

Point-wise Hierarchical Reconstruction for Discontinuous Galerkin and Finite Volume Methods for Solving Conservation Laws

Zhiliang Xu[‡], Yingjie Liu[§], Huijing Du,^{*}Guang Lin^{||} and Chi-Wang Shu[¶]

September 6, 2010

Abstract

We develop a new hierarchical reconstruction (HR) method [17, 29] for limiting solutions of the discontinuous Galerkin and finite volume methods up to fourth order without local characteristic decomposition for solving hyperbolic nonlinear conservation laws on triangular meshes. The new HR utilizes a set of point values when evaluating polynomials and remainders on neighboring cells, extending the technique introduced in Hu, Li and Tang [9]. The point-wise HR simplifies the implementation of the previous HR method which requires integration over neighboring cells and makes HR easier to extend to arbitrary meshes. We prove that the new point-wise HR method keeps the order of accuracy of the approximation polynomials. Numerical computations for scalar and system of nonlinear hyperbolic equations are performed on two-dimensional triangular meshes. We demonstrate that the new hierarchical reconstruction generates essentially non-oscillatory solutions for schemes up to fourth order on triangular meshes.

[‡](E-mail: zxu2@nd.edu)

Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556. Research supported in part by NSF grant DMS-0800612.

[§](E-mail: yingjie@math.gatech.edu)

School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332. Research supported in part by NSF grant DMS-0810913.

^{**}(E-mail: hdu@nd.edu)

Department of Applied and Computational Mathematics and Statistics, University of Notre Dame, Notre Dame, IN 46556.

^{||}(E-mail: guang.lin@pnl.gov)

Computational Mathematics Group, Pacific Northwest National Laboratory, 902 Battelle Boulevard, Richland, WA 99352. Research was supported by the Advanced Scientific Computing Research Program of the U.S. Department of Energy Office of Science.

[¶](E-mail: shu@dam.brown.edu)

Division of Applied Mathematics, Brown University, Providence RI 02912. Research supported in part by ARO grant W911NF-08-1-0520 and NSF grant DMS-0809086.

1 Introduction

The limiting techniques for eliminating spurious oscillations of the numerical solutions of nonlinear conservation laws have been actively studied for the past few decades and many highly successful methods have been developed such as the MUSCL scheme [13, 14, 15], the ENO [7, 25, 26] and WENO schemes [16, 10] etc. Robust limiting methods without excessive dissipation are essential for the success of the finite volume schemes, discontinuous Galerkin methods (DG) and many other numerical approximations of non-smooth solutions. In particular, limiting for high order methods on unstructured meshes, e.g. triangular meshes are useful in many real applications involving complex geometry. In [1, 8], the WENO schemes have been successfully developed on unstructured triangular meshes for nonlinear conservation laws. The DG method [22, 5, 4, 6] can be easily formulated for unstructured meshes and is compact, thus is very nice for implementation and parallelization.

Given a polynomial approximation to the solution (either obtained by a preliminary reconstruction from cell average values or evolved by DG), a limiting technique should ideally take advantage of all available information in the neighborhood of a cell in order to be compact. However, due to the Gibbs phenomenon, all high order information is more or less polluted near a discontinuity of the solution. It is a challenging problem to use adjacent high order information in the limiting procedure to remove spurious oscillations while keeping high resolution of the waves near a discontinuity of the solution, particularly without using local characteristic decomposition. The compact total variation bounded (TVB) projection limiter by Cockburn, Shu *et al.* [23, 5] limits the gradients of a polynomial in a cell by comparing it to the finite difference of adjacent lower moments, and truncates its higher order moments when local non-smoothness is detected. The moment limiter [2] takes the r -th order derivative of the Legendre polynomial in a cell (successively from high order to low order) and applies a similar strategy to limit the first moment of resulting polynomial. Since the finite difference of lower moments does not provide high order approximation to the gradient, it is used to formulate a bound for the gradient in nonsmooth region of the solution and it is critical in the relaxation and application of the bound, see [3, 11] for its further developments for DG and [30] for the spectral difference method on triangular meshes. The WENO finite volume reconstruction has also been developed for the DG method as a limiter, see [20, 21].

Similar to the moment limiters, the hierarchical reconstruction (HR) method [17] takes the r -th order derivative of the polynomial in a cell (successively from high order to low order) and modifies the linear part of the resulting polynomial. However, the modification takes a different approach from the moment limiters. The cell averages of this linear part over nearby cells are first estimated to sufficient order of accuracy which involves previously modified higher degree terms, then a non-oscillatory reconstruction of a linear polynomial out of these cell averages can be applied to update this linear part. As a result, HR maintains the approximation order of accuracy of the original polynomial in the cell. Since the reconstruction of a linear polynomial can be easily realized by utilizing information from adjacent cells e.g. the MUSCL reconstruction, HR can be formulated in multi-dimensions in a very compact manner, which only involves immediate neighbors of the target cell. HR does not use local characteristic decomposition and thus is convenient for unstructured meshes. Note that the traditional ENO/WENO reconstruction should be performed on characteristic variables (e.g., [7, 8, 19]) when the order of the scheme gets higher, otherwise spurious

oscillations may occur beyond third order.

There is still a lot of space for further improvement and understanding of HR. In [29], the partial neighboring cell idea is introduced for HR applied to the third order DG method on triangular meshes so that small overshoot/undershoots have been eliminated during interactions of discontinuities. In [9], a point-wise HR has been developed for limiting a very interesting point-valued scheme for solving the stationary Euler equations. Based on the two techniques, we are able to extend the work of [29] to limit numerical solutions computed by the fourth order DG and finite volume methods on triangular meshes.

In this paper, we combine the ideas in [9] and [29] to develop a new point-wise HR method which utilizes a set of point values when evaluating polynomials and remainders on neighboring cells. We prove and numerically show that the point-wise HR keeps the approximation order of accuracy of the original polynomial in the cell if the solution is smooth. We also present shock wave test problems to demonstrate that the point-wise HR limiting technique gives good resolutions of the numerical solutions of both finite volume and DG schemes. We apply the point-wise HR to limit the solutions computed by the 4th order finite volume scheme, the 3rd and 4th order RKDG and RKDG with conservation constraint schemes. We note that point-wise HR provides more flexibility than the previous HR [29], since the latter requires integration over the partial neighboring cells. For many occasions, creating suitable partial neighboring cells is not trivial for the more general polygonal type cells specially in 3D. Moreover, when applying the previous HR [29] to limiting the 4th order accurate DG solutions, we observed the returning of small overshoot/undershoot. With the new point-wise HR, the small overshoot/undershoots become essentially negligible. In fact, by using sufficiently accurate Gaussian quadrature, the average of a polynomial over a partial neighboring cell can be viewed as a weighted average of certain point-values of the polynomial (at Gaussian points). Therefore the improvement which the current point-wise HR has made implies that the average of more evenly distributed point values is better at removing residual overshoot/undershoots than the Gaussian quadrature while both of them keep the approximation order of accuracy of the polynomial intact when employed by HR. Finally, we would like to comment that although the description of the point-wise HR algorithm seems to be complex, its implementation is actually quite simple, since each stage only involves evaluation of some point-values on neighboring cells and the reconstruction of a linear polynomial for the conservative variables of the governing equations.

The paper is organized as follows. Section 2 describes the DG and finite volume solution procedures and the new point-wise HR limiting procedure. The details for implementing the point-wise HR for the 3rd and 4th order accurate schemes are also given in Section 2. Numerical tests are presented in Section 3. Concluding remarks and summary are included in Section 4.

2 Algorithm Formulation

In this section, we briefly outline the Runge-Kutta discontinuous Galerkin finite element method and the finite volume method for solving time dependent hyperbolic conservation

laws (2.1)

$$\begin{cases} \frac{\partial u_k}{\partial t} + \nabla \cdot \mathbf{F}_k(\mathbf{u}) = 0, & k = 1, \dots, p, \quad \text{in } \Omega \times (0, T), \\ \mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x}), \end{cases} \quad (2.1)$$

where $\Omega \subset R^d$, $\mathbf{x} = (x_1, \dots, x_d)$, d is the dimension, $\mathbf{u} = (u_1, \dots, u_p)^T$ and the flux vectors $\mathbf{F}_k(\mathbf{u}) = (F_{k,1}(\mathbf{u}), \dots, F_{k,d}(\mathbf{u}))$. More details of the methods can be found in [5, 4, 6, 24, 29].

The method of lines approach is used to evolve the solution on the triangulated domain. Specifically, TVD Runge-Kutta method [25] is used to update the solution. The hierarchical reconstruction is applied in the vicinities of discontinuities of the solution to remove spurious oscillations. The ideas of the new point-wise HR are described in Section 2.3, and the details for implementing the point-wise HR for the 3rd and 4th order accurate schemes follow.

2.1 Runge-Kutta discontinuous Galerkin method

We review the RKDG formulation here. The physical domain Ω is partitioned into a collection of \mathcal{N} triangular cells so that $\Omega = \bigcup_{i=1}^{\mathcal{N}} \mathcal{K}_i$ and let

$$\mathcal{T}_h = \{\mathcal{K}_i : i = 1, \dots, \mathcal{N}\} . \quad (2.2)$$

For simplicity, we assume that there are no hanging nodes. Let the basis function set which spans the finite element space on cell \mathcal{K}_i be

$$\mathcal{B}_i = \{\phi_m(\mathbf{x}) : m = 0, \dots, r\} .$$

In the present study, we choose the basis function set supported on the cell \mathcal{K}_i to be the monomials of degree q of multi-dimensional Taylor expansions about the cell centroid of \mathcal{K}_i so that $r = (q + 1)(q + 2)/2 + 1$.

In each cell \mathcal{K}_i , the approximate solution $u_{h,k}$ of the k^{th} equation of (2.1) is expressed as

$$u_{h,k} = \sum_{m=0}^r c_m(t) \phi_m(\mathbf{x}) . \quad (2.3)$$

The semi-discrete DG formulation of the k^{th} equation of (2.1) is to find an approximate piecewise polynomial solution u_h (neglecting its subscript k for convenience) of degree q such that

$$\frac{d}{dt} \int_{\mathcal{K}_i} u_h v_h d\mathbf{x} + \int_{\partial\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h) \cdot \mathbf{n}_i v_h d\Gamma - \int_{\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} = 0 , \quad (2.4)$$

for any $v_h \in \text{span}\{\mathcal{B}_i\}$. Here \mathbf{n}_i is the outer unit normal vector of \mathcal{K}_i . Since u_h is discontinuous between element interfaces, the interfacial fluxes are not uniquely determined. The flux function $\mathbf{F}_k(\mathbf{u}_h) \cdot \mathbf{n}_i$ appearing in Eq. (2.4) is replaced by a numerical flux function (the Lax-Friedrich flux, see e.g. [24]) defined as

$$h_k(\mathbf{x}, t) = h_k(\mathbf{u}_h^{\text{in}}, \mathbf{u}_h^{\text{out}}) = \frac{1}{2}(\mathbf{F}_k(\mathbf{u}_h^{\text{in}}) \cdot \mathbf{n}_i + \mathbf{F}_k(\mathbf{u}_h^{\text{out}}) \cdot \mathbf{n}_i) + \frac{\alpha}{2}(u_h^{\text{in}} - u_h^{\text{out}}) , \quad (2.5)$$

where α is the largest characteristic speed,

$$\begin{aligned} \mathbf{u}_h^{\text{in}}(\mathbf{x}, t) &= \lim_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \in \mathcal{K}_i^{\text{int}}} \mathbf{u}_h(\mathbf{y}, t) , \\ \mathbf{u}_h^{\text{out}}(\mathbf{x}, t) &= \lim_{\mathbf{y} \rightarrow \mathbf{x}, \mathbf{y} \notin \mathcal{K}_i} \mathbf{u}_h(\mathbf{y}, t) . \end{aligned}$$

Equation (2.4) now becomes

$$\frac{d}{dt} \int_{\mathcal{K}_i} u_h v_h d\mathbf{x} + \int_{\partial\mathcal{K}_i} h_k v_h d\Gamma - \int_{\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} = 0 . \quad (2.6)$$

The above systems of ordinary differential equations can be solved by a s -stage TVD Runge-Kutta method, which can be written in the form:

$$\int_{\mathcal{K}_i} u_h^{(j)} v_h d\mathbf{x} = \sum_{l=0}^{j-1} \alpha_{jl} \left(\int_{\mathcal{K}_i} u_h^{(l)} v_h d\mathbf{x} + \Delta t_n \beta_{jl} L(u_h^{(l)}, v_h) \right) , \quad j = 1, \dots, s \quad (2.7)$$

with

$$u_h^{(0)} = u_h^n, \quad u_h^{(s)} = u_h^{n+1} . \quad (2.8)$$

Here α_{jl} and β_{jl} are coefficients of the Runge-Kutta method at the j^{th} stage, and

$$L(u_h, v_h) = - \int_{\partial\mathcal{K}_i} h_k v_h d\Gamma + \int_{\mathcal{K}_i} \mathbf{F}_k(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} .$$

2.1.1 Runge-Kutta discontinuous Galerkin method with conservation constraints

In [28], we enforce a few additional local cell average conservation constraints on the RKDG method in order to obtain a larger CFL number. This idea is a simple technique connecting the DG and finite volume methods which are both compact and can take larger CFL numbers closer to those of the finite volume methods.

The idea of the new scheme is as follows: Let the edge adjacent neighbors of \mathcal{K}_i be collected as the set $\{\mathcal{K}_j : j = 1, 2, \dots, M.\}$ (which also contains cell \mathcal{K}_i). We assume that the degree of $u_h(\mathbf{x}, t) \geq M$. The reason for having this assumption will become evident later.

