Journal of Computational Physics 228 (2009) 2194–2212

Contents lists available at ScienceDirect

Journal of Computational Physics

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

Hierarchical reconstruction for discontinuous Galerkin methods on unstructured grids with a WENO-type linear reconstruction and partial neighboring cells

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 10 January 2008
Received in revised form 17 November 2008
Accepted 26 November 2008
Available online 7 December 2008

Keywords:
Hierarchical reconstruction
Discontinuous Galerkin methods
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] is applied to the piecewise quadratic discontinuous Galerkin method on two-dimensional unstructured triangular grids. A variety of limiter functions have been explored in the construction of piecewise linear polynomials in every hierarchical reconstruction stage. We show that on triangular grids, the use of center biased limiter functions is essential in order to recover the desired order of accuracy. Several new techniques have been developed in the paper: (a) we develop a WENO-type linear reconstruction in each hierarchical level, which solves the accuracy degeneracy problem of previous limiter functions and is essentially independent of the local mesh structure; (b) we find that HR using partial neighboring cells significantly reduces over/under-shoots, and further improves the resolution of the numerical solutions. The method is compact and therefore easy to implement. Numerical computations for scalar and systems of nonlinear hyperbolic equations are performed. We demonstrate that the procedure can generate essentially non-oscillatory solutions while keeping the resolution and desired order of accuracy for smooth solutions.

1. Introduction

The discontinuous Galerkin (DG) method was first introduced in 1973 by Reed and Hill [19] as a technique to solve neutron transport problems. A major development of the DG method to solve nonlinear time dependent hyperbolic conservation laws (1.1) was carried out by Cockburn in a series of papers [5,4,3,6], in which they built a framework with spatial DG and Runge–Kutta time discretization (RKDG) to solve nonlinear time dependent hyperbolic conservation laws

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

* 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 Grants DMS-0511815 and DMS-0810913.
[2] Research supported in part by NSF Grants DMS-0510345 and DMS-0809086.

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.11.025

Here $\Omega \subset R^d, \mathbf{x} = (x_1, \ldots, x_d), d$ is the spatial dimension, $\mathbf{u} = (u_1, \ldots, u_m)^T$ and $\mathbf{F}_k(\mathbf{u}) = (F_{k,1}(\mathbf{u}), \cdots, F_{k,d}(\mathbf{u}))$ is the flux. Since hyperbolic conservation laws may develop discontinuous solutions, in their work, the explicit, total variation diminishing (TVD) Runge–Kutta time discretizations [24] are used in time. Exact or approximate Riemann solvers are used to compute fluxes across cell edges and the TVB projection limiters [21,5] are used to eliminate spurious oscillations near discontinuities.

The limiting for RKDG has a close connection to similar techniques used in finite volume schemes. However, since RKDG is compact, and evolves a polynomial in each cell along time (rather than a cell average as in a finite volume scheme), it would be ideal to have a limiting technique that is also compact and takes advantage of all available information from adjacent polynomials. In [5], the TVB projection limiter limits the variation between a cell edge value and its cell average by the differences between cell averages of the current and neighboring cells. High order Legendre moments (orders higher than one) are truncated in a cell if non-smoothness in the cell is detected. Biswas [1] proposed a moment limiter on the orthogonal expansion by taking appropriate derivatives and detecting the non-smoothness from the highest order moment to the lowest order one. The limiting process is applied when necessary from higher to lower moments. An improved moment limiter can be found in [2]. These techniques limit the moment within certain range and could hurt the accuracy if they were mistakenly activated in smooth regions of the solution. Also the limiting range for high order moments usually needs to be set on characteristic variables. The authors of [17,33] use weighted essentially non-oscillatory (WENO) finite volume methodology as limiters for RKDG for solving nonlinear hyperbolic conservation laws on two-dimensional structured and unstructured meshes, where the polynomials supported in the "trouble cells" are reconstructed by the WENO procedure out of nearby cell averages. In [16], Luo develop a Hermite WENO-based limiter for the second order RKDG method on unstructured meshes following [18]. We would also like to mention a WENO-type finite volume scheme by Dumbser [7] on two-dimensional unstructured triangular meshes as well as on three-dimensional tetrahedral meshes, where the high order reconstructed polynomial is made up of a convex combination of polynomials of the same order on sub-stencils.

In [14], a non-oscillatory hierarchical reconstruction (HR) method is introduced by Liu for the staggered central DG method to reconstruct polynomials computed by the DG method. This method computes cell averages of various orders of derivatives of a polynomial and uses them in the reconstruction of non-oscillatory linear polynomials in each hierarchical stage. The coefficients of the reconstructed linear polynomials are used to update the corresponding ones of the original polynomial. HR is compact and utilizes all the information from the polynomials in the given cell and its neighbors, without first truncating these polynomials. For the limiting methods based on cell average values, the size of the stencil generally has to increase with the increasing order of the reconstructed polynomials. Compared to these limiting methods, HR can be essentially independent of the shapes of the mesh cells. In each HR stage, only non-oscillatory conservative linear reconstructions are needed, such as the MUSCL or second order ENO reconstruction used in [14], which simplifies the implementation.

In this paper, we extend the non-oscillatory hierarchical reconstruction method to reconstruct DG solutions on two-dimensional unstructured meshes and develop several new techniques. Our limiting procedure in a cell only involves adjacent cells sharing edges with it (same as DG), which is supposed to be independent of the degree of supported polynomials. In particular, (a) we introduce a weighted linear reconstruction for each hierarchical step in the spirit of the harmonic average of one-sided slope approximations [27,28], modified ENO [22] and the WENO schemes [13,10,7]; and (b) we develop a partial neighboring cell technique for HR, inspired by Wang and Liu and Wang et al. [30,31]. The basic idea for using partial neighboring cells is very simple. When apply HR to a cell, say $\mathcal{K}_0$, instead of using adjacent cells directly, we first divide these adjacent cells into subcells (e.g. with half of the original size), and then only use subcells adjacent to (sharing an edge with) cell $\mathcal{K}_0$. This procedure allows the remainder term in HR to be extended over a shorter distance and improves the result.

Numerical tests are presented. We show that this method is robust and is easy to implement.

This paper is organized as follows. Section 2 describes the RKDG procedure and the limiting procedure. Numerical tests are presented in Section 3. Concluding remarks are included in Section 4.

2. Algorithm formulation

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

2.1. Spatial discretization

First, the physical domain Ω is partitioned into a collection of $\mathcal{N}$ triangular cells

$$\Omega = \cup_{i=1}^{\mathcal{N}} \mathcal{K}_i \text{ and } \mathcal{T}_h = \{\mathcal{K}_i : i = 1, \ldots, \mathcal{N}\}. \tag{2.1}$$

We choose polynomial basis functions of degree q in a cell $\mathcal{K}_i$ to be monomials of multidimensional Taylor expansions about cell centroids. For the convenience of computation, in two-dimensional space, we consider a right-triangular reference cell $\mathcal{K}$ as shown in Fig. 1. For example, on $\mathcal{K}$, the basis functions in terms of (ξ, η) are

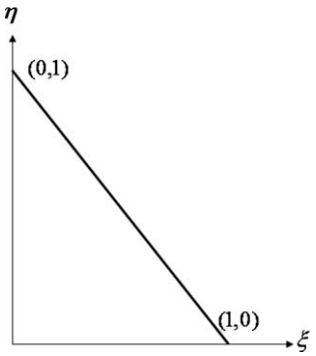


Fig. 1. Reference triangular cell $\mathcal{K}$.

$$B = \{b_m(\xi-\xi_0,\eta-\eta_0), m=1,\ldots,N_q\} = \{1,\xi-\xi_0,\eta-\eta_0,(\xi-\xi_0)^2,(\xi-\xi_0)(\eta-\eta_0),(\eta-\eta_0)^2,\ldots,(\eta-\eta_0)^q\}, \quad (2.2)$$

where $N_q=(q+1)(q+2)/2$, and (ξ_0,η_0) is the centroid of $\mathcal{K}$. Any function f can be approximated by the basis functions in $\mathcal{K}$ as

$$f(\xi,\eta)=\sum_{m=1}^{N_q} f_m b_m(\xi-\xi_0,\eta-\eta_0). \quad (2.3)$$

The inner product of b_m and b_n on $\mathcal{K}$ is

$$(b_m,b_n)=\int_0^1\int_0^{1-\xi} b_m b_n d\eta d\xi, \quad (2.4)$$

which can be computed by

$$\int_0^1\int_0^{1-\xi}\xi^m\eta^n d\eta d\xi = \frac{1}{n+1}\left[\sum_{l=0}^{n+1} C_{n+1}^l(-1)^l\frac{1}{m+l+1}\right]$$

where C_{n+1}^l are the binomial coefficients.

With the help of the reference cell, the integration of basis functions in the (x,y) coordinates can now be calculated easily. For a general triangular cell $\mathcal{K}_i$, the basis set in the (x,y) coordinates is

$$\begin{aligned}\mathcal{B} &= \{g_m(x-x_i,y-y_i), m=1,\ldots,N_q\}\\ &= \{1,x-x_i,y-y_i,(x-x_i)^2,(x-x_i)(y-y_i),(y-y_i)^2,\ldots,(y-y_i)^q\},\end{aligned} \quad (2.5)$$

where $\mathbf{x}_i\equiv(x_i,y_i)$ is the centroid of $\mathcal{K}_i$.

