = 1.8$ from the P^2 numerical solution is shown in Fig. 4(a). Fig. 4(a) is obtained by interpolating the numerical solution along the horizontal line $y = 0.05$ on 301 equally spaced points. From Fig. 4(a), we can clearly see that the resolution of the method is good. The details of the solution are almost exactly recovered. Fig. 4(b) is the 3D view of the solution.

3.5. Shock tube problems of Euler equations

We consider the solution of the Euler equations in a rectangular domain of $[-1, 1] \times [0, 0.2]$ with a triangulation of 101 vertices in the x-direction and 11 vertices in the y-direction. The triangular meshes as in Section 3.4 are used. The initial velocity along the y-direction is zero.

The first case is the Sod problem [19]. The initial data is

$$(\rho, u, p) = \begin{cases} (1, 0, 1), & \text{if } x \leqslant 0 \\ (0.125, 0, 0.1), & \text{if } x > 0. \end{cases} \tag{3.5}$$

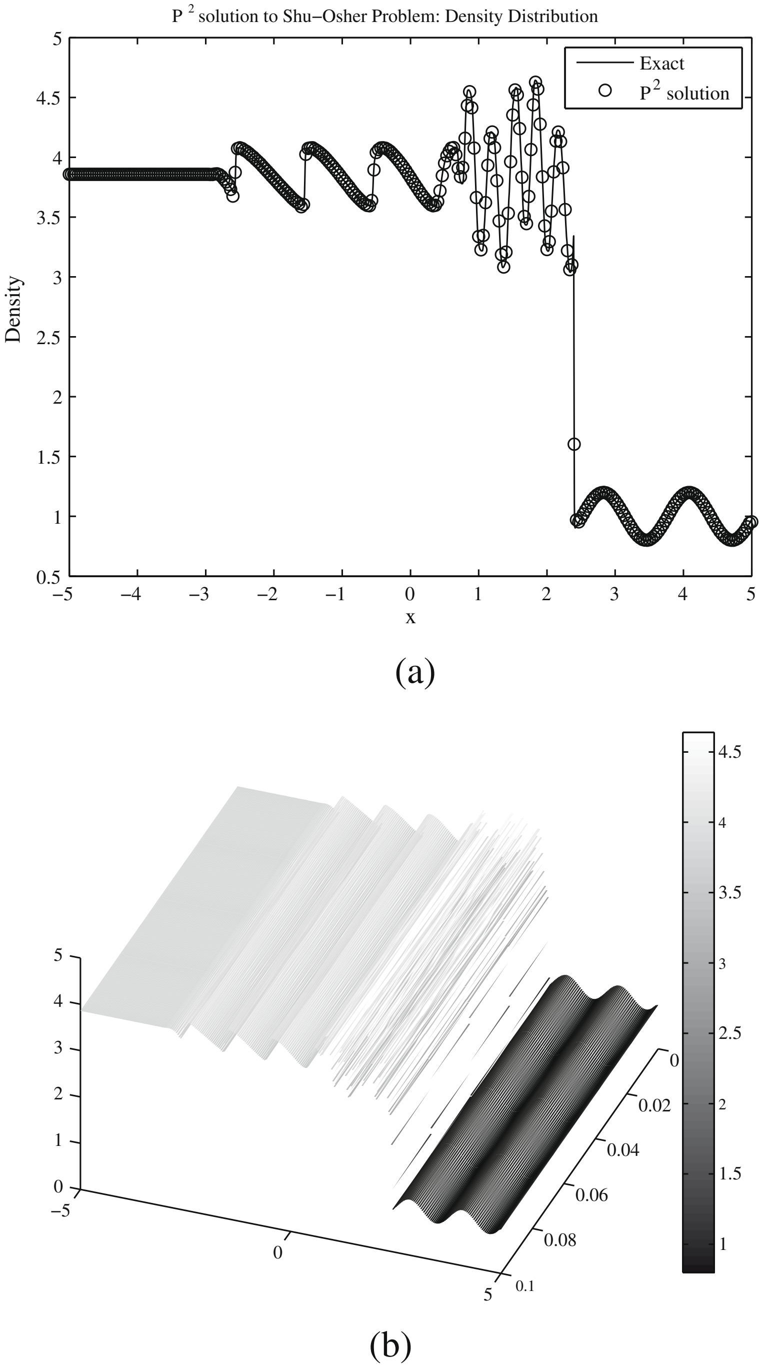


Fig. 4. Shu–Osher problem, third order spectral volume solution, density plot. (a) Cut line view by interpolating the solution data along the horizontal line $y = 0.05$ on 301 equally spaced points; (b) 3D view of the solution.

