

Hermite WENO schemes for Hamilton–Jacobi equations on unstructured meshes [☆]



Jun Zhu ^a, Jianxian Qiu ^{b,*}

^a College of Science, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu 210016, PR China

^b School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, PR China

ARTICLE INFO

Article history:

Received 1 November 2012

Received in revised form 16 July 2013

Accepted 23 July 2013

Available online 2 August 2013

Keywords:

HWENO scheme

WENO scheme

Hamilton–Jacobi equation

Unstructured mesh

High order accuracy

ABSTRACT

In this paper, we extend a class of the Hermite weighted essentially non-oscillatory (HWENO) schemes for solving the Hamilton–Jacobi equations by Qiu and Shu (2005) [24] to two dimensional unstructured meshes. The idea of the reconstruction in the HWENO schemes comes from the original WENO schemes, however both the function and its first two derivative values are evolved via time advancing and used in the reconstructions, while only the function values are evolved and used in the original WENO schemes which are nodal based approximations. The third and fourth order HWENO schemes using the combinations of second order approximations with nonlinear weights and TVD Runge–Kutta time discretization method are used here. Comparing with the original WENO schemes for Hamilton–Jacobi equations, one major advantage of HWENO schemes presented here is its compactness in the reconstructions. Extensive numerical tests are performed to illustrate the capability and high order accuracy of the methodologies.

© 2013 Elsevier Inc. All rights reserved.

1. Introduction

In this paper, we adopt the first order monotone Hamiltonian developed by Abgrall [2] in conjunction with the Hermite WENO (HWENO) reconstruction methodology on unstructured meshes. And then, it is used to solve the Hamilton–Jacobi (HJ) equations as:

$$\begin{cases} \phi_t + H(\nabla\phi) = 0, & (x, y, t) \in \Omega \times [0, \infty), \\ \phi(x, y, 0) = \phi_0(x, y), & (x, y) \in \Omega, \end{cases} \quad (1.1)$$

where $\nabla\phi = (\phi_x, \phi_y)^T$.

The Hamilton–Jacobi equations are often used in geometric optics, computer vision, material science, image processing and variational calculus [6,17,25]. Yet, the solutions to (1.1) are continuous but their derivatives can be discontinuous. And such solutions may not be unique unless using the physical implications and then getting the viscosity solutions [3]. It is well known that the HJ equations are closely related to conservation laws, hence successful numerical methods for conservation laws can be adapted for solving the HJ equations. Among successful numerical methods for conservation laws, we mention the essentially non-oscillatory (ENO) and weighted ENO (WENO) methods. The ENO methodology can be tracked back to the work proposed by Harten and Osher [8]. The key idea of ENO schemes is applying the most smooth stencil among all candidate stencils to approximate the solutions at cell boundaries to a high order of accuracy and avoid

[☆] The research was partially supported by NSFC grant 11002071, 10931004 and 91203110.

* Corresponding author.

E-mail addresses: zhujun@nuaa.edu.cn (J. Zhu), jxqiu@xmu.edu.cn (J. Qiu).

oscillations near discontinuities. In 1994, Liu et al. [18] proposed a WENO scheme that was constructed from the $(r + 1)$ -th order ENO schemes to obtain $(r + 2)$ -th order accuracy. Then in 1996, Jiang and Shu proposed the framework to construct finite difference WENO schemes from the $(r + 1)$ -th order (in L^1 norm sense) ENO schemes to get $(2r + 1)$ -th order accuracy, gave a new way of measuring the smoothness, and emulated the ideas of minimizing the total variation of the approximation in [12]. The first ENO and WENO papers on unstructured meshes were proposed by Abgrall in 1994 [1] and Friedrich in 1998 [7], respectively. The application of the WENO ideas and the whole reconstruction algorithm were also described in detail. And then, Shu and Hu gave two robust WENO schemes [9] which could maintain high order of accuracy by using a combination of lower algebraic polynomials in the finite volume formulation on unstructured meshes. The key idea of WENO schemes is applying all the stencils to approximate the variables at cell boundaries to an even higher order of accuracy in smooth regions and also can avoid oscillations near discontinuities. WENO type schemes can improve ENO type schemes in many aspects [27].

The framework of the ENO and WENO schemes is to evolve only one degree of freedom per cell, namely the cell average for the finite volume version or the point value at the center of the cell for the finite difference version. In [22,23,30,31], a class of WENO schemes based on Hermite polynomials, termed HWENO schemes, were presented for solving one and two dimensional nonlinear hyperbolic conservation law systems. The main difference between the Hermite WENO scheme designed in [22,23,31] and the traditional WENO schemes is that the former has a more compact stencil than the latter for the same order of accuracy. This compactness is achieved by evolving both the function and its first derivative values in time and they are both used in the reconstruction in HWENO schemes. Numerical examples in [22,23,30,31] demonstrate that HWENO schemes work well for solving hyperbolic conservation laws.

Based on the facts that the HJ equations are closely related to the conservation laws, many numerical methods for the conservation laws can be amended and used for solving the HJ equations. Along this line, Osher and Sethian [19] proposed a second order ENO scheme and Osher and Shu [20] presented high order ENO schemes to solve the Hamilton–Jacobi equations. Then, a high order WENO scheme was proposed by Jiang and Peng [11]. HWENO schemes for solving the Hamilton–Jacobi equations on structured meshes were presented by Qiu and Shu [21,24]. In 1996, Lafon and Osher [14] constructed the ENO schemes for solving the Hamilton–Jacobi equations on unstructured meshes. Zhang and Shu [29], Li and Chan [16] further developed high order WENO schemes for solving two dimensional Hamilton–Jacobi equations by using the nodal based weighted essentially non-oscillatory algebraic polynomial reconstructions on triangular meshes. And some finite element methods for arbitrary triangular meshes were developed in [4,5,10,15].

In this continuation paper, we would extend HWENO method to solve the Hamilton–Jacobi equations on unstructured meshes. We evolve both the viscosity solution ϕ at the barycenter of the triangular cell and the cell averages of its derivatives ϕ_x, ϕ_y over the cell. Both the point values of the solution and the cell averages of its derivatives are used to reconstruct the point values of the derivatives at the points of different inner sectors on the triangular cell and its boundaries. Comparing with the original WENO schemes of Jiang and Peng [11], Zhang and Shu [29] and Li and Chan [16], one major advantage of HWENO schemes is its compactness in the reconstructions, since both the solution and its derivatives are evolved in time.

The organization of this paper is as follows: In Section 2, we review and construct the third and fourth order HWENO schemes in details for solving Hamilton–Jacobi equations on unstructured meshes and present extensive numerical results in Section 3 to verify the accuracy and stability of these approaches. Concluding remarks are given in Section 4.

2. The construction of Hermite WENO schemes for the Hamilton–Jacobi equations

In this section, we firstly give the framework of solving the Hamilton–Jacobi equations briefly and then develop the procedures of the third and fourth order HWENO schemes on unstructured meshes for the Hamilton–Jacobi equations in details.

2.1. The framework

We take the governing equation (1.1), in which Ω has a triangulation consisting of cells $\Delta_i, i = 1, \dots, N$. We denote $(x_i, y_i), |\partial\Delta_{i\ell\ell}|, \ell\ell = 1, 2, 3$ and $|\Delta_i|$ to be the barycenter, the different edge length and the area of the triangular cell Δ_i , respectively. Let $u(x, y, t) = \phi_x(x, y, t)$ and $v(x, y, t) = \phi_y(x, y, t)$, and taking the x, y derivatives of (1.1), we can obtain the conservation laws:

$$\begin{cases} u_t + H(u, v)_x = 0, \\ u(x, y, 0) = \frac{\partial\phi_0(x, y)}{\partial x}, \end{cases} \tag{2.1}$$

and

$$\begin{cases} v_t + H(u, v)_y = 0, \\ v(x, y, 0) = \frac{\partial\phi_0(x, y)}{\partial y}. \end{cases} \tag{2.2}$$

We define $\phi_i = \phi(x_i, y_i, t)$ to be the numerical approximation to the viscosity solution of (1.1) at the barycenter of the target (triangular) cell Δ_i , the cell average of u as $\bar{u}_i(t) = \frac{1}{|\Delta_i|} \int_{\Delta_i} u(x, y, t) dx dy$ and the cell average of v as $\bar{v}_i(t) =$

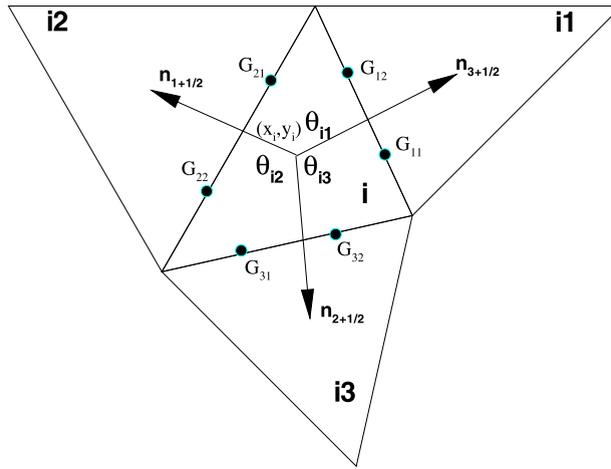


Fig. 2.1. Barycenter and boundary Gaussian points of Δ_i and its inner sectors.

$\frac{1}{|\Delta_i|} \int_{\Delta_i} v(x, y, t) dx dy$. Integrate (2.1) and (2.2) over the target cell Δ_i in conjunction with the semidiscrete scheme of (1.1), we obtain the modified formulations as:

$$\begin{cases} \frac{d}{dt} \phi_i(t) = -H(\nabla \phi_i), \\ \frac{d}{dt} \bar{u}_i(t) = -\frac{1}{|\Delta_i|} \int_{\partial \Delta_i} H(u(x, y, t), v(x, y, t)) n_x ds, \\ \frac{d}{dt} \bar{v}_i(t) = -\frac{1}{|\Delta_i|} \int_{\partial \Delta_i} H(u(x, y, t), v(x, y, t)) n_y ds, \end{cases} \quad (2.3)$$

where $n = (n_x, n_y)^T$ denotes the outward unit vector. The integrals in (2.3) can be discretized by a 2-point Gaussian integration formula on every edge (which has $G_1 = (\frac{1}{2} + \frac{\sqrt{3}}{6})T_1 + (\frac{1}{2} - \frac{\sqrt{3}}{6})T_2$, $G_2 = (\frac{1}{2} - \frac{\sqrt{3}}{6})T_1 + (\frac{1}{2} + \frac{\sqrt{3}}{6})T_2$ and quadrature coefficients $\sigma_{1,2} = \frac{1}{2}$ for the edge with end points T_1 and T_2), and we approximate (2.3) by the following schemes:

