Hierarchical Reconstruction for Discontinuous Galerkin Methods on Unstructured Grids with a WENO Type Linear Reconstruction

Zhiliang Xu [‡], Yingjie Liu [§] and Chi-Wang Shu [¶]

January 10, 2008

Abstract

The hierarchical reconstruction [11] is applied to discontinuous Galerkin method on the two-dimensional unstructured grids. We explore a variety of limiter functions used in the construction of piecewise linear polynomials. We show that due to the abrupt shift of stencils, the use of center biased limiter functions is essential in order to recover the desired order of accuracy. Furthermore, we develop a WENO type linear reconstruction in each hierarchical level. Numerical computations for scalar and system of nonlinear hyperbolic equations are performed. We demonstrate that the hierarchical reconstruction can generate essentially non-oscillatory solutions while keeping the resolution and desired order of accuracy for smooth solutions.

1 Introduction

The discontinuous Galerkin (DG) method was firstly introduced by Reed and Hill [14] as a technique to solve neutron transport problems. The DG method is a finite element method using piecewise solution and test spaces (usually piecewise polynomials of certain degree). A major development of the DG method was carried out by Cockburn, Shu *et al.* in a series of papers [5, 4, 3, 2], in which they built a framework to solve nonlinear time dependent hyperbolic conservation laws (1.1)

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

[‡](E-mail: zxu2@nd.edu)
Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556.
[§](E-mail: yingjie@math.gatech.edu)
School of Mathematics, Georgia Institute of Technology, Atlanta, GA 30332. Research supported in part by NSF grant DMS-0511815.
[¶](E-mail: shu@dam.brown.edu)
Division of Applied Mathematics, Brown University, Providence RI 02912. Research supported in part by NSF grant DMS-0510345.

where $\Omega \subset R^d$, $\mathbf{x} = (x_1, ..., x_d)$, d is the dimension, $\mathbf{u} = (u_1, ..., u_m)^T$ and the flux vectors $\mathbf{F}_k(\mathbf{u}) = (F_{k,1}(\mathbf{u}), ..., F_{k,d}(\mathbf{u}))$. Since the hyperbolic conservation laws may develop discontinuous solutions, in their work, the explicit, total variation diminishing (TVD) Runge-Kutta time discretizations [19] are used in time. In space, exact or approximate Riemann solvers are used to compute the interface fluxes and the TVB projection limiters [16, 5] are used to prevent or restrict the spurious Gibbs-like oscillations near discontinuities.

It is an active research field to seek a robust limiting method both to prevent oscillations and to maintain the desired accuracy and resolution. In [5], the projection limiter limits the variation between a cell edge value and its cell average by the differences between the cell averages of the current and neighboring cells. The high order Legendre moments (order > 1) are truncated in a cell if the non-smoothness is detected. Biswas *et al.* [1] proposed a moment limiter method based on the orthogonal basis by detecting the non-smoothness from the higher order moments to the lower order ones. The limiting process is applied when necessary from higher to lower moments. Qiu and Shu [13] used a WENO reconstruction procedure as a limiter to "fix" the trouble cells, where the polynomials defined at the quadrature points are reconstructed from the WENO procedure and are projected to the finite element space to replace the ones computed by the DG method. In [11], a non-oscillatory hierarchical reconstruction (HR) method is introduced by Liu *et al.* for the staggered central DG method to reconstruct polynomials computed by the DG method, in which the cell averages of various orders of derivatives of a polynomial are calculated and used in the reconstruction of non-oscillatory linear polynomials on each hierarchical stage. The coefficients of the reconstructed linear polynomials are used to update the corresponding moments of the original polynomial.

In this paper, we extend the non-oscillatory hierarchical reconstruction method to the DG method on the unstructured meshes and develop several new techniques on triangular meshes. In particular, we introduce a weighted linear reconstruction for each hierarchical step in the spirit of the harmonic average of one-sided slope approximations [22, 23], modified ENO [17] and the WENO schemes [10, 7]. 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 DG solution procedure and the limiting procedure. Numerical tests are presented in Section 3. Concluding remarks and a plan for future work are included in Section 4.

2 Algorithm Formulation

We use the method of lines approach to evolve the solution on the triangulated domain. The DG method is used to compute the piecewise polynomial solution in each time level followed by the hierarchical reconstruction to remove the 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 elements $\Omega = \cup_{i=1}^{\mathcal{N}} \mathcal{K}_i$ and

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

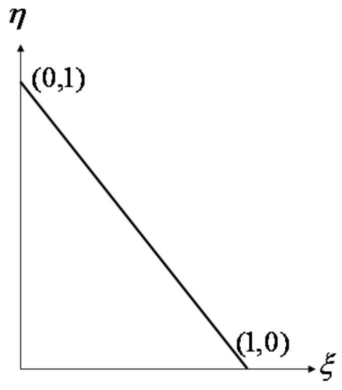


Figure 1: Reference triangular element $\mathcal{K}$

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

$$\begin{aligned} B &= \{b_m(\xi-\xi_0, \eta-\eta_0), m=1, ..., N_q\} \\ &= \{1, \xi-\xi_0, \eta-\eta_0, (\xi-\xi_0)^2, (\xi-\xi_0)(\eta-\eta_0), (\eta-\eta_0)^2, ..., (\eta-\eta_0)^q\} \ , \end{aligned} \tag{2.2}$$

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 basis functions in $\mathcal{K}$ as

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

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

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

which can be computed by

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

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

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

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

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

$$\begin{aligned} x &= (x_2-x_1)\xi+(x_3-x_1)\eta+x_1 \\ y &= (y_2-y_1)\xi+(y_3-y_1)\eta+y_1 \ , \end{aligned} \tag{2.6}$$

where (x_1,y_1), (x_2,y_2) and (x_3,y_3) are the coordinates of the vertices of element $\mathcal{K}_i$ for which the vertices are ordered counter clockwisely so that all double integrals are evaluated in the reference domain

