

High order finite difference hermite WENO schemes for the Hamilton–Jacobi equations on unstructured meshes

Feng Zheng^{a,1}, Chi-Wang Shu^{b,2,*}, Jianxian Qiu^{a,3}

^aSchool of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, PR China

^bDivision of Applied Mathematics, Brown University, Providence, RI 02912, USA

ARTICLE INFO

Article history:

Received 29 August 2018
Revised 31 January 2019
Accepted 14 February 2019
Available online 15 February 2019

Keywords:

HWENO Method
Hamilton–Jacobi equation
Finite difference method
Unstructured mesh

ABSTRACT

In this paper, a new type of high order Hermite weighted essentially non-oscillatory (HWENO) methods is proposed to solve the Hamilton–Jacobi (HJ) equations on unstructured meshes. We use a fourth order accurate scheme to demonstrate our procedure. Both the solution and its spatial derivatives are evolved in time. Our schemes have three advantages. First, they are more compact than the one in [38] as more information is used at each node which allows us to achieve the same high order accuracy with a more compact stencil. Second, the new HWENO approximation on the unstructured mesh allows arbitrary positive linear weights, which enhances the stability of our scheme. Third, the new HWENO procedure produces an approximation polynomial on each triangle, which allows us to compute all the spatial derivatives at the three nodes of each triangle based on this single polynomial, instead of computing each derivative individually with different linear weights in the classical HWENO framework, which improves the efficiency of our scheme. Extensive numerical experiments are performed to verify the accuracy, high resolution and efficiency of this new scheme.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

In this paper, we consider numerical methods for solving the Hamilton–Jacobi (HJ) equations

$$\begin{aligned} \phi_t + H(x, y, t, \phi_x, \phi_y) &= 0 \\ \phi(x, y, 0) &= \phi_0(x, y) \end{aligned} \quad (1.1)$$

The HJ equations can be used in many applications such as optimal control, differential games, image processing, computer vision and so on. It is well known that the solutions of HJ equations are always continuous, however their derivatives could become discontinuous even if the initial condition is smooth. With the definition of the viscosity solution by Crandall and Lions [6], we can obtain the unique weak solution for the HJ equations.

As the spatial derivative of the HJ solution satisfies a conservation law equation in the one dimensional case, the HJ equation has a close relationship with conservation laws. Many successful methods for conservation laws can be adapted to solve the HJ equations. In general, the HJ equations are easier to solve than their corresponding conservation laws because the solutions of HJ equations are smoother than those for conservation laws (continuous versus discontinuous).

There are many papers designing numerical schemes for HJ equations on structured meshes. Crandall and Lion [7] introduced first order monotone schemes and proved that the schemes can converge to the viscosity solution. Although monotone schemes are only first order accurate, they serve as the building blocks for most higher order schemes. Osher and Shu [25] designed high order essentially nonoscillatory (ENO) schemes for solving the HJ equations. Jiang and Peng [11] generalized them to weighted ENO (WENO) schemes, with tremendous success. The WENO schemes in [11] have become standard choices for solving HJ equations on structured meshes. Hermite WENO (HWENO) schemes and related methods have been developed in [27,29,40,44] to achieve more compact stencils for the same order of accuracy. Bryson and Levy [2] presented central schemes for solving the HJ equations. The schemes mentioned above are all designed in the finite difference framework, namely they approximate point values of the

* Corresponding author.

E-mail addresses: fzbz200808-31@163.com (F. Zheng), shu@dam.brown.edu (C.-W. Shu), jxqiu@xmu.edu.cn (J. Qiu).

¹ Research supported by NSFC grant 11571290 and the China Scholarship Council.

² Research supported by ARO grant W911NF-16-1-0103 and NSF grant DMS-1719410.