$$\begin{cases} \frac{d}{dt} \phi_i(t) \approx -\hat{H}_i, \\ \frac{d}{dt} \bar{u}_i(t) \approx -\frac{1}{|\Delta_i|} \sum_{\ell \ell=1}^3 |\partial \Delta_{i\ell\ell}| \sum_{\ell=1}^2 \sigma_\ell \tilde{H}(u(G_{\ell\ell\ell}, t), v(G_{\ell\ell\ell}, t)) \cdot n_{x\ell\ell}, \\ \frac{d}{dt} \bar{v}_i(t) \approx -\frac{1}{|\Delta_i|} \sum_{\ell \ell=1}^3 |\partial \Delta_{i\ell\ell}| \sum_{\ell=1}^2 \sigma_\ell \tilde{H}(u(G_{\ell\ell\ell}, t), v(G_{\ell\ell\ell}, t)) \cdot n_{y\ell\ell}, \end{cases} \quad (2.4)$$

where $G_{\ell\ell\ell}$, σ_ℓ , $\ell \ell = 1, 2, 3$, $\ell = 1, 2$ are the associated quadrature points and coefficients for different edges of the target cell Δ_i , respectively. \hat{H}_i is a global Lax-Friedrichs monotone Hamiltonian [2] (see Fig. 2.1) and defined as:

$$\hat{H}_i \approx \tilde{H} \left(\frac{\sum_{j=1}^3 \theta_{i_j} (\nabla \phi_i)_j}{2\pi} \right) - \frac{\alpha}{\pi} \sum_{j=1}^3 \beta_{j+1/2} \left(\frac{(\nabla \phi_i)_j + (\nabla \phi_i)_{j+1}}{2} \right) \cdot n_{j+1/2}, \quad (2.5)$$

where θ_{i_j} , $j = 1, 2, 3$ are the counterclockwise inner angle of sectors and $n_{j+1/2}$, $j = 1, 2, 3$ are the outward unit normal vectors. $\beta_{j+1/2} = \tan(\frac{\theta_{i_j}}{2}) + \tan(\frac{\theta_{i_{j+1}}}{2})$ and if $j = 3$ we change the lower index $j + 1$ to be 1. And we set $\alpha = \max\{\alpha_x, \alpha_y\} = \max\{\max|H_1|, \max|H_2|\}$, here H_1 and H_2 are the partial derivatives of H with respect to ϕ_x and ϕ_y , otherwise are the Lipschitz constants of H globally (if not differentiable). And $\tilde{H}(u(G_{\ell\ell\ell}, t), v(G_{\ell\ell\ell}, t)) \cdot n_{x\ell\ell}$, $\tilde{H}(u(G_{\ell\ell\ell}, t), v(G_{\ell\ell\ell}, t)) \cdot n_{y\ell\ell}$ are replaced by the numerical fluxes such as the Lax-Friedrichs fluxes:

$$\begin{aligned} \tilde{H}(u(G_{\ell\ell\ell}, t), v(G_{\ell\ell\ell}, t)) \cdot n_{x\ell\ell} \approx & \frac{1}{2} \{ (H(u^-(G_{\ell\ell\ell}, t), v^-(G_{\ell\ell\ell}, t)) + H(u^+(G_{\ell\ell\ell}, t), v^+(G_{\ell\ell\ell}, t))) \\ & \cdot n_{x\ell\ell} - \alpha_x (u^+(G_{\ell\ell\ell}, t) - u^-(G_{\ell\ell\ell}, t)) \}, \quad \ell \ell = 1, 2, 3, \ell = 1, 2, \end{aligned} \quad (2.6)$$

and

$$\begin{aligned} \tilde{H}(u(G_{\ell\ell\ell}, t), v(G_{\ell\ell\ell}, t)) \cdot n_{y\ell\ell} \approx & \frac{1}{2} \{ (H(u^-(G_{\ell\ell\ell}, t), v^-(G_{\ell\ell\ell}, t)) + H(u^+(G_{\ell\ell\ell}, t), v^+(G_{\ell\ell\ell}, t))) \\ & \cdot n_{y\ell\ell} - \alpha_y (v^+(G_{\ell\ell\ell}, t) - v^-(G_{\ell\ell\ell}, t)) \}, \quad \ell \ell = 1, 2, 3, \ell = 1, 2. \end{aligned} \quad (2.7)$$

We rewrite the ODEs (2.4) as the form:

$$U_t = L(U). \quad (2.8)$$

Then we use third order version TVD Runge–Kutta time discrete method [28]:

$$\begin{cases} U^{(1)} = U^n + \Delta t L(U^n), \\ U^{(2)} = \frac{3}{4}U^n + \frac{1}{4}U^{(1)} + \frac{1}{4}\Delta t L(U^{(1)}), \\ U^{n+1} = \frac{1}{3}U^n + \frac{2}{3}U^{(2)} + \frac{2}{3}\Delta t L(U^{(2)}), \end{cases} \quad (2.9)$$

to obtain fully discrete scheme both in space and time. And we would like to omit variable t in the following if not cause confusion.

We now give the crucial procedures of the HWENO schemes, from the point value ϕ_i and the different cell averages \bar{u}_i , \bar{v}_i to obtain the point values $\phi_{x_*}^\pm = u_*^\pm$, $\phi_{y_*}^\pm = v_*^\pm$ at the barycenter (the superscript “-” is used to indicate the point value lies inside of the inner sector and “+” to indicate the point value lies outside) and the boundary of the target cell Δ_i (“-” is for the point value lies inside of the boundary and “+” for the point value lies outside), respectively. These reconstructions should be both high order accurate and essentially non-oscillatory. We come up with the procedures of the reconstructions for both the third and fourth order accurate cases in the following.

2.2. The construction of Hermite WENO3 scheme

1. The reconstruction of functions $u_{i_j}^-$ and $v_{i_j}^-$ at the barycenter (x_i, y_i) in the inner sector whose inner angle is θ_{i_j} , $j = 1, 2, 3$ of the target cell Δ_i (see from Fig. 2.1 to Fig. 2.4).

Step 2.1.1. We select three different big stencils as $S = \{\Delta_i, \Delta_{i1}, \Delta_{i2}, \Delta_{i3}, \Delta_{ia}, \Delta_{ib}, \Delta_{ja}, \Delta_{jb}\}$ (for $u_{i_1}^-$ and $v_{i_1}^-$), $S = \{\Delta_i, \Delta_{i1}, \Delta_{i2}, \Delta_{i3}, \Delta_{ja}, \Delta_{jb}, \Delta_{ka}, \Delta_{kb}\}$ (for $u_{i_2}^-$ and $v_{i_2}^-$) and $S = \{\Delta_i, \Delta_{i1}, \Delta_{i2}, \Delta_{i3}, \Delta_{ia}, \Delta_{ib}, \Delta_{ka}, \Delta_{kb}\}$ (for $u_{i_3}^-$ and $v_{i_3}^-$), respectively. Then we construct different polynomials $P(x, y) \in span\{1, \frac{(x-x_i)}{\sqrt{|\Delta_i|}}, \frac{(y-y_i)}{\sqrt{|\Delta_i|}}, \frac{(x-x_i)^2}{|\Delta_i|}, \frac{(x-x_i)(y-y_i)}{|\Delta_i|}, \frac{(y-y_i)^2}{|\Delta_i|}, \frac{(x-x_i)^3}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)^2(y-y_i)}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)(y-y_i)^2}{\sqrt{|\Delta_i|^3}}, \frac{(y-y_i)^3}{\sqrt{|\Delta_i|^3}}\}$ by requiring that they match the point values of ϕ and cell averages of u, v on the target cell Δ_i and other cells in a least square sense [9,29].

Step 2.1.2. We then construct polynomials $q_l(x, y) \in span\{1, \frac{(x-x_i)}{\sqrt{|\Delta_i|}}, \frac{(y-y_i)}{\sqrt{|\Delta_i|}}, \frac{(x-x_i)^2}{|\Delta_i|}, \frac{(x-x_i)(y-y_i)}{|\Delta_i|}, \frac{(y-y_i)^2}{|\Delta_i|}\}$, $l = 1, \dots, \chi$:

$$q_l(x_\ell, y_\ell) = \phi(x_\ell, y_\ell), \quad (2.10)$$

$$\frac{1}{|\Delta_{\ell_x}|} \int_{\Delta_{\ell_x}} \frac{\partial q_l(x, y)}{\partial x} dx dy = \bar{u}_{\ell_x}, \quad (2.11)$$

and

$$\frac{1}{|\Delta_{\ell_y}|} \int_{\Delta_{\ell_y}} \frac{\partial q_l(x, y)}{\partial y} dx dy = \bar{v}_{\ell_y}. \quad (2.12)$$

Step 2.1.3. We find the two sets of combination coefficients, also called linear weights, denoted by $\gamma_1^x, \dots, \gamma_\chi^x$ and $\gamma_1^y, \dots, \gamma_\chi^y$, satisfying:

$$\frac{\partial}{\partial x} P(x_i, y_i) = \sum_{l=1}^\chi \gamma_l^x \frac{\partial}{\partial x} q_l(x_i, y_i), \quad (2.13)$$

and

$$\frac{\partial}{\partial y} P(x_i, y_i) = \sum_{l=1}^\chi \gamma_l^y \frac{\partial}{\partial y} q_l(x_i, y_i). \quad (2.14)$$

Then we can get the approximations of $u_{i_j}^-$ and $v_{i_j}^-$ at the barycenter (x_i, y_i) in different sectors. We know that (2.13) and (2.14) hold for any polynomials of degree at most 2, if $\sum_{l=1}^\chi \gamma_l^{x,y} = 1$. This is because every $q_l(x, y)$ reconstructs quadric polynomial exactly. We then ask for:

$$\min \left(\sum_{l=1}^\chi \gamma_l^{x,y} \right)^2. \quad (2.15)$$

By doing so, we can get the linear weights uniquely but cannot maintain them positively all the time, we can use the methods that presented in [26] to overcome this drawback.

Step 2.1.4. We compute the smoothness indicators, denote by β_l , $l = 1, \dots, \chi$, which measure how smooth the functions $q_l(x, y)$, $l = 1, \dots, \chi$ are in the target cell Δ_i . The smaller these smoothness indicators, the smoother the functions are in the target cell Δ_i . We use the similar recipe for the smoothness indicators as in [9,12]:

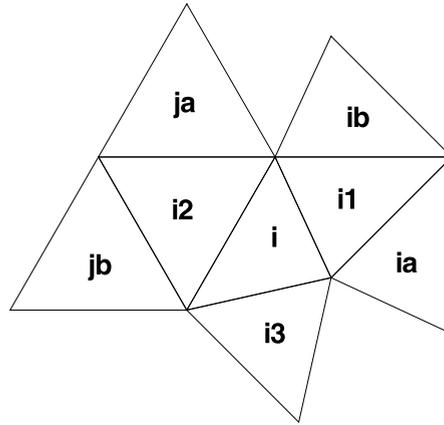


Fig. 2.2. The big partial stencil I.

$$\beta_l^x = \sum_{|\ell| \geq 1} |\Delta_i|^{|\ell|-1} \int_{\Delta_i} \left(\frac{\partial^{|\ell|}}{\partial x^{\ell_1} \partial y^{\ell_2}} \frac{\partial}{\partial x} q_l(x, y) \right)^2 dx dy, \quad l = 1, \dots, \chi, \tag{2.16}$$

and

$$\beta_l^y = \sum_{|\ell| \geq 1} |\Delta_i|^{|\ell|-1} \int_{\Delta_i} \left(\frac{\partial^{|\ell|}}{\partial x^{\ell_1} \partial y^{\ell_2}} \frac{\partial}{\partial y} q_l(x, y) \right)^2 dx dy, \quad l = 1, \dots, \chi, \tag{2.17}$$

where $\ell = (\ell_1, \ell_2)$.