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

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

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

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

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

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

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

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

by replacing u_h with Eq. (2.9). Since the approximated solution u_h is discontinuous between element interfaces, the interface 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. [18]) defined as

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

where α is the largest characteristic speed,

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

The domain and the boundary integrals in Eq. (2.10) are computed with $2q$ and $2q+1$ order accurate Gaussian quadrature rules respectively to preserve the $(q+1)^{th}$ order of

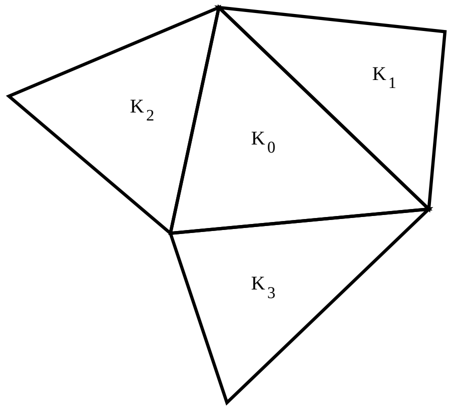


Figure 2: Schematic of 2D HR for cell $\mathcal{K}_0$

accuracy of the finite element space discretization. For a 3^{rd} order accurate scheme, the quadrature rule for the domain integral is

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

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

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

2.2 Time integration

Eq. (2.10) is integrated in time using the widely used TVD Runge-Kutta methods [19]. In particular, we use the three stage TVD Runge-Kutta method. 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 [11], which processes the DG solution at each Runge-Kutta stage to eliminate such spurious oscillations. We refer to [11, 12] for the summary of the HR steps and the implementations of HR for central and finite volume schemes.

Since we use 2^{nd} degree polynomials in our calculations, we describe the implementation of HR for piece-wise quadratic finite element space on the triangular elements and the new piece-wise linear polynomial reconstruction procedure in this section.

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

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

where (x_j, y_j) is the element centroid of $\mathcal{K}_j$. We will reconstruct a new polynomial in $\mathcal{K}_0$ with a point-wise error $\mathcal{O}(\triangle x^3)$.

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

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

Calculate the cell average of $L_j(x-x_j, y-y_j)$ on element $\mathcal{K}_j$ to obtain $\overline{L}_j = \partial_x u_j(0,0), j = 0,1,2,3$. We apply a non-oscillatory reconstruction procedure to the cell averages $\overline{L}_j$, which will be described at the end of this section, to obtain a new linear polynomial on element $\mathcal{K}_0$:

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

with $\partial_x\tilde{u}_0(0,0) = \overline{L}_0$. We then take the 1^{st} partial derivative with respect to y for $u_j(x-x_j, y-y_j), j = 0,1,2,3$ to redefine $L_j(x-x_j,y-y_j) = \partial_y u_j(0,0)+\partial_{xy}u_j(0,0)(x-x_j)+\partial_{yy}u_j(0,0)(y-y_j), j = 0,1,2,3$, and perform the same reconstruction procedure to obtain another polynomial on $\mathcal{K}_0$:

$$\tilde{L}_0(x-x_j,y-y_j) = \partial_y\tilde{u}_0(0,0)+\partial_{xy}\tilde{u}_0(0,0)(x-x_j)+\partial_{yy}\tilde{u}_0(0,0)(y-y_j) \ , \tag{2.16}$$

with $\partial_y\tilde{u}_{h,0}(0,0) = \overline{L}_0$, the cell average of L_0.

$\partial_{xx}\tilde{u}_0(0,0)$ and $\partial_{yy}\tilde{u}_0(0,0)$ will be the corresponding new coefficients of the reconstructed quadratic polynomial. $\partial_{xy}\tilde{u}_0(0,0)$ appears twice in the above procedures and is finalized by a limiter function which will be described later.

We then perform Step 2 of the algorithm. We compute the cell average of $u_{h,j}(x-x_j, y-y_j), j = 0,1,2,3$ to obtain $\overline{u}_{h,j}$ and compute the cell averages of the polynomial

$$\tilde{R}_0(x-x_0,y-y_0) = \frac{1}{2}\partial_{xx}\tilde{u}_0(0,0)(x-x_0)^2+\partial_{xy}\tilde{u}_0(0,0)(x-x_0)(y-y_0)+\frac{1}{2}\partial_{yy}\tilde{u}_0(0,0)(y-y_0)^2 \tag{2.17}$$

on elements $\{\mathcal{K}_0, \mathcal{K}_1, \mathcal{K}_2, \mathcal{K}_3\}$ to obtain $\overline{R}_0, \overline{R}_1, \overline{R}_2, \overline{R}_3$ respectively. Redefine $\overline{L}_j = \overline{u}_{h,j} - \overline{R}_j, j = 0,1,2,3$. The same reconstruction procedure is applied to the cell averages $\{\overline{L}_j\}$ to obtain new coefficients $\partial_x\tilde{u}_0(0,0)$ and $\partial_y\tilde{u}_0(0,0)$. Finally let the new coefficient $\tilde{u}_0(0,0) = \overline{L}_0$ to ensure conservation.