³ Research supported by NSFC grant 11571290, NSAF grant U1630247 and Science Challenge Project No. TZ2016002.

solution (and possibly also its spatial derivatives). There are also many schemes designed in the finite volume or discontinuous Galerkin (DG) framework, namely they solve for the cell averages of the solution (and possibly also its higher order moments), mostly for the conservation law equations satisfied by the spatial derivatives of the solution but some also for the HJ equations directly. These schemes can be defined both on structured meshes and on unstructured meshes. Hu and Shu [9] designed the first DG scheme of this type. Later, Li and Shu [18] reinterpreted and simplified the DG scheme in [9]. Li and Yakovlev [19] proposed central DG schemes to solve the HJ equations. Liu and Pollack [21] suggested the alternative evolution DG (AEDG) schemes. Zhu and Qiu [42] and Zheng and Qiu [39] designed finite volume schemes to solve the HJ equations.

We are particularly interested in numerical methods on unstructured meshes, because of their ability in dealing with problems in complicated domains. The first paper in this category is [1] by Abgrall, who designed a Lax-Friedrichs (LF) type monotone scheme on triangular meshes and proved its convergence to the viscosity solution. The monotone scheme in [1] is the building block for most higher order schemes for HJ equations on unstructured meshes. Li, Yan and Chan [20] also developed a monotone and convergent scheme based on the weak form of the viscous HJ equation. Lafon and Osher [14] proposed high order ENO methods on triangular meshes. Zhang and Shu [38] and Levy et al. [15] developed high order WENO and central WENO schemes on triangular meshes respectively. Zhu and Qiu [41,43,45] developed high order HWENO schemes and related schemes on triangular meshes. For more details of numerical solutions for HJ equations, we refer to the lecture notes [31].

The schemes that we construct in this paper belong to the class of HWENO schemes. HWENO schemes come from WENO schemes, originally designed for solving conservation laws in [12,22]. The WENO method has been applied in many areas, such as magnetohydrodynamics equations [4,5], shallow water equations [26,35], detonation waves [8,34], multi-phase flow [36,46], Euler equations [10,37] and so on. Comparing with earlier ENO schemes, WENO schemes use a convex combination of several candidate stencils, instead of using just one of them in the ENO procedure. WENO schemes can achieve high order accuracy in smooth regions and can keep sharp and non-oscillatory shock transition when discontinuities appear. Comparing with WENO schemes, HWENO schemes [28,29] use a more compact stencil for the same order of accuracy by evolving both the solution and its derivatives or first order moments in each cell. HWENO schemes can also achieve both high order accuracy and the essentially non-oscillatory property. However, one major difficulty in the classical WENO and HWENO methods is the computation of the linear weights. These linear weights depend on the particular mesh, and for triangular meshes, different cells and different quadrature points have different linear weights. When we implement WENO or HWENO methods on triangular meshes, we would need to compute and prestore the linear weights on every cell. This would be particularly expensive if we use moving meshes. Furthermore, the linear weights, which depend on the local mesh structure, could become negative, which could lead to instability. Even though there is a procedure to handle such negative weights [30], it may not always fix the stability problem when the negative linear weight is very large [23]. Worse still, in certain situations the linear weights for optimal accuracy may fail to exist. Recently, Zhu and Qiu proposed a new type of WENO method [44], with similar ideas in earlier work [3,16,17]. These WENO methods use a convex combination of a high order polynomial on the large stencil and several low order polynomials on small stencils. The high order polynomial determines the accuracy and the low order polynomials play a major role in ensuring the non-oscillatory performance when discontinuities appear.