The density at $t = 0.40$ is shown in Fig. 5(a).

The second case is the Lax problem [7]. The initial data is

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528), & \text{if } x \leqslant 0 \\ (0.5, 0, 0.571), & \text{if } x > 0. \end{cases} \tag{3.6}$$

The density at $t = 0.26$ is shown in Fig. 5(b).

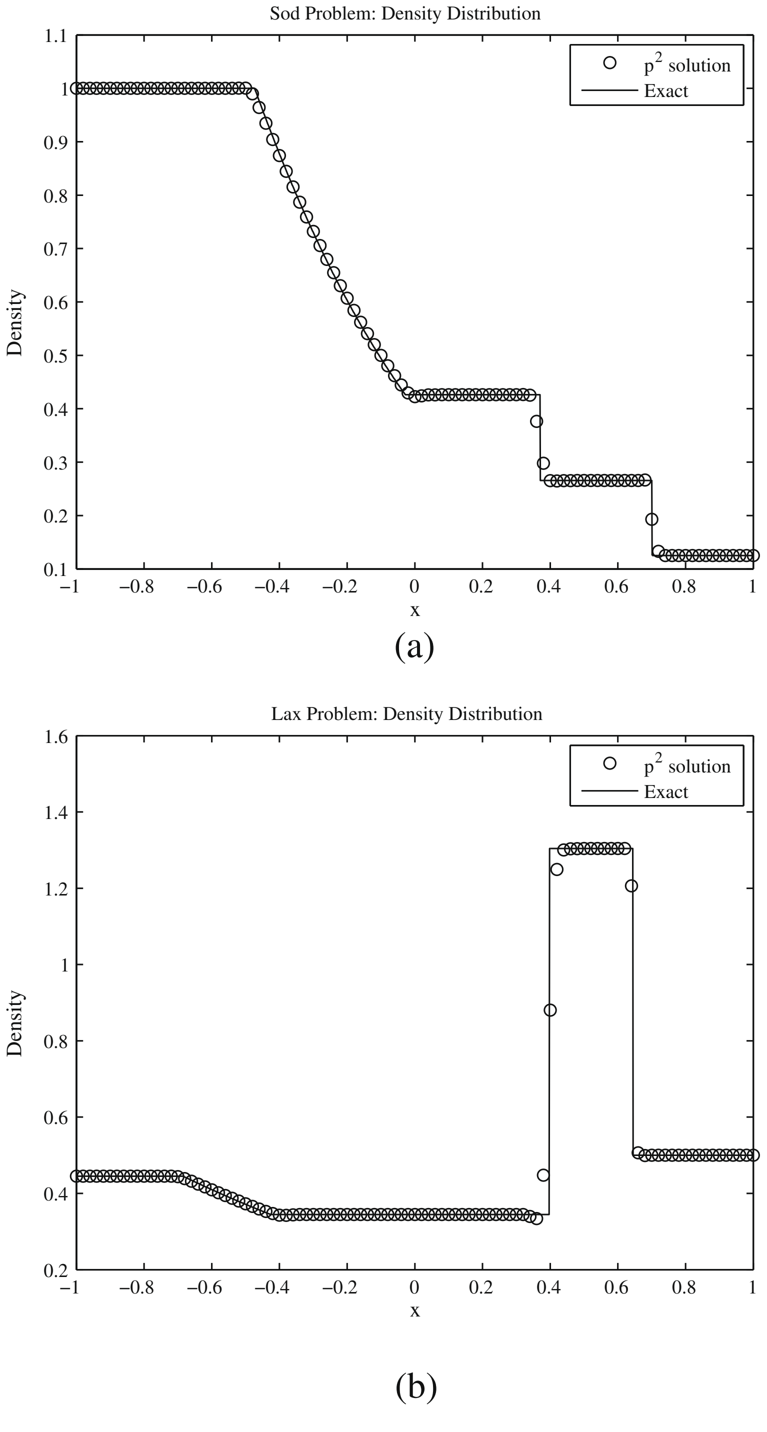
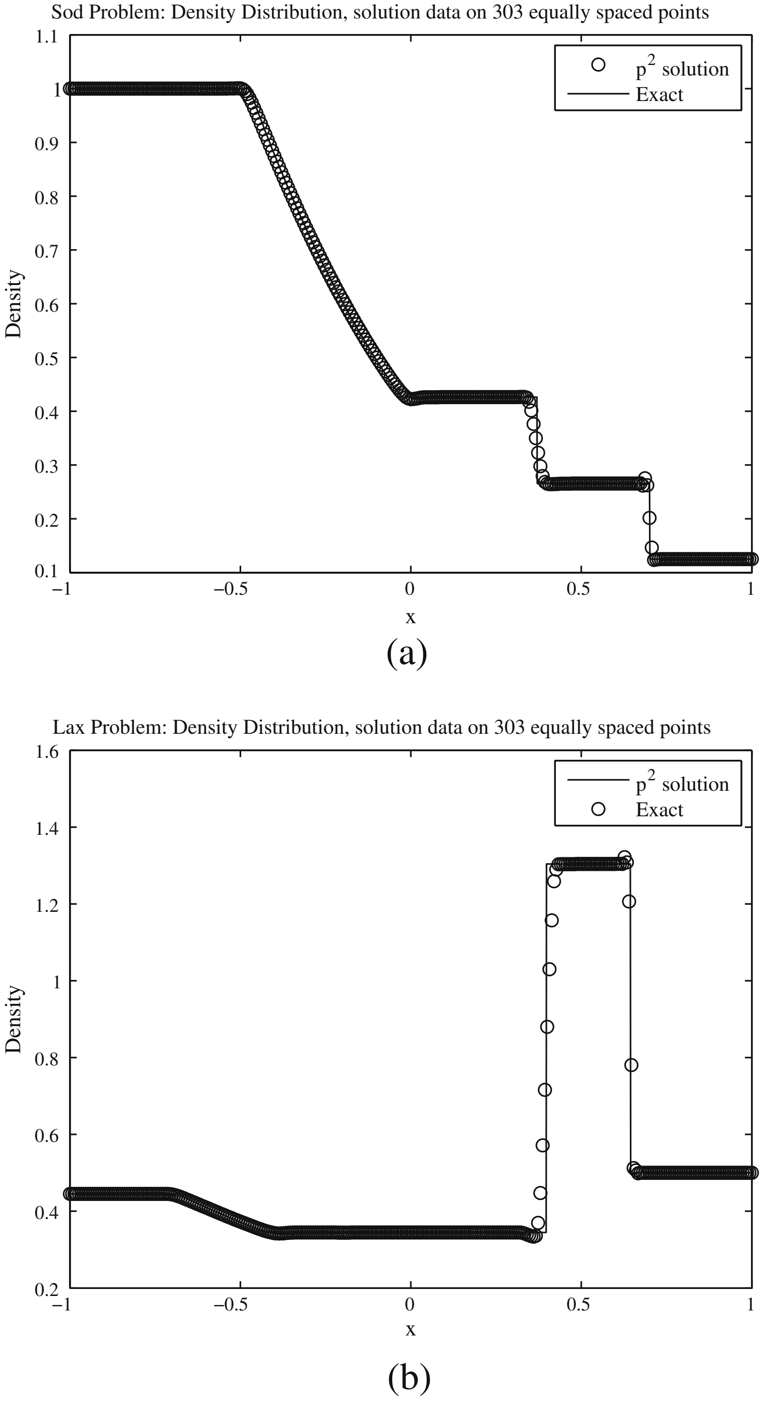


Fig. 5. P^2 spectral volume solutions to Shock tube problems: density plot; cut line view by interpolating the solution data along the horizontal line $y = 0.1$ on 101 equally spaced points. (a) Sod problem; (b) Lax problem.