Step 2.1.5. We compute the nonlinear weights based on the smoothness indicators:

$$\omega_l^{x,y} = \frac{\bar{\omega}_l^{x,y}}{\sum_{\ell=1}^{\chi} \bar{\omega}_\ell^{x,y}}, \quad \bar{\omega}_\ell^{x,y} = \frac{\gamma_\ell^{x,y}}{(\varepsilon + \beta_\ell^{x,y})^2}, \quad l = 1, \dots, \chi. \tag{2.18}$$

Here ε is a small positive number to avoid the denominator to become zero. We take $\varepsilon = 10^{-5}$ in our computation.

Step 2.1.6. The final approximations are then given by:

$$u^-(x_{i_j}, y_{i_j}) \approx \sum_{l=1}^{\chi} \omega_l^x \frac{\partial}{\partial x} q_l(x_i, y_i), \quad j = 1, 2, 3, \tag{2.19}$$

and

$$v^-(x_{i_j}, y_{i_j}) \approx \sum_{l=1}^{\chi} \omega_l^y \frac{\partial}{\partial y} q_l(x_i, y_i), \quad j = 1, 2, 3. \tag{2.20}$$

In this paper, we provide one type of the smaller stencils as follows:

For the reconstruction of functions $u_{i_1}^-$ and $v_{i_1}^-$ at the barycenter (x_i, y_i) (see Fig. 2.1 and Fig. 2.2):

- $l = 1, \ell = i, i1, i2, ia, ib, ja;$ $l = 2, \ell = i, i1, i2, ib, ja, jb;$
- $l = 3, \ell = i, i1, i2, i3, ia, jb;$ $l = 4, \ell = i, i1, i2, i3, ib, ja;$
- $l = 5, \ell = i, i1, i2, ja, jb, \ell_x = i1;$ $l = 6, \ell = i, i1, i2, ia, ib, \ell_x = i2;$
- $l = 7, \ell = i, i1, i2, ja, jb, \ell_y = i1;$ $l = 8, \ell = i, i1, i2, ia, ib, \ell_y = i2;$
- $l = 9, \ell = i, i1, ia, ib, \ell_x = i1, \ell_y = i1;$ $l = 10, \ell = i, i2, ja, jb, \ell_x = i2, \ell_y = i2;$
- $l = 11, \ell = i, i1, i2, i3, \ell_x = i1, \ell_y = i1;$ $l = 12, \ell = i, i1, i2, i3, \ell_x = i2, \ell_y = i2.$

Remarks. (1) The target cell Δ_i should retain downwind on the big partial stencil I (see Fig. 2.2) for the barycenter point value reconstruction in the inner sector whose inner angle is θ_{i_1} (see Fig. 2.1). (2) The small stencils should harbor the target cell Δ_i and their neighboring cells mainly in the fan-shaped region of the inner sector. (3) When the cell Δ_{i3} (which is purely located outside of the inner sector) is used in some small stencils, the cells $(\Delta_{i1}$ and $\Delta_{i2})$ neighboring the target cell Δ_i inside of the inner sector should be used simultaneously to combine associated small stencils.

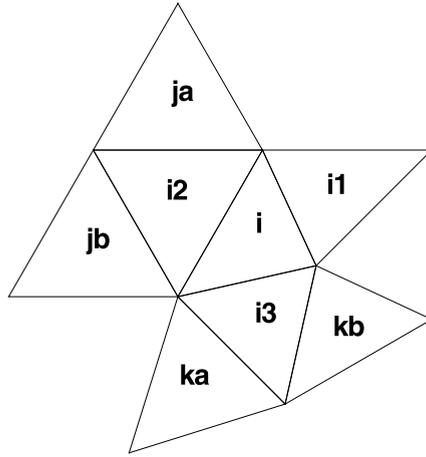


Fig. 2.3. The big partial stencil II.

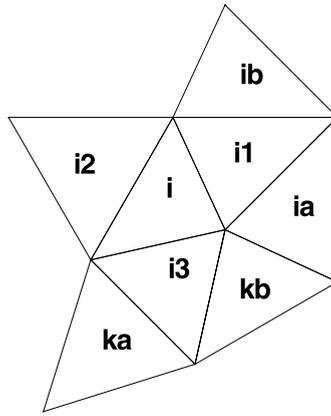


Fig. 2.4. The big partial stencil III.

For the reconstruction of functions $u_{i_2}^-$ and $v_{i_2}^-$ at the barycenter (x_i, y_i) (see Fig. 2.1 and Fig. 2.3):

- $l = 1, \ell = i, i2, i3, ja, jb, ka;$ $l = 2, \ell = i, i2, i3, jb, ka, kb;$
- $l = 3, \ell = i, i1, i2, i3, ja, kb;$ $l = 4, \ell = i, i1, i2, i3, jb, ka;$
- $l = 5, \ell = i, i2, i3, ka, kb, \ell_x = i2;$ $l = 6, \ell = i, i2, i3, ja, jb, \ell_x = i3;$
- $l = 7, \ell = i, i2, i3, ka, kb, \ell_y = i2;$ $l = 8, \ell = i, i2, i3, ja, jb, \ell_y = i3;$
- $l = 9, \ell = i, i2, ja, jb, \ell_x = i2, \ell_y = i2;$ $l = 10, \ell = i, i3, ka, kb, \ell_x = i3, \ell_y = i3;$
- $l = 11, \ell = i, i1, i2, i3, \ell_x = i2, \ell_y = i2;$ $l = 12, \ell = i, i1, i2, i3, \ell_x = i3, \ell_y = i3.$

For the reconstruction of functions $u_{i_3}^-$ and $v_{i_3}^-$ at the barycenter (x_i, y_i) (see Fig. 2.1 and Fig. 2.4):

- $l = 1, \ell = i, i1, i3, ia, ib, kb;$ $l = 2, \ell = i, i1, i3, ia, ka, kb;$
- $l = 3, \ell = i, i1, i2, i3, ib, ka;$ $l = 4, \ell = i, i1, i2, i3, ia, kb;$
- $l = 5, \ell = i, i1, i3, ka, kb, \ell_x = i1;$ $l = 6, \ell = i, i1, i3, ia, ib, \ell_x = i3;$
- $l = 7, \ell = i, i1, i3, ka, kb, \ell_y = i1;$ $l = 8, \ell = i, i1, i3, ia, ib, \ell_y = i3;$
- $l = 9, \ell = i, i1, ia, ib, \ell_x = i1, \ell_y = i1;$ $l = 10, \ell = i, i3, ka, kb, \ell_x = i3, \ell_y = i3;$
- $l = 11, \ell = i, i1, i2, i3, \ell_x = i1, \ell_y = i1;$ $l = 12, \ell = i, i1, i2, i3, \ell_x = i3, \ell_y = i3.$

2. The reconstruction of functions $u_{G_{\ell\ell\ell}}^-$ and $v_{G_{\ell\ell\ell}}^-$ at different Gaussian quadrature points $(x_{G_{\ell\ell\ell}}, y_{G_{\ell\ell\ell}})$ of three boundaries of the target cell Δ_i (see Fig. 2.1 and Fig. 2.5).

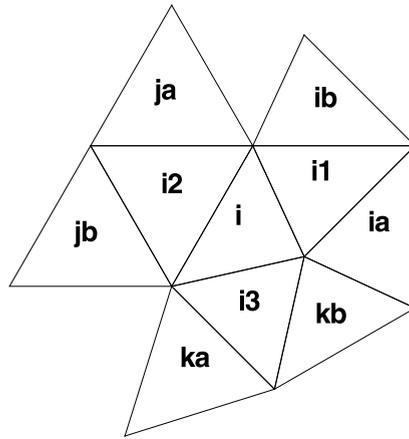


Fig. 2.5. The big central stencil.

Step 2.2.1. We select the big stencil as $S = \{\Delta_i, \Delta_{i1}, \Delta_{i2}, \Delta_{i3}, \Delta_{ia}, \Delta_{ib}, \Delta_{ja}, \Delta_{jb}, \Delta_{ka}, \Delta_{kb}\}$. Then we construct a polynomial $Q(x, y) \in \text{span}\{1, \frac{(x-x_i)}{\sqrt{|\Delta_i|}}, \frac{(y-y_i)}{\sqrt{|\Delta_i|}}, \frac{(x-x_i)^2}{|\Delta_i|}, \frac{(x-x_i)(y-y_i)}{|\Delta_i|}, \frac{(y-y_i)^2}{|\Delta_i|}, \frac{(x-x_i)^3}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)^2(y-y_i)}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)(y-y_i)^2}{\sqrt{|\Delta_i|^3}}, \frac{(y-y_i)^3}{\sqrt{|\Delta_i|^3}}\}$ by requiring that it matches the point values of ϕ and cell averages of u, v on the target cell Δ_i and others in a least square sense [9,29].

Step 2.2.2. We then construct polynomials $q_l(x, y) \in \text{span}\{1, \frac{(x-x_i)}{\sqrt{|\Delta_i|}}, \frac{(y-y_i)}{\sqrt{|\Delta_i|}}, \frac{(x-x_i)^2}{|\Delta_i|}, \frac{(x-x_i)(y-y_i)}{|\Delta_i|}, \frac{(y-y_i)^2}{|\Delta_i|}\}$, $l = 1, \dots, \chi\chi$ (which satisfy both the point values of ϕ and the cell averages of variables u and v on the associated cells) that satisfying (2.10), (2.11) and (2.12).

For simplicity, we omit the similar steps and then obtain the final approximations as:

$$u^-(x_{G_{\ell\ell\ell}}, y_{G_{\ell\ell\ell}}) \approx \sum_{l=1}^{\chi\chi} \omega_l^x \frac{\partial}{\partial x} q_l(x_{G_{\ell\ell\ell}}, y_{G_{\ell\ell\ell}}), \quad \ell\ell = 1, 2, 3, \ell = 1, 2, \tag{2.21}$$

and

$$v^-(x_{G_{\ell\ell\ell}}, y_{G_{\ell\ell\ell}}) \approx \sum_{l=1}^{\chi\chi} \omega_l^y \frac{\partial}{\partial y} q_l(x_{G_{\ell\ell\ell}}, y_{G_{\ell\ell\ell}}), \quad \ell\ell = 1, 2, 3, \ell = 1, 2. \tag{2.22}$$

In this paper, we provide one type of the smaller stencils as follows:

$$\begin{aligned} l = 1, \ell = i, i1, ia, ib, \ell_x = i, \ell_y = i; & \quad l = 2, \ell = i, i1, i3, ia, \ell_x = i, \ell_y = i; \\ l = 3, \ell = i, i1, i2, ib, \ell_x = i, \ell_y = i; & \quad l = 4, \ell = i, i2, ja, jb, \ell_x = i, \ell_y = i; \\ l = 5, \ell = i, i1, i2, ja, \ell_x = i, \ell_y = i; & \quad l = 6, \ell = i, i2, i3, jb, \ell_x = i, \ell_y = i; \\ l = 7, \ell = i, i3, ka, kb, \ell_x = i, \ell_y = i; & \quad l = 8, \ell = i, i2, i3, ka, \ell_x = i, \ell_y = i; \\ l = 9, \ell = i, i1, i3, kb, \ell_x = i, \ell_y = i. & \end{aligned} \tag{2.23}$$

Remarks. The target cell Δ_i should retain central to avoid serious downwind on the big central stencil for the boundary point value reconstruction. When some triangles merge in the stencil, we can always use the next layer of neighboring triangles to overcome this drawback.