An important property of these WENO methods is that the linear weights can be chosen as arbitrary positive numbers provided that they sum to one, thus the shortcomings of classical WENO schemes mentioned above can be avoided. In this paper, we exploit this idea in designing a new type of high order HWENO methods for solving HJ equations. Our method belongs to the class of finite difference schemes, in evolving approximations to the point values of the solution and its first order spatial derivatives at nodes. Only the evolution of the solution itself is written in numerical Hamiltonian form (corresponding to the conservative form for solving conservation laws). The evolution of the spatial derivatives is performed in a non-conservative fashion, thus leading to a much simpler and more efficient algorithm comparing with finite volume type schemes. We note that this will not affect convergence to viscosity solutions (correct kink location) when the scheme converges. The main procedure of these HWENO schemes is as follows. First, we take the spatial derivatives of the original HJ equation to get a system of partial differential equations (PDEs) satisfied by these spatial derivatives. Second, we replace the nonlinear terms in the original and derived PDEs with numerical Hamiltonian and approximate the derivatives using the new type of HWENO procedure. Finally, we evolve the solution and its spatial derivatives by the Runge–Kutta method. The constructed HWENO schemes has the following advantages. The scheme is more compact than the WENO method for the same order of accuracy as it uses information not only from the solution but also from its spatial derivatives. We use the new type of HWENO approximation procedure, which allows arbitrary positive linear weights as long as they sum to one, thus simplifying the algorithms significantly on triangular meshes. The new type of HWENO methods produces an approximation polynomial on each triangle, which allows us to compute all the spatial derivatives at the three nodes of each triangle based on this single polynomial, instead of computing each derivative individually with different linear weights in the classical WENO or HWENO framework. This results in a significant saving of computational cost. Our scheme is analyzed for its formal high order of accuracy, measured by local truncation errors, in Section 2.2.1, see Remark 1 there.

The organization of this paper is as follows. In Section 2, we introduce our new HWENO scheme in detail. In Section 3, we present numerical results to demonstrate the performance of our HWENO schemes. Conclusion remarks are given in Section 4.

2. The HWENO algorithm for 2D unstructured meshes

In this section, we describe in detail the framework of our HWENO schemes for solving HJ equations and the HWENO approximation procedure on triangular meshes.

2.1. The framework

We consider the governing Eq. (1.1) solved on a domain Ω , which is partitioned into non-overlapping triangles denoted by $\Delta_l, l = 1 \dots N$. We define $|\Delta_l|$ and (x_l, y_l) as the area and the barycenter of the triangle Δ_l respectively. For every node i , we define the angular sectors which share the same node i as T_0, \dots, T_{k_i} in the anticlockwise order. $\vec{n}_{l+\frac{1}{2}}$ is the unit vector of the half-line $D_{l+\frac{1}{2}} = T_l \cap T_{l+1}$, and θ_l is the inner angle of the sector $T_l, 0 \leq l \leq k_i$. See Fig. 1.

We define ϕ_i as the numerical approximation to the viscosity solution of (1.1) at node i , and we denote (u_i, v_i) as the numerical approximation to the spatial derivatives $\nabla\phi$ at node i . By taking spatial derivatives on both sides of the Eq. (1.1), we obtain the

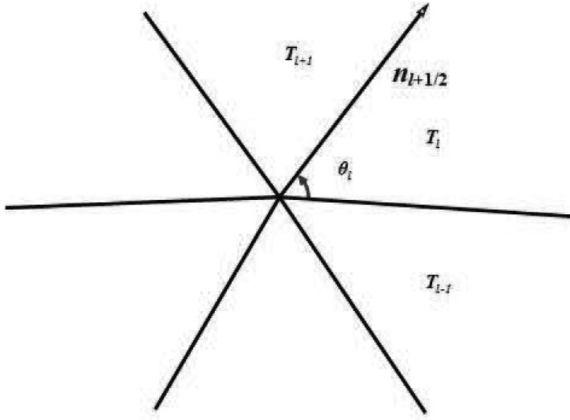


Fig. 1. Node i and its sectors.

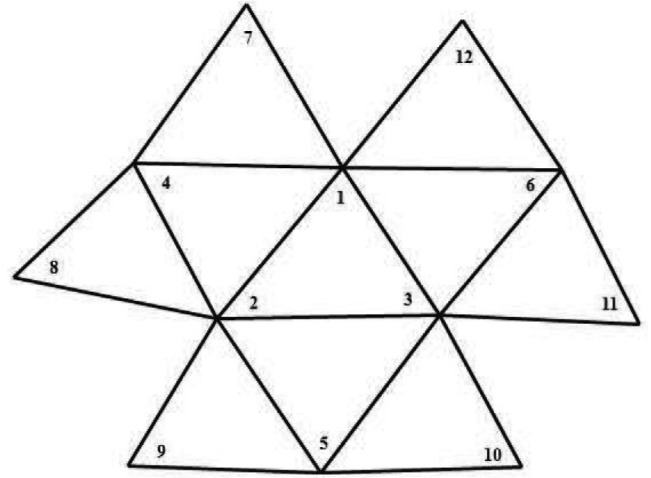


Fig. 2. The nodes used for the big stencil.