Fig. 5(a) and (b) are obtained by interpolating the numerical solutions along the horizontal line $y = 0.1$ on 101 equally spaced points. They show almost negligible over/under-shoots and sharp contact profile.

For the purpose of comparison with standard finite volume method, profiles of the numerical solutions along the horizontal line $y = 0.1$ on 303 equally spaced points are plotted in Fig. 6(a) and (b). The quality of the P^2 spectral volume solution is comparable with those from standard finite volume method.



(a)

(b)

Fig. 6. P^2 spectral volume solutions to Shock tube problems: density plot; cut line view by interpolating the solution data along the horizontal line $y = 0.1$ on 303 equally spaced points. (a) Sod problem; (b) Lax problem.

3.6. 2D shock vortex interactions

This test case is taken from [16] to investigate the ability of the scheme to resolve the vortex and the interaction. The computational domain is $[0,2] \times [0,2]$. Initially a Mach 1.1 shock is positioned at $x = 0.5$ and normal to the x-axis. Its left state is $(\rho, u, v, P) = (1, 1.1\sqrt{\gamma}, 0, 1)$. The vortex is described by a perturbation to the velocity (u, v), temperature $\left(T = \frac{P}{\rho}\right)$ and entropy $\left(S = \ln \frac{P}{\rho^{\gamma}}\right)$ of the mean flow and has the values:

$$
\begin{aligned}
&\tilde{u} = \epsilon\tau e^{\alpha(1-r^2)} \sin\theta,\\
&\tilde{v} = -\epsilon\tau e^{\alpha(1-r^2)} \cos\theta,\\
&\widetilde{T} = -\frac{(\gamma-1)\epsilon^2 e^{2\alpha(1-r^2)}}{4\alpha\gamma},\\
&\widetilde{S} = 0,
\end{aligned}
\tag{3.7}
$$

where $\tau = \frac{r}{r_c}$, $r = \sqrt{(x-x_c)^2 + (y-y_c)^2}$, $(x_c, y_c) = (0.25, 0.5)$, the strength of the vortex ϵ is equal to 0.3, $r_c = 0.05$ and $\alpha = 0.204$. The triangular meshes used are as in Fig. 3. The triangle edge length is roughly equal to 1/100.

The density profile in $[0,2] \times [0,1]$ is plotted at $T = 0.35$ in Fig. 7, with 30 equally spaced contours. It shows that the profile of the vortex after passing the stationary shock is preserved.

3.7. Double Mach reflection

The Double Mach reflection problem is taken from [24]. We solve the Euler equations on a rectangular computational domain of $[0,4] \times [0,1]$. A reflecting wall lies at the bottom of the domain starting from $x = \frac{1}{6}$. Initially a right-moving Mach 10 shock is located at $x = \frac{1}{6},\ y = 0$. The shock makes a 60° angle with the x axis and extends to the top of the computational domain at $y = 1$. The reflective boundary condition is used at the wall. The region from $x = 0$ to $x = \frac{1}{6}$ along the boundary $y = 0$ is always set with the exact post-shock solution, so is the left-side boundary. At the right-side boundary, the flow through boundary condition is used. At the top boundary, the flow values are set to describe the exact motion of the initial Mach 10 shock.

Triangular meshes are used with the triangle edge length roughly equal to $\frac{1}{200}$ and $\frac{1}{400}$, respectively. The portion of the mesh shown in Fig. 8 corresponds to the case with edge length $\frac{1}{200}$. The results are plotted at $t = 0.2$. The density contour of the flow at the time $t = 0.2$ in $[0,3] \times [0,1]$ is shown with 30 equally spaced contour lines.

Fig. 9 is the contour plot of the P^2 spectral volume solutions with triangle edge lengths $\frac{1}{200}$ and $\frac{1}{400}$. Fig. 10 shows the "blown-up" portion around the Double Mach region. With finer mesh, more roll-ups are captured as expected.