Eqn. (2.6) can be solved by a TVD Runge-Kutta method which can be viewed as a convex combinations of several forward Euler schemes. The additional conservation constraints are added within each of the component forward Euler scheme. A forward Euler scheme for solving (2.6) can be written as

$$\int_{\mathcal{K}_i} u_h^{n+1} v_h d\mathbf{x} = \int_{\mathcal{K}_i} u_h^n v_h d\mathbf{x} - \Delta t_n \int_{\partial\mathcal{K}_i} h_k v_h d\Gamma + \Delta t_n \int_{\mathcal{K}_i} \mathbf{F}_k^n(\mathbf{u}_h) \cdot \nabla v_h d\mathbf{x} , \quad (2.9)$$

where the superscript n denotes the time level t_n , $\Delta t_n = t_{n+1} - t_n$. In particular, letting $v_h \equiv 1$, we obtain the cell average of u_h^{n+1} over cell \mathcal{K}_i , $\overline{u_i^{n+1}}$, just as with a finite volume scheme.

Now suppose the cell averages $\{\overline{u_i^{n+1}}\}$ have been computed on all cells. We do not compute the rest of the moments of u_h^{n+1} on cell \mathcal{K}_i by using equation (2.9). Instead, we let u_h^{n+1} on cell \mathcal{K}_i minimize an energy functional (variational to (2.9)) subject to that it conserves additional given cell averages not only in cell \mathcal{K}_i but also in some of its neighbors. Rewrite (2.9) in cell \mathcal{K}_i as

$$\int_{\mathcal{K}_i} u_h^{n+1} v_h d\mathbf{x} = \mathcal{L}(v_h) , \quad (2.10)$$

where $\mathcal{L}(v_h)$ represents the right-hand-side of (2.9) which is a linear functional defined on the finite element space on \mathcal{K}_i with respect to the test function v_h . The variational form of (2.10) is to find u_h^{n+1} in the finite element space on \mathcal{K}_i such that it minimizes the energy functional

$$E(v_h) = \frac{1}{2} \int_{\mathcal{K}_i} (v_h)^2 d\mathbf{x} - \mathcal{L}(v_h) . \quad (2.11)$$

Finally, the RKDG with conservation constraints scheme on cell \mathcal{K}_i can be described as finding u_h^{n+1} in each stage of the TVD Runge-Kutta method in the finite element space on \mathcal{K}_i , such that

$$\begin{aligned} E(u_h^{n+1}) = & \text{Minimizing}\{E(v_h) : v_h \text{ in the finite element space on } \mathcal{K}_i\}, \\ & \text{subject to } \frac{1}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} u_h^{n+1} d\mathbf{x} = \overline{u_j^{n+1}}, \quad j = 1, \dots, M . \end{aligned} \quad (2.12)$$

This constrained minimization problem is solved by the method of Lagrange multipliers as follows

$$\begin{aligned} \int_{\mathcal{K}_i} u_h^{n+1} v_h d\mathbf{x} - \mathcal{L}(v_h) &= \sum_{j=1}^M \frac{\lambda_j}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} v_h d\mathbf{x}, \\ \frac{1}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} u_h^{n+1} d\mathbf{x} &= \overline{u_j^{n+1}}, \quad j = 1, \dots, M , \end{aligned} \quad (2.13)$$

where $\{\lambda_j\}$ are Lagrangian multipliers. The moments of u_h^{n+1} are determined by (2.13). We note that we assume “the degree of $u_h(\mathbf{x}, t) \geq M$ ” for the minimization problem to be well-posed.

2.2 Finite volume method

Taking the cell \mathcal{K}_i in partition (2.2) as the control volume, the semi-discrete finite volume method for solving Eqn. (2.1) is formulated by integrating (2.1) over the cell \mathcal{K}_i :

$$\frac{d}{dt} \bar{u}_{k,i}(t) + \frac{1}{|\mathcal{K}_i|} \int_{\partial\mathcal{K}_i} \mathbf{F}_k \cdot \mathbf{n}_i d\Gamma = 0 , \quad (2.14)$$

where $\bar{u}_{k,i}(t)$ is the cell average of u_k on \mathcal{K}_i , and \mathbf{n}_i is the outward unit normal of the boundary of cell \mathcal{K}_i . We can evaluate the flux integral by Gaussian quadrature rule with $\mathbf{F}_k \cdot \mathbf{n}_i$ being replaced by the Lax-Friedrich flux function (2.5). We obtain the following semi-discrete numerical scheme:

$$\frac{d}{dt} \bar{u}_{h,k,i}(t) + \frac{1}{|\mathcal{K}_i|} \int_{\partial\mathcal{K}_i} h_k d\Gamma = 0 , \quad (2.15)$$

where $\bar{u}_{h,k,i}(t)$ is the approximate cell average. Equation (2.15) is solved by a s -stage TVD Runge-Kutta method.

2.2.1 Preliminary reconstruction

Given the numerical cell averages $\{\bar{u}_{h,i} : i = 1, \dots, \mathcal{N}\}$ at a time t (again neglecting the subscript k for convenience), we first construct piecewise polynomial function $u_{h,i}(\mathbf{x} - \mathbf{x}_i)$ in the form of Taylor series expansion about the cell centroid \mathbf{x}_i for all cells $\{\mathcal{K}_i : i = 1, \dots, \mathcal{N}\}$.

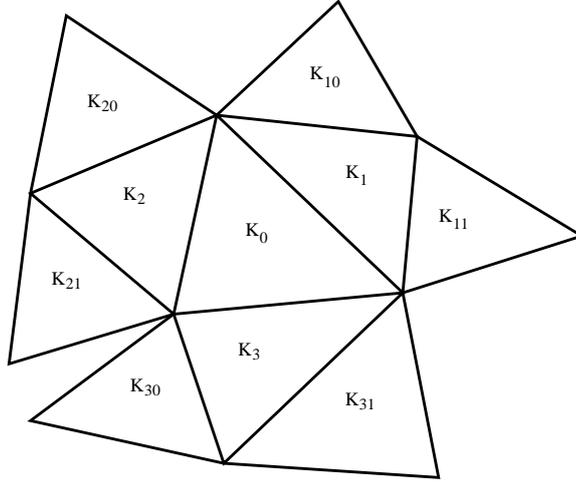


Figure 1: The stencil for preliminary construction for the finite volume scheme on cell \mathcal{K}_0 .

The constructed polynomial function may contain spurious oscillations in non-smooth regions of the solution, we use HR to limit or reconstruct these oscillatory polynomial functions.

Fourth Order Case. The constructed polynomial function is a piecewise third degree polynomial ($q = 3$). The coefficients of this polynomial can be determined by solving the following linear system:

$$\frac{1}{|\mathcal{K}_j|} \int_{\mathcal{K}_j} u_{h,j}(\mathbf{x} - \mathbf{x}_j) dx = \bar{u}_{h,j} ,$$

for all cell indices j such that $j = 0, 1, 2, 3, 10, 11, 20, 21, 30$ and 31 (see also Fig. 1).

2.3 Limiting by point-wise hierarchical reconstruction

To prevent non-physical oscillations in the vicinity of discontinuities, we develop the point-wise HR method to limit the numerical solution. The point-wise HR method incorporates a further developed point-wise HR in [9] to extend the work of [29], resulting in more flexibility and reduced computational complexity.

We first outline the point-wise HR algorithm. The details for implementing it for the piece-wise quadratic and cubic polynomial approximation solutions on the triangular meshes and the corresponding piece-wise linear polynomial reconstruction procedure follow. We also refer to [17, 18] for the summary of the HR algorithm which utilizes average values of polynomials and remainders on the neighboring cells and [29] for the partial neighboring cell technique of HR on triangular meshes.

Suppose we have computed a piecewise polynomial (of degree q) numerical solution at a time. Let \mathcal{K}_I be the target cell under consideration and the set $\{\mathcal{K}_J\}$ be the collection of cells adjacent to cell \mathcal{K}_I . Let \mathbf{x}_i , $i = I, J$ be the cell centroids of cells \mathcal{K}_I and $\{\mathcal{K}_J\}$ respectively.

We write the polynomial solution in the form of Taylor series expansion

$$u_{h,i}(\mathbf{x} - \mathbf{x}_i) = \sum_{m=0}^q \sum_{|\mathbf{m}|} \frac{1}{\mathbf{m}!} u_i^{\mathbf{m}}(0) (\mathbf{x} - \mathbf{x}_i)^{\mathbf{m}} \quad (2.16)$$

The generalized cell average of a polynomial v_h over a cell \mathcal{K}_J , denoted as P-average, is defined to be $(\sum_{s=1}^M v_h(\mathbf{x}_{s,J}))/M$ where $\{\mathbf{x}_{s,J}\}$ is a set of points in cell \mathcal{K}_J to be explained in more details; the coordinates of the P-centroid of cell \mathcal{K}_J is defined as $(\sum_{s=1}^M \mathbf{x}_{s,J})/M$.

The general procedure of point-wise HR to limit solution $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$, *i.e.*, to reconstruct coefficients $u_I^{\mathbf{m}}(0)$ to obtain a new set of coefficients $\tilde{u}_I^{\mathbf{m}}(0)$ and therefore a smoother $\tilde{u}_{h,I}(\mathbf{x} - \mathbf{x}_I)$ proceeds as follows:

Algorithm 1 Step 1. Suppose $q \geq 2$. For $m = q, q - 1, \dots, 1$, do the following:

(a) Take a $(m - 1)^{th}$ order partial derivative for each of $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ and $\{u_{h,J}(\mathbf{x} - \mathbf{x}_J)\}$ to obtain polynomials $\partial^{m-1}u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ and $\{\partial^{m-1}u_{h,J}(\mathbf{x} - \mathbf{x}_J)\}$ respectively. In particular, denote $\partial^{m-1}u_{h,I}(\mathbf{x} - \mathbf{x}_I) = L_{m,I}(\mathbf{x} - \mathbf{x}_I) + R_{m,I}(\mathbf{x} - \mathbf{x}_I)$, where $L_{m,I}(\mathbf{x} - \mathbf{x}_I)$ is the linear part of $\partial^{m-1}u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ and $R_{m,I}(\mathbf{x} - \mathbf{x}_I)$ is the remainder.

(b) Calculate the cell average of $\partial^{m-1}u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ on cell \mathcal{K}_I to obtain $\overline{\partial^{m-1}u_{h,I}}$. Calculate the cell average (P-average) of $\partial^{m-1}u_{h,J}(\mathbf{x} - \mathbf{x}_J)$ on cell \mathcal{K}_J to obtain $\overline{\partial^{m-1}u_{h,J}}$.

(c) Let $\tilde{R}_{m,I}(\mathbf{x} - \mathbf{x}_I)$ be the $R_{m,I}(\mathbf{x} - \mathbf{x}_I)$ with its coefficients replaced by the corresponding new values¹. For cell \mathcal{K}_i , $i \in \{I, J\}$, calculate the cell average (P-average, if it's used on cell \mathcal{K}_J in (b)) of $\tilde{R}_{m,I}(\mathbf{x} - \mathbf{x}_I)$ on cell \mathcal{K}_i to obtain $\overline{\tilde{R}_{m,i}}$.

(d) Let $\bar{L}_{m,i} = \overline{\partial^{m-1}u_{h,i}} - \overline{\tilde{R}_{m,i}}$ for cell \mathcal{K}_i , $i \in \{I, J\}$.

(e) Form stencils out of the new approximate cell averages $\{\bar{L}_{m,i}\}$ by using a MUSCL, second order ENO or other non-oscillatory strategies. Each stencil will determine² a set of candidates for the coefficients in the first degree terms of $L_{m,I}(\mathbf{x} - \mathbf{x}_I)$, which are also candidates for the corresponding $u_{h,I}^{(\mathbf{m})}(\mathbf{0})$'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 m^{th} degree terms of $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ have been computed. For each of these coefficients, say $\frac{1}{\mathbf{m}!}u_{h,I}^{(\mathbf{m})}(\mathbf{0})$, $|\mathbf{m}| = m$, let the new value $\tilde{u}_{h,I}^{(\mathbf{m})}(\mathbf{0}) = F(\text{candidates of } u_{h,I}^{(\mathbf{m})}(\mathbf{0}))$, where F is a limiter function (*e.g.*, the minmod limiter) returns a convex average of its arguments.

Step 2. The new coefficient in the 0^{th} degree term of $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ is chosen so that the cell average of $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ on cell \mathcal{K}_I is invariant with the new coefficients. At this stage all the new coefficients of $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ have been found.

We now prove that if the **Condition 1** (given below) is satisfied, the point-wise HR retains the approximation order of accuracy of the original polynomial.

Condition 1 Let $\{\mathbf{x}_{J_0}, \mathbf{x}_{J_1}, \dots, \mathbf{x}_{J_d}\}$ be the $d + 1$ cell centroids (P-centroids wherever P-averages are used) of a stencil. Then there is a point among them, say \mathbf{x}_{J_0} , such that the

¹At this stage, we have already found new values for all coefficients in the terms of $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ of degree above m . These coefficients remain in $R_{m,I}(\mathbf{x} - \mathbf{x}_I)$ (after taking a $(m - 1)^{th}$ order partial of $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$). When they are replaced by their corresponding new values, $R_{m,I}(\mathbf{x} - \mathbf{x}_I)$ becomes $\tilde{R}_{m,I}(\mathbf{x} - \mathbf{x}_I)$.

²For example, in 2D a stencil contains 3 new approximate cell averages. A linear polynomial can be determined by letting it equal each new approximate cell average at the corresponding cell centroid (P-centroid, if P-average is used in calculating the new approximate cell average). This linear polynomial approximates $L_{m,I}(\mathbf{x} - \mathbf{x}_I)$, thus provides candidate values of its coefficients.