2.3. The construction of Hermite WENO4 scheme

1. The reconstruction of functions $u_{i_j}^-$ and $v_{i_j}^-$ at the barycenter (x_i, y_i) in the inner sector whose inner angle is θ_{i_j} , $j = 1, 2, 3$ of the target cell Δ_i (see from Fig. 2.1 to Fig. 2.4).

Step 3.1.1. We select the same big stencils as $S = \{\Delta_i, \Delta_{i1}, \Delta_{i2}, \Delta_{i3}, \Delta_{ia}, \Delta_{ib}, \Delta_{ja}, \Delta_{jb}\}$ (for $u_{i_1}^-$ and $v_{i_1}^-$), $S = \{\Delta_i, \Delta_{i1}, \Delta_{i2}, \Delta_{i3}, \Delta_{ja}, \Delta_{jb}, \Delta_{ka}, \Delta_{kb}\}$ (for $u_{i_2}^-$ and $v_{i_2}^-$) and $S = \{\Delta_i, \Delta_{i1}, \Delta_{i2}, \Delta_{i3}, \Delta_{ia}, \Delta_{ib}, \Delta_{ka}, \Delta_{kb}\}$ (for $u_{i_3}^-$ and $v_{i_3}^-$), respectively. Then we construct different polynomials $P(x, y) \in \text{span}\{1, \frac{(x-x_i)}{\sqrt{|\Delta_i|}}, \frac{(y-y_i)}{\sqrt{|\Delta_i|}}, \frac{(x-x_i)^2}{|\Delta_i|}, \frac{(x-x_i)(y-y_i)}{|\Delta_i|}, \frac{(y-y_i)^2}{|\Delta_i|}, \frac{(x-x_i)^3}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)^2(y-y_i)}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)(y-y_i)^2}{\sqrt{|\Delta_i|^3}}, \frac{(y-y_i)^3}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)^4}{|\Delta_i|^2}, \frac{(x-x_i)^3(y-y_i)}{|\Delta_i|^2}, \frac{(x-x_i)^2(y-y_i)^2}{|\Delta_i|^2}, \frac{(x-x_i)(y-y_i)^3}{|\Delta_i|^2}, \frac{(y-y_i)^4}{|\Delta_i|^2}\}$ by requiring that they match the point values of ϕ and cell averages of u, v on the target cell Δ_i and other cells in a least square sense [9,29].

Step 3.1.2. We then construct polynomials $q_l(x, y) \in \text{span}\{1, \frac{(x-x_i)}{\sqrt{|\Delta_i|}}, \frac{(y-y_i)}{\sqrt{|\Delta_i|}}, \frac{(x-x_i)^2}{|\Delta_i|}, \frac{(x-x_i)(y-y_i)}{|\Delta_i|}, \frac{(y-y_i)^2}{|\Delta_i|}, \frac{(x-x_i)^3}{\sqrt{|\Delta_i|^3}}, \frac{(y-y_i)^3}{\sqrt{|\Delta_i|^3}}\}$, $l = 1, \dots, \chi$ that satisfying (2.10), (2.11) and (2.12).

Step 3.1.3. We find the two sets of linear weights, denoted by $\gamma_1^x, \dots, \gamma_\chi^x$ and $\gamma_1^y, \dots, \gamma_\chi^y$, satisfying (2.13) and (2.14). Then we can get the approximations of $u_{i_j}^-$ and $v_{i_j}^-$ at the point (x_i, y_i) . And there are some free parameters need to be determined by asking for (2.15) holds.

Step 3.1.4. We compute the smoothness indicators by using (2.16) and (2.17).

Step 3.1.5. We compute the nonlinear weights by using (2.18).

Step 3.1.6. The final approximations are then given by (2.19) and (2.20).

In this paper, we provide one type of the smaller stencils as follows:

For the reconstruction of functions $u_{i_1}^-$ and $v_{i_1}^-$ at the barycenter (x_i, y_i) (see Fig. 2.1 and Fig. 2.2):

- $l = 1, \ell = i, i1, i2, ia, ib, ja, \ell_x = i1, \ell_y = i1;$
- $l = 2, \ell = i, i1, i2, ia, ib, ja, \ell_x = ib, \ell_y = ib;$
- $l = 3, \ell = i, i1, i2, ia, ib, jb, \ell_x = i1, \ell_y = i1;$
- $l = 4, \ell = i, i1, i2, ia, ib, jb, \ell_x = ib, \ell_y = ib;$
- $l = 5, \ell = i, i1, i2, ib, ja, jb, \ell_x = ja, \ell_y = ja;$
- $l = 6, \ell = i, i1, i2, ib, ja, jb, \ell_x = i2, \ell_y = i2;$
- $l = 7, \ell = i, i1, i2, i3, ib, ja, \ell_x = ib, \ell_y = ib;$
- $l = 8, \ell = i, i1, i2, i3, ib, ja, \ell_x = ja, \ell_y = ja;$
- $l = 9, \ell = i, i1, i2, i3, ia, ja, \ell_x = i1, \ell_y = i1;$
- $l = 10, \ell = i, i1, i2, i3, ia, ja, \ell_x = ja, \ell_y = ja;$
- $l = 11, \ell = i, i1, i2, i3, ib, jb, \ell_x = ib, \ell_y = ib;$
- $l = 12, \ell = i, i1, i2, i3, ib, jb, \ell_x = i2, \ell_y = i2;$
- $l = 13, \ell = i, i1, i2, i3, ia, ib, ja, jb;$
- $l = 14, \ell = i, i1, i2, ia, ja, jb, \ell_x = ja, \ell_y = ja;$
- $l = 15, \ell = i, i1, i2, ia, ja, jb, \ell_x = i2, \ell_y = i2.$

For the reconstruction of functions $u_{i_2}^-$ and $v_{i_2}^-$ at the barycenter (x_i, y_i) (see Fig. 2.1 and Fig. 2.3):

- $l = 1, \ell = i, i2, i3, ja, jb, ka, \ell_x = i2, \ell_y = i2;$
- $l = 2, \ell = i, i2, i3, ja, jb, ka, \ell_x = jb, \ell_y = jb;$
- $l = 3, \ell = i, i2, i3, ja, jb, kb, \ell_x = i2, \ell_y = i2;$
- $l = 4, \ell = i, i2, i3, ja, jb, kb, \ell_x = jb, \ell_y = jb;$
- $l = 5, \ell = i, i2, i3, jb, ka, kb, \ell_x = ka, \ell_y = ka;$
- $l = 6, \ell = i, i2, i3, jb, ka, kb, \ell_x = i3, \ell_y = i3;$
- $l = 7, \ell = i, i2, i3, ja, ka, kb, \ell_x = ka, \ell_y = ka;$
- $l = 8, \ell = i, i2, i3, ja, ka, kb, \ell_x = i3, \ell_y = i3;$
- $l = 9, \ell = i, i1, i2, i3, jb, ka, \ell_x = jb, \ell_y = jb;$
- $l = 10, \ell = i, i1, i2, i3, jb, ka, \ell_x = ka, \ell_y = ka;$
- $l = 11, \ell = i, i1, i2, i3, ja, ka, \ell_x = i2, \ell_y = i2;$
- $l = 12, \ell = i, i1, i2, i3, ja, ka, \ell_x = ka, \ell_y = ka;$
- $l = 13, \ell = i, i1, i2, i3, jb, kb, \ell_x = jb, \ell_y = jb;$
- $l = 14, \ell = i, i1, i2, i3, jb, kb, \ell_x = i3, \ell_y = i3;$
- $l = 15, \ell = i, i1, i2, i3, ja, jb, ka, kb.$

For the reconstruction of functions $u_{i_3}^-$ and $v_{i_3}^-$ at the barycenter (x_i, y_i) (see Fig. 2.1 and Fig. 2.4):

- $l = 1, \ell = i, i1, i3, ia, ib, kb, \ell_x = ia, \ell_y = ia;$
- $l = 2, \ell = i, i1, i3, ia, ib, kb, \ell_x = i1, \ell_y = i1;$
- $l = 3, \ell = i, i1, i3, ia, ib, ka, \ell_x = ia, \ell_y = ia;$
- $l = 4, \ell = i, i1, i3, ia, ib, ka, \ell_x = i1, \ell_y = i1;$
- $l = 5, \ell = i, i1, i3, ia, ka, kb, \ell_x = i3, \ell_y = i3;$
- $l = 6, \ell = i, i1, i3, ia, ka, kb, \ell_x = kb, \ell_y = kb;$
- $l = 7, \ell = i, i1, i3, ib, ka, kb, \ell_x = i3, \ell_y = i3;$
- $l = 8, \ell = i, i1, i3, ib, ka, kb, \ell_x = kb, \ell_y = kb;$
- $l = 9, \ell = i, i1, i2, i3, ia, kb, \ell_x = ia, \ell_y = ia;$
- $l = 10, \ell = i, i1, i2, i3, ia, kb, \ell_x = kb, \ell_y = kb;$
- $l = 11, \ell = i, i1, i2, i3, ia, ka, \ell_x = ia, \ell_y = ia;$
- $l = 12, \ell = i, i1, i2, i3, ia, ka, \ell_x = i3, \ell_y = i3;$
- $l = 13, \ell = i, i1, i2, i3, ib, kb, \ell_x = i1, \ell_y = i1;$
- $l = 14, \ell = i, i1, i2, i3, ib, kb, \ell_x = kb, \ell_y = kb;$
- $l = 15, \ell = i, i1, i2, i3, ia, ib, ka, kb.$

2. The reconstruction of functions $u_{G_{\ell\ell\ell}}^-$ and $v_{G_{\ell\ell\ell}}^-$ at different Gaussian quadrature points $(x_{G_{\ell\ell\ell}}, y_{G_{\ell\ell\ell}})$ of three boundaries of the target cell Δ_i (see Fig. 2.1 and Fig. 2.5).