following system for the approximation:

$$\begin{cases} \frac{d\phi_i}{dt} = -H(\nabla\phi_i) \\ \frac{du_i}{dt} = -H_1(\nabla\phi_i)u_{x_i} - H_2(\nabla\phi_i)v_{x_i} \\ \frac{dv_i}{dt} = -H_1(\nabla\phi_i)u_{y_i} - H_2(\nabla\phi_i)v_{y_i} \end{cases} \quad (2.1)$$

where $H_1(u, v) = \frac{\partial H}{\partial u}$ and $H_2(u, v) = \frac{\partial H}{\partial v}$. As $u_y = v_x$, we can also rewrite the system as the following:

$$\begin{cases} \frac{d\phi_i}{dt} = -H(\nabla\phi_i) \\ \frac{du_i}{dt} = -H_1(\nabla\phi_i)u_{x_i} - H_2(\nabla\phi_i)u_{y_i} \\ \frac{dv_i}{dt} = -H_1(\nabla\phi_i)v_{x_i} - H_2(\nabla\phi_i)v_{y_i} \end{cases} \quad (2.2)$$

We introduce approximations to the right hand sides of (2.2) using hats, and obtain the semi-discrete scheme as

$$\begin{cases} \frac{d\phi_i}{dt} = -\widehat{H}_i \\ \frac{du_i}{dt} = -H_1 u_{x_i} + \widehat{H}_2 u_{y_i} \\ \frac{dv_i}{dt} = -H_1 v_{x_i} + \widehat{H}_2 v_{y_i} \end{cases} \quad (2.3)$$

Here, \widehat{H}_i is the LF type monotone Hamiltonian for triangular meshes introduced by Abgrall [1]. It is an important building block for our schemes and is defined as follows:

$$\widehat{H}_i = H\left(\frac{\sum_{l=0}^{k_i} \theta_l(\nabla\phi_l)}{2\pi}\right) - \frac{\alpha}{\pi} \sum_{l=0}^{k_i} \beta_{l+\frac{1}{2}} \left(\frac{\nabla\phi_l + \nabla\phi_{l+1}}{2}\right) \cdot \vec{n}_{l+\frac{1}{2}} \quad (2.4)$$

where $\beta_{l+\frac{1}{2}} = \tan\left(\frac{\theta_l}{2}\right) + \tan\left(\frac{\theta_{l+1}}{2}\right)$, $\alpha =$

$$\max \left\{ \max_{\substack{A \leq u \leq B \\ C \leq v \leq D}} |H_1(u, v)|, \max_{\substack{A \leq u \leq B \\ C \leq v \leq D}} |H_2(u, v)| \right\}.$$

Here $\nabla\phi_l$ is the numerical approximation to $\nabla\phi$ at node l in sector T_l . $[A, B]$ is the range of the value ϕ_{x_i} , and $[C, D]$ is the range of the value ϕ_{y_i} , over $0 \leq l \leq k_i$ for the local LF Hamiltonian, and over $0 \leq l \leq k_i$ and $0 \leq i \leq N$ for the global LF Hamiltonian. In this paper, we use the global LF Hamiltonian.