We employ a linear transformation to map (ξ,η) of $\mathcal{K}$ to (x,y) of a cell $\mathcal{K}_i$

$$\begin{aligned}x &= (\hat{x}_2-\hat{x}_1)\xi+(\hat{x}_3-\hat{x}_1)\eta+\hat{x}_1\\ y &= (\hat{y}_2-\hat{y}_1)\xi+(\hat{y}_3-\hat{y}_1)\eta+\hat{y}_1,\end{aligned} \quad (2.6)$$

where $(\hat{x}_1,\hat{y}_1),(\hat{x}_2,\hat{y}_2)$ and $(\hat{x}_3,\hat{y}_3)$ are the coordinates of the three vertices of cell $\mathcal{K}_i$ (ordered counter-clock-wisely). All double integrals are evaluated in the reference domain

$$\int\int_{\mathcal{K}_i} dydx = \int\int_{\mathcal{K}}\left|\frac{\partial(x,y)}{\partial(\xi,\eta)}\right| d\xi d\eta. \quad (2.7)$$

The semi-discrete DG formulation of the kth equation of (1.1) is to find a piecewise polynomial solution u_h (neglecting its subscript k for convenience) of degree q such that

$$\frac{d}{dt}\int_{\mathcal{K}_i} u_h v_h d\mathbf{x} + \int_{\partial\mathcal{K}_i}\mathbf{F}_k(\mathbf{u}_h)\cdot\mathbf{n}_i v_h d\Gamma - \int_{\mathcal{K}_i}\mathbf{F}_k(\mathbf{u}_h)\cdot\nabla v_h d\mathbf{x} = 0, \quad (2.8)$$

for any piecewise polynomial function v_h of degree q. Here $\mathbf{n}_i$ is the outward unit normal vector of the cell boundary $\partial\mathcal{K}_i$. Let u_h be expressed as

$$u_h(\mathbf{x},t)=\sum_{m=1}^{N_q} u_{m,i}(t) g_m(x-x_i,y-y_i), \mathbf{x}\in\mathcal{K}_i, \quad i=1,\ldots,\mathcal{N}. \quad (2.9)$$

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

Taking the test function v_h to be one of the basis functions g_n in the cell $\mathcal{K}_i$, we obtain a system of N_q equations for $\mathcal{K}_i$

$$\sum_{m=1}^{N_q} \frac{du_m}{dt} \int_{\mathcal{K}_i} g_m g_n d\mathbf{x} + \int_{\partial\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h)\cdot\mathbf{n}_i g_n d\Gamma - \int_{\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h)\cdot\nabla g_n d\mathbf{x} = 0, 1 \leqslant n \leqslant N_q \tag{2.10}$$

by substituting u_h with (2.9). Since the approximated solution u_h is discontinuous across cell edges, fluxes are not uniquely determined. The flux function $\mathbf{F}_k(\mathbf{u}_h)\cdot\mathbf{n}_i$ appearing in Eq. (2.10) is replaced by a numerical flux function (the Lax-Friedrich flux, see e.g. [23]) defined by

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

where α is the largest characteristic speed,

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

The domain and the boundary integrals in Eq. (2.10) are computed with $(2q)$th and $(2q+1)$th order accurate Gaussian quadrature rules respectively to preserve the $(q+1)$th order of accuracy of the finite element space discretization. For a 3rd order accurate scheme, the quadrature rule for the domain integral is

$$\int_{\mathcal{K}} g(\mathbf{x})d\mathbf{x} = \sum_{i=1}^{3} g(a^i)\frac{|\mathcal{K}|}{20} + \sum_{1\leqslant i<j\leqslant 3}^{3} g(a^{ij})\frac{2|\mathcal{K}|}{15} + g(a^0)\frac{9|\mathcal{K}|}{20}, \tag{2.11}$$

where a^0 is the centroid, a^i is the vertex, and a^{ij} is the mid-point of the edge connecting a^i and a^j. The quadrature rule for the boundary integral is

$$\int_{-1}^{1} f(x)dx = \frac{5f\left(-\sqrt{3/5}\right) + 8f(0) + 5f\left(-\sqrt{3/5}\right)}{9}. \tag{2.12}$$

2.2. Time integration

Eq. (2.10) is integrated in time by the widely used three-stage (3rd order) TVD Runge–Kutta method [24]. The CFL number is chosen to be 0.1 which is less than $\frac{1}{2q+1}$ to satisfy the stability requirement.

2.3. Limiting by hierarchical reconstruction

Without an appropriate limiting procedure, the DG method will produce non-physical oscillations in the vicinity of discontinuities. We use the hierarchical reconstruction introduced in [14], which processes the DG solution at each Runge–Kutta stage to eliminate such spurious oscillations. We refer to [14,15] for the implementations of HR for central DG and finite volume schemes. In this section, we first give a general description of HR in Section 2.3.1. We then describe HR without using partial neighboring cells in Section 2.3.2. HR with partial neighboring cells is described in Section 2.4.

2.3.1. A general description of hierarchical reconstruction

Suppose we are given a piecewise polynomial (of degree q) numerical solution at a time. Let $\mathcal{K}_I$ in R^d be the cell under consideration and the set $\{\mathcal{K}_J\}$ be the collection of cells adjacent to cell $\mathcal{K}_I$ and $\mathcal{K}_I$ itself. Let $\mathbf{x}_J$ be the cell centroid of cell $\mathcal{K}_J$. The polynomial solution in cell $\mathcal{K}_J$ can be written in the Taylor expansion

$$u_J(\mathbf{x}-\mathbf{x}_J) = \sum_{m=0}^{q}\sum_{|\mathbf{m}|=m} \frac{1}{\mathbf{m}!} u_J^{(\mathbf{m})}(\mathbf{0})(\mathbf{x}-\mathbf{x}_J)^{\mathbf{m}}.$$

$u_I(\mathbf{x}-\mathbf{x}_I)$ could be oscillatory if located near a discontinuity of the weak solution. Therefore, all its coefficients need to be recomputed. Corresponding to $u_I^{(\mathbf{m})}(\mathbf{0})$, the recomputed new coefficient is denoted by $\tilde{u}_I^{(\mathbf{m})}(\mathbf{0})$. Sometimes multiple new values are computed for $u_I^{(\mathbf{m})}(\mathbf{0})$. They are called candidates, still denoted by $\tilde{u}_I^{(\mathbf{m})}(\mathbf{0})$ to avoid creating too many new notations, and a limiter is used in the end to obtain the new coefficient from its candidates.

If the polynomials were linear, we could have used their cell averages in cells $\{\mathcal{K}_J\}$ to recompute the gradient of the linear polynomial in cell $\mathcal{K}_I$, through a MUSCL, second order ENO or other compact linear reconstructions, which usually is easy to adapt to irregular meshes. HR [14] decomposes the job of recomputing coefficients of a high order polynomial into a series of smaller jobs, each of which is to recompute the gradient of a linear polynomial in cell $\mathcal{K}_I$ by using its approximate cell averages in adjacent cells $\{\mathcal{K}_J\}$. For example, to recompute coefficients in the mth degree terms of $u_I(\mathbf{x}-\mathbf{x}_I)$, we take a $(m-1)$th partial derivative of $u_I(\mathbf{x}-\mathbf{x}_I)$ to obtain $\partial^{m-1}u_I(\mathbf{x}-\mathbf{x}_I) = L_I(\mathbf{x}-\mathbf{x}_I) + R_I(\mathbf{x}-\mathbf{x}_I)$, where $L_I(\mathbf{x}-\mathbf{x}_I)$ is the linear part and $R_I(\mathbf{x}-\mathbf{x}_I)$ is the remainder. It is easy to see that the gradient of $L_I(\mathbf{x}-\mathbf{x}_I)$ is made up of some $u_I^{(\mathbf{m})}(\mathbf{0})$ subject to $|\mathbf{m}| = m$. Therefore reconstructing a new gradient for L_I also recomputes new values for some coefficients in the mth degree terms of $u_I(\mathbf{x}-\mathbf{x}_I)$. In order to reconstruct a new gradient for L_I, we need to figure out average values of L_I over cells $\{\mathcal{K}_J\}$ by using

the given polynomials on $\{\mathcal{K}_J\}$ and the remainder R_J. These cell averages provide redundant information for a non-oscillatory reconstruction of the new gradient of L_I. HR recomputes the new coefficients of $u_I(\mathbf{x}-\mathbf{x}_I)$ iteratively from the highest to the lowest degree terms. We describe the general procedure of HR as follows:

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

(a) For all J, take the same $(m-1)$th order partial derivative for $u_J(\mathbf{x}-\mathbf{x}_J)$ to obtain the polynomial $\partial^{m-1}u_J(\mathbf{x}-\mathbf{x}_J)$. In particular, denote $\partial^{m-1}u_I(\mathbf{x}-\mathbf{x}_I) = L_I(\mathbf{x}-\mathbf{x}_I) + R_I(\mathbf{x}-\mathbf{x}_I)$, where $L_I(\mathbf{x}-\mathbf{x}_I)$ is the linear part of $\partial^{m-1}u_I(\mathbf{x}-\mathbf{x}_I)$ and $R_I(\mathbf{x}-\mathbf{x}_I)$ is the remainder.

(b) For all J, calculate the average value of $\partial^{m-1}u_J(\mathbf{x}-\mathbf{x}_J)$ on cell $\mathcal{K}_J$ to obtain $\overline{\partial^{m-1}u_J}$.

(c) Let $\widetilde{R}_I(\mathbf{x}-\mathbf{x}_I)$ be the $R_I(\mathbf{x}-\mathbf{x}_I)$ with its coefficients replaced by the recomputed new coefficients. Compute the average value of $\widetilde{R}_I(\mathbf{x}-\mathbf{x}_I)$ on cell $\mathcal{K}_J$ to obtain $\overline{R}_J$, for all J.

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

(e) Use a non-oscillatory procedure to reconstruct a new gradient of $L_I(\mathbf{x}-\mathbf{x}_I)$ by using cell averages $\{\overline{L}_J\}$, which also provides candidates for the corresponding $u_I^{(\mathbf{m})}(0)$'s, $|\mathbf{m}| = m$.

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

Step 2. In order to recompute the new coefficients in the zero-th and first degree terms of $u_I(\mathbf{x}-\mathbf{x}_I)$, we perform the procedure of step 1 (a)–(f) with $m = 1$, and the new coefficient in the zero-th degree term of $u_I(\mathbf{x}-\mathbf{x}_I)$ is determined by the invariance of the cell average of $u_I(\mathbf{x}-\mathbf{x}_I)$ on the cell $\mathcal{K}_I$ with new coefficients.

2.3.2. Hierarchical reconstruction for the P^2 DG solution

Since we use 2nd degree polynomials in our calculation, we describe the implementation of HR for piecewise quadratic finite element space on the triangular cells and the new piecewise linear polynomial reconstruction procedure in this section.

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

$$\begin{aligned} u_j(x-x_j, y-y_j) = {} & u_j(0,0) + \partial_x u_j(0,0)(x-x_j) + \partial_y u_j(0,0)(y-y_j) \\ & + \frac{1}{2}\partial_{xx}u_j(0,0)(x-x_j)^2 + \partial_{xy}u_j(0,0)(x-x_j)(y-y_j) \\ & + \frac{1}{2}\partial_{yy}u_j(0,0)(y-y_j)^2, \end{aligned} \tag{2.13}$$

where (x_j, y_j) is the cell centroid of $\mathcal{K}_j, j = 0, \ldots, 3$. We will reconstruct a new polynomial in $\mathcal{K}_0$ with a point-wise error $\mathcal{O}(h^3)$, where h is the triangle edge length.