Now we describe the reconstruction procedure. In [12], three types of limiter functions are used in the reconstruction procedure, and are applied to the candidates of new coefficients of various orders from solving linear equations such as Eq. (2.21). The *minmod* limiter function defined by

$$m(c_1,c_2,...,c_r) = \begin{cases} \min\{c_1,c_2,...,c_r\}, & \text{if } c_1,c_2,...,c_r > 0 \ , \\ \max\{c_1,c_2,...,c_r\}, & \text{if } c_1,c_2,...,c_r < 0 \ , \\ 0, & \text{otherwise} \ , \end{cases} \tag{2.18}$$

gives a MUSCL reconstruction [22, 24]; the limiter function defined by

$$\mathrm{m}_2(c_1, c_2, ..., c_r) = c_j, \text{ if } \min\{|c_1|, |c_2|, ..., |c_r|\} \ , \tag{2.19}$$

gives the ENO [6] reconstruction; and the center biased *minmod* limiter m_b and ENO limiter m_{2b} can be formulated as

$$\begin{aligned} \mathrm{m_b}(c_1, c_2, ..., c_r) &= m\big((1+\varepsilon)m(c_1, c_2, ..., c_r), \tfrac{1}{r}\textstyle\sum_{i=1}^{r} c_i\big) \ , \\ \mathrm{m_{2b}}(c_1, c_2, ..., c_r) &= m_2\big((1+\varepsilon)m_2(c_1, c_2, ..., c_r), \tfrac{1}{r}\textstyle\sum_{i=1}^{r} c_i\big) \ , \end{aligned} \tag{2.20}$$

where $\varepsilon > 0$ is a small perturbation number. The reconstruction procedure using these limiter functions works very well on the rectangular and staggered grids [12]. However, our numerical experiments show that on the unstructured triangular mesh, this reconstruction procedure with *minmod* and ENO limiter functions fails to give the desired order of accuracy. The reason of failure stems from the abrupt shift of stencils which reduces the smoothness of the numerical flux [15, 17]. While the reconstruction procedure with the center biased *minmod* limiter function with a large value of ε gives the desired order of accuracy, the large value of ε introduces significant overshoots and undershoots. See Sec. 3 for the numerical results of test problems.

To remedy the abrupt shift of stencils, we introduce a new weighted combination of functions which follows the line of [22, 23, 17, 10, 7]. The new reconstruction procedure proceeds as follows.

Take the reconstruction of polynomial (2.15) as an example. In detail, we 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 solve the following equations for $\partial_{xx}\tilde{\tilde{u}}_{0,1}(0,0)$ and $\partial_{xy}\tilde{\tilde{u}}_{0,1}(0,0)$

$$\frac{1}{|\mathcal{K}_j|}\int_{\mathcal{K}_0} \tilde{L}_0(x-x_j, y-y_j)dxdy = \overline{L}_0 + \partial_{xx}\tilde{\tilde{u}}_{0,1}(0,0)(x-x_j) + \partial_{xy}\tilde{\tilde{u}}_{0,1}(0,0)(y-y_j) = \overline{L}_j \ , \tag{2.21}$$

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

Denote the linear polynomials computed from these three stencils to be $L_{0,1}(x - x_j, y - y_j)$, $L_{0,2}(x-x_j, y-y_j)$ and $L_{0,3}(x-x_j, y-y_j)$ respectively. The corresponding coefficients of these linear polynomials are $\partial_{xx}\tilde{\tilde{u}}_{0,1}, \partial_{xy}\tilde{\tilde{u}}_{0,1}$; $\partial_{xx}\tilde{\tilde{u}}_{0,2}, \partial_{xy}\tilde{\tilde{u}}_{0,2}$; and $\partial_{xx}\tilde{\tilde{u}}_{0,3}, \partial_{xy}\tilde{\tilde{u}}_{0,3}$ respectively.

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

$$\tilde{L}_0(x - x_j, y - y_j) = \sum_{r=1}^{3} w_r L_{0,r} \ . \tag{2.22}$$

The weight w_r depends on $L_{0,r}$ and satisfies

$$w_r \geq 0, \quad \sum_{r=1}^{3} w_r = 1 \tag{2.23}$$

for stability and consistency. Other considerations for designing the weights are that when a stencil contains a discontinuity of the solution, the corresponding weight will be essentially 0, and the weights are smooth functions of involved cell averages. The weights are set as follows:

$$w_r = \frac{\alpha_r}{\sum_{s=1}^{3}\alpha_s}, \quad r = 1, 2, 3, \tag{2.24}$$

where α_s are to be defined later. Let

$$d_r = \frac{1/\theta_r}{\sum_{s=1}^{3} 1/\theta_s} \ , \tag{2.25}$$

where θ_r is the condition number of the corresponding stencil, which is $||A||||A^{-1}||$, where A is the coefficient matrix of systems of Eq. (2.21), $||\cdot||$ denotes the $1-$norm. This choice of d_r puts the condition numbers of stencils into consideration and the candidates of new coefficients computed from a "bad" stencil have less weights. Let

$$\alpha_r = \frac{d_r}{1 + h\beta_r} \ , \tag{2.26}$$

where

$$\beta_r = (\partial_{xx}\tilde{\tilde{u}}_{0,r}(0,0))^2 + (\partial_{xy}\tilde{\tilde{u}}_{0,r}(0,0))^2 \ . \tag{2.27}$$

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

$$\partial_{xx}\tilde{u}_0 = \begin{cases} \sum_{r=1}^{3} w_r \partial_{xx}\tilde{\tilde{u}}_{0,r}, & \text{if } \overline{L}_{\min} < \overline{L}_0 < \overline{L}_{\max}, \\ 0, & \text{otherwise.} \end{cases} \tag{2.28}$$

Here $\overline{L}_{\min} = \min\{\overline{L}_j : j = 0, \cdots, 3\}$; $\overline{L}_{\max} = \max\{\overline{L}_j : j = 0, \cdots, 3\}$. $\overline{L}_{\min} < \overline{L}_0 < \overline{L}_{\max}$ is an extreme value detector, which is used to further reduce oscillations. The candidate coefficient $\partial_{xy}\tilde{u}_0$ is determined similarly.

The reconstruction of function (2.16) follows the above procedure. After the reconstruction of functions (2.15) and (2.16), $\partial_{xx}\tilde{u}_0$ and $\partial_{yy}\tilde{u}_0$ are the corresponding new coefficients respectively for the function $u_0(x - x_0, y - y_0)$ as in (2.13). However, the reconstruction of functions (2.15) and (2.16) leaves us two choices for the coefficient $\partial_{xy}u_0$, each of which comes from (2.15) and (2.16) respectively. We use the center biased ENO limiter function m_{2b} to determine the new coefficient $\partial_{xy}\tilde{u}_0$ from them, in which ε is set to be 0.01.