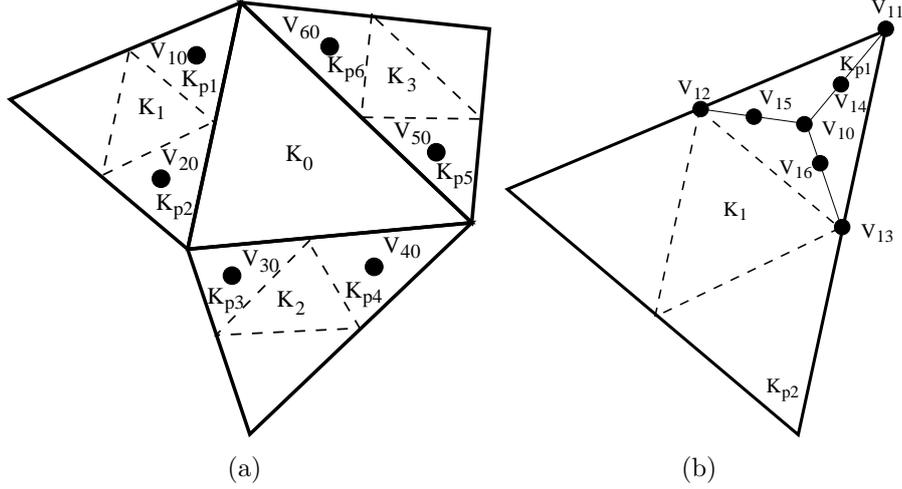


Figure 2: Schematic of the stencils for the 2D 3^{rd} order accurate point-wise HR on the target cell \mathcal{K}_0 . Cells $\mathcal{K}_{p1}, \dots, \mathcal{K}_{p6}$ are partial neighboring cells. (a) Stencil for reconstructing quadratic terms; V_{10}, \dots, V_{60} are the P-centroids, which are the centroids of partial neighboring cells $\mathcal{K}_{p1}, \dots, \mathcal{K}_{p6}$ respectively. (b) The P-centroid defined on the partial cell \mathcal{K}_{p1} of \mathcal{K}_1 for point-wise HR reconstructing the linear parts of the polynomial. The centroid V_{10} of \mathcal{K}_{p1} , its three vertices V_{11}, V_{12}, V_{13} , and the middle points V_{14}, V_{15}, V_{16} between V_{10} and $V_{1i}, i = 1, 2, 3$ respectively are averaged to give the P-centroid on \mathcal{K}_{p1} . Locations of points are marked by black dots “•” in (a) and (b).

matrix $A = \frac{1}{h}[\mathbf{x}_{J_1} - \mathbf{x}_{J_0}, \mathbf{x}_{J_2} - \mathbf{x}_{J_0}, \dots, \mathbf{x}_{J_d} - \mathbf{x}_{J_0}]$ is non singular. Further, there is a constant $\beta > 0$ independent of the mesh size h such that $\|A^{-1}\| \leq \beta$.

This condition requires that the distribution of the stencil centroids (P-centroids) are uniformly non singular for the interpolation of a linear polynomial in d dimensions.

Theorem 1 Suppose $u_{h,i}(\mathbf{x} - \mathbf{x}_i)$ in Algorithm 1 approximates a C^{r+1} function $u(\mathbf{x})$ with point-wise error $\mathcal{O}(h^{r+1})$ within cell $\mathcal{K}_i, i \in \{I, J\}$, and all cells in $\{\mathcal{K}_I, \mathcal{K}_J\}$ are contained in a circle centered at \mathbf{x}_I with radius $\mathcal{O}(h)$. Let the $d + 1$ cell centroids in every stencil used in Algorithm 1 satisfy Condition 1. Then after the application of Algorithm 1, the polynomial $\tilde{u}_{h,I}(\mathbf{x} - \mathbf{x}_I)$, i.e. $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$ with its coefficients replaced by the corresponding new values also approximates the function $u(\mathbf{x})$ with point-wise error $\mathcal{O}(h^{r+1})$ within cell \mathcal{K}_I . The cell average of $\tilde{u}_{h,I}(\mathbf{x} - \mathbf{x}_I)$ on cell \mathcal{K}_I is the same as that of $u_{h,I}(\mathbf{x} - \mathbf{x}_I)$.

Proof. The proof follows [17] exactly with cell averages replaced by P-averages and cell centroids replaced by P-centroids wherever they are used.

We now catalogue the details for implementing point-wise HR on triangular meshes for the 3^{rd} and 4^{th} order accurate schemes tested in the paper.

2.3.1 Point-wise HR for the 3^{rd} order solution (HR3)

We first describe the detailed steps for implementing the point-wise HR for limiting the 3^{rd} order solution polynomials on triangular meshes.

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

$$u_j(x - x_j, y - y_j) = u_j(0, 0) + \partial_x u_j(0, 0)(x - x_j) + \partial_y u_j(0, 0)(y - y_j) + \frac{1}{2} \partial_{xx} u_j(0, 0)(x - x_j)^2 + \partial_{xy} u_j(0, 0)(x - x_j)(y - y_j) + \frac{1}{2} \partial_{yy} u_j(0, 0)(y - y_j)^2, \quad (2.17)$$

where (x_j, y_j) is the centroid of cell \mathcal{K}_j . Here we neglect the subscript h for convenience. We will reconstruct a new polynomial in \mathcal{K}_0 with a point-wise error $\mathcal{O}(h^3)$, where h is the mesh size (or triangle edge length).

HR3 step 1: we conduct point-wise HR3 for stage $m = 2$.

We first take the $(m - 1 = 1)$ 1st partial derivative with respect to x for $u_j(x - x_j, y - y_j)$ to obtain

$$L_{2,j}(x - x_j, y - y_j) = \partial_x u_j(0, 0) + \partial_{xx} u_j(0, 0)(x - x_j) + \partial_{xy} u_j(0, 0)(y - y_j), \quad j = 0, \dots, 3. \quad (2.18)$$

On cell \mathcal{K}_0 , we calculate the cell average of $L_{2,0}(x - x_0, y - y_0)$ to obtain

$$\bar{L}_{2,0} = \partial_x u_0(0, 0).$$

On each of cells $\mathcal{K}_1, \mathcal{K}_2$ and \mathcal{K}_3 , we define two P-centroids. E.g., on cell \mathcal{K}_1 , we connect middle points of edges to form four smaller triangles $\mathcal{K}_{p1}, \mathcal{K}_{p2}, \dots$, (they are called partial neighboring cells in [29]). The two P-centroids on \mathcal{K}_1 , denoted by $P_{2,c1}$ and $P_{2,c2}$, are the centroids of two of these triangles $\mathcal{K}_{p1}, \mathcal{K}_{p2}$, sharing edges with \mathcal{K}_0 . Thus

$$P_{2,c1} = V_{10};$$

and

$$P_{2,c2} = V_{20}.$$

The P-centroids on cells \mathcal{K}_2 and \mathcal{K}_3 are similarly defined. See also Figure 2(a).

At these P-centroids $\{P_{2,cj} : j = 1, \dots, 6\}$, evaluate the P-averages of $L_{2,j}(x - x_j, y - y_j)$ to obtain

$$\bar{L}_{2,j} = \partial_x u_s(0, 0), \quad j = 1, \dots, 6.$$

Here

$$s = \begin{cases} 1, & \text{if } j = 1, 2, \\ 2, & \text{if } j = 3, 4, \\ 3, & \text{if } j = 5, 6. \end{cases}$$

Remark: at this stage, the point-wise HR and the previous HR with partial neighboring cells [29] are identical.

Apply the weighted non-oscillatory linear reconstruction procedure [29] with cell averages replaced by P-averages and cell centroids replaced by P-centroids wherever they are used to $\{\bar{L}_{2,j} : j = 0, \dots, 6\}$ to obtain a new linear polynomial on cell \mathcal{K}_0 :

$$\tilde{L}_{2,0}(x - x_0, y - y_0) = \partial_x \tilde{u}_0(0, 0) + \partial_{xx} \tilde{u}_0(0, 0)(x - x_0) + \partial_{xy} \tilde{u}_0(0, 0)(y - y_0), \quad (2.19)$$

with $\partial_x \tilde{u}_0(0, 0) = \bar{L}_{2,0}$.

Similarly we take the 1st partial derivative with respect to y for $u_j(x - x_j, y - y_j)$, $j = 0, \dots, 6$ to redefine

$$L_{2,j}(x - x_j, y - y_j) = \partial_y u_j(0, 0) + \partial_{xy} u_j(0, 0)(x - x_j) + \partial_{yy} u_j(0, 0)(y - y_j), j = 0, \dots, 6,$$

and perform the same weighted non-oscillatory linear reconstruction procedure to obtain another linear polynomial on \mathcal{K}_0 (which is still denoted as $\tilde{L}_{2,0}$ to avoid introducing too many notations):

$$\tilde{L}_{2,0}(x - x_0, y - y_0) = \partial_y \tilde{u}_0(0, 0) + \partial_{xy} \tilde{u}_0(0, 0)(x - x_0) + \partial_{yy} \tilde{u}_0(0, 0)(y - y_0). \quad (2.20)$$

$\partial_{xx} \tilde{u}_0(0, 0)$, $\partial_{xy} \tilde{u}_0(0, 0)$ and $\partial_{yy} \tilde{u}_0(0, 0)$ will be the corresponding new coefficients of the reconstructed quadratic polynomial. $\partial_{xy} \tilde{u}_0(0, 0)$ appears twice in the above procedure and is finalized by a limiter function identical to the one used in [29].

Remark:

1. *Reconstructed polynomials in equations (2.19) and (2.20) are all denoted as $\tilde{L}_{2,0}(x - x_0, y - y_0)$ to avoid introducing too many notations. The same notation rule also applies to the other hierarchical reconstruction stages.*

HR3 step 2: we conduct point-wise HR3 for stage $m = 1$.

On cell \mathcal{K}_0 , we first compute cell average of $u_0(x - x_0, y - y_0)$ to obtain \bar{u}_0 , and compute cell average of the polynomial (remainder on cell \mathcal{K}_0)

$$\tilde{R}_{1,0}(x - x_0, y - y_0) = \frac{1}{2} \partial_{xx} \tilde{u}_0(0, 0)(x - x_0)^2 + \partial_{xy} \tilde{u}_0(0, 0)(x - x_0)(y - y_0) + \frac{1}{2} \partial_{yy} \tilde{u}_0(0, 0)(y - y_0)^2 \quad (2.21)$$

on cell \mathcal{K}_0 to obtain $\overline{\tilde{R}_{1,0}}$.

Define $\bar{L}_{1,0} = \bar{u}_0 - \overline{\tilde{R}_{1,0}}$.

On each of cells $\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$, we redefine two P-centroids. E.g., on the partial neighboring cell \mathcal{K}_{p1} of \mathcal{K}_1 , the P-centroid $P_{1,c1}$ is defined to be the average of the centroid V_{10} of \mathcal{K}_{p1} , its three vertices V_{11}, V_{12}, V_{13} , and the middle points V_{14}, V_{15}, V_{16} between V_{10} and V_{1i} , $i = 1, 2, 3$ respectively (see Figure 2(b)). Thus

$$P_{1,c1} = \frac{1}{7} \sum_{s=0}^6 V_{1s}.$$

On the P-centroids $\{P_{1,cj} : j = 1, \dots, 6\}$, we define $\bar{L}_{1,j}$ by computing the P-average

$$\bar{L}_{1,j} = \frac{1}{7} \sum_{s=0}^6 \left(u_i(x_{j,s} - x_j, y_{j,s} - y_j) - \tilde{R}_{1,0}(x_{j,s} - x_0, y_{j,s} - y_0) \right), \quad (2.22)$$

where $(x_{j,s}, y_{j,s})$ are points made of the j^{th} P-centroid $P_{1,cj}$. And

$$i = \begin{cases} 1, & \text{if } j = 1, 2, \\ 2, & \text{if } j = 3, 4, \\ 3, & \text{if } j = 5, 6. \end{cases}$$

The weighted non-oscillatory linear reconstruction procedure with appropriate weight functions [29] with cell averages replaced by P-averages and cell centroids replaced by P-centroids wherever they are used, is applied to $\{\bar{L}_{1,j} : j = 0, \dots, 6\}$ to obtain the new linear polynomial

$$\tilde{L}_{1,0}(x - x_0, y - y_0) = \bar{L}_{1,0} + \partial_x \tilde{u}_0(0, 0)(x - x_0) + \partial_y \tilde{u}_0(0, 0)(y - y_0) \quad (2.23)$$

with the new coefficients $\partial_x \tilde{u}_0(0, 0)$ and $\partial_y \tilde{u}_0(0, 0)$. Finally we let the new coefficient $\tilde{u}_0(0, 0) = \bar{L}_{1,0}$ to ensure conservation.

This completes the reconstruction for the 2^{nd} degree polynomial $u_0(x - x_0, y - y_0)$.

2.3.2 Point-wise HR for the 4^{th} order DG solution (HR4)

We now catalogue the detailed steps for implementing the point-wise HR for reconstructing the 4^{th} order RKDG solution polynomials on triangular meshes. The implementation for limiting the 4^{th} order finite volume solution is briefly discussed at the end of this section.

Suppose on each cell $\mathcal{K}_j \in \{\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ in Figure 3, a cubic polynomial approximate solution is given in the form of a two-dimensional Taylor expansion

$$\begin{aligned} u_j(x - x_j, y - y_j) = & u_j(0, 0) + \partial_x u_j(0, 0)(x - x_j) + \partial_y u_j(0, 0)(y - y_j) + \\ & \frac{1}{2} \partial_{xx} u_j(0, 0)(x - x_j)^2 + \partial_{xy} u_j(0, 0)(x - x_j)(y - y_j) + \\ & \frac{1}{2} \partial_{yy} u_j(0, 0)(y - y_j)^2 + \frac{1}{6} \partial_{xxx} u_j(0, 0)(x - x_j)^3 + \\ & \frac{1}{2} \partial_{xxy} u_j(0, 0)(x - x_j)^2(y - y_j) + \frac{1}{2} \partial_{xyy} u_j(0, 0)(x - x_j)(y - y_j)^2 + \\ & \frac{1}{6} \partial_{yyy} u_j(0, 0)(y - y_j)^3, \end{aligned} \quad (2.24)$$

where (x_j, y_j) is the centroid of cell \mathcal{K}_j . Here we neglect the subscript h for convenience. We will reconstruct a new polynomial in \mathcal{K}_0 with a point-wise error $\mathcal{O}(h^4)$, where h is the mesh size.