According to HR, we first take the 1st partial derivative with respect to x for $u_j(x-x_j, y-y_j)$ to obtain

$$L_j(x-x_j, y-y_j) = \partial_x u_j(0,0) + \partial_{xx}u_j(0,0)(x-x_j) + \partial_{xy}u_j(0,0)(y-y_j), \quad j = 0,1,2,3. \tag{2.14}$$

Calculate the cell average of $L_j(x-x_j, y-y_j)$ on cell $\mathcal{K}_j$ to obtain $\overline{L}_j = \partial_x u_j(0,0), \quad j = 0,1,2,3$. Then we apply a WENO-type reconstruction procedure to the cell averages $\overline{L}_j$, which is described in Section 2.3.3, to obtain a new linear polynomial $\widetilde{L}_0(x-x_0, y-y_0)$ on cell $\mathcal{K}_0$:

$$\widetilde{L}_0(x-x_0, y-y_0) = \partial_x \tilde{u}_0(0,0) + \partial_{xx}\tilde{u}_0(0,0)(x-x_0) + \partial_{xy}\tilde{u}_0(0,0)(y-y_0) \tag{2.15}$$

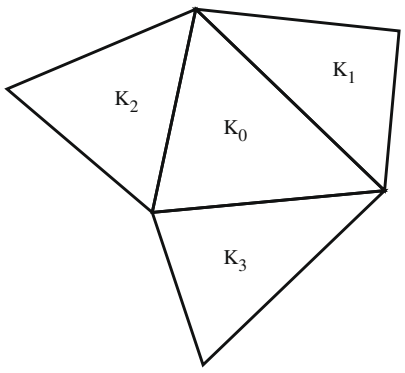


Fig. 2. Schematic of 2D HR for cell $\mathcal{K}_0$.

with recomputed coefficients $\partial_{xx}\tilde{u}_0(0,0), \partial_{xy}\tilde{u}_0(0,0)$ and $\partial_x\tilde{u}_0(0,0) = \bar{L}_0$. Here $\partial_{xx}\tilde{u}_0(0,0)$ and $\partial_{xy}\tilde{u}_0(0,0)$ are candidates for coefficients $\partial_{xx}u_0(0,0)$ and $\partial_{xy}u_0(0,0)$ of the reconstructed polynomial, respectively.

We then take the 1st partial derivative with respect to y for $u_j(x-x_j, y-y_j)$ to redefine $L_j(x-x_j, y-y_j)$ to be

$$L_j(x-x_j, y-y_j) = \partial_y u_j(0,0) + \partial_{xy}u_j(0,0)(x-x_j) + \partial_{yy}u_j(0,0)(y-y_j), \quad j = 0,\ldots,3, \tag{2.16}$$

and similarly calculate their cell averages and perform the same WENO-type reconstruction procedure to obtain another linear polynomial (still denoted by $\widetilde{L}_0$) on $\mathcal{K}_0$,

$$\widetilde{L}_0(x-x_0, y-y_0) = \partial_y\tilde{u}_0(0,0) + \partial_{xy}\tilde{u}_0(0,0)(x-x_0) + \partial_{yy}\tilde{u}_0(0,0)(y-y_0), \tag{2.17}$$

with the recomputed coefficients $\partial_{xy}\tilde{u}_0(0,0)$ and $\partial_{yy}\tilde{u}_0(0,0)$ being candidates for the coefficients $\partial_{xy}u_0(0,0)$ and $\partial_{yy}u_0(0,0)$ of the reconstructed polynomial, respectively.

From the above procedure, each of $\partial_{xx}u_0(0,0)$ and $\partial_{yy}u_0(0,0)$ of the reconstructed polynomial has only one candidate, which will be its new value. However, $\partial_{xy}u_0(0,0)$ has two candidates. Even though $\partial_{xy}\tilde{u}_0(0,0)$ appears in both Eqs. (2.15) and (2.17) for simplicity of notations, it represents different candidate values of $\partial_{xy}u_0(0,0)$ of the reconstructed polynomial. The new value of $\partial_{xy}u_0(0,0)$ (still denoted by $\partial_{xy}\tilde{u}_0(0,0)$) can be obtained by applying a limiter function to its candidates, which will be described later in Section 2.3.3.

To perform Step 2 of HR, we first compute the cell average of $u_j(x-x_j, y-y_j)$ on cell $\mathcal{K}_j$ to obtain $\bar{u}_j, \quad j = 0,1,2,3$. And then compute cell averages of the updated remainder polynomial

$$\widetilde{R}_0(x-x_0, y-y_0) = \frac{1}{2}\partial_{xx}\tilde{u}_0(0,0)(x-x_0)^2 + \partial_{xy}\tilde{u}_0(0,0)(x-x_0)(y-y_0) + \frac{1}{2}\partial_{yy}\tilde{u}_0(0,0)(y-y_0)^2 \tag{2.18}$$

on cells $\{\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ to obtain $\bar{R}_0, \bar{R}_1, \bar{R}_2, \bar{R}_3$, respectively. Note that it is the remainder supported on cell $\mathcal{K}_0$ whose cell averages on adjacent cells are being taken. Also the coefficients of the remainder have been updated with the corresponding new values. Redefine $\bar{L}_j = \bar{u}_j - \bar{R}_j, \quad j = 0,1,2,3$. The same WENO-type reconstruction procedure is applied to cell averages $\{\bar{L}_j\}$ to obtain new coefficients $\partial_x\tilde{u}_0(0,0)$ and $\partial_y\tilde{u}_0(0,0)$. Finally let the new coefficient $\tilde{u}_0(0,0) = \bar{L}_0$ to ensure invariance of the cell average of $u_0(x-x_0, y-y_0)$ on cell $\mathcal{K}_0$ with the new coefficients.

2.3.3. Weighted essentially non-oscillatory (WENO)-type reconstruction procedure

Now we describe the WENO-type linear reconstruction procedure used in Section 2.3.2 in each hierarchical level to obtain the new coefficients $\partial_{xx}\tilde{u}_0(0,0)$ and $\partial_{yy}\tilde{u}_0(0,0), \partial_{xy}\tilde{u}_0(0,0), \partial_x\tilde{u}_0(0,0)$ and $\partial_y\tilde{u}_0(0,0)$ respectively. The polynomial reconstruction using this WENO-type linear reconstruction procedure is referred to as HR with the WENO-type linear reconstruction in the numerical example section – Section 3.

In [15], three types of limiter functions are used in the linear reconstruction procedure to obtain these new coefficients at each hierarchical level. The minmod limiter function defined by

$$\mathrm{m}(c_1, c_2, \ldots, c_r) = \begin{cases} \min\{c_1, c_2, \ldots, c_r\}, & \text{if } c_1, c_2, \ldots, c_r > 0, \\ \max\{c_1, c_2, \ldots, c_r\}, & \text{if } c_1, c_2, \ldots, c_r < 0, \\ 0, & \text{otherwise}, \end{cases} \tag{2.19}$$

gives the MUSCL reconstruction [27,29]; the limiter function defined by

$$\mathrm{m}_2(c_1, c_2, \ldots, c_r) = c_j, \text{where } j \in [1,r] \text{satisfies } |c_j| = \min\{|c_1|, |c_2|, \ldots, |c_r|\}, \tag{2.20}$$

gives the ENO [8] reconstruction; and the center biased minmod limiter m_b and ENO limiter m_{2b} can be formulated as

$$\begin{aligned} \mathrm{m}_b(c_1, c_2, \ldots, c_r) &= m\left((1+\varepsilon)m(c_1, c_2, \ldots, c_r), \tfrac{1}{r}\sum_{i=1}^{r} c_i\right), \\ \mathrm{m}_{2b}(c_1, c_2, \ldots, c_r) &= m_2\left((1+\varepsilon)m_2(c_1, c_2, \ldots, c_r), \tfrac{1}{r}\sum_{i=1}^{r} c_i\right), \end{aligned} \tag{2.21}$$

where $\varepsilon > 0$ is a small perturbation number. The linear reconstruction procedure using these limiter functions works very well on the rectangular and staggered grids [15], and the polynomial reconstruction using the minmod or ENO type limiter functions is referred to as HR with the minmod or ENO type limiter functions below and in the numerical example section – Section 3.

However, our numerical experiments show that on triangular meshes, HR using the minmod or ENO limiter function often fails to give the desired order of accuracy. The reason of this failure might stem from the abrupt shift of stencils which reduces the smoothness of the numerical flux [20,22], which seems to be more significant for triangular meshes. While the linear reconstruction procedure with a center biased minmod limiter or ENO limiter function could give the desired order of accuracy, the value of ε often needs to be large on triangular meshes, which introduces significant overshoots and undershoots. See Section 3 for related numerical experiments.

To overcome this problem, we introduce a new linear reconstruction method based upon weighted combination of functions which follows the line of [27,28,22,13,10]. The new WENO-type reconstruction procedure proceeds as follows.

We use the reconstruction of the polynomial (2.15) as an example. First, we use cells $\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2$ and $\mathcal{K}_3$ to form three stencils: $\{\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2\}, \{\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_3\}$ and $\{\mathcal{K}_0, \mathcal{K}_2, \mathcal{K}_3\}$. On the first stencil, we can reconstruct a linear polynomial (denoted by $L_{0,1}$) by letting it have the three average values of the stencil, i.e., by solving the following equations for $\partial_{xx}u_{0,1}(0,0)$ and $\partial_{xy}u_{0,1}(0,0)$

$$\frac{1}{|\mathcal{K}_j|}\int_{\mathcal{K}_j} L_{0,1}(x-x_0, y-y_0)dxdy \equiv \bar{L}_0 + \partial_{xx}u_{0,1}(0,0)(x_j - x_0) + \partial_{xy}u_{0,1}(0,0)(y_j - y_0) = \bar{L}_j, \tag{2.22}$$

where $j = 1, 2$, similarly for the other two stencils.