In Step 2 of HR, for the linear reconstruction involving $\partial_x u_0$ and $\partial_y u_0$, the following weight from [18] is used:

$$\alpha_r = \frac{d_r}{(\epsilon + \beta_r)^2} \ , \tag{2.29}$$

where β_r is the "smoothness indicator" of the r^{th} stencil similar to those in the WENO scheme,

$$\beta_r = (\partial_x\tilde{\tilde{u}}_{0,r}(0,0))^2 + (\partial_y\tilde{\tilde{u}}_{0,r}(0,0))^2 \ , \tag{2.30}$$

$\partial_x\tilde{\tilde{u}}_{0,r}(0,0)$ and $\partial_y\tilde{\tilde{u}}_{0,r}(0,0))$ are the first degree coefficients determined in stencil r by an equation similar to (2.21). ϵ is introduced to avoid the denominator to become 0. Note that in Step 2 it also can adopt the weight

$$\alpha_r = \frac{d_r}{1 + (\beta_r)^2} \tag{2.31}$$

which is similar to the form of Eq. (2.26). We found that weight (2.31) gave slightly bigger overshoot/undershoot.

A function similar to Eq. (2.28) is used to determine the new coefficients $\partial_x \tilde{u}_0(0,0)$ and $\partial_y \tilde{u}_0(0,0)$ for the function $u_0(x-x_0, y-y_0)$ as in (2.13). However, the extreme value detector (*i.e.*, the "0" case in (2.28)) is not applied here.

The reason that we choose different forms of weights is as the following: 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 effectively without the loss of accuracy. On the other hand, the high order coefficients ($\partial_{xx}u$, $\partial_{xy}u$, $\partial_{yy}u$ in the present paper) are less accurate and are more sensitive to the shift of stencils. We therefore want these high order coefficients to be closer to the mean values of the ones computed on different stencils where the solution is smooth, to reduce the abrupt shift of stencils.

Moreover, an error analysis shows that $w_r \partial_{xx} \tilde{\tilde{u}}_{0,r}$ is of $\mathcal{O}(h)$ where there is a discontinuity, which damps the oscillation to the required approximation error size of the second degree coefficients. In fact, Since $\partial_{xx}\tilde{\tilde{u}}$, $\partial_{xy}\tilde{\tilde{u}}$ and $\partial_{yy}\tilde{\tilde{u}}$ are of $\mathcal{O}(\frac{1}{h^2})$ at discontinuities, $\sqrt{\beta_r}$ is between $\mathcal{O}(1)$ if the solution is smooth, and $\mathcal{O}(\frac{1}{h^2})$ if there is a discontinuity. Therefore α_r is between $\mathcal{O}(h^3)$ and $\mathcal{O}(1)$ from Eq. (2.26). And $w_r \partial_{xx}\tilde{\tilde{u}}$ 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 region, Eq. (2.28) effectively damps out spurious oscillations. Our numerical experiments show that solutions to equations like Eq. (2.21) have the same sign most of the time as long as the DG solution is smooth, because the solutions to Eq. (2.21) are $\mathcal{O}(\triangle x)$ approximations to the gradient.

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

2.4 Local limiting procedure

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

Denote

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

where $\mathbf{x}$ is the middle point on an edge, $\bar{u}$ 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 of the adjacent element sharing the edge, and $\mu > 1$. We take $\mu = 1.2$ in our numerical runs. If the minmod function returns other than the first argument, this cell is identified as a "bad cell", and the computed DG solution is regarded to be oscillatory and marked for reconstructions. The limiting process is applied to these elements while keeping the computed DG solutions unchanged for other elements.

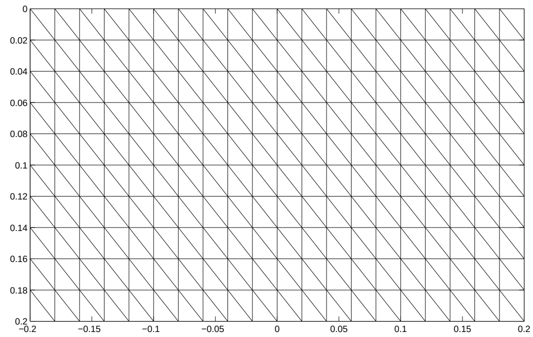


Figure 3: Mesh for accuracy test of scalar equations and for the Riemann problems.

3 Numerical Examples

We first study the limiter functions and test the capability of the method to achieve the desired 3^{rd} order 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 conservation form

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

where $\mathbf{u} = (\rho, \rho u, \rho v, E)$, $f(\mathbf{u}) = (\rho u, \rho u^2 + p, \rho uv, u(E + p))$, 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 Numerical Errors for Smooth Solutions

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

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

where the domain Ω is the square $[-1, 1] \times [-1, 1]$. At $T = 0.1$ the exact solution is smooth. For simplicity, the uniform triangular meshes are obtained by adding one diagonal line in each rectangle. The structure of the mesh is shown in Fig. 3. The errors presented are those of the cell averages of u.

ENO limiter function

The accuracy results are shown in Table 1 for reconstruction with the ENO limiter function. The accuracy results are shown in Tables 2, 3, and 4 for reconstruction with center biased ENO limiter function. This test problem shows that the ENO reconstruction of piecewise quadratic polynomial is 2^{nd} order accurate, since the abrupt shift of stencils reduces the desired order of accuracy of the method. As expected, this problem can be remedied by a center biased selection of stencils which is also confirmed by this numerical

Table 1: Accuracy for 2D Burgers equation with ENO limiter.