HR4 step 1: we first conduct point-wise HR4 for stage $m = 3$.

We first take the $(m - 1 = 2)$ 2^{nd} partial derivative with respect to x for $u_j(x - x_j, y - y_j)$ to obtain

$$L_{3,j}(x - x_j, y - y_j) = \partial_{xx} u_j(0, 0) + \partial_{xxx} u_j(0, 0)(x - x_j) + \partial_{xxy} u_j(0, 0)(y - y_j), \quad j = 0, \dots, 3. \quad (2.25)$$

On cell \mathcal{K}_0 , we calculate the cell average of $L_{3,0}(x - x_0, y - y_0)$ to obtain

$$\bar{L}_{3,0} = \partial_{xx} u_0(0, 0).$$

On each of cells $\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$, we build two P-centroids. E.g., on cell \mathcal{K}_1 , we connect middle points of edges to form four partial neighboring triangles $\mathcal{K}_{p1}, \mathcal{K}_{p2}, \dots$. The two P-centroids on \mathcal{K}_1 , denoted by $P_{2,c1}$ and $P_{2,c2}$, are the centroids of two of these triangles $\mathcal{K}_{p1}, \mathcal{K}_{p2}$, sharing edges with \mathcal{K}_0 . Thus

$$P_{3,c1} = V_{10};$$

and

$$P_{3,c2} = V_{20}.$$

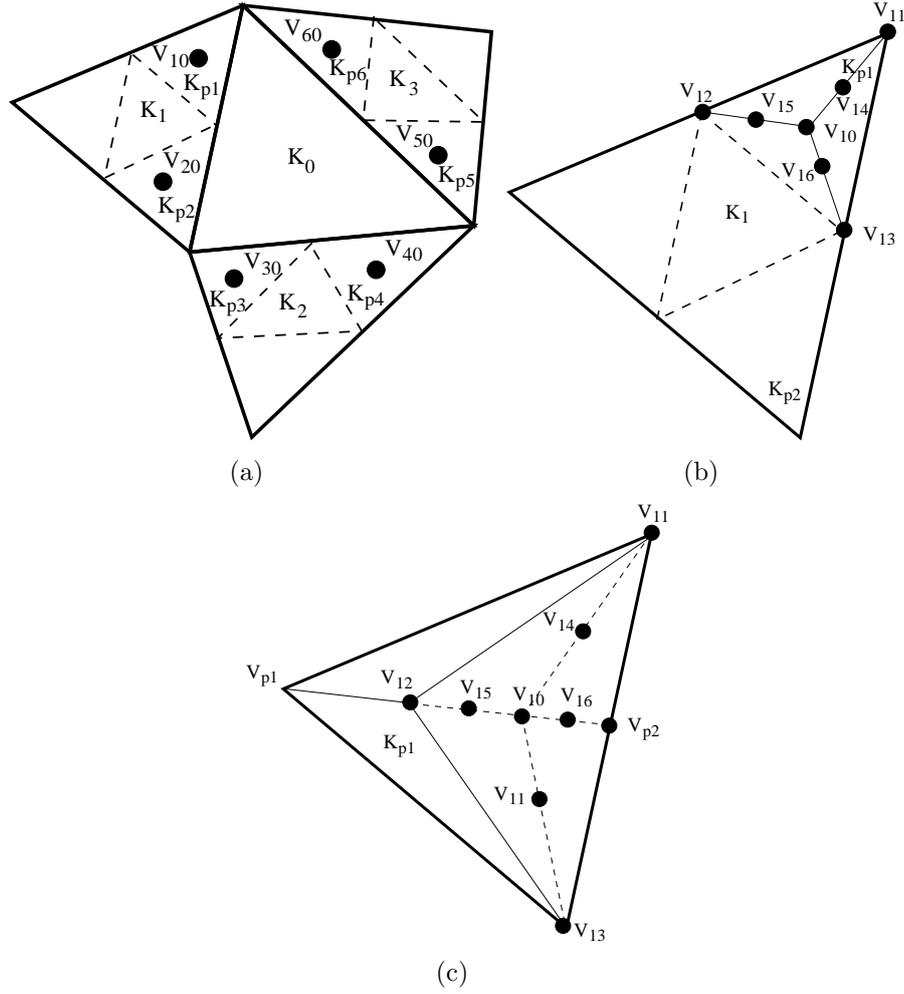


Figure 3: Schematic of the stencils for the 2D point-wise HR for reconstructing the 4th order accurate DG solution on the target cell \mathcal{K}_0 . Cells $\mathcal{K}_{p1}, \dots, \mathcal{K}_{p6}$ are partial neighboring cells. (a) Stencil for reconstructing cubic terms; V_{10}, \dots, V_{60} are the P-centroids, which are the centroids of partial neighboring cells $\mathcal{K}_{p1}, \dots, \mathcal{K}_{p6}$ respectively. (b) The P-centroid defined on the partial cell \mathcal{K}_{p1} of \mathcal{K}_1 for point-wise HR reconstructing the quadratic part of the polynomial. The centroid V_{10} of \mathcal{K}_{p1} , its three vertices V_{11}, V_{12}, V_{13} , and the middle points V_{14}, V_{15}, V_{16} between V_{10} and $V_{1i}, i = 1, 2, 3$ respectively are averaged to give the P-centroid on \mathcal{K}_{p1} . (c) The P-centroid defined on the partial neighboring cell \mathcal{K}_{p1} for reconstructing the linear part of the polynomial. The partial cell with vertices V_{11}, V_{12} and V_{13} is constructed in \mathcal{K}_{p1} . The P-centroid defined on \mathcal{K}_{p1} is the average of the centroid V_{10} of the partial cell of \mathcal{K}_{p1} , its three vertices V_{11}, V_{12}, V_{13} , and the middle points between V_{10} and $V_{1i}, i = 1, 2, 3$ respectively. Locations of points are marked by black dots “•” in (a), (b) and (c). *Note that the same notations are used for different points in (b) and (c) to avoid introducing too many notations. They should be understood as different points for the P-centroids used at different HR stages.*

The P-centroids on cells \mathcal{K}_2 and \mathcal{K}_3 are similarly defined. See also Figure 3(a).

Remark: The P-centroid at this stage of the 4th order HR is identical to that of the 3rd order HR3 with stage $m = 2$.

At the P-centroids $\{P_{3,cj} : j = 1, \dots, 6\}$, calculate the P-average of $L_{3,j}(x - x_j, y - y_j)$ to obtain

$$\bar{L}_{3,j} = \partial_{xx}u_s(0, 0), \quad j = 1, \dots, 6.$$

Here

$$s = \begin{cases} 1, & \text{if } j = 1, 2, \\ 2, & \text{if } j = 3, 4, \\ 3, & \text{if } j = 5, 6. \end{cases}$$

Then we apply a weighted non-oscillatory linear reconstruction procedure, which is similar to [29] and will be described in Section 2.3.3, to cell averages $\{\bar{L}_{3,j} : j = 0, \dots, 6\}$ to obtain a new linear polynomial on cell \mathcal{K}_0 :

$$\tilde{L}_{3,0}(x - x_0, y - y_0) = \partial_{xx}\tilde{u}_0(0, 0) + \partial_{xxx}\tilde{u}_0(0, 0)(x - x_0) + \partial_{xxy}\tilde{u}_0(0, 0)(y - y_0), \quad (2.26)$$

with $\partial_{xx}\tilde{u}_0(0, 0) = \bar{L}_0$.

Similarly we take the 2nd partial derivative with respect to x and y for $u_j(x - x_j, y - y_j)$, $j = 0, \dots, 3$ to redefine

$$L_{3,j}(x - x_j, y - y_j) = \partial_{xy}u_j(0, 0) + \partial_{xxy}u_j(0, 0)(x - x_j) + \partial_{xyy}u_j(0, 0)(y - y_j), \quad j = 0, \dots, 3,$$

and perform the same reconstruction procedure to obtain another linear polynomial on \mathcal{K}_0 (still denoted by $\tilde{L}_{3,0}$):

$$\tilde{L}_{3,0}(x - x_0, y - y_0) = \partial_{xy}\tilde{u}_0(0, 0) + \partial_{xxy}\tilde{u}_0(0, 0)(x - x_0) + \partial_{xyy}\tilde{u}_0(0, 0)(y - y_0). \quad (2.27)$$

Finally, we take the 2nd partial derivative with respect to y for $u_j(x - x_j, y - y_j)$, $j = 0, \dots, 3$ to redefine

$$L_{3,j}(x - x_j, y - y_j) = \partial_{yy}u_j(0, 0) + \partial_{xyy}u_j(0, 0)(x - x_j) + \partial_{yyy}u_j(0, 0)(y - y_j), \quad j = 0, \dots, 3,$$

and perform the same reconstruction procedure to obtain the 3rd linear polynomial on \mathcal{K}_0 (still denoted by $\tilde{L}_{3,0}$ to avoid too many notations):

$$\tilde{L}_{3,0}(x - x_0, y - y_0) = \partial_{yy}\tilde{u}_0(0, 0) + \partial_{xyy}\tilde{u}_0(0, 0)(x - x_0) + \partial_{yyy}\tilde{u}_0(0, 0)(y - y_0). \quad (2.28)$$

$\partial_{xxx}\tilde{u}_0(0, 0)$, $\partial_{xxy}\tilde{u}_0(0, 0)$, $\partial_{xyy}\tilde{u}_0(0, 0)$ and $\partial_{yyy}\tilde{u}_0(0, 0)$ will be the corresponding new coefficients of the reconstructed cubic polynomial. $\partial_{xxy}\tilde{u}_0(0, 0)$ and $\partial_{xyy}\tilde{u}_0(0, 0)$ appear twice in the above procedures (see equations (2.26), (2.27) and (2.28)) and is finalized by a limiter function described in Section 2.3.3.

Remark:

1. For reconstructing coefficients of the q^{th} degree terms, point-wise HR is identical to the original HR.

HR4 step 2: we now conduct HR4 for stage $m = 2$.

For example, taking the $(m - 1 = 1)$ 1st partial derivative with respect to x gives

$$\begin{aligned}\partial_x u_j(x - x_j, y - y_j) &= \partial_x u_{h,j}(0, 0) + \partial_{xx} u_j(0, 0)(x - x_j) + \\ &\quad \partial_{xy} u_j(0, 0)(y - y_j) + \frac{1}{2} \partial_{xxx} u_j(0, 0)(x - x_j)^2 + \\ &\quad \partial_{xxy} u_j(0, 0)(x - x_j)(y - y_j) + \frac{1}{2} \partial_{xyy} u_j(0, 0)(y - y_j)^2 \\ &= L_{2,j}(x - x_j, y - y_j) + R_{2,j}(x - x_j, y - y_j), \quad j = 0, \dots, 3,\end{aligned}\quad (2.29)$$

where $L_{2,j}(x - x_j, y - y_j)$ is the linear part of $\partial_x u_j(x - x_j, y - y_j)$ and $R_{2,j}(x - x_j, y - y_j)$ is the remainder.

On cell \mathcal{K}_0 , we compute cell average of $\partial_x u_0(x - x_0, y - y_0)$ to obtain $\overline{\partial_x u_0}$ and compute cell average of the polynomial (remainder on cell \mathcal{K}_0)

$$\begin{aligned}\widetilde{R}_{2,0}(x - x_0, y - y_0) &= \frac{1}{2} \partial_{xxx} \tilde{u}_0(0, 0)(x - x_0)^2 + \\ &\quad \partial_{xxy} \tilde{u}_0(0, 0)(x - x_0)(y - y_0) + \frac{1}{2} \partial_{xyy} \tilde{u}_0(0, 0)(y - y_0)^2,\end{aligned}\quad (2.30)$$

to obtain $\overline{\widetilde{R}_{2,0}}$.

We then define $\overline{L}_{2,0} = \overline{\partial_x u_0} - \overline{\widetilde{R}_{2,0}}$.

On the P-centroids $\{P_{2,cj} : j = 1, \dots, 6\}$ (see Figure 3(b)), we define $\overline{L}_{2,j}$ by computing P-averages

$$\overline{L}_{2,j} = \frac{1}{7} \sum_{s=0}^6 \left(\partial_x u_i(x_{j,s} - x_j, y_{j,s} - y_j) - \widetilde{R}_{2,0}(x_{j,s} - x_0, y_{j,s} - y_0) \right), \quad (2.31)$$

where $(x_{j,s}, y_{j,s})$ are points made of the j^{th} P-centroid $P_{2,cj}$. And

$$i = \begin{cases} 1, & \text{if } j = 1, 2, \\ 2, & \text{if } j = 3, 4, \\ 3, & \text{if } j = 5, 6. \end{cases}$$

A weighted non-oscillatory reconstruction procedure which will be described in Section 2.3.3 is applied to cell averages $\{\overline{L}_{2,j} : j = 0, \dots, 6\}$ to obtain the new linear polynomial

$$\widetilde{L}_{2,0}(x - x_0, y - y_0) = \overline{L}_{2,0} + \partial_{xx} \tilde{u}_0(0, 0)(x - x_0) + \partial_{xy} \tilde{u}_0(0, 0)(y - y_0) \quad (2.32)$$

with new coefficient $\partial_{xx} \tilde{u}_0(0, 0)$, and a candidate for $\partial_{xy} \tilde{u}_0(0, 0)$.