Let the linear polynomials computed from these three stencils be $L_{0,1}(x-x_0, y-y_0), L_{0,2}(x-x_0, y-y_0)$ and $L_{0,3}(x-x_0, y-y_0)$ respectively. The first degree coefficients of these linear polynomials are $\partial_{xx}u_{0,1}(0,0), \partial_{xy}u_{0,1}(0,0); \partial_{xx}u_{0,2}(0,0), \partial_{xy}u_{0,2}(0,0)$; and $\partial_{xx}u_{0,3}(0,0), \partial_{xy}u_{0,3}(0,0)$, respectively.

The reconstructed linear polynomial (2.15) is a convex combination of these computed polynomials, i.e.,

$$\widetilde{L}_0(x-x_0, y-y_0) = \sum_{r=1}^{3} w_r L_{0,r}(x-x_0, y-y_0). \tag{2.23}$$

For stability and consistency, the weight w_r depends on $L_{0,r}$ and satisfies

$$w_r \geqslant 0, \text{for } r = 1, 2, 3; \text{and } \sum_{r=1}^{3} w_r = 1. \tag{2.24}$$

Other considerations for designing the weights are: (1) when a stencil contains a discontinuity of the solution, the corresponding weight will be essentially 0; and (2) the weights are smooth functions of involved cell averages. We set

$$w_r = \frac{\alpha_r}{\sum_{s=1}^{3}\alpha_s}, \quad r = 1, 2, 3, \tag{2.25}$$

where α_s are to be defined later. Let

$$d_r = \frac{1/\theta_r}{\sum_{s=1}^{3} 1/\theta_s}, \tag{2.26}$$

where $\theta_r = ||A||||A^{-1}||$ is the condition number of the rth stencil, A is the coefficient matrix of the linear system (2.22) for the rth stencil, $||\cdot||$ denotes the 1-norm. This choice of d_r puts condition numbers of stencils into consideration so that candidates of new coefficients computed from an ill-conditioned stencil have less weights. Let

$$\alpha_r = \frac{d_r}{1 + h\beta_r}, \tag{2.27}$$

where

$$\beta_r = (\partial_{xx}u_{0,r}(0,0))^2 + (\partial_{xy}u_{0,r}(0,0))^2. \tag{2.28}$$

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

$$\partial_{xx}\tilde{u}_0(0,0) = \begin{cases} \min_{r=1}^{3} w_r \partial_{xx}u_{0,r}(0,0), & \text{if } \bar{L}_{\min} < \bar{L}_0 < \bar{L}_{\max}, \\ 0, & \text{otherwise}, \end{cases} \tag{2.29}$$

where $\bar{L}_{\min} = \min\{\bar{L}_j : j = 0, \cdots, 3\}$ and $\bar{L}_{\max} = \max\{\bar{L}_j : j = 0, \cdots, 3\}$. In fact, violation of $\bar{L}_{\min} < \bar{L}_0 < \bar{L}_{\max}$ indicates a local extreme value. The candidate $\partial_{xy}\tilde{u}_0$ is similarly determined.

The reconstruction of function (2.17) also follows the above procedure. After the reconstruction of functions (2.15) and (2.17), $\partial_{xx}\tilde{u}_0(0,0)$ and $\partial_{yy}\tilde{u}_0(0,0)$ are the corresponding new values in (2.13) for the polynomial $u_0(x-x_0, y-y_0)$. However, the reconstruction of functions (2.15) and (2.17) leaves us two choices for the coefficient $\partial_{xy}u_0(0,0)$, each of which comes from (2.15) and (2.17), respectively. 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, for the linear reconstruction involving $\partial_x u_0(0,0)$ and $\partial_y u_0(0,0)$, the following weight from [23] is used:

$$\alpha_r = \frac{d_r}{(\epsilon_1 + \beta_r)^2}, \tag{2.30}$$

where β_r is the "smoothness indicator" of the rth stencil similar to those in the WENO scheme,

$$\beta_r = (\partial_x u_{0,r}(0,0))^2 + (\partial_y u_{0,r}(0,0))^2, \tag{2.31}$$

$\partial_x u_{0,r}(0,0)$ and $\partial_y u_{0,r}(0,0)$) are the first degree coefficients determined in stencil r by an equation similar to (2.22) as we repeat the linear reconstruction procedure in Step 1 of HR. $\epsilon_1 > 0$ is a small number keeping the denominator away from 0. (Note that in Step 2 one can also adopt the weight

$$\alpha_r = \frac{d_r}{1 + (\beta_r)^2} \tag{2.32}$$

which is similar to (2.27). However it gives slightly bigger overshoots/undershoots.)

A function similar to (2.29) is used to determine the new coefficients $\partial_x \tilde{u}_0(0,0)$ and $\partial_y \tilde{u}_0(0,0)$ in (2.13) for the function $u_0(x - x_0, y - y_0)$. However, the extreme value detector (i.e., the "0" case in (2.29)) is not applied here.

The reason that we choose different forms of weights is as follows. Since the low order coefficients are more accurate and less sensitive to the shift of stencils, we can put more weights to the smoother stencil to damp out oscillations without the loss of accuracy. On the other hand, the high order coefficients (the second degree coefficients in the present paper) are less accurate and are more sensitive to the shift of stencils. We want these high order coefficients to be set closer to the mean values of the ones computed on different stencils, which reduces the abrupt shift of stencils.

An error analysis shows that $w_r \partial_{xx} u_{0,r}(0,0)$ is of $\mathcal{O}(h)$ where there is a discontinuity, provided that there is at least one other stencil in smooth region. This damps the oscillatory term to the required approximation error size of the second degree coefficients. In fact, Since $\partial_{xx} u_{0,r}(0,0), \partial_{xy} u_{0,r}(0,0)$ and $\partial_{yy} u_{0,r}(0,0)$ are of $\mathcal{O}\left(\frac{1}{h^2}\right)$ at discontinuities, β_r is between $\mathcal{O}(1)$ if the solution is smooth, and $\mathcal{O}(\frac{1}{h^4})$ if there is a discontinuity. Therefore α_r is between $\mathcal{O}(h^3)$ and $\mathcal{O}(1)$ from (2.27). And $w_r \partial_{xx} u_{0,r}(0,0)$ is of $\mathcal{O}(h)$ at a discontinuity, provided that at least one of the other stencils is in smooth region. When all stencils are in non-smooth regions, the zero case of (2.29) effectively damps out spurious oscillations.

For systems, we perform the reconstruction on conservative variables (component-wise) and achieve satisfactory results.

2.4. Limiting by hierarchical reconstruction using partial neighboring cells

To further improve the resolution of numerical solutions and reduce over/under-shoots, we introduce the idea of HR with partial neighboring cells, which is a slight modification of the method introduced in Section 2.3. HR using partial neighboring cells proceeds as follows (see Fig. 3).

Suppose we are going to apply HR on cell $\mathcal{K}_0$. First partition cells $\mathcal{K}_i \in \{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ into four similar subcells, denoted by $\mathcal{K}_{ij}, j = 0, 1, 2, 3$. These subcells are formed by connecting the mid-points of edges of triangular elements. The polynomial defined on a subcell is the same polynomial defined on the original cell containing the subcell.

To perform HR on cell $\mathcal{K}_0$, only subcells sharing an edge with cell $\mathcal{K}_0$ are used. Therefore, the set of involved cells consists of $\{\mathcal{K}_0, \mathcal{K}_{10}, \mathcal{K}_{12}, \mathcal{K}_{20}, \mathcal{K}_{22}, \mathcal{K}_{30}, \mathcal{K}_{32}\}$. We then apply the reconstruction algorithm described in Section 2.3 to this set of cells to reconstruct a new polynomial on cell $\mathcal{K}_0$. For example, we take the 1st partial derivative with respect to x for the polynomial supported on each cell in this set to obtain (2.14). The new reconstructed linear polynomial (2.15) now is a convex combination of six linear polynomials, which come from stencils $\{\mathcal{K}_0, \mathcal{K}_{10}, \mathcal{K}_{12}\}, \{\mathcal{K}_0, \mathcal{K}_{12}, \mathcal{K}_{20}\}, \{\mathcal{K}_0, \mathcal{K}_{20}, \mathcal{K}_{22}\}, \{\mathcal{K}_0, \mathcal{K}_{22}, \mathcal{K}_{30}\}$, $\{\mathcal{K}_0, \mathcal{K}_{30}, \mathcal{K}_{32}\}$ and $\{\mathcal{K}_0, \mathcal{K}_{32}, \mathcal{K}_{10}\}$, respectively.

2.5. Local limiting procedure

Since shock waves and 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 use a local limiting procedure which adopts the limiter in [6] to identify "bad cells", i.e., cells which may contain oscillatory solutions.

We examine the DG solution in a cell by checking its value at the cell edge,

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

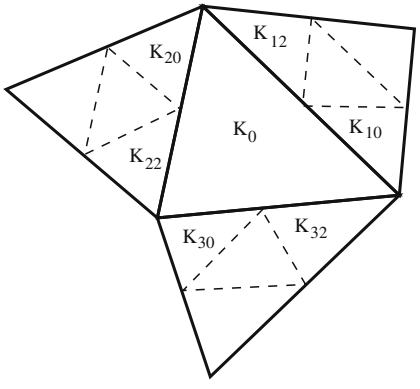


Fig. 3. Schematic of 2D HR using labeled partial neighboring cells for element $\mathcal{K}_0$.

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

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

where m is the minmod function, $\triangle\bar{u} = \bar{u}_1 - \bar{u}$, $\bar{u}_1$ is the cell average value in the adjacent cell sharing the edge, and $\mu > 1$. We take $\mu = 1.2$ in our numerical examples. If the minmod function returns other than the first argument, this cell is identified as a "bad cell". The limiting process is applied to these "bad cells" while keeping the computed DG solutions unchanged in other cells. With this local procedure, the limiting process is usually within 10% of the total CPU time for our numerical examples.

3. Numerical examples

In this section, we present results from HR with and without using partial neighboring cells, respectively.

We first study the limiter functions and test the capability of the method to achieve the desired 3rd order accuracy, using the scalar Burgers' equation and the Euler equations for gas dynamics. In two-dimensional space, the Euler equations can be expressed in the conservative form

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

where $\mathbf{u} = (\rho, \rho u, \rho v, E), f(\mathbf{u}) = (\rho u, \rho u^2 + p, \rho uv, u(E+p))$, and $g(\mathbf{u}) = (\rho v, \rho uv, \rho v^2 + p, v(E+p))$. 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 equations for gas dynamics.