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

Table 2: Accuracy for 2D Burgers equation with biased ENO limiter (2.20),$\varepsilon = 0.1$.

h	L_1 error	order	L_∞ error	order
1/4	1.66E-01	-	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

Table 3: Accuracy for 2D Burgers equation with biased ENO limiter, $\varepsilon = 1$.

h	L_1 error	order	L_∞ error	order
1/4	1.22E-01	-	6.28E-2	-
1/8	2.72E-2	2.17	1.68E-2	1.90
1/16	6.69E-3	2.02	5.16E-3	1.70
1/32	1.81E-3	1.88	2.04E-3	1.34

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

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

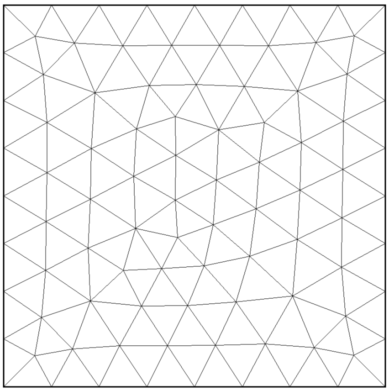


Figure 4: Mesh for accuracy test for the Euler equations.

test. We clearly see that the order of accuracy is improved with respect to increasing values of ε for the center biased ENO limiter.

We also test the center biased *minmod* limiter function. The test results show the same behavior as that of the center biased ENO limiter, and ε needs to take a slightly bigger value to achieve the 3^{rd} order accuracy. However, we found that both the center biased limiters are not stable for the shock wave problems with too large ε.

WENO type reconstruction

The accuracy results are shown in Table 5 for WENO type reconstruction. The 3^{rd} order accuracy is achieved.

Table 5: Accuracy for 2D Burgers equation with WENO type reconstruction.

h	L_1 error	order	L_∞ error	order
1/4	3.36E-2	-	4.31E-2	-
1/8	3.92E-3	3.10	9.51E-3	2.18
1/16	3.49E-4	3.39	1.88E-3	2.34
1/32	4.46E-5	2.99	5.44E-4	1.79

For the remaining test problems we will use the WENO type reconstruction.

3.2 Accuracy test for smooth inviscid compressible flow

A two-dimensional test problem [18] for the Euler equations is used, for ideal gas with $\gamma = 1.4$. The exact solution is given by $\rho = 1 + 0.5\sin(x + y - (u + v)t)$, $u = 1.0$, $v = -0.7$ and $p = 1$. The convergence test is conducted on irregular triangular meshes on the spatial domain $[0, 1] \times [0, 1]$ from the time $T = 0$ to $T = 0.1$, see Fig. 4 for a typical mesh. The triangle size is roughly equal to a rectangular element case of $\triangle x = \triangle y = h$, as indicated in Table 6. The accuracy results are shown in Table 6. The errors presented are those of the cell averages of density.

3.3 Riemann problems of Euler equations

The unsteady compressible inviscid flow problems are considered as test cases. The two dimensional triangular DG methods with HR are applied to one-dimensional shock tube problems. We consider the solution of the Euler equations in a rectangular domain of $[-1,1]\times[0,0.2]$ with a triangulation of 101 vertices in the x-direction and 11 vertices in the y-direction. The uniform triangular meshes are used, see Fig. 3. The periodic boundary condition is used in the y-direction, and the flow-through boundary conditions are used at the two ends of the boundaries in the x-direction. The initial value of the velocity component in the y-direction is zero. Figures 5(a) and 5(b) are obtained by interpolating the numerical solution data along the horizontal line $y = 0.1$ on 101 equally spaced points.

The first case is the Sod problem [21]. The initial data is

$$(\rho, u, p) = \begin{cases} (1, 0, 1), & \text{if } x \le 0 \\ (0.125, 0, 0.1), & \text{if } x > 0 \ . \end{cases} \tag{3.3}$$

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

The second case is the Lax problem [8]. The initial data is

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

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

3.4 Shu-Osher problem

The Shu-Osher problem [20] is tested. It is the Euler equations with an initial data

$$(\rho, u, p) = \begin{cases} (3.857143, 2.629369, 10.333333) & \text{if } x \le -4 \\ (1 + 0.2\sin(5x), 0, 1) & \text{if } x \ge -4 \ . \end{cases} \tag{3.5}$$

We consider the solution of the Euler equations in a rectangular domain of $[-5,5]\times[0,0.1]$ with a triangulation of 301 vertices in the x-direction and 4 vertices in the y-direction. The uniform triangular meshes are used. The initial value of the velocity component in the y-direction is zero. The density at $t = 1.8$ is shown in Fig. 6.

Table 6: Accuracy for 2D Euler equation with smooth sine evolution, on triangular meshes.

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

3.5 2D Riemann problem

A two-dimensional Riemann problem [9] for the Euler equations is computed. The computational domain is $[0,1]\times[0,1]$. The initial states are constants within each of the 4 quadrants. Counter-clock-wisely from the upper right quadrant, these states are labelled as (ρ_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. 7, with 30 equally spaced contours. The density profile along $x = 0.8$ is plotted in Fig. 8. The unstructured triangular mesh is used. The triangle edge length is roughly equal to 1/400.