We then take the 1st partial derivative with respect to y and repeat the above procedure to obtain the new coefficient $\partial_{yy} \tilde{u}_0(0, 0)$, and another candidate for $\partial_{xy} \tilde{u}_0(0, 0)$. We denote the obtained new linear polynomial

$$\widetilde{L}_{2,0}(x - x_0, y - y_0) = \overline{L}_{2,0} + \partial_{xy} \tilde{u}_0(0, 0)(x - x_0) + \partial_{yy} \tilde{u}_0(0, 0)(y - y_0) \quad (2.33)$$

for the future references.

Finally, $\partial_{xy} \tilde{u}_0(0, 0)$ is chosen from these two candidates in (2.32) and (2.33) by a limiter function described in Section 2.3.3.

HR4 step 3: we now conduct HR4 for stage $m = 1$.

On cell \mathcal{K}_0 , we first compute cell average of $u_0(x - x_0, y - y_0)$ to obtain \bar{u}_0 , and compute cell average of the polynomial (remainder on cell \mathcal{K}_0)

$$\begin{aligned} \tilde{R}_{1,0}(x - x_0, y - y_0) = & \frac{1}{2}\partial_{xx}\tilde{u}_0(0,0)(x - x_0)^2 + \partial_{xy}\tilde{u}_0(0,0)(x - x_0)(y - y_0) + \\ & \frac{1}{2}\partial_{yy}\tilde{u}_0(0,0)(y - y_0)^2 + \frac{1}{6}\partial_{xxx}\tilde{u}_0(0,0)(x - x_0)^3 + \\ & \frac{1}{2}\partial_{xxy}\tilde{u}_0(0,0)(x - x_0)^2(y - y_0) + \frac{1}{2}\partial_{xyy}\tilde{u}_0(0,0)(x - x_0)(y - y_0)^2 + \\ & \frac{1}{6}\partial_{yyy}\tilde{u}_0(0,0)(y - y_0)^3 \end{aligned} \quad (2.34)$$

on cell \mathcal{K}_0 to obtain $\overline{\tilde{R}_{1,0}}$.

We redefine $\bar{L}_{1,0} = \bar{u}_0 - \overline{\tilde{R}_{1,0}}$.

On each of partial neighboring cells $\{\mathcal{K}_{p1}, \dots, \mathcal{K}_{p6}\}$, we redefine a P-centroid. E.g. the P-centroid defined on the partial neighboring cell \mathcal{K}_{p1} for reconstructing the linear part of the polynomial. Vertices of \mathcal{K}_{p1} are labeled as V_{p1} , V_{11} and V_{13} . We first build a partial cell of the partial neighboring cell \mathcal{K}_{p1} . The two of the vertices of this partial cell of \mathcal{K}_{p1} are the two endpoints V_{11} and V_{13} of the edge of \mathcal{K}_{p1} neighboring \mathcal{K}_0 . The third vertex V_{12} is computed as follows: Let V_{p2} be the middle point between V_{11} and V_{13} . $V_{12} = V_{p2} + \frac{7}{20}(V_{p1} - V_{p2})$. The P-centroid defined on \mathcal{K}_{p1} is the average of the centroid V_{10} of the partial cell of \mathcal{K}_{p1} , its three vertices V_{11} , V_{12} , V_{13} , and the middle points between V_{10} and V_{1i} , $i = 1, 2, 3$ respectively. We determine the location of V_{12} based on numerical experiments. When V_{12} is close to V_{p1} , we observed increased undershoots/overshoots in numerical solutions (see also Figure 3(c)).

On the P-centroids $\{P_{1,cj} : j = 1, \dots, 6\}$ (see Figure 3(c)), we redefine $\bar{L}_{1,j}$ by computing the P-average

$$\bar{L}_{1,j} = \frac{1}{7} \sum_{s=0}^6 \left(u_i(x_{j,s} - x_j, y_{j,s} - y_j) - \tilde{R}_{1,0}(x_{j,s} - x_0, y_{j,s} - y_0) \right), \quad (2.35)$$

where $(x_{j,s}, y_{j,s})$ are points made of the j^{th} P-centroid $P_{1,cj}$. And

$$i = \begin{cases} 1, & \text{if } j = 1, 2, \\ 2, & \text{if } j = 3, 4, \\ 3, & \text{if } j = 5, 6. \end{cases}$$

The weighted non-oscillatory reconstruction procedure described in Section 2.3.3 is applied to cell averages $\{\bar{L}_{1,j}\}$ to obtain the new linear polynomial

$$\tilde{L}_{1,0}(x - x_0, y - y_0) = \bar{L}_{1,0} + \partial_x \tilde{u}_0(0,0)(x - x_0) + \partial_y \tilde{u}_0(0,0)(y - y_0) \quad (2.36)$$

with the new coefficients $\partial_x \tilde{u}_0(0,0)$ and $\partial_y \tilde{u}_0(0,0)$.

Finally we let the new coefficient $\tilde{u}_0(0,0) = \bar{L}_{1,0}$ to ensure conservation. This completes the reconstruction for the 3^{rd} degree polynomial $u_0(x - x_0, y - y_0)$.

2.3.3 Weighted linear reconstruction procedure for HR4

We now describe the weighted linear reconstruction procedure for the HR4 for the DG solution case. The idea of weighted linear reconstruction procedure and the choice of weight functions for the 4^{th} order point-wise HR (HR4) follow [29] with the following modification.

Let's denote the centroid of \mathcal{K}_0 be P_0 . To compute new coefficients $\tilde{u}_0^{\mathbf{m}}(0)$ for the reconstructed polynomial

$$u_{h,0}(\mathbf{x} - \mathbf{x}_0) = \sum_{m=0}^q \sum_{|\mathbf{m}|} \frac{1}{\mathbf{m}!} \tilde{u}_0^{\mathbf{m}}(0) (\mathbf{x} - \mathbf{x}_0)^{\mathbf{m}}, \quad (2.37)$$

we form six stencils which are collected in the set \mathcal{S} :

$\mathcal{S} \equiv \{\{P_0, P_{m,c1}, P_{m,c2}\}, \{P_0, P_{m,c2}, P_{m,c3}\}, \{P_0, P_{m,c3}, P_{m,c4}\}, \{P_0, P_{m,c4}, P_{m,c5}\}, \{P_0, P_{m,c5}, P_{m,c6}\}$ and $\{P_0, P_{m,c6}, P_{m,c1}\}\}$ for the m^{th} HR stage and perform the following sub-steps:

Sub-step 1:

Denote $L_{m,0,j}(x - x_0, y - y_0) \equiv \bar{L}_{m,0} + a_{0,j}(0,0)(x - x_0) + b_{0,j}(0,0)(y - y_0)$. At each HR stage $m = 3, 2, 1$, on each of the stencil $\{P_0, P_{m,cj}, P_{m,c(j+1)\%6}\}, j = 1, \dots, 6$, we solve a linear equation for $a_{0,j}(0,0), b_{0,j}(0,0)$ in the form:

$$L_{m,0,j} \equiv \bar{L}_{m,0} + a_{0,j}(0,0)(x_{i,m} - x_0) + b_{0,j}(0,0)(y_{i,m} - y_0) = \bar{L}_{m,i}, \quad (2.38)$$

with $i = 1, 2, \dots, 6$ respectively. Here $a_{0,j}(0,0), b_{0,j}(0,0)$ represents candidates for finalizing the choice of new coefficients $\tilde{u}_0^{\mathbf{m}}(0)$; (x_0, y_0) is the coordinates of P_0 ; and $(x_{i,m}, y_{i,m})$ is the coordinates of the P-centroid used to evaluate $\bar{L}_{m,i}$.

Sub-step 2:

The reconstructed linear polynomial for each of (2.26), (2.27), (2.28), (2.32), (2.33) and (2.23) is a convex combination of these computed linear polynomials $L_{m,0,j}(x - x_0, y - y_0)$. We catalogue the formulas of the convex combination for each HR4 stage in **Weight function selection**.

This completes the sub-steps for the weighted linear reconstruction procedure.

Here we explain the choice of weight functions for HR4.

Weight function selection for HR4, stage $m = 3$

Consider the reconstruction of polynomial (2.26) as an example. In **Sub-step 1**, on the first stencil $\{P_0, P_{3,c1}, P_{3,c2}\}$, we solve the following equations for $a_{0,1}(0,0) \equiv \partial_{xxx} \tilde{u}_{0,1}(0,0)$ and $b_{0,1}(0,0) \equiv \partial_{xxy} \tilde{u}_{0,1}(0,0)$

$$\bar{L}_{3,0} + \partial_{xxx} \tilde{u}_{0,1}(0,0)(x_{i,q} - x_0) + \partial_{xxy} \tilde{u}_{0,1}(0,0)(y_{i,q} - y_0) = \bar{L}_{3,i}, \quad (2.39)$$

where $i = 1, 2$ and $q = 3$.

Repeat the procedure for solving equation (2.39) for the rest of stencils in \mathcal{S} . The corresponding gradients of these linear polynomials are $\{\partial_{xxx} \tilde{u}_{0,j}(0,0), \partial_{xxy} \tilde{u}_{0,j}(0,0)\}, j = 1, \dots, 6$, respectively.

The reconstructed linear polynomial (2.26) is a convex combination of these computed linear polynomials $L_{3,0,j}(x - x_0, y - y_0)$. The weights in **Sub-step 2** are determined as follows.

Let the weights w_r be smooth functions of involved cell averages $\bar{L}_{3,j}$, and be set as follows:

$$w_r = \frac{\alpha_r}{\sum_{s=1}^6 \alpha_s}, \quad r = 1, \dots, 6, \quad (2.40)$$

where α_r are to be defined later. Let

$$d_r = \frac{1/\theta_r}{\sum_{s=1}^6 1/\theta_s}, \quad (2.41)$$

where $\theta_r = \|A\| \|A^{-1}\|$ is the condition number, A is the coefficient matrix of Eq. (2.39) for the corresponding stencil r , $\|\cdot\|$ denotes the 1-norm. This choice of d_r puts the condition numbers of stencils into consideration, and candidates of new coefficients computed from a stencil with bad condition number have less weights. Let

$$\alpha_r = \frac{d_r}{1 + h\beta_r^2}, \quad (2.42)$$

where the smoothness indicator

$$\beta_r = (\partial_{xxx}\tilde{u}_{0,r}(0,0))^2 + (\partial_{xxy}\tilde{u}_{0,r}(0,0))^2. \quad (2.43)$$

After the weights w_r are computed, the new coefficient $\partial_{xxx}\tilde{u}_0(0,0)$ is defined to be

$$\partial_{xxx}\tilde{u}_0(0,0) = \begin{cases} \sum_{r=1}^3 w_r \partial_{xxx}\tilde{u}_{0,r}(0,0), & \text{if } \bar{L}_{\min} < \bar{L}_{3,0} < \bar{L}_{\max}, \\ 0, & \text{otherwise,} \end{cases} \quad (2.44)$$

where $\bar{L}_{\min} = \min\{\bar{L}_{3,j} : j = 0, \dots, 6\}$ and $\bar{L}_{\max} = \max\{\bar{L}_{3,j} : j = 0, \dots, 6\}$. Violation of $\bar{L}_{\min} < \bar{L}_{3,0} < \bar{L}_{\max}$ detects an extreme value, hence the gradient of (2.26) is set to be zero to further reduce oscillations. The candidate coefficient $\partial_{xxy}\tilde{u}_0(0,0)$ is determined similarly.

The reconstruction of functions (2.27) and (2.28) follows the **Sub-steps 1 and 2**.

After the reconstruction of functions (2.26), (2.27) and (2.28), $\partial_{xxx}\tilde{u}_0(0,0)$ and $\partial_{yyy}\tilde{u}_0(0,0)$ are corresponding new coefficients for function $u_0(x - x_0, y - y_0)$ as in (2.24). It also leaves us two choices for each of the new coefficients $\partial_{xxy}\tilde{u}_0(0,0)$ and $\partial_{xyy}\tilde{u}_0(0,0)$. We put these choices into arguments of the center biased ENO limiter function m_{2b} to obtain the new coefficient. Here,

$$\begin{aligned} m_2(c_1, c_2, \dots, c_r) &= c_j, \text{ if } c_j = \min\{|c_1|, |c_2|, \dots, |c_r|\}, \\ m_{2b}(c_1, c_2, \dots, c_r) &= m_2\left((1 + \varepsilon)m_2(c_1, c_2, \dots, c_r), \frac{1}{r} \sum_{i=1}^r c_i\right), \end{aligned} \quad (2.45)$$

where ε is a small perturbation number and is set to be 0.01. This completes the computation for $\partial_{xxx}\tilde{u}_0(0,0)$, $\partial_{xxy}\tilde{u}_0(0,0)$ and $\partial_{xyy}\tilde{u}_0(0,0)$.

Weight function selection for HR4, stage $m = 2$

To compute the new coefficients $\partial_{xx}\tilde{u}_0(0,0)$, $\partial_{yy}\tilde{u}_0(0,0)$ and $\partial_{xy}\tilde{u}_0(0,0)$ in equations (2.32) and (2.33) respectively, **Sub-step 1 and 2** are repeated for $m = 2$. Take (2.32) as the example. We solve equation (2.38) for $m = 2$, and $\bar{L}_{2,j}$ are obtained by taking 1st partial derivative with respect to x in the HR4 with stage $m = 2$. Here

$$a_{0,j}(0,0) \equiv \partial_{xx}\tilde{u}_0(0,0), \quad b_{0,j}(0,0) \equiv \partial_{xy}\tilde{u}_0(0,0),$$

which represent candidates for finalizing the choice of $\partial_{xx}\tilde{u}_0(0,0)$ and $\partial_{xy}\tilde{u}_0(0,0)$ respectively.