3.1. Accuracy test cases for HR without using partial neighboring cells

3.1.1. Numerical errors for smooth solutions of the Burgers' equation

We start with the two-dimensional Burgers' equation with periodic boundary conditions:

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

where the domain $\Omega = [-1,1]\times[-1,1]$. At $T = 0.1$ the exact solution is smooth. For simplicity, HR with the ENO-type limiter functions (2.20) and (2.21) to do the linear reconstruction is performed on structural triangular meshes obtained by adding one diagonal line in each rectangle, see Fig. 4(a). h in Tables 1–6 represents the triangle edge length. The errors presented are for the cell averages of u.

HR with ENO limiter functions. The accuracy results are shown in Table 1 for HR using the ENO type limiter function (2.20). The accuracy results are shown in Tables 2–4 for HR with the center biased ENO limiter function in Eq. (2.21). This test problem shows that with the ENO limiter function, only 2nd order accuracy is achieved due to the abrupt shift of stencils. This problem can be overcome by using the center biased ENO limiter function. We clearly see that the order of accuracy increases with increasing values of ε in the center biased ENO limiter. However, to achieve 3rd order accuracy (in L_1 norm), we need $\varepsilon = 3$ in Table 4, which is too large and produces excessive oscillations when the solution contains discontinuities. The center biased minmod limiter function in (2.21) also has similar behavior.

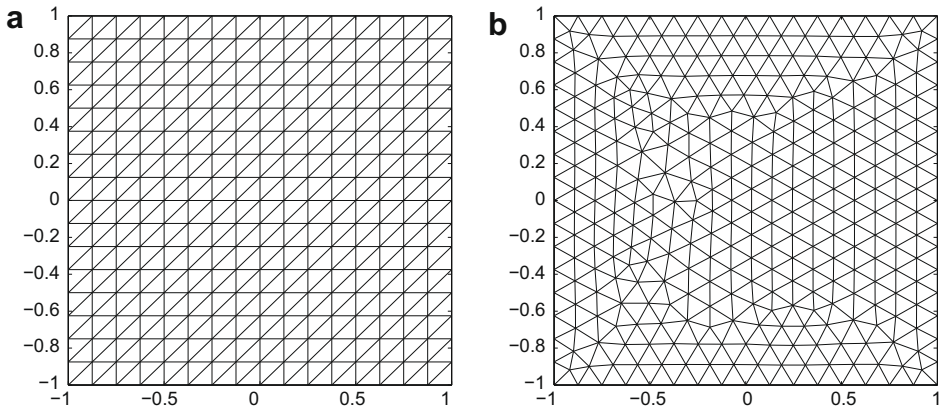


Fig. 4. Representative meshes for accuracy test: (a) Mesh for accuracy test of the Burgers' equation with ENO and center biased ENO-type limiter functions and (b) mesh for accuracy test of the Burgers' equation and the Euler equations with WENO-type reconstruction.

Table 1
Accuracy of ENO limiter (2.20) for 2D Burgers' equation.

h	L_1 error	Order	L_∞ error	Order
1/4	1.82E−1	–	9.28E−2	–
1/8	6.11E−2	1.57	3.17E−2	1.55
1/16	1.89E−2	1.69	1.39E−2	1.19
1/32	5.06E−3	1.90	5.96E−3	1.22
1/64	1.63E−3	1.64	3.10E−3	0.94
1/128	4.69E−4	1.80	1.48E−3	1.07

Table 2
Accuracy of biased ENO limiter (2.21) for 2D Burgers' equation, $\varepsilon = 0.1$.

h	L_1 error	Order	L_∞ error	Order
1/4	1.66E−1	–	8.53E−2	–
1/8	5.05E−2	1.72	2.81E−2	1.60
1/16	1.45E−2	1.80	1.23E−2	1.19
1/32	3.68E−3	1.99	5.25E−3	1.23
1/64	1.03E−3	1.84	2.83E−3	0.89
1/128	2.79E−4	1.88	1.74E−3	0.70

Table 3
Accuracy of biased ENO limiter (2.21) for 2D Burgers' equation, $\varepsilon = 1$.

h	L_1 error	Order	L_∞ error	Order
1/4	1.22E−1	–	6.28E−2	–
1/8	2.84E−2	2.10	1.90E−2	1.72
1/16	7.77E−3	1.87	6.28E−3	1.60
1/32	1.98E−3	1.97	2.37E−3	1.41
1/64	5.12E−4	1.95	7.75E−4	1.61
1/128	1.28E−4	2.00	2.80E−4	1.47

Table 4
Accuracy of biased ENO limiter for 2D Burgers' equation, $\varepsilon = 3$.

h	L_1 error	Order	L_∞ error	Order
1/4	8.61E−2	–	4.95E−2	–
1/8	1.55E−2	2.47	1.12E−2	2.14
1/16	1.49E−3	3.38	2.29E−3	2.29
1/32	1.51E−4	3.30	5.13E−4	2.16
1/64	1.66E−5	3.19	1.38E−4	1.89
1/128	1.70E−6	3.29	2.79E−5	2.31

Table 5
Accuracy of WENO-type reconstruction on irregular triangular meshes for 2D Burgers' equation (HR without using partial neighboring cells).

h	L_1 error	Order	L_∞ error	Order
1/4	2.73E−2	–	2.07E−2	–
1/8	3.20E−3	3.09	6.09E−3	1.76
1/16	3.86E−4	3.05	1.21E−3	2.33
1/32	4.72E−5	3.03	9.37E−5	3.69
1/64	5.81E−6	3.02	2.68E−5	1.81
1/128	7.28E−7	3.00	8.49E−6	1.66

Table 6
Accuracy for 2D Euler equations with smooth solution on triangular meshes (HR without using partial neighboring cells).

h	L_1 error	Order	L_∞ error	Order
1/8	1.03E−5	–	4.37E−5	–
1/16	1.40E−6	2.88	6.31E−6	2.79
1/32	1.79E−7	2.97	1.15E−6	2.46
1/64	1.94E−8	3.21	1.81E−7	2.67

Table 7
Accuracy of HR using partial neighboring cells for 2D Burgers' equation on irregular triangular meshes.

h	L_1 error	Order	L_∞ error	Order
1/4	2.66E−2	–	2.15E−2	–
1/8	2.25E−3	3.56	3.38E−3	2.67
1/16	2.08E−4	3.44	3.82E−4	3.15
1/32	2.32E−5	3.16	3.83E−5	3.32
1/64	2.74E−6	3.08	6.50E−6	2.56
1/128	3.35E−7	3.03	6.24E−7	3.38

HR with WENO-type reconstruction. To demonstrate the robustness of HR with the WENO-type reconstruction, this polynomial reconstruction is performed on irregular triangular meshes as shown in Fig. 4(b). The accuracy results are shown in Table 5. The 3rd order accuracy is achieved in the L_1 norm.

For remaining test problems we will use the WENO-type reconstruction in each hierarchical level.

3.1.2. Accuracy test for smooth inviscid compressible flow

In order to test the accuracy for a system, a two-dimensional test problem [23] for the Euler equations is used. 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 $T = 0.1$. The meshes used in this study are similar to the representative mesh in Fig. 4(b) with the size of the meshes properly scaled to fit the domain. The accuracy results are shown in Table 6. Clearly, 3rd order accuracy (in L_1 norm) is achieved for the system. The errors presented are those of the cell averages of density.

3.2. Accuracy test cases for HR using partial neighboring cells

3.2.1. Numerical errors for smooth solutions of the Burgers' equation reconstructed by HR using partial neighboring cells

We use exactly the same setup as in Section 3.1.1 to test the accuracy of HR with partial neighboring cell for the case of scalar equations. The accuracy results are shown in Table 7. The desired 3rd order accuracy is achieved in L_1 norm. In addition, better accuracy is achieved in L_∞ norm comparing to HR without using partial neighboring cells (see Table 5).

3.2.2. Accuracy test for HR using partial neighboring cells with smooth inviscid compressible flow

We use the same setup as in Section 3.1.2 to test the accuracy of HR using partial neighboring cells for the case of systems of equations.

The desired 3rd order accuracy is achieved in L_1 norm (see Table 8). Again, better accuracy is achieved in L_∞ norm comparing to HR without using partial neighboring cells (see Table 6).

Remark: We also tested center biased limiter functions in each hierarchical level of HR with partial neighboring cells. We found that in order to achieve the desired order of accuracy for smooth solutions, the ε value in these limiter functions (e.g. in (2.21)) could be smaller than in the case without using partial neighboring cells. However, the ε value is still too large for discontinuous solutions. When HR (using center biased limiter functions in each hierarchical level) is applied only in non-smooth regions, we may not need to care about the accuracy issue and choose a small ε value, e.g. $\varepsilon = 0.01$. We have also conducted preliminary tests in this case and found its resolution slightly worse than HR with the WENO-type linear reconstruction. More detailed comparison of all these linear reconstruction methods for HR with partial neighboring cells will be documented in the future.

3.2.3. Accuracy test for HR using partial neighboring cells for vortex evolution problem

This test problem is taken from [23] 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

Table 8
Accuracy of HR using partial neighboring cells for 2D Euler equations with smooth solution on triangular meshes.

h	L_1 error	Order	L_∞ error	Order
1/4	4.1E−5	–	1.21E−4	–
1/8	6.18E−6	2.73	1.77E−5	2.77
1/16	7.68E−7	3.01	2.47E−6	2.84
1/32	9.50E−8	3.02	3.58E−7	2.79
1/64	1.10E−8	3.11	4.72E−8	2.92

Table 9
Accuracy of HR using partial neighboring cells for 2D vortex evolution problem.

h	L_1 density error	Order	L_∞ density error	Order	L_1 energy error	Order	L_∞ energy error	Order
1/2	6.18E−2	–	1.21E−2	–	2.73E−1	–	6.51E−2	–
1/4	7.24E−3	3.09	1.57E−3	2.95	3.11E−2	3.13	7.60E−3	3.10
1/8	8.65E−4	3.07	2.05E−4	2.94	3.70E−3	3.07	9.07E−4	3.07
1/16	1.07E−4	3.02	2.44E−5	3.07	4.64E−4	3.00	1.11E−4	3.03
1/32	1.33E−5	3.01	2.98E−6	3.03	6.37E−5	2.87	2.39E−5	2.22

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

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 9 that the desired 3rd order accuracy is achieved in L_1 norm.