3.6 2D shock vortex interactions

This test case is taken from [18] to investigate the ability of the scheme to resolve the vortex and the interaction. The computational domain is $[0,2]\times[0,2]$. A stationary Mach 1.1 shock is positioned at $x = 0.5$ and normal to the x-axis. Its left state is $(\rho, u, v, P) = (1, 1.1\sqrt{\gamma}, 0, 1)$. The vortex is described by a perturbation to the velocity (u, v), temperature $(T = \frac{P}{\rho})$ and entropy $(S = \ln\frac{P}{\rho^\gamma})$ of the mean flow and has the values:

$$\begin{aligned}
&\tilde{u} = \epsilon\tau e^{\alpha(1-r^2)}\sin\theta\ , \\
&\tilde{v} = -\epsilon\tau e^{\alpha(1-r^2)}\cos\theta\ , \\
&\tilde{T} = -\tfrac{(\gamma-1)\epsilon^2 e^{2\alpha(1-r^2)}}{4\alpha\gamma}\ , \\
&\tilde{S} = 0\ ,
\end{aligned} \tag{3.6}$$

where $\tau = \frac{r}{r_c}$ and $r = \sqrt{(x-x_c)^2 + (y-y_c)^2}$. The strength of the vortex ϵ is equal to 0.3. $r_c = 0.05$ and $\alpha = 0.204$. The density profile in $[0,2]\times[0,1]$ is plotted at $T = 0.35$ in Fig. 9, with 30 equally spaced contours. The unstructured triangular mesh is used. The triangle edge length is roughly equal to 1/100.

3.7 Double Mach reflection

The Double Mach reflection problem is taken from [25]. 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 shock is located at $x = \frac{1}{6}, y = 0$. The shock makes a 60^0 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 $\frac{1}{300}$ and $\frac{1}{500}$ respectively. 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. 10 is the contour plot with triangle edge length $\frac{1}{300}$. Fig. 11 is the contour plot with triangle edge length $\frac{1}{500}$. The "blown-up" portion around the double Mach region is shown in Fig. 12 to see that the fine details of the complicated flow structure under the triple Mach stem is captured.

Strong shocks of the double Mach problem introduce the negative pressure problem due to the undershoots. To fix this problem, we employ a scaling technique to remove the negative pressure. If at a quadrature point of an element, the negative pressure remains after reconstruction with the reconstructed polynomial u_h, We redefine the new polynomial u_h^* to be: $u_h^* = \bar{u}_h + 0.5(u_h - \bar{u}_h)$, where $\bar{u}_h$ is the cell average value of u_h. The negative pressure is removed after 1 or 2 iterations of the scaling normally.

4 Concluding Remarks

We have developed the HR reconstruction procedure and used it as a limiter for the discontinuous Galerkin method on the unstructured triangular meshes. The HR reconstruction with the WENO type reconstruction of piecewise linear polynomials maintains the desired order of accuracy and resolution, and effectively reduces spurious oscillations for discontinuous solutions.

In the future, we will experiment with more sophisticated troubled-cell indicators, and further study the WENO type reconstruction for piecewise polynomials of higher degree.

References

[1] R. Biswas, K. Devine and J. Flaherty. Parallel, adaptive finite element methods for conservation laws. *Appl. Numer. Math.*, 14:255–283, 1994.

[2] 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.

[3] B. Cockburn, S. Hou and C.-W. Shu. The TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. *Math. Comp.*, 54:545–581, 1990.

[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] 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.

[7] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228, 1996.

[8] P. Lax. Weak solutions of nonlinear hyperbolic equations and their numerical computations. *Comm. Pure Appl. Math.*, 7:159, 1954.

[9] P. Lax and X.-D. Liu. Solution of two dimensional Riemann problem of gas dynamics by positive schemes. *SIAM J. Sci. Comput.*, 19:319–340, 1998.

[10] X.-D. Liu, S. Osher and T. Chan. Weighted essentially non-oscillatory schemes. *J. Comput. Phys.*, 115:200–212,1994.

[11] 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.

[12] 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.

[13] J. Qiu and C.-W. Shu. Runge-Kutta discontinuous Galerkin method using WENO limiters. *SIAM J. Sci. Comput.*, 26:907–929, 2005.

[14] W. Reed and T. Hill. Triangular mesh methods for the neutron transport equation. Tech. report la-ur-73-479, Los Alamos Scientific Laboratory, 1973.

[15] A.M. Rogerson and E. Meiburg. A numerical study of the convergence properties of ENO schemes. *J. Sci. Comput.*, 5:151–167, 1990.

[16] C.-W. Shu. TVB uniformly high-order schemes for conservation laws. *Math. Comp.*, 49:105–121, 1987.

[17] C.-W. Shu. Numerical experiments on the accuracy of ENO and modified ENO schemes. *J. Sci. Comput.*, 5:127–149, 1990.

[18] 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.

[19] C.-W. Shu and S. Osher. Efficient Implementation of essentially non-oscillatory shock capturing schemes. *J. Comput. Phys.*, 77:439–471, 1988.

[20] C.-W. Shu and S. Osher. Efficient Implementation of essentially non-oscillatory shock capturing schemes, II. *J. Comput. Phys.*, 83:32–78, 1989.

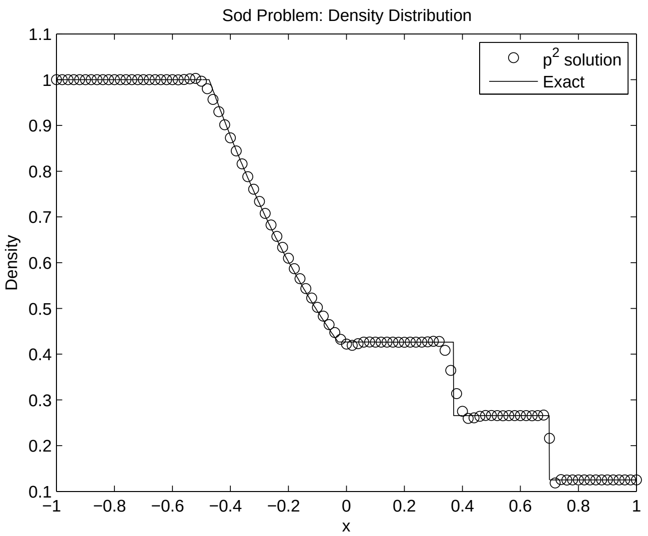
[21] G. Sod. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.*, 27:1, 1978.

[22] 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.