Remark on scaling to remove negative pressure. We noticed that strong shocks in the Double Mach reflection problem introduce the negative pressure at quadrature points if they are very close to the shock front due to some small undershoots. The negative pressure may still exist after applying HR. The scaling technique [25] is applied at the end of HR if necessary to remove the negative pressure. The idea is as follows: if at a quadrature point of a CV, the negative pressure remains after reconstruction with the reconstructed polynomial u_h, we redefine the new polynomial u_h^* to be: $u_h^* = \bar{u}_h + 0.5(u_h - \bar{u}_h)$, where $\bar{u}_h$ is the average value of u_h on the CV. The negative pressure is removed after 1 or 2 iterations of the scaling normally.

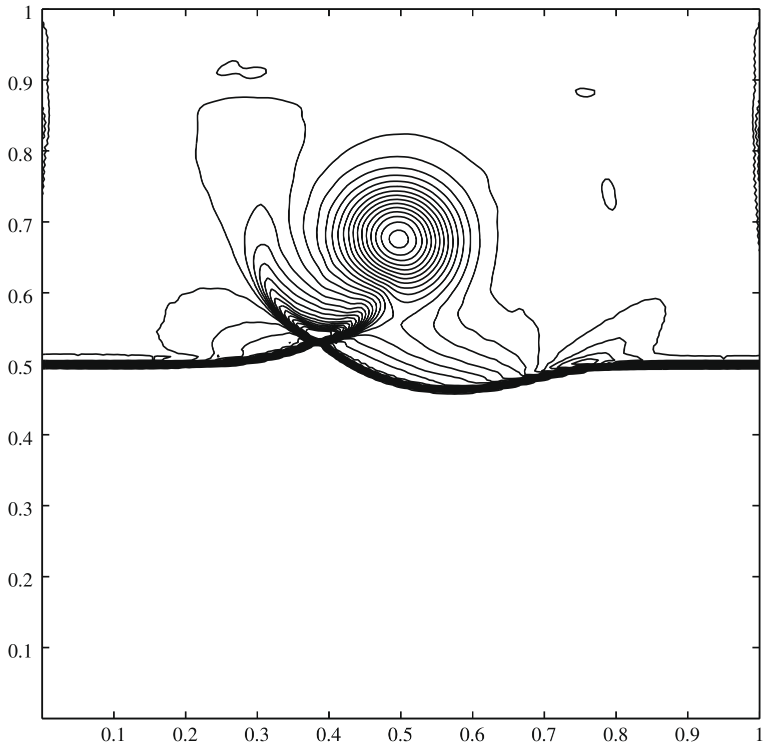
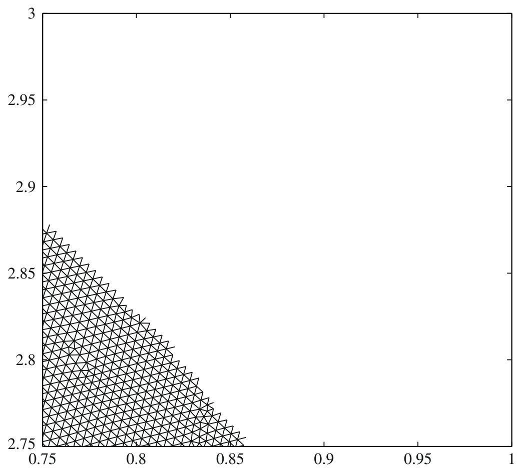


Fig. 7. 2D shock vortex interaction. Density contour at $t = 0.35$ with 30 equally spaced contour lines.