Step 3.2.1. We select the big stencil as $S = \{\Delta_i, \Delta_{i1}, \Delta_{i2}, \Delta_{i3}, \Delta_{ia}, \Delta_{ib}, \Delta_{ja}, \Delta_{jb}, \Delta_{ka}, \Delta_{kb}\}$. Then we construct polynomial $Q(x, y) \in \text{span}\{1, \frac{(x-x_i)}{\sqrt{|\Delta_i|}}, \frac{(y-y_i)}{\sqrt{|\Delta_i|}}, \frac{(x-x_i)^2}{|\Delta_i|}, \frac{(x-x_i)(y-y_i)}{|\Delta_i|}, \frac{(y-y_i)^2}{|\Delta_i|}, \frac{(x-x_i)^3}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)^2(y-y_i)}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)(y-y_i)^2}{\sqrt{|\Delta_i|^3}}, \frac{(y-y_i)^3}{\sqrt{|\Delta_i|^3}}, \frac{(x-x_i)^4}{|\Delta_i|^2}, \frac{(x-x_i)^3(y-y_i)}{|\Delta_i|^2}, \frac{(x-x_i)^2(y-y_i)^2}{|\Delta_i|^2}, \frac{(x-x_i)(y-y_i)^3}{|\Delta_i|^2}, \frac{(y-y_i)^4}{|\Delta_i|^2}\}$ by requiring that it matches the point values of ϕ and cell averages of u, v on the target cell Δ_i and other cells in a least square sense [9,29].

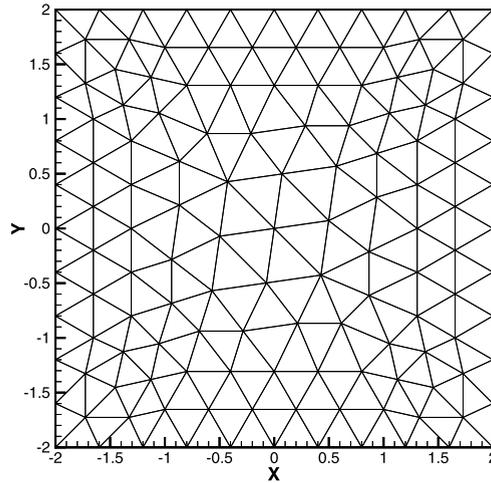


Fig. 3.1. The coarsest unstructured mesh with boundary triangle size: $h = \frac{4}{10}$.

Step 3.2.2. We then construct polynomials $q_l(x, y) \in \text{span}\{1, \frac{(x-x_i)}{\sqrt{|\Delta_l|}}, \frac{(y-y_i)}{\sqrt{|\Delta_l|}}, \frac{(x-x_i)^2}{|\Delta_l|}, \frac{(x-x_i)(y-y_i)}{|\Delta_l|}, \frac{(y-y_i)^2}{|\Delta_l|}\}$, $l = 1, \dots, \chi\chi$ that satisfying (2.10), (2.11) and (2.12).

For simplicity, we omit the similar steps if won't cause confusions. And in this paper, we provide one type of the smaller stencils as follows:

$$\begin{aligned}
 l = 1, \ell = i, i1, ia, ib, \ell_x = i, \ell_y = i; & \quad l = 2, \ell = i, i2, ja, jb, \ell_x = i, \ell_y = i; \\
 l = 3, \ell = i, i3, ka, kb, \ell_x = i, \ell_y = i; & \quad l = 4, \ell = i, i1, i2, ib, \ell_x = i, \ell_y = i; \\
 l = 5, \ell = i, i1, i2, ja, \ell_x = i, \ell_y = i; & \quad l = 6, \ell = i, i2, i3, jb, \ell_x = i, \ell_y = i; \\
 l = 7, \ell = i, i2, i3, ka, \ell_x = i, \ell_y = i; & \quad l = 8, \ell = i, i1, i3, ia, \ell_x = i, \ell_y = i; \\
 l = 9, \ell = i, i1, i3, kb, \ell_x = i, \ell_y = i; & \quad l = 10, \ell = i, i1, ia, \ell_x = i, \ell_y = i, \ell_x = ia; \\
 l = 11, \ell = i, i1, ib, \ell_x = i, \ell_y = i, \ell_y = ia; & \quad l = 12, \ell = i, i2, ja, \ell_x = i, \ell_y = i, \ell_x = ja; \\
 l = 13, \ell = i, i2, jb, \ell_x = i, \ell_y = i, \ell_y = ja; & \quad l = 14, \ell = i, i3, ka, \ell_x = i, \ell_y = i, \ell_x = ka; \\
 l = 15, \ell = i, i3, kb, \ell_x = i, \ell_y = i, \ell_y = ka. &
 \end{aligned}$$

Remarks. In this paper, the number of small stencils is more than that is needed for solving the linear weights. But if some triangles distort severely (can't be used to construct polynomials in 2D on unstructured meshes), we simply discard using the small stencils which harbor such distorted triangles and won't pollute the numerical accuracy resolutions in smooth regions and can obtain essentially non-oscillatory property nearby the discontinuities simultaneously. When the number of good small stencils (can be used to construct polynomials in 2D) are not rich enough for solving the linear weights, we can also use the next layer of neighboring triangles to combine new small stencils and overcome such difficulty.

3. Numerical tests

In this section, we present the results of numerical tests for the schemes in the previous sections. We will show the results by setting CFL number to be 0.6 for both the HWENO3 and HWENO4 schemes. For the temporal discretization, we use the third-order TVD Runge–Kutta scheme [28].

Example 3.1. We solve the following Hamilton–Jacobi equation [16]:

$$\phi_t - \cos(\phi_x + \phi_y + 1) = 0, \quad -2 \leq x, y < 2, \quad (3.1)$$

with the initial condition $\phi(x, y, 0) = -\cos(\pi(x + y)/2)$ and the periodic boundary conditions. When $t = 0.5/\pi^2$ the solution is smooth. The coarsest unstructured mesh with boundary triangle size $h = \frac{4}{10}$ is shown in Fig. 3.1. The errors and numerical orders of accuracy by HWENO3 and HWENO4 schemes are shown in Table 3.1. We can see that both schemes achieve their designed order of accuracy.

Table 3.1

$\phi_t - \cos(\phi_x + \phi_y + 1) = 0$. $\phi(x, y, 0) = -\cos(\pi(x + y)/2)$. Periodic boundary conditions. $t = 0.5/\pi^2$.

Cells	HWENO3				HWENO4			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
232	3.42E-3		1.28E-2		2.89E-3		1.99E-2	
928	8.88E-4	1.95	5.99E-3	1.10	5.16E-4	2.49	3.79E-3	2.39
3712	1.72E-4	2.37	1.22E-3	2.29	3.80E-5	3.76	8.65E-4	2.13
14 848	2.27E-5	2.92	2.22E-4	2.46	1.37E-6	4.79	4.32E-5	4.32
59 392	1.79E-6	3.66	1.83E-5	3.60	7.59E-8	4.17	2.35E-6	4.20

Table 3.2

$\phi_t + \frac{(\phi_x + \phi_y + 1)^2}{2} = 0$. $\phi(x, y, 0) = -\cos(\pi(x + y)/2)$. Periodic boundary conditions. $t = 0.5/\pi^2$.

Cells	HWENO3				HWENO4			
	L^1 error	Order	L^∞ error	Order	L^1 error	Order	L^∞ error	Order
232	1.19E-2		4.19E-2		1.32E-2		8.88E-2	
928	3.31E-3	1.85	1.33E-2	1.65	1.89E-3	2.80	1.30E-2	2.77
3712	7.44E-4	2.16	4.36E-3	1.62	1.80E-4	3.39	1.10E-3	3.56
14 848	8.78E-5	3.08	8.62E-4	2.34	8.68E-6	4.38	7.06E-5	3.96
59 392	6.63E-6	3.72	6.18E-5	3.80	5.00E-7	4.12	4.45E-6	3.99

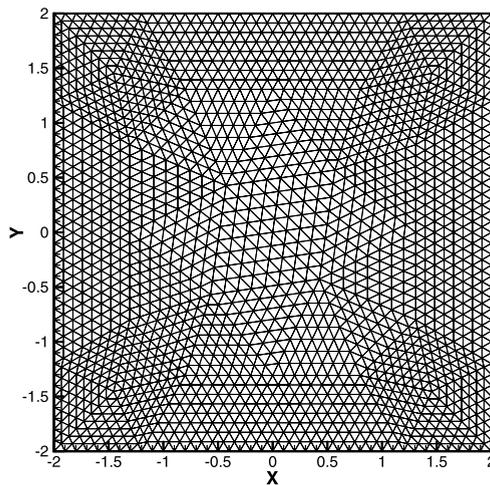


Fig. 3.2. Unstructured mesh for two dimensional Burgers' equation. Boundary triangle size: $h = \frac{4}{40}$.

Example 3.2. We solve the following nonlinear scalar two dimensional Burgers' equation:

$$\phi_t + \frac{(\phi_x + \phi_y + 1)^2}{2} = 0, \quad -2 \leq x, y < 2, \tag{3.2}$$

with the initial condition $\phi(x, y, 0) = -\cos(\pi(x + y)/2)$ and periodic boundary conditions. When $t = 0.5/\pi^2$ the solution is still smooth. The coarsest unstructured mesh with boundary triangle size $h = \frac{4}{10}$ is also shown in Fig. 3.1. The errors and numerical orders of accuracy by the HWENO schemes are shown in Table 3.2. Again, we can see that both schemes achieve their designed order of accuracy.

Example 3.3. We solve the same two dimensional nonlinear Burgers' equation (3.2) as in Example 3.2 with the same initial condition $\phi(x, y, 0) = -\cos(\pi(x + y)/2)$, except that we now plot the results at $t = 1.5/\pi^2$ with boundary triangle size $h = \frac{4}{40}$ (see Fig. 3.2) and are presented in Fig. 3.3 and Fig. 3.4 when the discontinuous derivative has already appeared in the solution. We observe good resolutions for this example.

Example 3.4. The two dimensional Riemann problem with a non-convex flux:

$$\begin{cases} \phi_t + \sin(\phi_x + \phi_y) = 0, & -1 \leq x, y < 1, \\ \phi(x, y, 0) = \pi(|y| - |x|). \end{cases} \tag{3.3}$$

The solution of the HWENO schemes are plotted at $t = 1$ with boundary triangle size $h = \frac{2}{40}$ (see Fig. 3.5) are shown in Fig. 3.6 and Fig. 3.7. We can also observe good resolutions for these two schemes' numerical simulations.

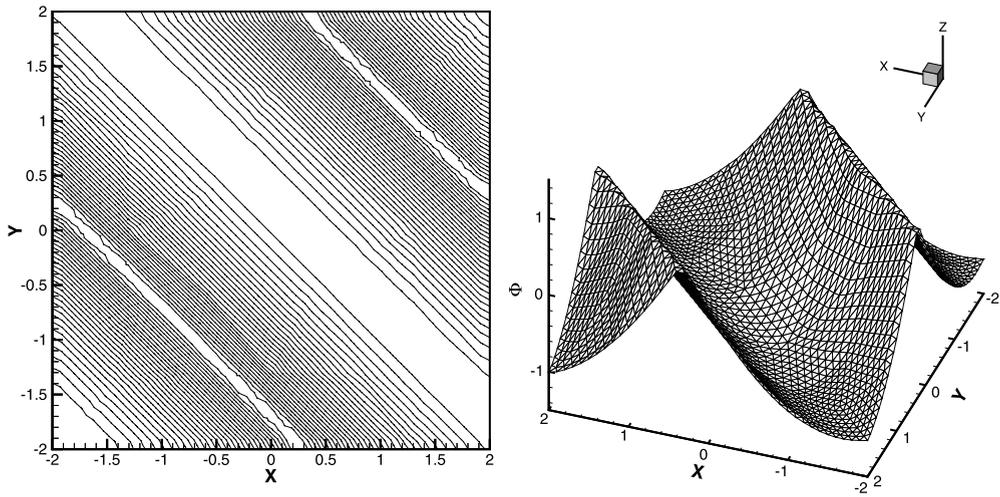


Fig. 3.3. Two dimensional Burgers' equation. Boundary triangle size: $h = \frac{4}{40}$. $t = 1.5/\pi^2$. HWENO3 scheme. Left: contours of the solution; right: the surface of the solution.