3.3. Riemann problems of the Euler equations

The two-dimensional triangular DG methods with HR are applied to one-dimensional shock tube problems. We solve the Euler equations in a rectangular domain of $[-1,1]\times[0,0.2]$, with a triangulation of approximately 101 vertices in the x-direction and 11 vertices in the y-direction. An irregular triangular mesh is used, see Fig. 5(a). Initially, the y-component of the velocity is zero. Figs. 6(a) and 7(a) are obtained by interpolating the numerical solution along the line $y = 0.1$ on 101 equally spaced points.

The first test is the Sod problem [26]. The initial data is

$$(\rho,u,p) = \begin{cases}(1,0,1) & \text{if } x \leqslant 0\\ (0.125,0,0.1) & \text{if } x > 0.\end{cases} \tag{3.4}$$

The density at $t = 0.4$ is shown in Fig. 6.

The second test is the Lax problem [11]. The initial data is

$$(\rho,u,p) = \begin{cases}(0.445,0.698,3.528) & \text{if } x \leqslant 0\\ (0.5,0,0.571) & \text{if } x > 0.\end{cases} \tag{3.5}$$

The density at $t = 0.26$ is shown in Fig. 7.

We can see that both methods of HR with and without using partial neighboring cells give satisfactory results. In particular, the numerical solutions of HR using partial neighboring cells have narrower contact width and smaller over/undershoots. For the Lax problem, the contact discontinuity computed by HR with the WENO-type linear reconstruction is narrower than those obtained by HR with minmod and ENO limiters [15], and by the 3rd order WENO [9].

3.4. Shu–Osher problem

The Shu–Osher problem [25] is the Euler equations with an initial data

$$(\rho,u,p) = \begin{cases}(3.857143,2.629369,10.333333) & \text{if } x \leqslant -4\\ (1+0.2\sin(5x),0,1) & \text{if } x \geqslant -4.\end{cases} \tag{3.6}$$

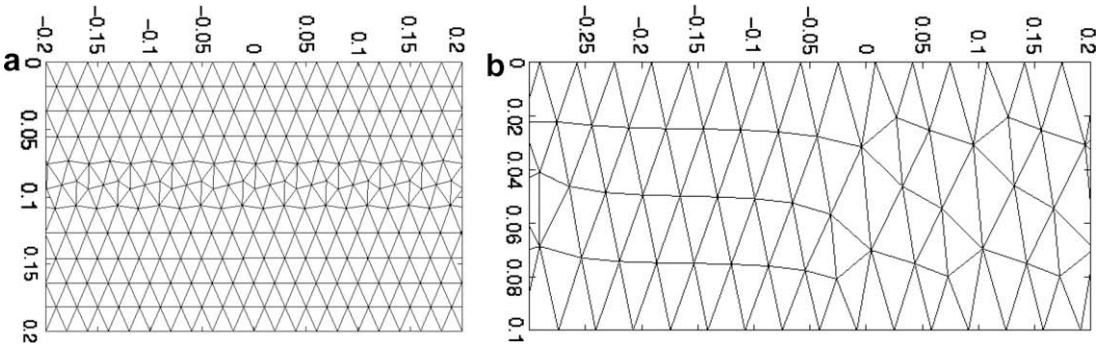


Fig. 5. Meshes for Riemann problems and Shu–Osher problem of the Euler equations. The x-axis is in the horizontal direction. (a) Portion of mesh for Lax and Sod problems and (b) portion of mesh for Shu–Osher problem.

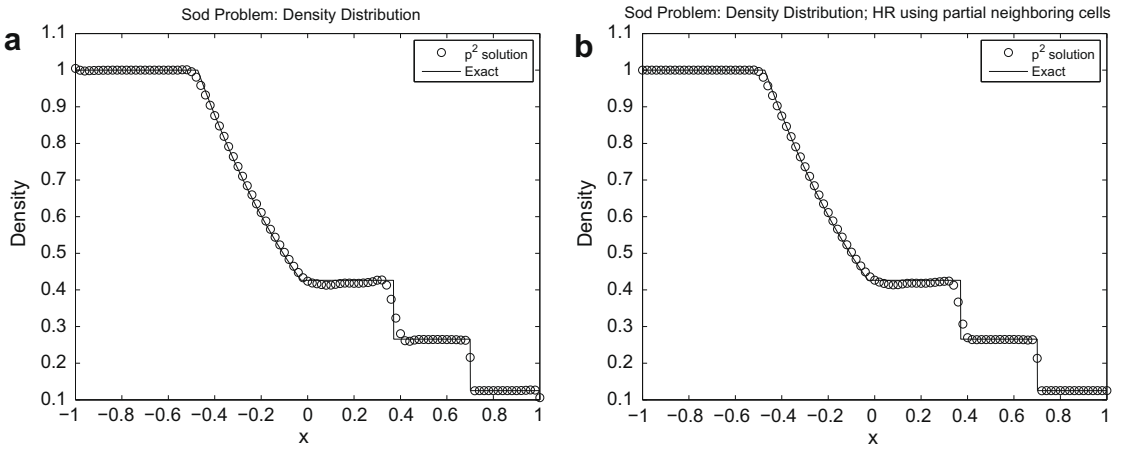


Fig. 6. Density profiles of solutions to the Sod problem. (a) HR without using partial neighboring cells and (b) HR using partial neighboring cells.

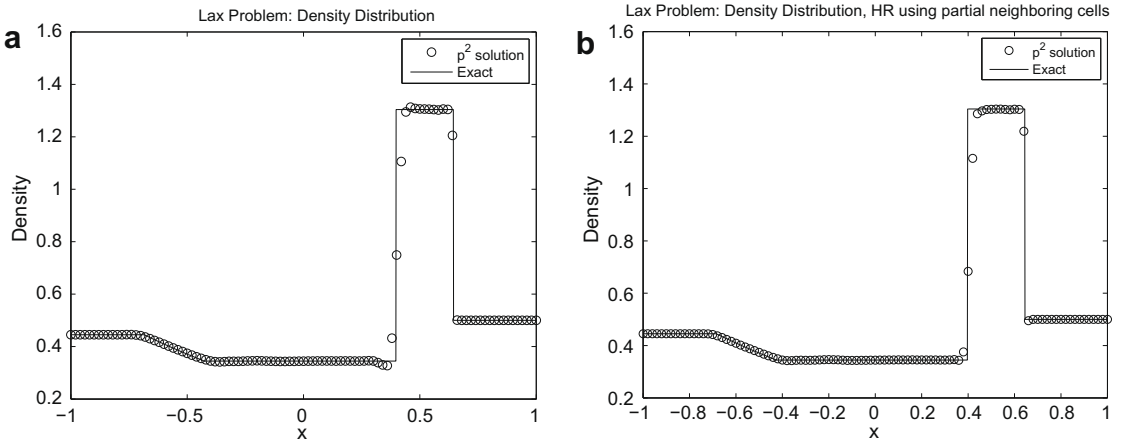


Fig. 7. Density profiles of solutions to the Lax problem. (a) HR without using partial neighboring cells and (b) HR using partial neighboring cells.

We solve the Euler equations in a rectangular domain of $[-5,5]\times[0,0.1]$ with a triangulation of about 301 vertices in the x-direction and four vertices in the y-direction, see Fig. 5(b). Initially, the y-velocity is zero. At $t=1.8$, the density profiles along $y=0.05$ is shown in Fig. 8(a)–(c). Fig. 8(d) gives the 2D profile.

We can see that both of the P^2 solutions have high resolution and almost no noise with the Euler equations computed component-wisely. In particular, HR using partial neighboring cells provides better resolution.

3.5. A 2D Riemann problem

A two-dimensional Riemann problem [12] for the Euler equations is computed. The computational domain is $[0,1]\times[0,1]$. The initial states are constants within each of the four quadrants. Counter-clock-wisely from the upper right quadrant, these states are labeled as $(\rho_i,u_i,v_i,p_i), i=1,2,3,4$. Initially, $\rho_1=1.1, u_1=0, v_1=0, p_1=1.1; \rho_2=0.5065, u_2=0.8939, v_2=0, p_2=0.35; \rho_3=1.1, u_3=0.8939, v_3=0.8939, p_3=1.1; \rho_4=0.5065, u_4=0, v_4=0.8939, p_4=0.35$. The density profile is plotted at $T=0.25$ in Fig. 9, with 30 equally spaced contours. The density profile along $x=0.8$ is plotted in Fig. 10. An unstructured triangular mesh is used. The distribution and connectivity of triangles are similar to those of triangles shown in Fig. 4(b). The triangle edge length is roughly equal to 1/400. We can see the numerical solutions are essentially non-oscillatory with the Euler system computed component-wisely. HR using partial neighboring cells produces sharper contacts and almost no over/under-shoots.

3.6. 2D shock vortex interactions

This test problem is taken from [23] to investigate the ability of the scheme to resolve the vortex interacting with a shock. The computational domain is $[0,2]\times[0,2]$. A stationary shock of Mach 1.1 is positioned at $x=0.5$ and perpendicular to the x-

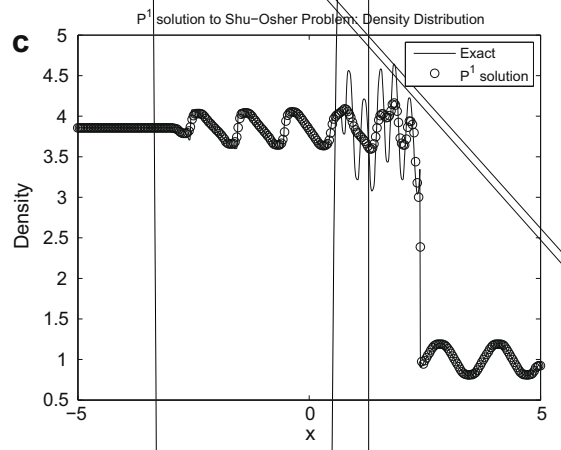
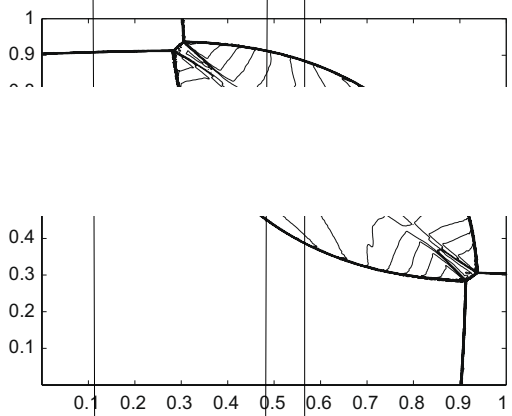


Fig. 8. Density profiles of solutions to the Shu–Osher problem. (a) P^2 solution; HR without using partial neighboring cells, (b) P^2 solution; HR using partial neighboring cells, (c) P^1 solution; HR without using partial neighboring cells and (d) 2D profile of the P^2 solution.