3
2.95
2.9
2.85
2.8
2.75
0.75
0.8
0.85
0.9
0.95
1

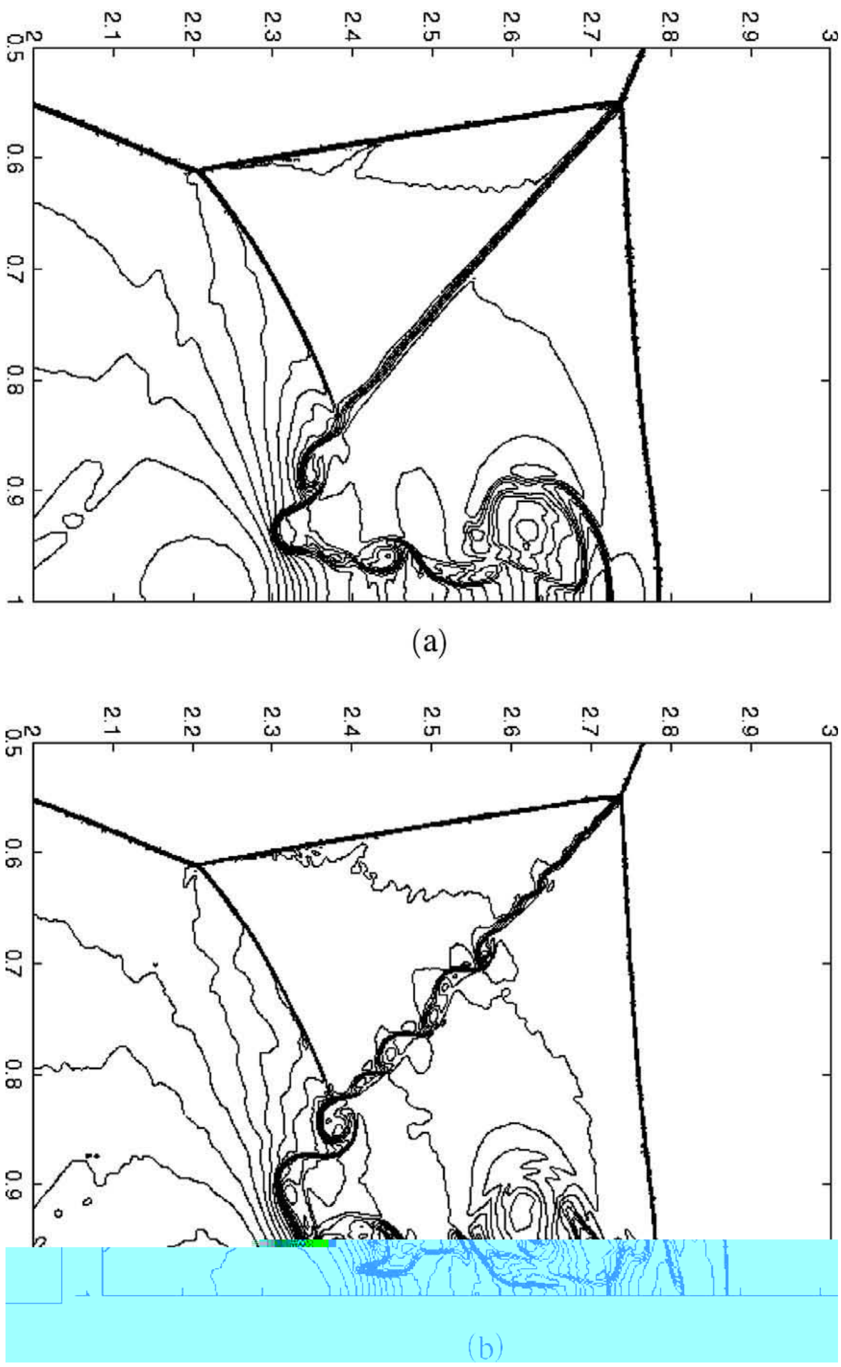


Fig. 10. Double Mach reflection problem, third order spectral volume solution. Blown-up region around the Double Mach stems. Density contour. (a) Solution with element edge length roughly $\frac{1}{200}$; (b) solution with element edge length roughly $\frac{1}{400}$.

3.8. Remark on computational cost of HR

To estimate the computational cost of HR, we use the Burgers' equation with smooth solution as a test case. See Section 3.1 for a description of the setup of the problem. The CPU time is about 2 h 44 min on a 2.4 GHz processor (AMD Opteron 250) by the P^2 spectral volume method with HR on a mesh with triangle edge length roughly equal to $\frac{1}{128}$ (in this case, HR is applied on every control volume). The CPU time is about 1 h 59 min for the same test without HR. Therefore, HR takes about 27% of the total computational time. The code is written in C and is compiled with "g++−O3".

4. Concluding remarks

We have developed the HR reconstruction procedure and used it as a limiter for the spectral volume method on unstructured triangular meshes. The HR reconstruction with the WENO-type reconstruction of linear polynomials maintains the desired order of accuracy and resolution, and effectively reduces spurious oscillations for discontinuous solutions. We also would like to point out that HR seems to reduces the magnitude of the error as well based on the accuracy tests.