We define $H_1 u_{x_i} + H_2 u_{y_i}$ in a similar way:

$$\begin{aligned} H_1 u_{x_i} + \widehat{H}_2 u_{y_i} &= H_1 \left(\frac{\sum_{l=0}^{k_i} \theta_l(\nabla\phi_l)}{2\pi} \right) \frac{\sum_{l=0}^{k_i} \theta_l(u_{x_i})}{2\pi} + H_2 \left(\frac{\sum_{l=0}^{k_i} \theta_l(\nabla\phi_l)}{2\pi} \right) \frac{\sum_{l=0}^{k_i} \theta_l(u_{y_i})}{2\pi} \\ &\quad - \frac{\alpha}{\pi} \sum_{l=0}^{k_i} \beta_{l+\frac{1}{2}} \left(\frac{\nabla u_l + \nabla u_{l+1}}{2} \right) \cdot \vec{n}_{l+\frac{1}{2}} \end{aligned} \quad (2.5)$$

in which the definition of the parameters is the same as before. Similarly, we can define $H_1 v_{x_i} + H_2 v_{y_i}$.

After we complete the spatial discretization, we can rewrite the semi-discrete scheme as $U_t = \mathcal{L}(U)$, where \mathcal{L} denotes the operator of the spatial discretization. As to the time derivative, we can use the third-order total variation diminishing (TVD) Runge–Kutta time discretization [32] to solve the semi-discrete form (2.3):

$$\begin{cases} U^{(1)} = U^n + \Delta t \mathcal{L}(U^n) \\ U^{(2)} = \frac{3}{4}U^n + \frac{1}{4}(U^{(1)} + \Delta t \mathcal{L}(U^{(1)})) \\ U^{n+1} = \frac{1}{3}U^n + \frac{2}{3}(U^{(2)} + \Delta t \mathcal{L}(U^{(2)})) \end{cases} \quad (2.6)$$

Now, we have completed the description of our scheme, except for the approximation of $\nabla\phi$, ∇u and ∇v , which would need to be obtained by the HWENO method to maintain high order accuracy in the smooth regions and sharp and non-oscillatory performance when derivative discontinuities appear. In the following section, we will describe the detailed procedure of the HWENO approximation method, using the fourth order version as an example.

2.2. Fourth order HWENO approximation

In this section, we follow similar ideas as in Zhu and Qiu [44,45] to construct fourth order HWENO approximation to the nodes 1,2,3 of the target cell Δ_0 , as shown in Fig. 2, for $\nabla\phi$, ∇u and ∇v .

2.2.1. Fourth order HWENO approximation for $\nabla\phi$

Step 1. In order to get a fourth order approximation to $\nabla\phi$, we would like to first construct a fourth degree interpolation or least square polynomial $p_0(x, y)$. Let (x_0, y_0) be the barycenter of the target cell Δ_0 . We define $\xi = \frac{(x-x_0)}{\sqrt{|\Delta_0|}}$, $\eta = \frac{(y-y_0)}{\sqrt{|\Delta_0|}}$. Then, we can write the polynomial $p_0(x, y)$ as

$$p_0(x, y) = \sum_{j=0}^4 \sum_{s+r=j} a_{rj} \xi^s \eta^r \quad (2.7)$$

It has 15 degrees of freedom, so we would need to use at least 15 conditions from the nodes.

Step 2. Given a big stencil $S_0 = \{1, 2, \dots, 12\}$ as shown in Fig. 2, we would like to obtain the fourth degree polynomial $p_0(x, y)$ such that

$$p_0(x_l, y_l) = \phi_l \quad l = 1, 2, 3 \quad (2.8)$$

and

$$p_0 = \operatorname{argmin} \left(\sum_l (p(x_l, y_l) - \phi_l)^2 + |\Delta_0| \sum_l \left(\frac{\partial}{\partial x} p(x_l, y_l) - u_l \right)^2 + |\Delta_0| \sum_l \left(\frac{\partial}{\partial y} p(x_l, y_l) - v_l \right)^2 \right) \quad (2.9)$$

where $l = 4, 5, \dots, 12$, and the minimum is taken over all polynomials p of degree at most 4. The $|\Delta_0|$ factor in front of the derivative terms is introduced to get the correct scaling with the mesh size.