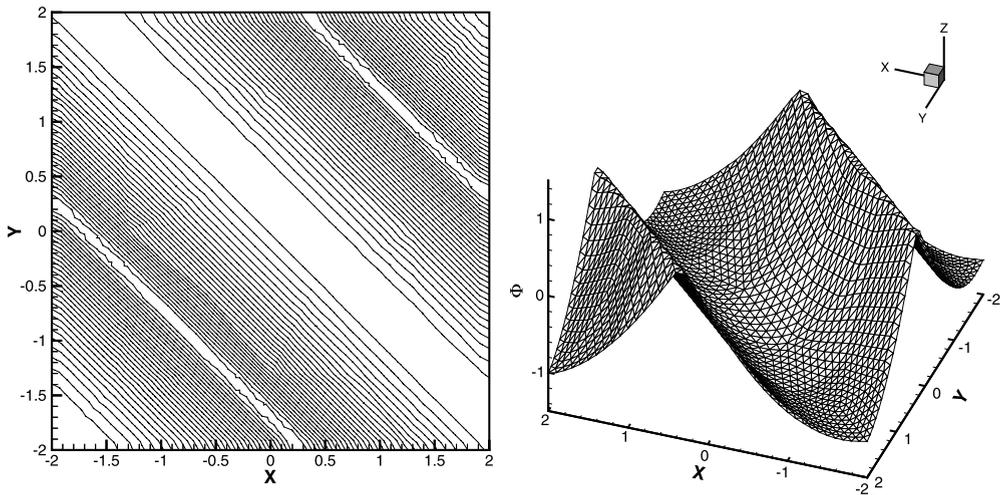


Fig. 3.4. Two dimensional Burgers' equation. Boundary triangle size: $h = \frac{4}{40}$. $t = 1.5/\pi^2$. HWENO4 scheme. Left: contours of the solution; right: the surface of the solution.

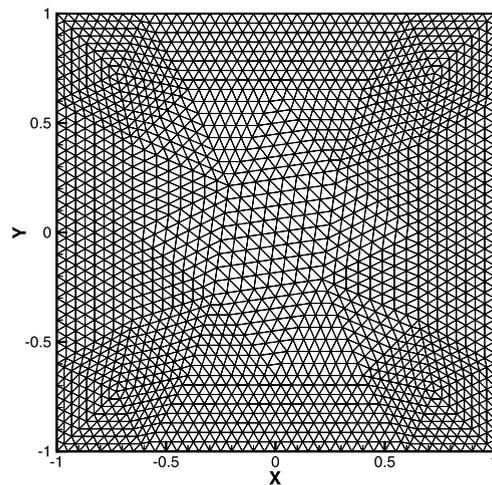


Fig. 3.5. Unstructured mesh for two dimensional Riemann problem with a non-convex flux. Boundary triangle size: $h = \frac{2}{40}$.

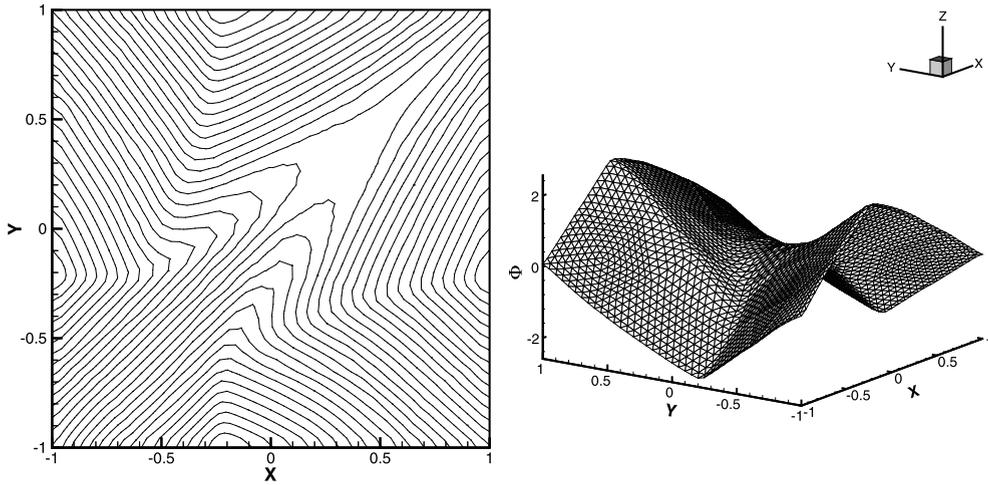


Fig. 3.6. Two dimensional Riemann problem with a non-convex flux $H(\phi_x, \phi_y) = \sin(\phi_x + \phi_y)$. Boundary triangle size: $h = \frac{2}{40}$. $t = 1$. HWENO3 scheme. Left: contours of the solution; right: the surface of the solution.

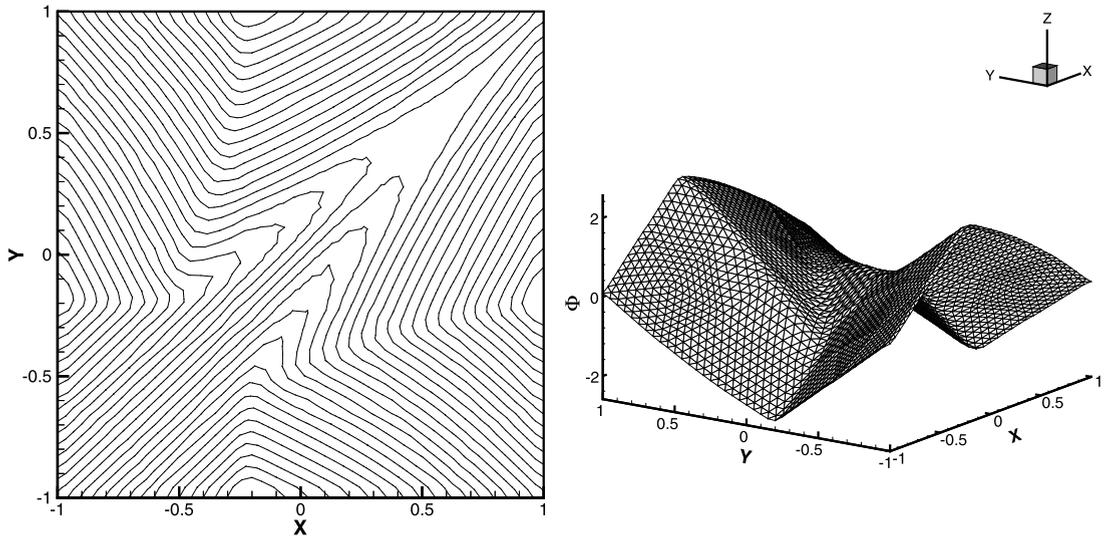


Fig. 3.7. Two dimensional Riemann problem with a non-convex flux $H(\phi_x, \phi_y) = \sin(\phi_x + \phi_y)$. Boundary triangle size: $h = \frac{2}{40}$. $t = 1$. HWENO4 scheme. Left: contours of the solution; right: the surface of the solution.

Example 3.5. A problem from optimal control:

$$\begin{cases} \phi_t + \sin(y)\phi_x + (\sin(x) + \text{sign}(\phi_y))\phi_y - \frac{1}{2} \sin(y)^2 - (1 - \cos(x)) = 0, & -\pi \leq x, y < \pi, \\ \phi(x, y, 0) = 0, \end{cases} \quad (3.4)$$

with periodic conditions, see [20]. The solutions of the HWENO schemes are plotted at $t = 1$ with boundary triangle size $h = \frac{2\pi}{40}$ (see Fig. 3.8) and the optimal control $\omega = \text{sign}(\phi_y)$ are shown in Fig. 3.9 and Fig. 3.10.

Example 3.6. A two dimensional Eikonal equation with a non-convex Hamiltonian, which arises in geometric optics [13], is given by:

$$\begin{cases} \phi_t + \sqrt{\phi_x^2 + \phi_y^2 + 1} = 0, & 0 \leq x, y < 1, \\ \phi(x, y, 0) = \frac{1}{4} (\cos(2\pi x) - 1)(\cos(2\pi y) - 1) - 1. \end{cases} \quad (3.5)$$

The solutions of the HWENO schemes are plotted at $t = 0.6$ with boundary triangle size $h = \frac{1}{40}$ (see Fig. 3.11) and are presented in Fig. 3.12 and Fig. 3.13. Good resolutions are observed with the different two HWENO schemes.

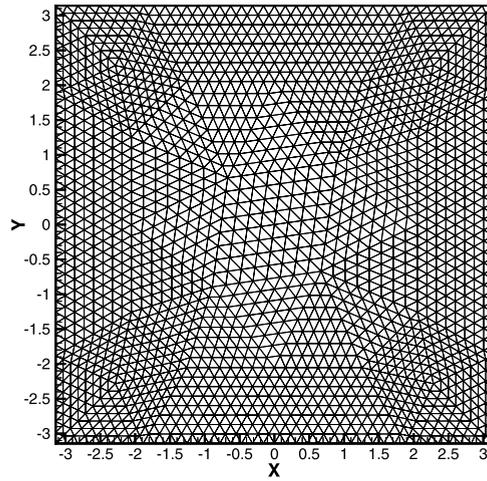


Fig. 3.8. Unstructured mesh for the optimal control problem. Boundary triangle size: $h = \frac{2\pi}{40}$.

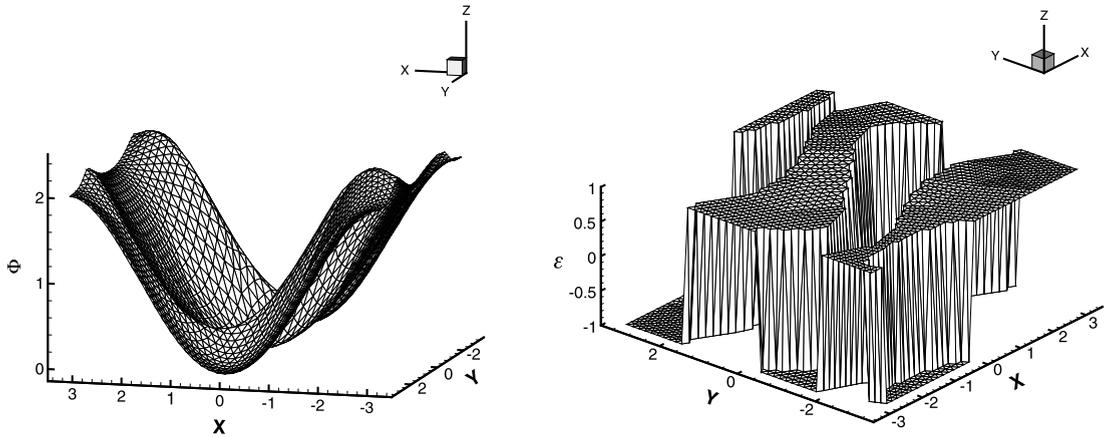


Fig. 3.9. The optimal control problem. Boundary triangle size: $h = \frac{2\pi}{40}$. $t = 1$. HWENO3 scheme. Left: the surface of the solution; right: the optimal control $\omega = \text{sign}(\phi_y)$.