In the future, we will explore how HR stabilizes the P^2 spectral method with the partition utilized in the present paper and improves the accuracy.

Acknowledgments

We thank Dr. Zhi Jian Wang for referring us to Ref. [1] about the stability of the CV partition.

All simulations presented in the paper were performed on the Notre Dame Biocomplexity Cluster supported in part by NSF MRI Grant No. DBI-0420980.

References

[1] Kris Van den Abeele, Chris Lacor, An accuracy and stability study of the 2D spectral volume method, J. Comput. Phys. 226 (2007) 1007–1026.
[2] B. Cockburn, C.-W. Shu, The TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws. V: Multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[3] B. Cockburn, C.-W. Shu, Runge–Kutta discontinuous Galerkin methods for convection-dominated problems, J. Sci. Comput. 16 (2001) 173–261.
[4] A. Harten, High resolution schemes for hyperbolic conservation laws, J. Comput. Phys. 49 (1983) 357–393.
[5] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes. III, J. Comput. Phys. 71 (1987) 231–303.
[6] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[7] P. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computations, Commun. Pure Appl. Math. 7 (1954) 159.
[8] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[9] B. van Leer, Toward the ultimate conservative difference scheme. II: Monotonicity and conservation combined in a second order scheme, J. Comput. Phys. 14 (1974) 361–370.
[10] B. van Leer, Towards the ultimate conservative difference scheme. IV: A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.
[11] B. van Leer, Towards the ultimate conservative difference scheme. V: A second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 101–136.
[12] Y.-J. Liu, C.-W. Shu, E. Tadmor, M.-P. Zhang, Central discontinuous Galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction, SIAM J. Numer. Anal. 45 (2007) 2442–2467.
[13] Y.-J. Liu, C.-W. Shu, E. Tadmor, M.-P. Zhang, Non-oscillatory hierarchical reconstruction for central and finite volume schemes, Commun. Comput. Phys. 2 (2007) 933–963.
[14] Y.-J. Liu, C.-W. Shu, E. Tadmor, M.-P. Zhang, L2 stability analysis of the central discontinuous Galerkin method and a comparison between the central and regular discontinuous Galerkin methods, ESAIM: Math. Model. Numer. Anal. 42 (2008) 593–607.
[15] C.-W. Shu, Numerical experiments on the accuracy of ENO and modified ENO schemes, J. Sci. Comput. 5 (1990) 127–149.
[16] C.-W. Shu, Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws, in: B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, A. Quarteroni (Eds.), Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lecture Notes in Mathematics, vol. 1697, Springer, Berlin, 1998.
[17] C.-W. Shu, S. Osher, Efficient Implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[18] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes. II, J. Comput. Phys. 83 (1989) 32–78.
[19] G. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1978) 1.
[20] Y. Sun, Z.J. Wang, Evaluation of discontinuous Galerkin and spectral volume methods for scalar and system conservation laws on unstructured grids, Int. J. Numer. Method Fluid 45 (2004) 819–838.
[21] Z.J. Wang, Spectral (finite) volume method for conservation laws on unstructured grids: basic formulation, J. Comput. Phys. 178 (2002) 210–251.
[22] Z.J. Wang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids. II: Extension to two-dimensional scalar equation, J. Comput. Phys. 179 (2002) 665–697.
[23] Z.J. Wang, L. Zhang, Y. Liu, Spectral (finite) volume method for conservation laws on unstructured grids. IV: Extension to two-dimensional systems, J. Comput. Phys. 194 (2004) 716–741.
[24] P. Woodward, P. Colella, Numerical simulation of two-dimensional fluid flows with strong shocks, J. Comput. Phys. 54 (1984) 115.
[25] Z.-L. Xu, Y.-J. Liu, C.-W. Shu, Hierarchical reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO type linear reconstruction and partial neighboring cells, J. Comput. Phys. 228 (2009) 2194–2212.
[26] M. Zhang, C.-W. Shu, An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods, Comput. Fluid 34 (2005) 581–592.