In the **Sub-step 2** for computing the new linear polynomial (2.32), the following weights are used:

$$\alpha_r = \frac{d_r}{1 + \beta_r} . \quad (2.46)$$

For example, in the case of taking $\partial_x u_j(x - x_j, y - y_j)$, the smoothness indicator β_r now is

$$\beta_r = (\partial_{xx} \tilde{u}_{0,r}(0,0))^2 + (\partial_{xy} \tilde{u}_{0,r}(0,0))^2 . \quad (2.47)$$

We again have two choices of the new coefficient $\partial_{xy} \tilde{u}_0(0,0)$, each of which is from (2.32) and (2.33) respectively. We use the limiter function m_{2b} to finalize the value of $\partial_{xy} \tilde{u}_0(0,0)$. This completes the computation for $\partial_{xx} \tilde{u}_0(0,0)$, $\partial_{xy} \tilde{u}_0(0,0)$ and $\partial_{yy} \tilde{u}_0(0,0)$.

Weight function selection for HR4, stage $m = 1$

To compute the new coefficients $\partial_x \tilde{u}_0(0,0)$, $\partial_y \tilde{u}_0(0,0)$ in (2.23), **Sub-step 1** and **2** are repeated for $m = 1$. The following weights similar to those in [24] are used:

$$\alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2} , \quad (2.48)$$

where the smoothness indicator

$$\beta_r = (\partial_x \tilde{u}_{0,r}(0,0))^2 + (\partial_y \tilde{u}_{0,r}(0,0))^2 . \quad (2.49)$$

Here ϵ is a small positive number introduced to avoid the denominator to become 0.

For the reconstruction of the third degree polynomials, the extreme value detector (*i.e.*, the "0" case in (2.44)) is applied at every HR stage, which is different from the reconstruction of the second degree polynomial in [29].

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

2.3.4 Point-wise HR for the 4th order finite volume solution

We briefly discuss the limiting procedure for the 4th order finite volume solution using the point-wise HR. The only difference between HR for the finite volume solution and HR for the DG solution exists in the choice of P-centroids. We also note that in the weighted linear reconstruction procedure described in Section 2.3.3, we only have three P-centroids at stages $m = 3$ and $m - 2$. Therefore, we focus on the discussion of P-centroid for the finite volume solution case in this section.

When we conduct HR for stage $m = 3$, the centroids of each of cells $\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ are chosen to be the P-centroids (see Figure 4(a)). Thus we have

$$P_{3,cj} = V_{j0}, \quad j = 1, 2, 3 . \quad (2.50)$$

When we conduct HR for stage $m = 2$, the P-centroids of each of cells $\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ are the average of seven points respectively. E.g., on \mathcal{K}_1 , the centroid V_{10} of \mathcal{K}_1 , its three vertices V_{11}, V_{12}, V_{13} , and the middle points V_{14}, V_{15}, V_{16} between V_{10} and $V_{1i}, i = 1, 2, 3$ respectively

are averaged to give the P-centroid on \mathcal{K}_1 , (see Figure 4(b)). Therefore we have $\{P_{2,cj} : j = 1, 2, 3\}$, where

$$P_{2,cj} = \frac{1}{7} \sum_{s=0}^6 V_{j,s} . \quad (2.51)$$

When we conduct HR for stage $m = 1$, on each of cells $\{\mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$, we construct two P-centroids. E.g., on cell \mathcal{K}_1 , we connect middle points of edges to form four partial neighboring cells. \mathcal{K}_{p1} and \mathcal{K}_{p2} are two partial neighboring cells in \mathcal{K}_1 , which are adjacent to \mathcal{K}_0 . The P-centroid defined on \mathcal{K}_{p1} for reconstructing the linear part of the polynomial is the average of the centroid V_{10} of the partial cell of \mathcal{K}_{p1} , its three vertices V_{11}, V_{12}, V_{13} , and the middle points between V_{10} and V_{1i} , $i = 1, 2, 3$ respectively, (see Figure 4(c)). At this stage, we have $\{P_{1,cj} : j = 1, \dots, 6\}$, where

$$P_{1,cj} = \frac{1}{7} \sum_{s=0}^6 V_{j,s} . \quad (2.52)$$

Remark:

The same notations are used for different points in Figures (4(a)), (4(b)) and (4(c)) as well as Eqns. (2.50), (2.51) and (2.52) to avoid introducing too many notations. They should be understood as different points for the P-centroids used at different HR stages.

2.4 Local limiting procedure

We again employ the local limiting procedure in [29] to speed up the computation except for the accuracy test problems. First we identify cells may contain spurious oscillations at the beginning of the limiting procedure. Then we apply HR in the limiting procedure to solutions supported on these cells.

3 Numerical Examples

We first test the limiter's ability to achieve the desired order of accuracy, using the scalar Burgers' equation and the Euler equation for gas dynamics. In the two-dimensional space, the Euler equation can be expressed in a conservative form as

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

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 equation for gas dynamics.

3.1 Accuracy test for the 2D Burgers' equation with smooth solution

We start with the 2D Burgers' equation

$$\partial_t u + \partial_x \left(\frac{u^2}{2} \right) + \partial_y \left(\frac{u^2}{2} \right) = 0, \quad \text{in } (0, T) \times \Omega , \quad (3.2)$$

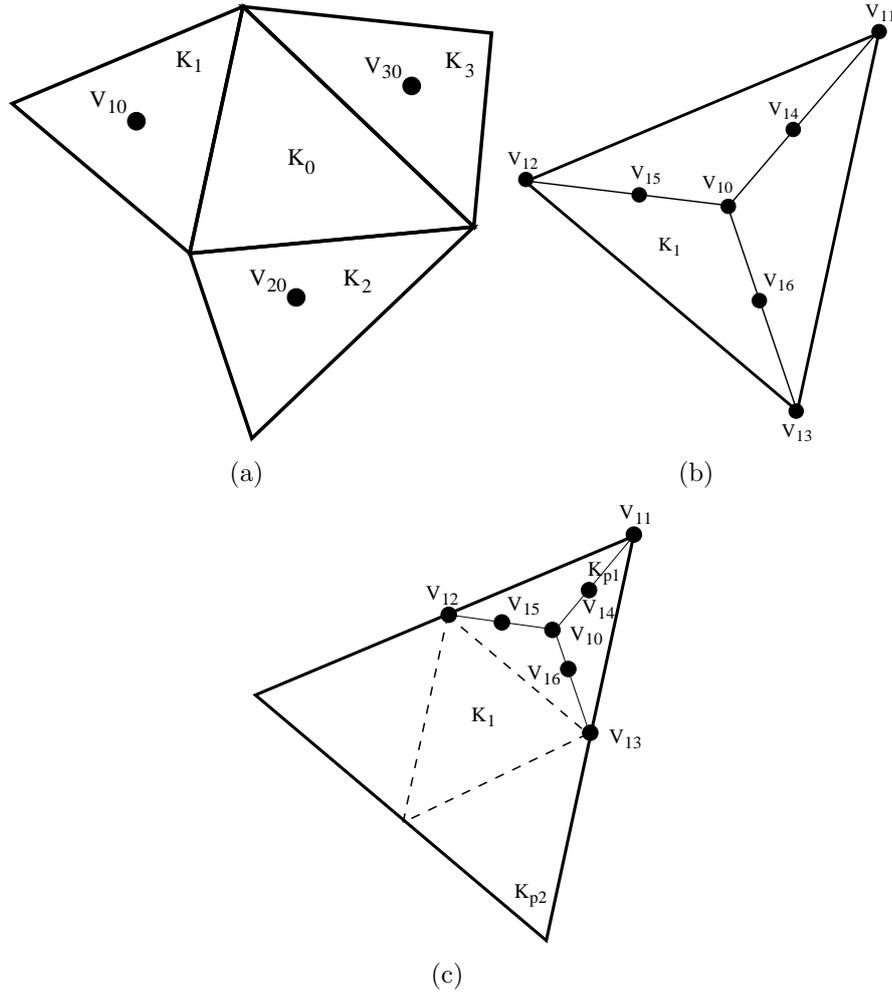


Figure 4: Schematic of the stencils for the 2D point-wise HR for reconstructing the 4th order accurate finite volume solution on the target cell \mathcal{K}_0 . (a) Stencil for reconstructing cubic terms; V_{10}, \dots, V_{30} are the P-centroids, which are the centroids of cells $\mathcal{K}_1, \mathcal{K}_2$ and \mathcal{K}_3 respectively. (b) The P-centroid defined on the cell \mathcal{K}_1 for point-wise HR reconstructing the quadratic part of the polynomial. The centroid V_{10} of \mathcal{K}_1 , its three vertices V_{11}, V_{12}, V_{13} , and the middle points V_{14}, V_{15}, V_{16} between V_{10} and $V_{1i}, i = 1, 2, 3$ respectively are averaged to give the P-centroid on \mathcal{K}_1 . (c) Partial neighboring cells are formed in $\mathcal{K}_1, \mathcal{K}_2$ and \mathcal{K}_3 by connecting middle points on cell edges respectively. \mathcal{K}_{p1} and \mathcal{K}_{p2} are two partial neighboring cells in \mathcal{K}_1 . The P-centroid defined on \mathcal{K}_{p1} for reconstructing the linear part of the polynomial is the average of the centroid V_{10} of the partial cell of \mathcal{K}_{p1} , its three vertices V_{11}, V_{12}, V_{13} , and the middle points between V_{10} and $V_{1i}, i = 1, 2, 3$ respectively. Locations of points are marked by black dots “•” in (a), (b) and (c). *Note that the same notations are used for different points in (a), (b) and (c) to avoid introducing too many notations. They should be understood as different points for the P-centroids used at different HR stages.*

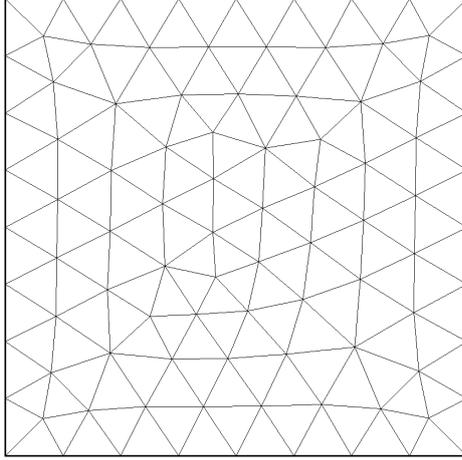


Figure 5: The typical mesh for the accuracy test for the 2D Burgers' equation and the 2D Euler equations.

with the following initial condition

$$u(t = 0, x, y) = \frac{1}{4} + \frac{1}{2} \sin(2\pi(x + y)), \quad (x, y) \in \Omega .$$

Here the domain Ω is the square $[0, 1] \times [0, 1]$. At $T = 0.1$ the exact solution is smooth. The structure of a typical mesh is shown in Fig. 5. The typical triangle edge length, denoted by h , is listed in all the Tables shown in this section. The errors presented are those of the cell averages of u . Also numerical solutions are limited by the point-wise HR. Table 1 shows the accuracy test results for the 4th order finite volume method solving the 2D Burgers' equations. Tables 2 and 3 show the accuracy test results for the 3rd order RKDG and RKDG with conservation constraint methods. Table 4 and 5 show the accuracy test results for the 4th order RKDG and RKDG with conservation constraint methods. We can see that the desired order of accuracy is retained for all the cases.

Table 1: Accuracy test results of the 4th order finite volume method solving the 2D Burgers' equation and the solution is limited by HR. CFL = 0.3.

h	L_1 error	order	L_∞ error	order
1/8	1.25E-2	-	6.93E-2	-
1/16	1.89E-3	2.73	2.55E-2	1.44
1/32	1.68E-4	3.49	2.96E-3	3.11
1/64	9.70E-6	4.11	2.62E-4	3.50
1/128	5.01E-7	4.28	2.44E-5	3.42
1/256	2.95E-8	4.09	1.46E-6	4.06

Table 2: Accuracy test results of the 3rd order RKDG method solving the 2D Burgers' equation and the solution is limited by HR. CFL = 0.2.

h	L_1 error	order	L_∞ error	order
1/8	1.34E-2	-	7.14E-2	-
1/16	2.21E-3	2.60	2.58E-2	1.47
1/32	3.15E-4	2.81	4.61E-3	2.48
1/64	3.40E-5	3.21	6.59E-4	2.81
1/128	3.67E-6	3.21	9.53E-5	2.80

Table 3: Accuracy test results of the 3rd order RKDG with conservation constraints method solving the 2D Burgers' equation and the solution is limited by HR. CFL = 0.2.

h	L_1 error	order	L_∞ error	order
1/8	1.35E-2	-	7.65E-2	-
1/16	2.31E-3	2.55	2.84E-2	1.43
1/32	3.16E-4	2.87	4.85E-3	2.55
1/64	3.49E-5	3.18	7.24E-4	2.74
1/128	3.75E-6	3.22	1.18E-4	2.62

Table 4: Accuracy test results of the 4th order RKDG method solving the 2D Burgers' equation and the solution is limited by HR. CFL = 0.1.

h	L_1 error	order	L_∞ error	order
1/8	7.98E-3	-	6.39E-2	-
1/16	1.20E-3	2.73	1.84E-2	1.80
1/32	7.57E-5	3.99	1.64E-3	3.49
1/64	3.89E-6	4.28	9.08E-5	4.17
1/128	2.15E-7	4.18	6.27E-6	3.86
1/256	1.36E-8	3.98	4.87E-7	3.69

3.2 Accuracy test for 2D Euler equations with smooth solution

A two-dimensional test problem [24] 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 in the spatial domain $[0, 1] \times [0, 1]$ from the time $T = 0$ to $T = 0.1$, see Fig. 5 for a typical mesh. The typical triangle edge length, denoted by h , is listed in all the Tables shown in this section. The errors presented are for the density. All numerical solutions are limited by the point-wise HR. Table 6 shows the accuracy test results for the 4th order finite volume method solving the 2D Euler equations. Tables 7 and 8 show the accuracy test results for the 3rd order RKDG and RKDG with conservation constraint methods solving the 2D Euler equations. Tables 9 and 10 show the accuracy test results for the 4th order RKDG and RKDG with conservation constraint methods solving the 2D Euler equations. We can the desired order of accuracy is retained for all test cases after applying the point-wise HR.