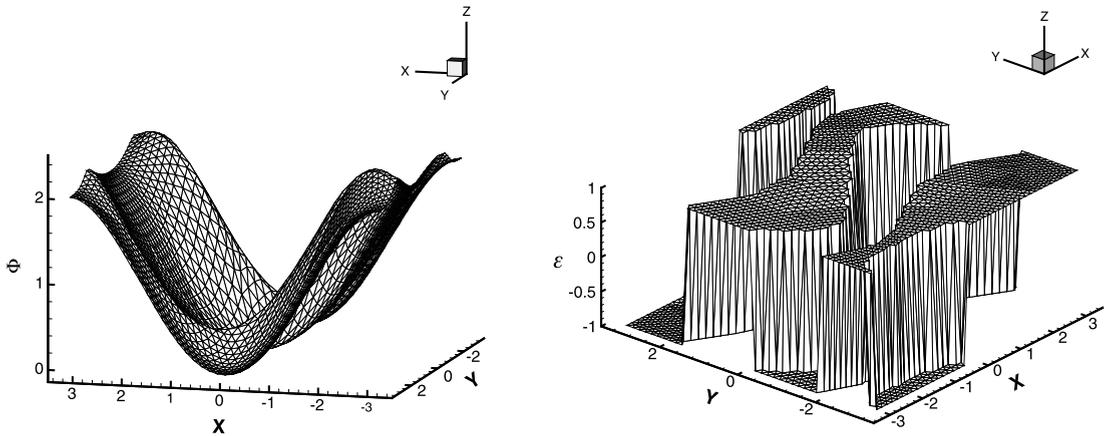


Fig. 3.10. The optimal control problem. Boundary triangle size: $h = \frac{2\pi}{40}$. $t = 1$. HWENO4 scheme. Left: the surface of the solution; right: the optimal control $\omega = \text{sign}(\phi_y)$.

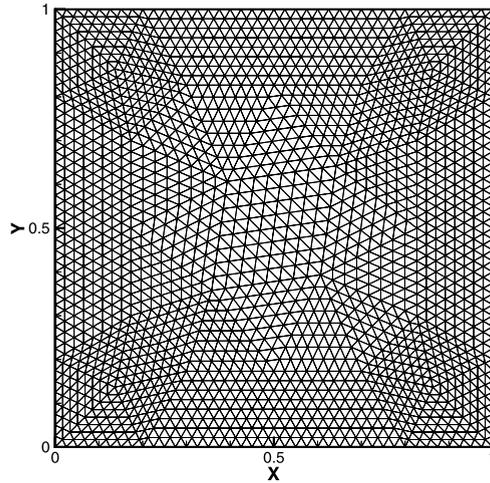


Fig. 3.11. Unstructured mesh for Eikonal equation with a non-convex Hamiltonian and the problem of propagating surface. Boundary triangle size: $h = \frac{1}{40}$.

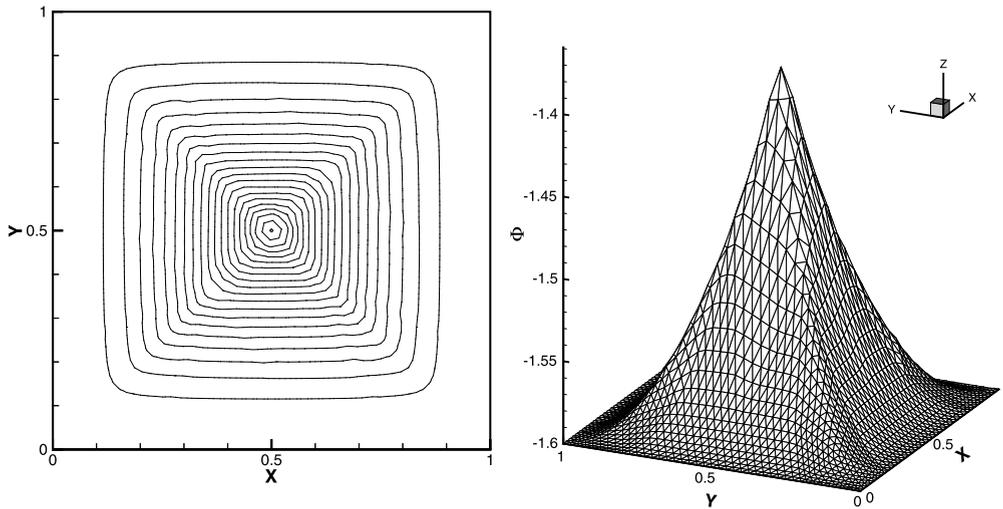


Fig. 3.12. Eikonal equation with a non-convex Hamiltonian. Boundary triangle size: $h = \frac{1}{40}$. $t = 0.6$. HWENO3 scheme. Left: contours of the solution; right: the surface of the solution.

Example 3.7. The problem of a propagating surface [19]:

$$\begin{cases} \phi_t - (1 - \varepsilon K)\sqrt{\phi_x^2 + \phi_y^2 + 1} = 0, & 0 \leq x, y < 1, \\ \phi(x, y, 0) = 1 - \frac{1}{4}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1), \end{cases} \quad (3.6)$$

where K is the mean curvature defined by:

$$K = -\frac{\phi_{xx}(1 + \phi_y)^2 - 2\phi_{xy}\phi_x\phi_y + \phi_{yy}(1 + \phi_x^2)}{(1 + \phi_x^2 + \phi_y^2)^{3/2}},$$

and ε is a small constant. A periodic boundary conditions are used. The results of $\varepsilon = 0$ (pure convection) and $\varepsilon = 0.1$ by the HWENO schemes with boundary triangle size $h = \frac{1}{40}$ (see Fig. 3.11) are presented in Fig. 3.14 and Fig. 3.15. The surfaces at $t = 0$ for $\varepsilon = 0$ and for $\varepsilon = 0.1$, and at $t = 0.1$ for $\varepsilon = 0.1$, are shifted downward in order to show the detail of the solution at later time.

4. Concluding remarks

In this paper, we have constructed a new class of the third and fourth order HWENO schemes for solving two dimensional Hamilton–Jacobi equations on unstructured meshes. The constructions of HWENO schemes for Hamilton–Jacobi

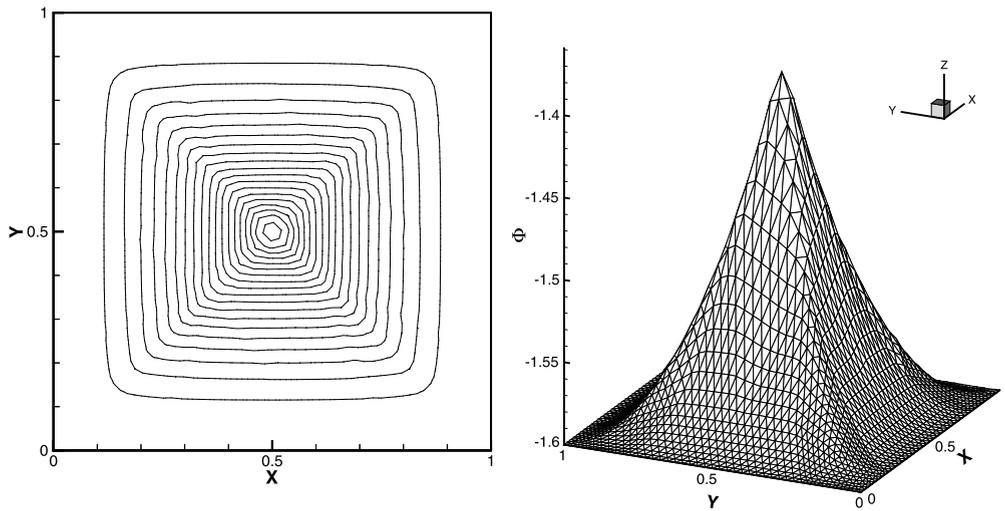


Fig. 3.13. Eikonal equation with a non-convex Hamiltonian. Boundary triangle size: $h = \frac{1}{40}$. $t = 0.6$. HWENO4 scheme. Left: contours of the solution; right: the surface of the solution.

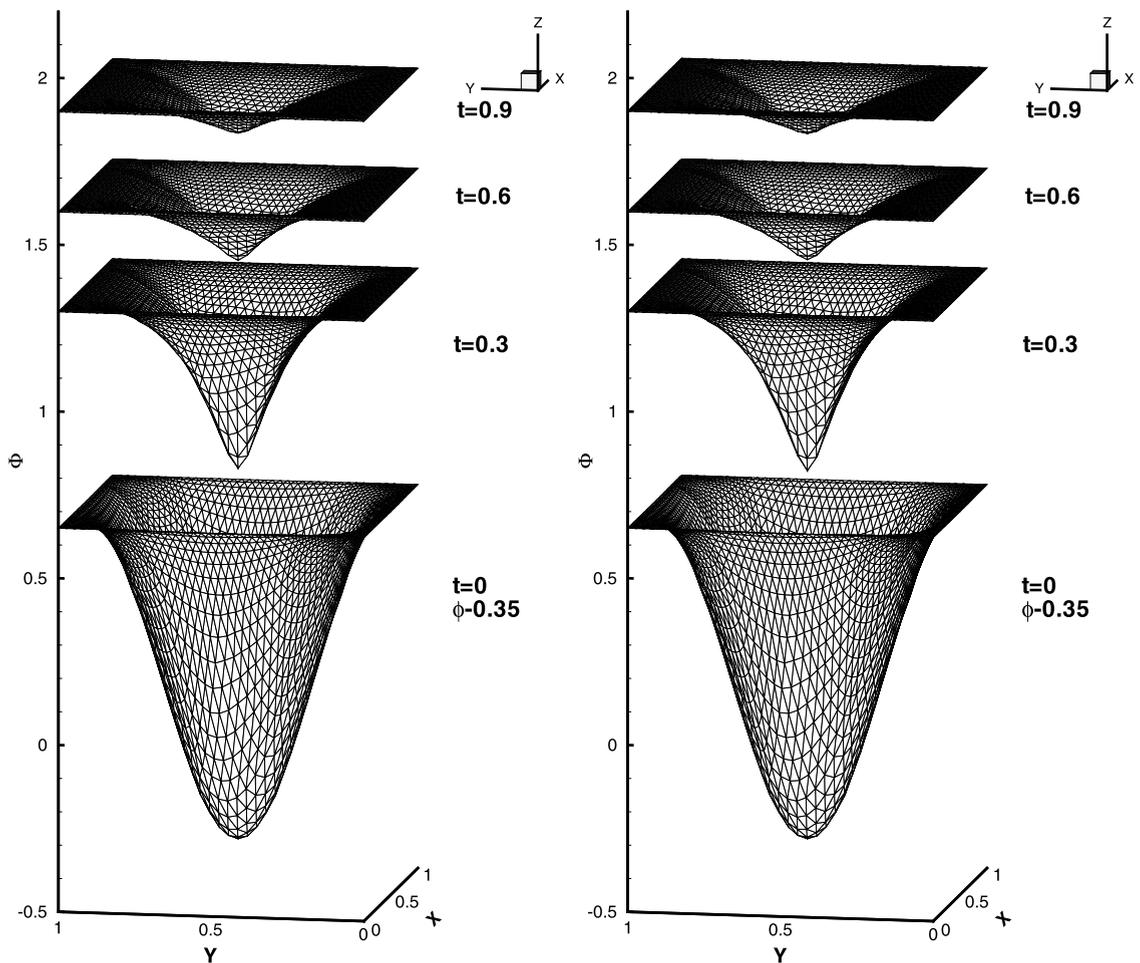


Fig. 3.14. Propagating surface. Boundary triangle size: $h = \frac{1}{40}$. $\varepsilon = 0$. Left: HWENO3 scheme; right: HWENO4 scheme.