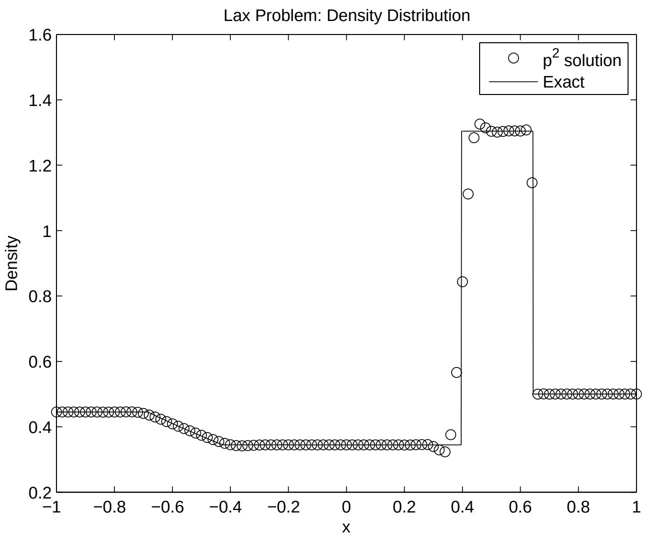
[23] B. van Leer. Towards the ultimate conservative difference scheme: IV. A new approach to numerical convection. *J. Comput. Phys.*, 23:276–299, 1977.

[24] 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.

[25] P. Woodward and P. Colella. Numerical simulation of two-dimensional fluid flows with strong shocks. *J. Comput. Phys.*, 54:115, 1984.

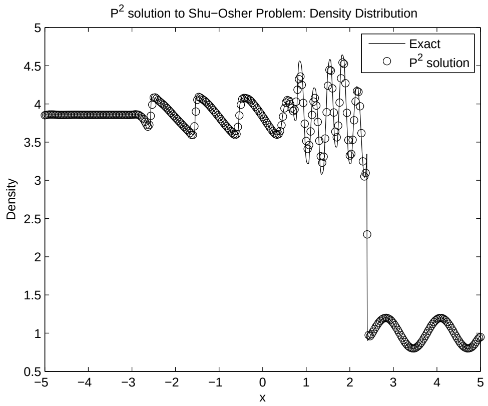


(a)

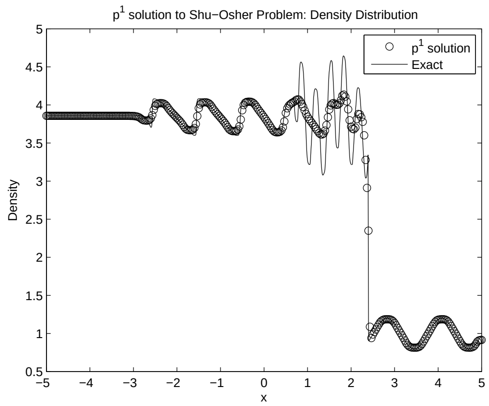


(b)

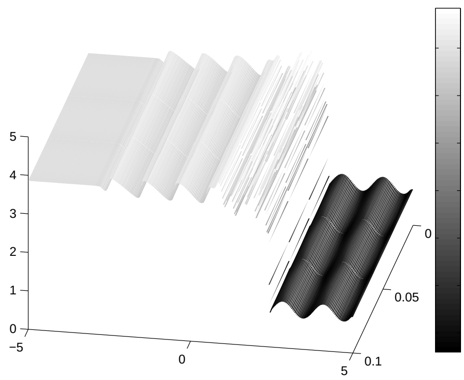
Figure 5: Shock tube problems: Density, (a) Sod problem (b) Lax problem



(a)



(b)



(c)

Figure 6: Shu-Osher problem: Density, (a) P^2 solution, (b) P^1 solution (c) 3D view of P^2 solution