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,

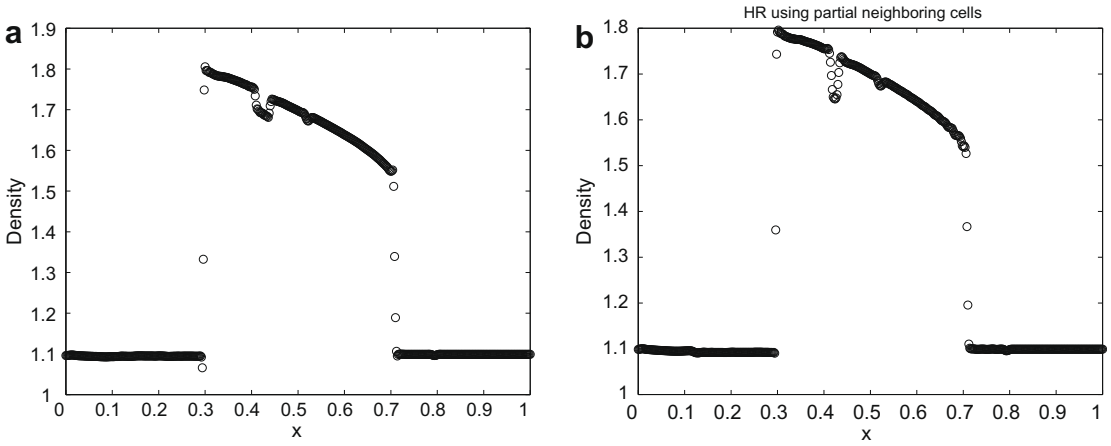


Fig. 10. Cross-section view of the solution to the 2D Riemann problem along the line $x = 0.8$. (a) HR without using partial neighboring cells and (b) HR using partial neighboring cells.

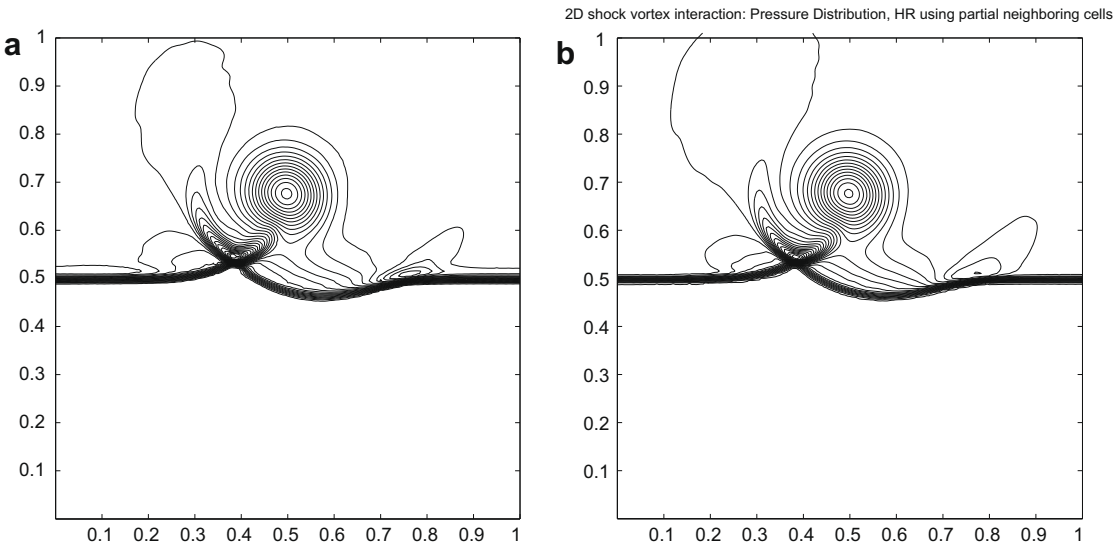


Fig. 11. 2D shock vortex interaction problem: Pressure contours at $t = 0.35$ with 30 equally spaced contour lines. (a) HR without using partial neighboring cells and (b) HR using partial neighboring cells.

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

where $\tau = \frac{r}{r_c}$ and $r = \sqrt{(x-x_c)^2 + (y-y_c)^2}$, $(x_c, y_c) = (0.25, 0.5)$, $r_c = 0.05$, $\alpha = 0.204$, and the strength of the vortex ϵ is equal to 0.3. The pressure profile in $[0,2]\times[0,1]$ is plotted at $T = 0.35$ in Fig. 11, with 30 equally spaced contours. An unstructured triangular mesh as in Fig. 4(b) is used. The triangle edge length is roughly equal to 1/100. Compared to Fig. 5.7 in Ref. [23], we can see that the resolution of the vortex and the noise-free performance by HR with partial neighboring cells is comparable to the result by WENO.

3.7. Double Mach reflection

The double Mach reflection problem is taken from [32]. We solve the Euler equations in 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

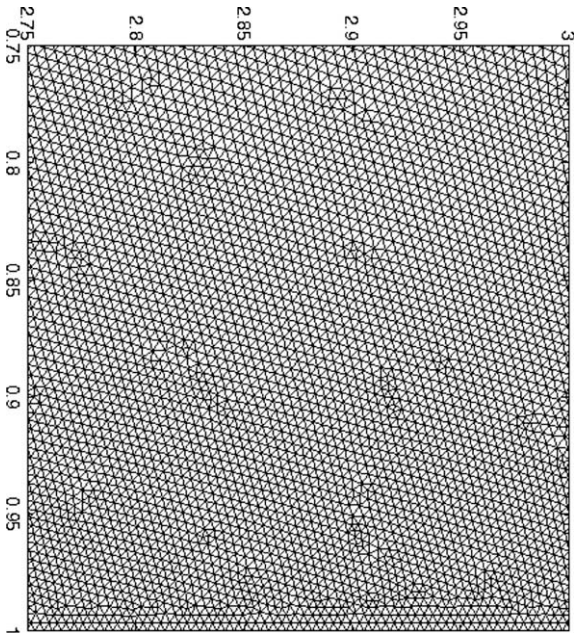


Fig. 12. Mesh for double Mach reflection problem. The portion shown here corresponds to case in which the triangle edge length roughly equals to $\frac{1}{250}$. The x-axis is in the horizontal direction. Finer triangles are used at the bottom reflection line.

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.

We test our method on unstructured triangular meshes with the triangle edge length roughly equal to $\frac{1}{250}$ and $\frac{1}{500}$, respectively. The portion of mesh shown in Fig. 12 corresponds to the case of edge length $\frac{1}{250}$. To speed up the calculation, coarser triangles are used at the upper half of the domain. 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. 13 is the contour plot with triangle edge length $\frac{1}{250}$. Fig. 14 is the contour plot with triangle edge length $\frac{1}{500}$. The "blown-up" portion around the double Mach region is shown in Fig. 15. We can see that fine details of the complicated flow structure under the triple Mach stem are captured.

Strong shocks of the double Mach problem introduce the negative pressure problem due to undershoots. To fix this problem, we employ a simple scaling technique to remove the negative pressure. If at a quadrature point of a cell, the negative pressure remains after reconstruction with the reconstructed polynomial u_h in the cell, we redefine the new polynomial $u_h^{(l+1)}$ to be

$$u_h^{(l+1)} = \bar{u}_h + 0.5(u_h^{(l)} - \bar{u}_h),$$

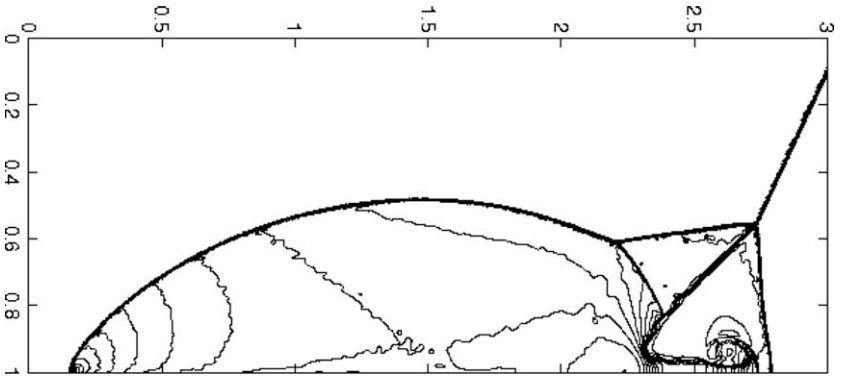


Fig. 13. Double Mach reflection problem: density contour, $t = 0.2, h = \frac{1}{250}$; 3rd order HR without using partial neighboring cells.

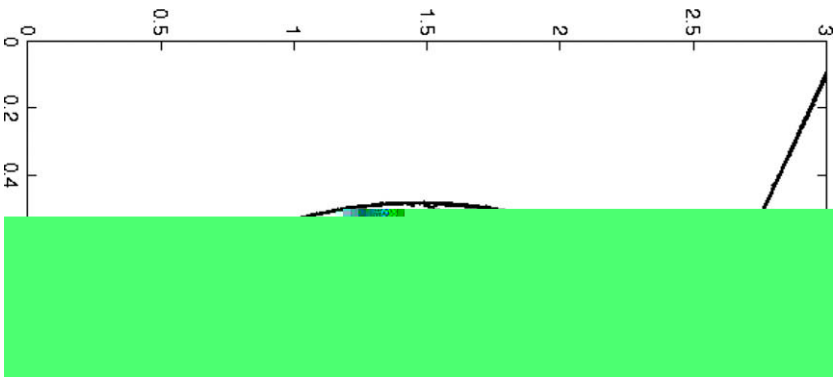


Fig. 14. Double Mach reflection problem: density contour, $t = 0.2, h = \frac{1}{500}$; 3rd order HR without using partial neighboring cells.