3.3 Shu-Osher problem

The Shu-Osher problem [26] is considered as a benchmark for the resolution of high order methods near discontinuities. This one-dimensional problem is extended to a two-

Table 5: Accuracy test results of the 4th order RKDG with conservation constraints method solving the 2D Burgers' equation and the solution is limited by HR. CFL = 0.1.

h	L_1 error	order	L_∞ error	order
1/8	8.06E-3	-	6.34E-2	-
1/16	1.25E-3	2.69	1.82E-2	1.80
1/32	8.45E-5	3.89	1.48E-3	3.62
1/64	4.05E-6	4.38	9.00E-5	4.04
1/128	2.22E-7	4.19	5.78E-6	3.96
1/256	1.31E-8	4.08	4.87E-7	3.57

Table 6: Accuracy test results of the 4th order finite volume method solving the 2D Euler equations and the solution is limited by HR. CFL = 0.3.

h	L_1 error	order	L_∞ error	order
1/4	8.38E-6	-	2.93E-5	-
1/8	7.38E-7	3.51	3.07E-6	3.25
1/16	4.32E-8	4.09	2.67E-7	3.52
1/32	2.24E-9	4.27	2.06E-8	3.70
1/64	1.41E-10	3.99	1.94E-9	3.41

Table 7: Accuracy test results of the 3rd order RKDG method solving the 2D Euler equations and the solution is limited by HR. CFL = 0.2.

h	L_1 error	order	L_∞ error	order
1/4	4.59E-5	-	1.26E-4	-
1/8	7.16E-6	2.68	1.95E-5	2.69
1/16	9.01E-7	2.99	2.61E-6	2.90
1/32	1.14E-7	2.98	4.37E-7	2.58
1/64	1.31E-8	3.12	5.39E-8	3.02

Table 8: Accuracy test results of the 3rd order RKDG with conservation constraints method solving the 2D Euler equations and the solution is limited by HR. CFL = 0.2.

h	L_1 error	order	L_∞ error	order
1/4	5.07E-5	-	1.68E-4	-
1/8	7.46E-6	2.76	2.51E-5	2.74
1/16	9.25E-7	3.01	3.54E-6	2.83
1/32	1.15E-7	3.01	4.92E-7	2.85
1/64	1.33E-8	3.11	5.96E-8	3.05

Table 9: Accuracy test results of the 4th order RKDG method solving the 2D Euler equations and the solution is limited by point-wise HR. CFL = 0.1.

h	L_1 error	order	L_∞ error	order
1/4	3.05E-6	-	9.97E-6	-
1/8	2.94E-7	3.37	8.97E-7	3.48
1/16	2.00E-8	3.88	7.19E-8	3.64
1/32	1.21E-9	4.05	4.63E-9	3.96
1/64	7.57E-11	4.00	4.07E-10	3.51

Table 10: Accuracy test results of the 4th order RKDG with conservation constraints method solving the 2D Euler equations and the solution is limited by HR. CFL = 0.1.

h	L_1 error	order	L_∞ error	order
1/4	3.44E-6	-	1.21E-5	-
1/8	2.57E-7	3.74	8.43E-7	3.84
1/16	1.73E-8	3.89	7.01E-8	3.59
1/32	1.04E-9	4.06	5.55E-9	3.66
1/64	6.57E-11	3.98	4.64E-10	3.58

dimensional triangular mesh with initial value

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if } x \leq -4 \\ (1 + 0.2 \sin(5x), 0, 1) & \text{if } x \geq -4. \end{cases} \quad (3.3)$$

The solutions of the Euler equations are computed in a rectangular domain of $[-5, 5] \times [0, 0.1]$ with a uniform triangulation of 301 vertices in the x -direction and 4 vertices in the y -direction. The initial value of the velocity component in the y -direction is zero. The density along a line parallel to x -axis is plotted in Figure 6 at $t = 1.8$, against a fine grid solution, which is treated as the “exact” solution. We can see that the all numerical schemes capture the solution profile of the Shu-Osher problem nicely; while the 4th order methods give the better resolution than that of the 3rd order methods as expected.

3.4 Shock tube problem

The Lax problem [12] is considered as a benchmark problem to assess the robustness and non-oscillatory property of the numerical methods. This one-dimensional problem is extended to a two-dimensional triangulated domain of $[-1, 1] \times [0, 0.2]$ with 101 vertices in the x -direction and 11 vertices in the y -direction. The solutions of the 2D Euler equations are computed. The initial data is

$$(\rho, u, p) = \begin{cases} (0.445, 0.698, 3.528), & \text{if } x \leq 0 \\ (0.5, 0, 0.571), & \text{if } x > 0. \end{cases} \quad (3.4)$$

The initial value of the velocity component in the y -direction is zero. The density at $t = 0.26$ is shown in Figure 7. We can see that the all numerical schemes capture the solution profile of the Lax problem with negligible overshoots/undershoots. A very sharp transition zone of contact discontinuities is captured by all schemes.

3.5 Flow past a forward facing step

This hydrodynamic flow problem is taken from [27]. The setup of the problem is the following: a right-going Mach 3 uniform flow enters a wind tunnel of 1 unit wide and 3 units long. The step is 0.2 units high and is located 0.6 units from the left side of the tunnel. The problem is initialized by a uniform, right-going Mach 3 flow, which has density 1.4, pressure 1.0, and velocity 3.0. The initial state of the gas is also used at the left side boundary. At the right side boundary, the out-flow boundary condition is applied there. Reflective boundary condition is applied along the walls of the tunnel.

The corner of the step is a singularity. Unlike in [27] and in other studies, we do not modify our scheme near the corner, which is known to lead to an erroneous entropy layer at the downstream bottom wall, as well as a spurious Mach stem at the bottom wall. Instead, we use the approach taken in [6], which is to locally refine the mesh near the corner, to decrease these artifacts. The edge length of the triangle away from the corner is roughly equal to $\frac{1}{160}$. Near the corner, the edge length of the triangle is roughly equal to $\frac{1}{320}$. Figures 8 and 9 show the contour plot of the numerical solutions computed by all the tested numerical methods. Comparing results in Figures 8 and 9, we can see that the resolution of the numerical solution

improves with the increase of the order of accuracy of the numerical schemes. Moreover, the solutions of the 4th order RKDG and RKDG with conservation constraints methods seems to capture more details in the vertex sheet than that of the 4th order finite volume method, which is expected.

4 Concluding Remarks

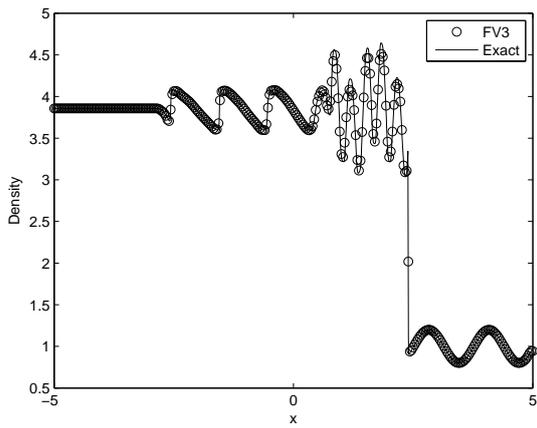
In this paper, we develop a new point-wise HR method for limiting solutions of the DG and finite volume methods up to fourth order for solving hyperbolic nonlinear conservation laws. The new HR utilizes a set of point values when evaluating polynomials and remainders on neighboring cells. We show that the new HR method keeps the approximation order of accuracy intact when applied to a polynomial. We numerically demonstrate that the new point-wise HR generates essentially non-oscillatory solutions for schemes up to the fourth order on triangular meshes with simplified implementation. When choosing the points in a neighboring cell, we first partition it into four identical partial cells and only uses the two which are closest to the cell under HR limiting. In each of these partial neighboring cells, we try to evenly distribute evaluation points so that the P-centroid is the same as the geometric centroid. The exceptions occur when re-calculating highest degree coefficients and first degree coefficients. In the former case we may want use a larger neighboring cell (i.e., the whole neighboring cell). In the latter case we tend to use a set of points in the neighboring cell which are closer to the cell under HR limiting, with the same consideration as in [29] that the remainder (it's now of highest possible degree and is computed for the cell under HR limiting) is sampled as close to the cell under HR limiting as possible. Finally, we note that HR with a stencil covering a large area seems to work better for the finite volume method, this could be related to the large wave length from the finite volume reconstruction than that of the DG.

References

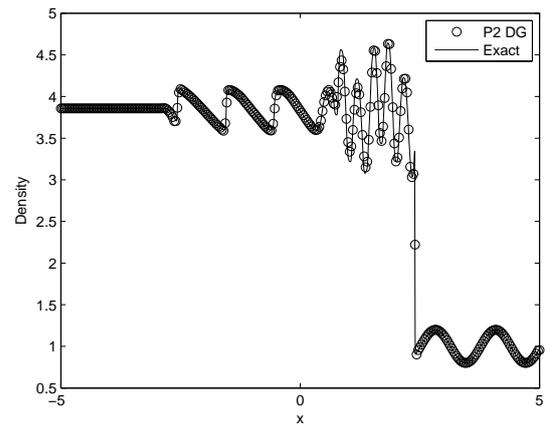
- [1] T. Barth and P. Frederickson, High order solution of the Euler equations on unstructured grids using quadratic reconstruction. *AIAA Paper*, No. 90-0013, 1990.
- [2] R. Biswas, K. Devine and J. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Appl. Numer. Math.*, 14:255–283, 1994.
- [3] A. Burbeau, P. Sagaut and C.H. Bruneau. A problem-independent limiter for high-order Runge-Kutta discontinuous Galerkin methods. *J. Comput. Phys.*, 169:111–150, 2001.
- [4] B. Cockburn, S.-Y. Lin and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one dimensional systems. *J. Comput. Phys.*, 52:411–435, 1989.
- [5] B. Cockburn and C.-W. Shu. TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. *Math. Comp.*, 52:411–435, 1989.

- [6] B. Cockburn and C.-W. Shu. The TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws V: multidimensional systems. *J. Comput. Phys.*, 141:199–224, 1998.
- [7] A. Harten, B. Engquist, S. Osher and S. Chakravarthy. Uniformly High Order Accurate Essentially Non-oscillatory Schemes, III. *J. Comput. Phys.*, 71:231–303, 1987.
- [8] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *J. Comp. Phys.*, 150:97–127, 1999.
- [9] G.-H. Hu, R. Li, and T. Tang, A robust high-order residual distribution type scheme for steady Euler equations on unstructured grids. *J. Comput. Phys.*, 229:1681-1697, 2010.
- [10] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.
- [11] L. Krivodonova. Limiters for high-order discontinuous Galerkin methods. *J. Comput. Phys.*, 226:879–896, 2007.
- [12] P. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computations. *Comm. Pure Appl. Math.*, 7:159, 1954.
- [13] B. van Leer. Toward the ultimate conservative difference scheme: II. Monotonicity and conservation combined in a second order scheme. *J. Comput. Phys.*, 14:361–370, 1974.
- [14] B. van Leer. Towards the ultimate conservative difference scheme: IV. A new approach to numerical convection. *J. Comput. Phys.*, 23:276–299, 1977.
- [15] B. van Leer. Towards the ultimate conservative difference scheme: V. A second order sequel to Godunov’s method. *J. Comput. Phys.*, 32:101–136, 1979.
- [16] X.-D. Liu, S. Osher and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115:200–212,1994.
- [17] Y.-J. Liu, C.-W. Shu, E. Tadmor and M.-P. Zhang. Central discontinuous Galerkin methods on overlapping cells with a non-oscillatory hierarchical reconstruction. *SIAM J. Numer. Anal.*, 45:2442-2467, 2007.
- [18] Y.-J. Liu, C.-W. Shu, E. Tadmor and M.-P. Zhang. Non-oscillatory hierarchical reconstruction for central and finite volume schemes. *Comm. Comput. Phys.*, 2:933–963, 2007.
- [19] J. Qiu and C.-W. Shu, On the construction, comparison, and local characteristic decomposition for high-order central WENO schemes, *J. Comput. Phys.*, 183:187–209, 2002.
- [20] J. Qiu and C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method. II. Two dimensional case.”, *Comput. Fluids*, 34:642–663, 2005.

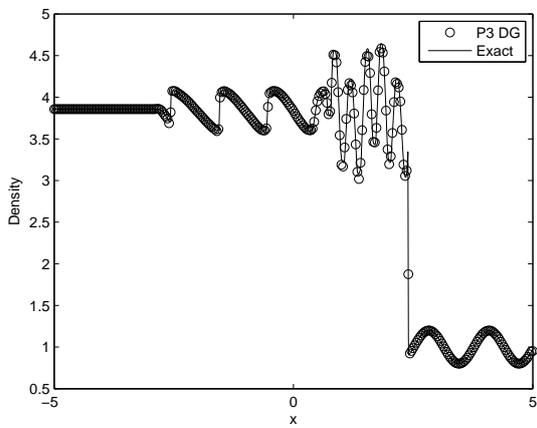
- [21] J. Qiu and C.-W. Shu, Runge-Kutta discontinuous Galerkin method using WENO limiters, *SIAM J. Sci. Comput.*, 26:907–929, 2005.
- [22] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Tech. report la-ur-73-479, Los Alamos Scientific Laboratory, 1973.
- [23] C.-W. Shu. TVB uniformly high-order schemes for conservation laws. *Math. Comp.*, 49:105–121, 1987.
- [24] C.-W. Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*, B. Cockburn, C. Johnson, C.-W. Shu and E. Tadmor (Editor: A. Quarteroni), *Lecture Notes in Mathematics*, Berlin. Springer. , 1697, 1998.
- [25] C.-W. Shu and S. Osher. Efficient Implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.
- [26] C.-W. Shu and S. Osher. Efficient Implementation of essentially non-oscillatory shock capturing schemes, II. *J. Comput. Phys.*, 83:32–78, 1989.
- [27] P. Woodward and P. Colella. Numerical simulation of two-dimensional fluid flows with strong shocks. *J. Comput. Phys.*, 54:115, 1984.
- [28] Z.-L. Xu and Y.-J. Liu. A Conservation Constrained Runge-Kutta Discontinuous Galerkin Method with the Improved CFL Condition for Conservation Laws. *Submitted to SIAM J. Sci. Comput.*, 2010.
- [29] Z.-L. Xu and Y.-J. Liu and C.-W. Shu. Hierarchical Reconstruction for Discontinuous Galerkin Methods on Unstructured Grids with a WENO Type Linear Reconstruction. *J. Comput. Phys.*, 228:2194-2212, 2009.
- [30] M. Yang and Z.J. Wang. A Parameter-Free Generalized Moment Limiter for High-Order Methods on Unstructured Grids. *AIAA-2009-605*.