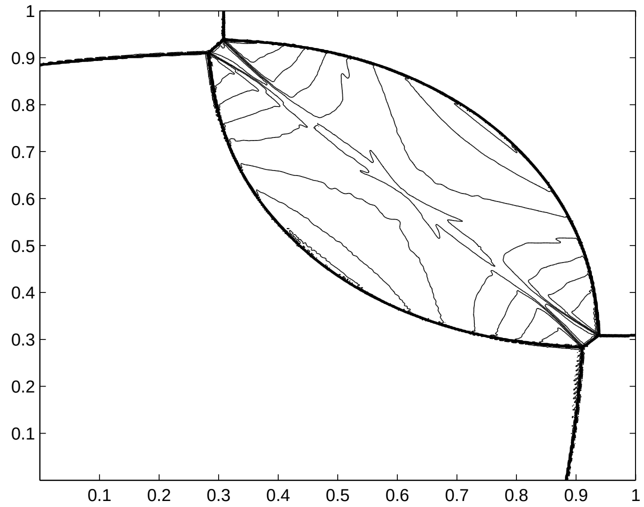


Figure 7: Density contour of a 2D Riemann problem at $t = 0.25$

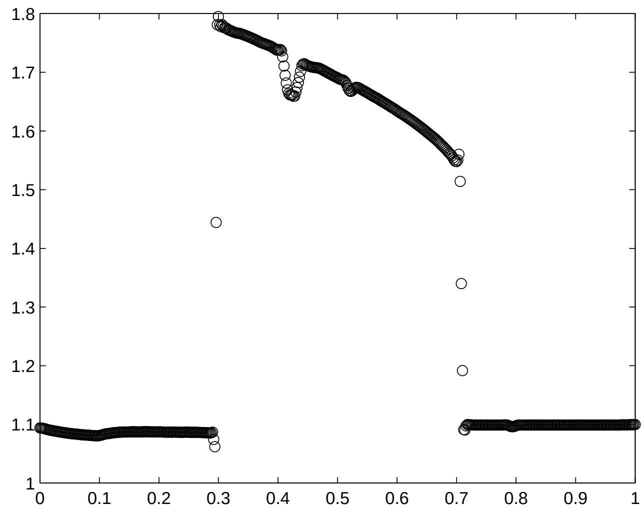
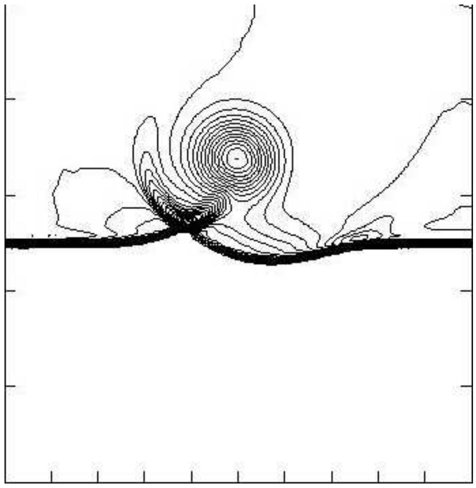


Figure 8: Density contour of a 2D Riemann problem along the line $x = 0.8$ at $t = 0.25$



(a)

Figure 9: 2D shock vortex interaction. Pressure contour at $t = 0.35$ with 30 equally spaced contour lines from 1.02 to 1.5.

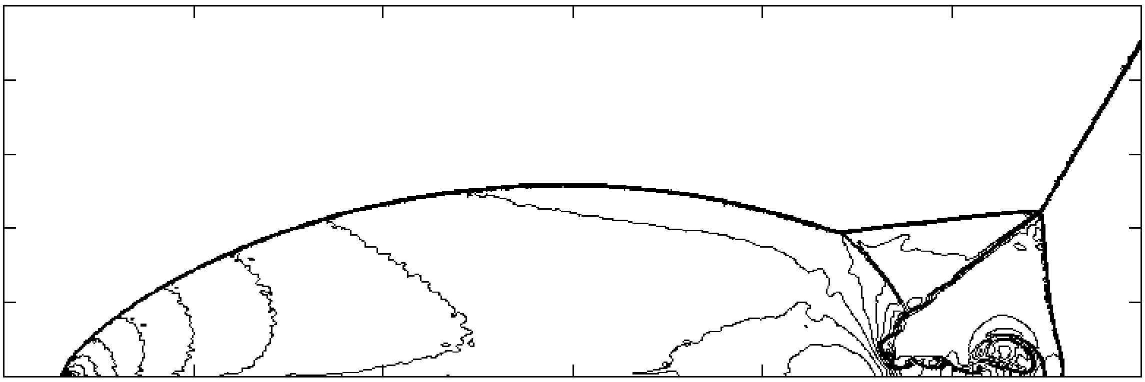


Figure 10: Double Mach reflection: density contour, $t = 0.2, h = \frac{1}{300}$

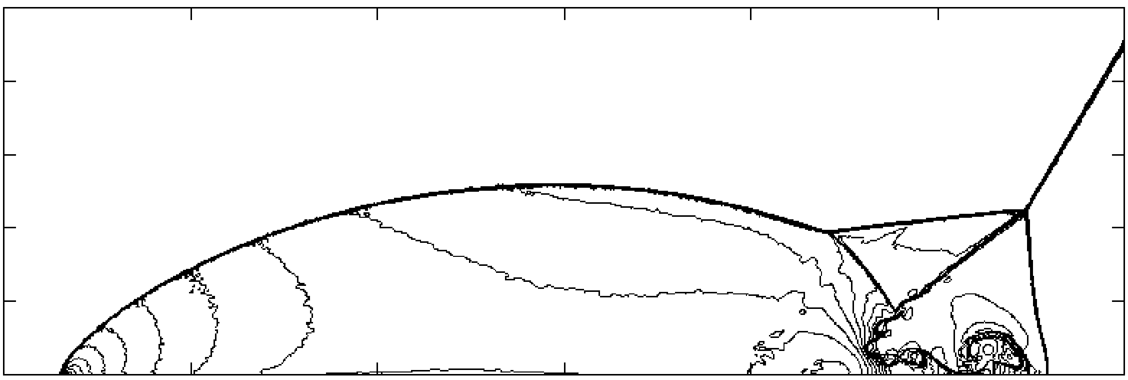
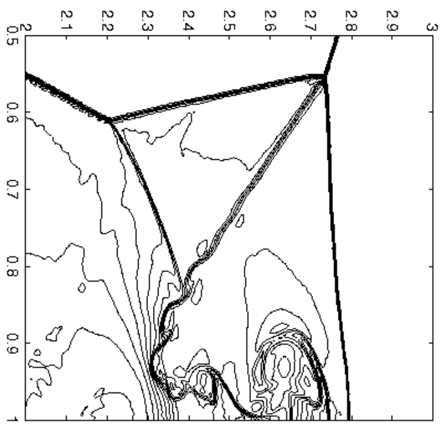
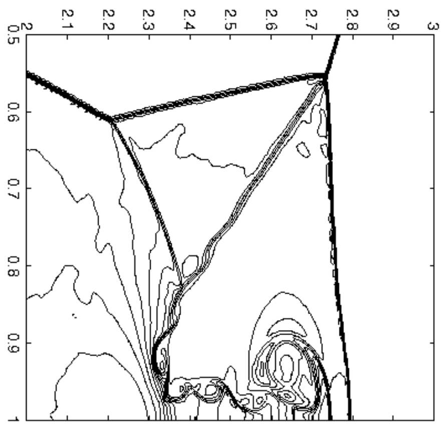


Figure 11: Double Mach reflection: density contour, $t = 0.2, h = \frac{1}{500}$



(a)



(b)

Figure 12: Double Mach reflection problem. Blown-up region around the double Mach stems. Density ρ. (a) Third-order P^2 with element edge length $\frac{1}{500}$; (b) Third-order P^2 with element edge length $\frac{1}{300}$.