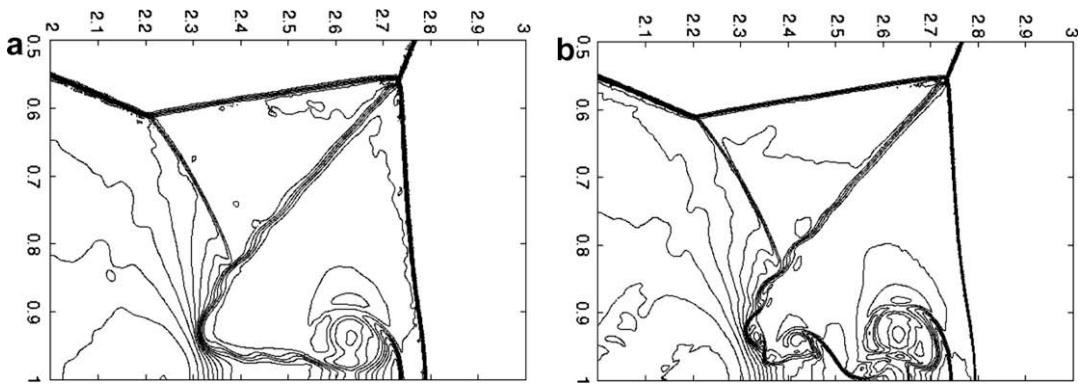


Fig. 15. Double Mach reflection problem: Blown-up region around the double Mach stems; density ρ. (a) 3rd order HR with cell edge length $\frac{1}{250}$ and (b) 3rd order HR with cell edge length $\frac{1}{500}$. HR without using partial neighboring cells.

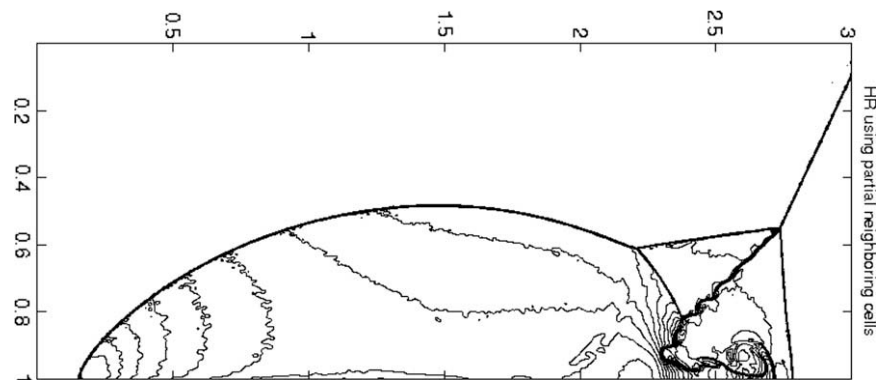


Fig. 16. Double Mach reflection problem: density contour, $t = 0.2, h = \frac{1}{400}$; 3rd order HR using partial neighboring cells.

where $\bar{u}_h$ is the cell average value of $u_h, u_h^{(0)} = u_h, l = 0, 1, 2, \ldots$. The negative pressure is removed after 1 or 2 iterations of the scaling normally. This procedure gradually "flattens" the polynomial in the cell to a constant without changing its cell average. It is easy to implement and locally reduces the order of accuracy to first order. Users will only need to check quadrature point values on triangle edges after applying HR and use the above scaling procedure if negative pressure is detected. This scaling procedure is only restricted to the "bad" cell, with no communication to or from neighbors. Moreover, the scaling is activated only in a very small region near the shock front, which does not affect the resolution of the computed solution. We would like to point out that HR is based on conservative variables instead of characteristic variables, which is probably the reason that this scaling procedure is needed and that the contours are more noisy. Among results obtained with limiters in conservative or primitive variables, our method is quite robust.

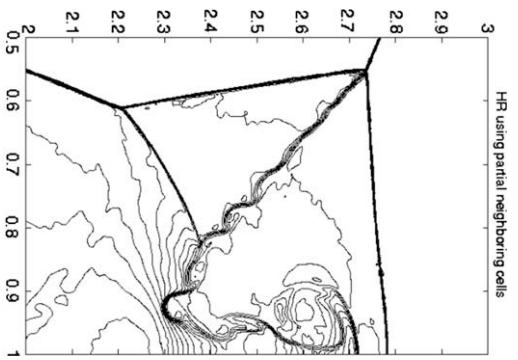


Fig. 17. Double Mach reflection problem: Blown-up region around the double Mach stems; density $\rho; t = 0.2, h = \frac{1}{400}$, 3rd order HR using partial neighboring cells.

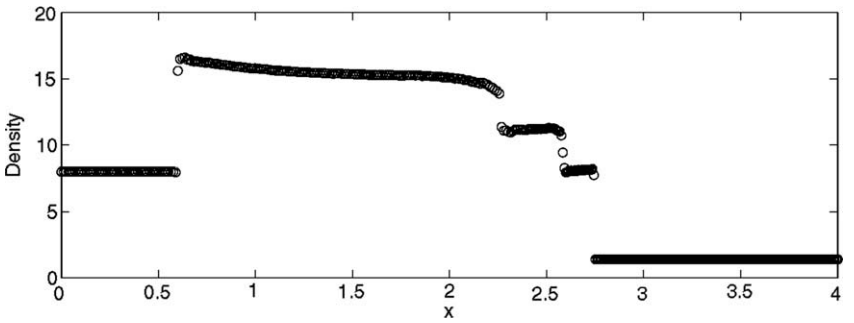


Fig. 18. Cross-section view of the solution to double Mach reflection problem along the line $y = \frac{1}{3}$ from the bottom of the domain. Density profile sampled with 400 points. HR using partial neighboring cells. $t = 0.2, h = \frac{1}{400}$.

We can see that both solutions by HR with and without using partial neighboring cells have high resolution, see Figs. 13–18. In particular, HR with partial neighboring cells performs better, and the negative pressure fix is much less frequently called during the run.

We also compare our results to [9]. Our result by HR with partial neighboring cell (on the 3rd order DG) is better than those by the 3rd and 4th order component-wise WENO, and seems to be between those of the 4th order component-wise WENO and the 4th order characteristic WENO on a similar mesh. See Fig. 17 and figures in [9].

4. Concluding remarks

We have further developed the HR reconstruction procedure and used it as a limiter for the discontinuous Galerkin method on the unstructured triangular meshes. In particular, the HR with the WENO-type reconstruction of linear polynomials maintains the desired order of accuracy and resolution on triangular meshes. A partial neighboring cell technique further improves resolution and reduces over/under-shoots for solutions containing interactions of discontinuities. These techniques provide us insight for future study of the application of HR to higher order DG methods.

References

[1] R. Biswas, K. Devine, J. Flaherty, Parallel, adaptive finite element methods for conservation laws, Appl. Numer. Math. 14 (1994) 255–283.
[2] A. Burbeau, P. Sagaut, C.H. Bruneau, A problem-independent limiter for high-order Runge–Kutta discontinuous Galerkin methods, J. Comput. Phys. 169 (2001) 111–150.
[3] B. Cockburn, S. Hou, C.-W. Shu, The TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Math. Comput. 54 (1990) 545–581.
[4] B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 52 (1989) 411–435.
[5] B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411–435.
[6] 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.
[7] M. Dumbser, M. Käser, V.A. Titarev, E.F. Toro, Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems, J. Comput. Phys. 226 (2007) 204–243.
[8] A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III, J. Comput. Phys. 71 (1987) 231–303.

[9] Changqing Hu, Chi-Wang Shu, Weighted essentially non-oscillatory schemes on triangular meshes, J. Comput. Phys. 150 (1999) 97–127.
[10] G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1996) 202–228.
[11] P. Lax, Weak solutions of nonlinear hyperbolic equations and their numerical computations, Commun. Pure Appl. Math. 7 (1954) 159.
[12] P. Lax, X.-D. Liu, Solution of two-dimensional Riemann problem of gas dynamics by positive schemes, SIAM J. Sci. Comput. 19 (1998) 319–340.
[13] X.-D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[14] 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.
[15] 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.
[16] H. Luo, J.D. Baum, R. Lohner, A Hermite WENO-based limiter for discontinuous Galerkin method on unstructured grids, J. Comput. Phys. 225 (2007) 686–713.
[17] J. Qiu, C.-W. Shu, Runge–Kutta discontinuous Galerkin method using WENO limiters, SIAM J. Sci. Comput. 26 (2005) 907–929.
[18] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: two-dimensional case, Comput. Fluids 34 (2005) 642C–663.
[19] W. Reed, T. Hill, Triangular mesh methods for the neutron transport equation. Technical Report la-ur-73-479, Los Alamos Scientific Laboratory, 1973.
[20] A.M. Rogerson, E. Meiburg, A numerical study of the convergence properties of ENO schemes, J. Sci. Comput. 5 (1990) 151–167.
[21] C.-W. Shu, TVB uniformly high-order schemes for conservation laws, Math. Comput. 49 (1987) 105–121.
[22] C.-W. Shu, Numerical experiments on the accuracy of ENO and modified ENO schemes, J. Sci. Comput. 5 (1990) 127–149.
[23] 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.
[24] C.-W. Shu, S. Osher, Efficient Implementation of essentially non-oscillatory shock capturing schemes, J. Comput. Phys. 77 (1988) 439–471.
[25] C.-W. Shu, S. Osher, Efficient Implementation of essentially non-oscillatory shock capturing schemes II, J. Comput. Phys. 83 (1989) 32–78.
[26] G. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws, J. Comput. Phys. 27 (1978) 1.
[27] 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.
[28] B. van Leer, Towards the ultimate conservative difference scheme: IV. A new approach to numerical convection, J. Comput. Phys. 23 (1977) 276–299.
[29] 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.
[30] 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.
[31] 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.
[32] P. Woodward, P. Colella, Numerical simulation of two-dimensional fluid flows with strong shocks, J. Comput. Phys. 54 (1984) 115.
[33] J. Zhu, J.-X. Qiu, C.-W. Shu, M. Dumbser, Runge–Kutta discontinuous Galerkin method using WENO limiters II: unstructured meshes, J. Comput. Phys. 227 (2008) 4330–4353.