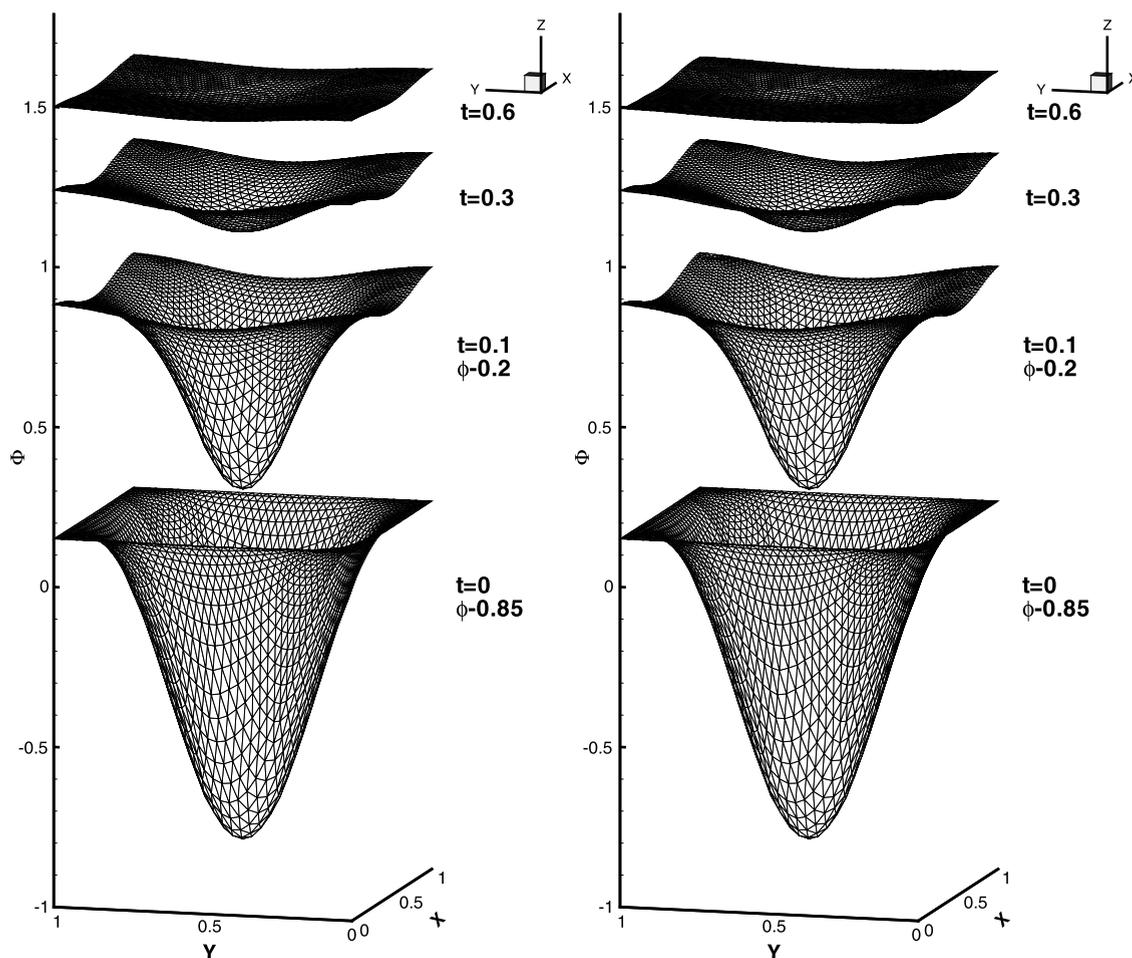


Fig. 3.15. Propagating surface. Boundary triangle size: $h = \frac{1}{40}$. $\varepsilon = 0.1$. Left: HWENO3 scheme; right: HWENO4 scheme.

equations are based on Hermite WENO interpolation in spatial field and then TVD Runge–Kutta discretization is used for the ODEs. The idea of the reconstructions for the HWENO schemes comes from the nodal WENO reconstructions [11,12]. In the HWENO schemes, both the nodal solution and its first two cell averaged derivatives are evolved via time marching and used in the reconstructions, in contrast to the regular WENO schemes [11,16,29] where only the nodal solution value is evolved in time and used in the reconstructions. These methods have high order accuracy for the smooth regions, can obtain high resolutions for the singularities of the derivatives and can converge to the physical viscosity solutions compactly and robustly. Extensive numerical experiments are performed to illustrate the capability of these two unstructured HWENO schemes.

References

- [1] R. Abgrall, On essentially non-oscillatory schemes on unstructured meshes: Analysis and implementation, *J. Comput. Phys.* 114 (1994) 45–58.
- [2] R. Abgrall, Numerical discretization of the first-order Hamilton–Jacobi equation on triangular meshes, *Comm. Pure Appl. Math.* 49 (1996) 1339–1373.
- [3] R. Abgrall, T. Sonar, On the use of Muehlbach expansions in the recovery step of ENO methods, *Numer. Math.* 76 (1997) 1–25.
- [4] S. Augoula, R. Abgrall, High order numerical discretization for Hamilton–Jacobi equations on triangular meshes, *J. Sci. Comput.* 15 (2000) 198–229.
- [5] T.J. Barth, J.A. Sethian, Numerical schemes for the Hamilton–Jacobi and level set equations on triangulated domains, *J. Comput. Phys.* 145 (1998) 1–40.
- [6] C.K. Chan, K.S. Lau, B.L. Zhang, Simulation of a premixed turbulent flame with the discrete vortex method, *Internat. J. Numer. Meth. Engrg.* 48 (2000) 613–627.
- [7] O. Friedrich, Weighted essentially non-oscillatory schemes for the interpolation of mean values on unstructured grids, *J. Comput. Phys.* 144 (1998) 194–212.
- [8] A. Harten, S. Osher, Uniformly high-order accurate non-oscillatory schemes, IMRC Technical Summary Rept. 2823, Univ. of Wisconsin, Madison, WI, May 1985.
- [9] C. Hu, C.-W. Shu, Weighted essentially non-oscillatory schemes on triangular meshes, *J. Comput. Phys.* 150 (1999) 97–127.
- [10] C. Hu, C.-W. Shu, A discontinuous Galerkin finite element method for Hamilton–Jacobi equations, *SIAM J. Sci. Comput.* 21 (1999) 666–690.
- [11] G.S. Jiang, D. Peng, Weighted ENO schemes for Hamilton–Jacobi equations, *SIAM J. Sci. Comput.* 21 (2000) 2126–2143.
- [12] G.S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes, *J. Comput. Phys.* 126 (1996) 202–228.
- [13] S. Jin, Z. Xin, Numerical passage from systems of conservation laws to Hamilton–Jacobi equations, and relaxation schemes, *SIAM J. Numer. Anal.* 35 (1998) 2163–2186.

- [14] F. Lafon, S. Osher, High order two dimensional nonoscillatory methods for solving Hamilton–Jacobi scalar equations, *J. Comput. Phys.* 123 (1996) 235–253.
- [15] O. Lepsky, C. Hu, C.-W. Shu, Analysis of the discontinuous Galerkin method for Hamilton–Jacobi equations, *Appl. Numer. Math.* 33 (2000) 423–434.
- [16] X.G. Li, C.K. Chan, High-order schemes for Hamilton–Jacobi equations on triangular meshes, *J. Comput. Appl. Math.* 167 (2004) 227–241.
- [17] P.L. Lions, *Generalized Solutions of Hamilton–Jacobi Equations*, Pitman, London, 1982.
- [18] X.D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes, *J. Comput. Phys.* 115 (1994) 200–212.
- [19] S. Osher, J. Sethian, Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations, *J. Comput. Phys.* 79 (1988) 12–49.
- [20] S. Osher, C.-W. Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, *SIAM J. Numer. Anal.* 28 (1991) 907–922.
- [21] J. Qiu, Hermite WENO schemes with Lax–Wendroff type time discretizations for Hamilton–Jacobi equations, *J. Comp. Math.* 25 (2007) 131–144.
- [22] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one dimensional case, *J. Comput. Phys.* 193 (2004) 115–135.
- [23] J. Qiu, C.-W. Shu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method II: two-dimensional case, *Comput. Fluids* 34 (2005) 642–663.
- [24] J. Qiu, C.-W. Shu, Hermite WENO schemes for Hamilton–Jacobi equations, *J. Comput. Phys.* 204 (2005) 82–99.
- [25] J.A. Sethian, *Level Set Methods and Fast Marching Methods, Evolving Interface, Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*, Cambridge University Press, Cambridge, 1999.
- [26] J. Shi, C. Hu, C.-W. Shu, A technique of treating negative weights in WENO schemes, *J. Comput. Phys.* 175 (2002) 108–127.
- [27] C.-W. Shu, High order weighted essentially non-oscillatory schemes for convection dominated problems, *SIAM Rev.* 51 (2009) 82–126.
- [28] C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock capturing schemes, *J. Comput. Phys.* 77 (1988) 439–471.
- [29] Y.T. Zhang, C.-W. Shu, High-order WENO schemes for Hamilton–Jacobi equations on triangular meshes, *SIAM J. Sci. Comput.* 24 (2003) 1005–1030.
- [30] J. Zhu, J. Qiu, A class of the fourth order finite volume Hermite weighted essentially non-oscillatory schemes, *Sci. China Ser. A Math.* 51 (2008) 1549–1560.
- [31] J. Zhu, J. Qiu, Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method III: Unstructured meshes, *J. Sci. Comput.* 39 (2009) 293–321.