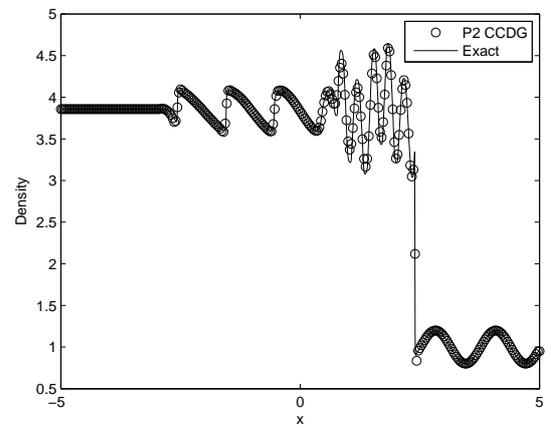
(a)



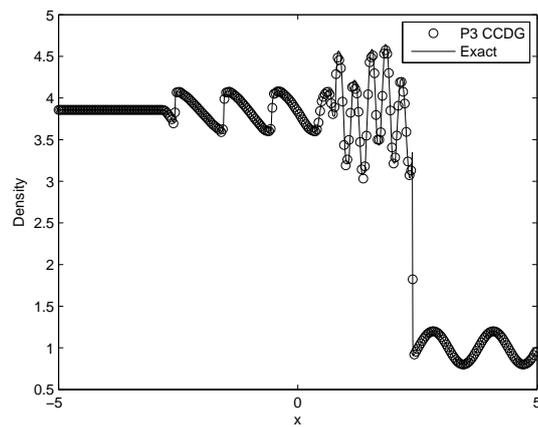
(b)



(c)

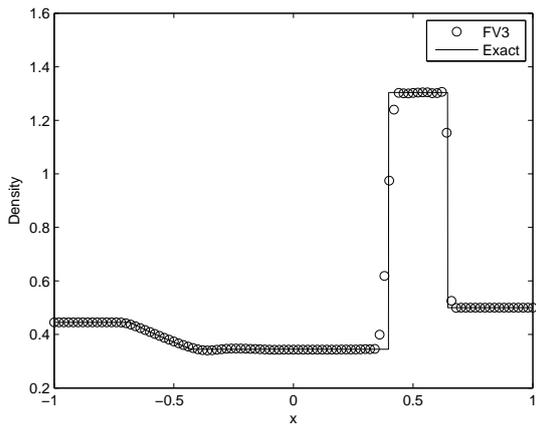


(d)

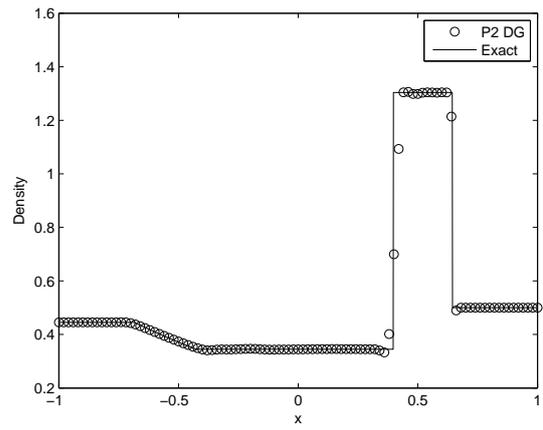


(e)

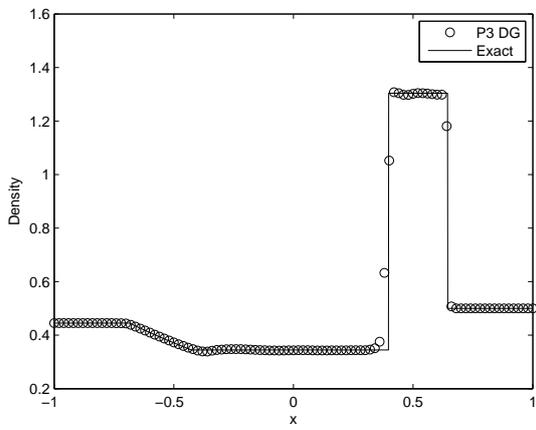
Figure 6: (a) Fourth order finite volume solution; (b) Third order DG solution; (c) Fourth order DG solution; (d) Third order DG with conservation constraints solution; (e) Fourth order DG with conservation constraints solution.



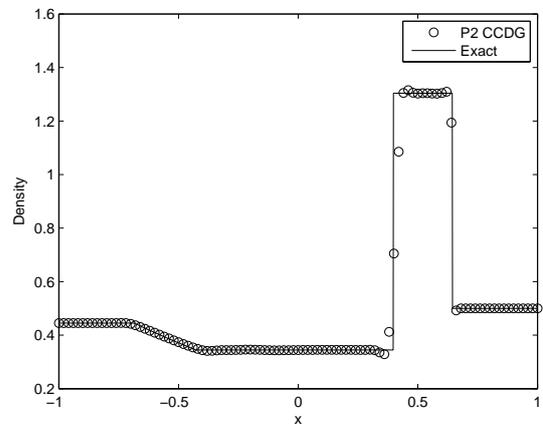
(a)



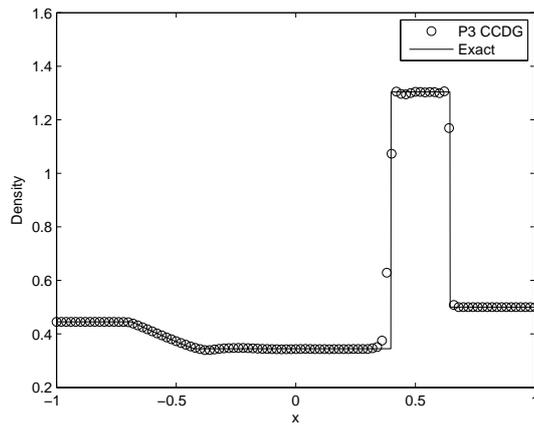
(b)



(c)

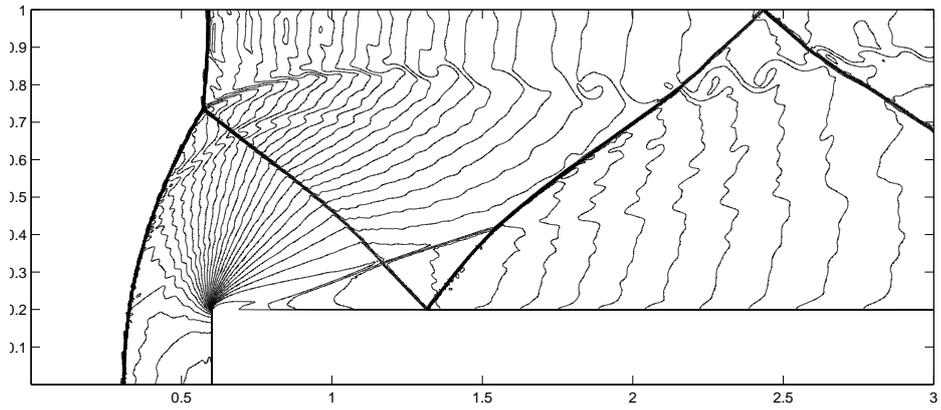


(d)

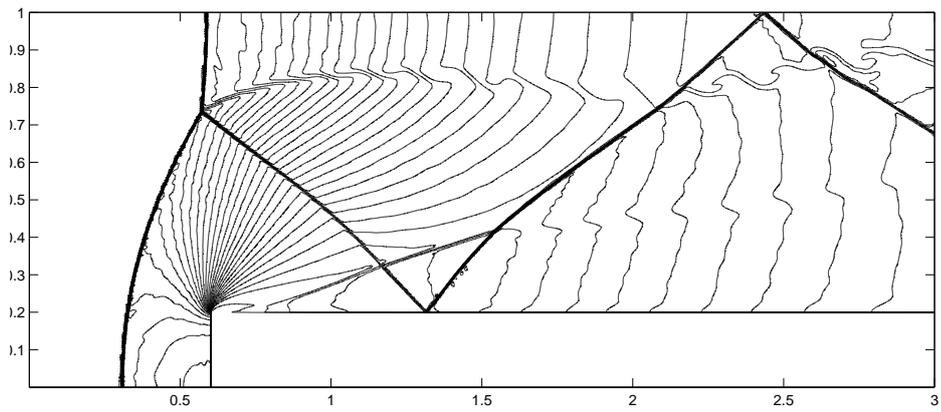


(e)

Figure 7: (a) Fourth order finite volume solution; (b) Third order DG solution; (c) Fourth order DG solution; (d) Third order DG with conservation constraints solution; (e) Fourth order DG with conservation constraints solution.

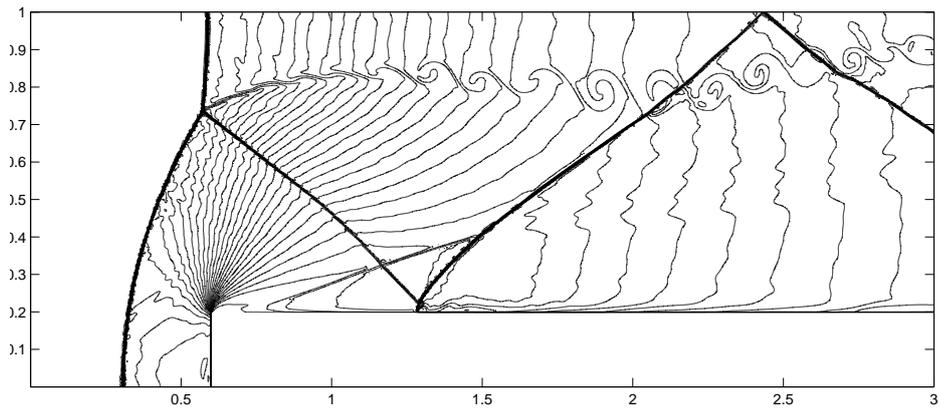


(a)

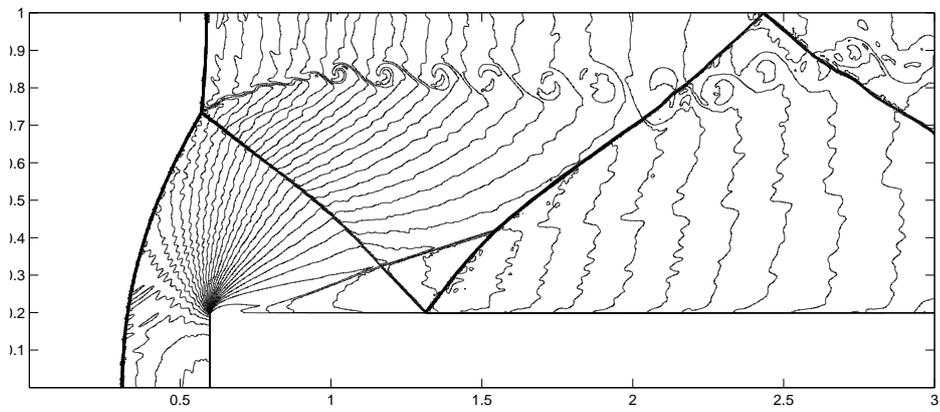


(b)

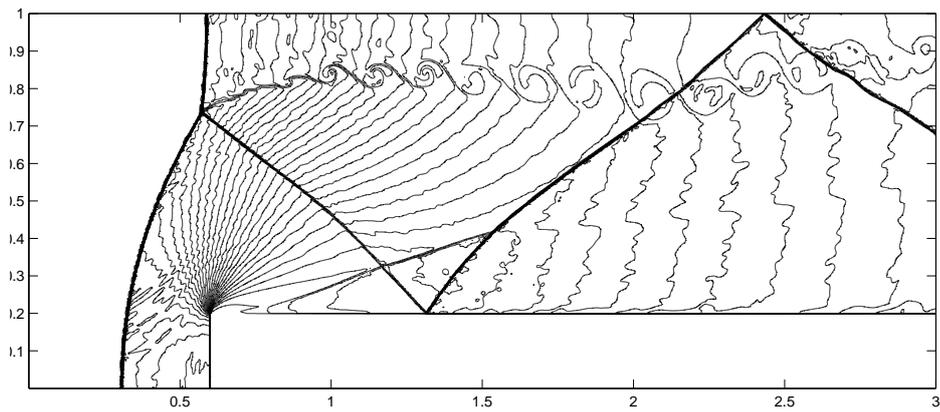
Figure 8: Forward step problem. Thirty equally spaced density contours from 0.32 to 6.15. (a) Third order DG solution; (b) Third order DG with conservation constraints solution.



(a)



(b)



(c)

Figure 9: Forward step problem. Thirty equally spaced density contours from 0.32 to 6.15. (a) fourth order finite volume solution; (b) Fourth order DG solution; (c) Fourth order DG with conservation constraints solution.