We can rewrite the above problem in the following matrix form:

$$\begin{aligned} \mathbf{a} &= \operatorname{argmin}_{\mathbf{x}} \|\mathbf{B}\mathbf{x} - \mathbf{f}\|_2^2 \\ \text{s.t. } \mathbf{A}\mathbf{a} &= \mathbf{b} \end{aligned} \quad (2.10)$$

where the \mathbf{A} is a 3×15 matrix coming from the coefficients of the Eq. (2.8), \mathbf{b} is a 3×1 matrix coming from right hand side of the Eq. (2.8), \mathbf{B} is a 27×15 matrix coming from the Eq. (2.9) and \mathbf{f} is 27×1 matrix coming from the information of $\phi_l, \sqrt{|\Delta_0|}u_l, \sqrt{|\Delta_0|}v_l$ in the Eq. (2.9). Here \mathbf{a} is the vector of coefficients of $p_0(x, y)$ defined in (2.7).

In order to solve this constraint least square problem, we can define the following Lagrange function:

$$L(\mathbf{x}, \lambda) = c \left(\frac{1}{2} \mathbf{x}^T \mathbf{B}^T \mathbf{B} \mathbf{x} - \mathbf{f}^T \mathbf{B} \mathbf{x} \right) - \lambda^T (\mathbf{A} \mathbf{x} - \mathbf{b})$$

where the second part of the Lagrange function comes from the constraints and the first part comes from the objective function:

$$\begin{aligned} \|\mathbf{B}\mathbf{x} - \mathbf{f}\|_2^2 &= (\mathbf{B}\mathbf{x} - \mathbf{f})^T (\mathbf{B}\mathbf{x} - \mathbf{f}) \\ &= \mathbf{x}^T \mathbf{B}^T \mathbf{B} \mathbf{x} - 2\mathbf{f}^T \mathbf{B} \mathbf{x} + \mathbf{f}^T \mathbf{f} \end{aligned} \quad (2.11)$$

Here, c is a parameter used to reduce the condition number numerically, and it is taken as

$$c = \frac{1}{27 \max_{i,j} |\mathbf{B}(i, j)|},$$

where 27 is the number of the rows in the matrix \mathbf{B} and $\mathbf{B}(i, j)$ refers to the elements of \mathbf{B} .

By requiring $L_{\mathbf{x}}(\mathbf{x}, \lambda) = 0$ and $L_{\lambda}(\mathbf{x}, \lambda) = 0$, we have the following linear system:

$$\begin{cases} c\mathbf{B}^T \mathbf{B} \mathbf{x} - \mathbf{A}^T \lambda = c\mathbf{B}^T \mathbf{f} \\ \mathbf{A} \mathbf{x} = \mathbf{b} \end{cases} \quad (2.12)$$

We can rewrite the linear system in the matrix form:

$$\begin{pmatrix} c\mathbf{B}^T \mathbf{B} & -\mathbf{A}^T \\ -\mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \lambda \end{pmatrix} = \begin{pmatrix} c\mathbf{B}^T \mathbf{f} \\ -\mathbf{b} \end{pmatrix} \quad (2.13)$$

Solving this linear system, we get $\mathbf{a} = \mathbf{x}$ as the coefficients of the fourth degree polynomial p_0 , for which $\nabla p_0(x, y)$ will be a fourth order approximation to $\nabla \phi(x, y)$.

Step 3. We also need to construct four second degree polynomials depending on four different small stencils in order to form the HWENO approximation. We select these small stencils as $S_1 = \{1, 2, 3, 4, 5, 6\}$, $S_2 = \{1, 2, 3, 4, 7, 8\}$, $S_3 = \{1, 2, 3, 5, 9, 10\}$ and $S_4 = \{1, 2, 3, 6, 11, 12\}$. We would like to construct the interpolation polynomials $p_k(x, y)$

$$p_k(x, y) = \sum_{j=0}^2 \sum_{s+r=j} a_{rj}^k \xi^s \eta^r, \quad k = 1, 2, 3, 4$$

such that

$$p_k(x_l, y_l) = \phi_l, \quad l \in S_k$$

Then, each $\nabla p_k(x, y)$ is a second order approximation to $\nabla \phi(x, y)$.

Step 4. We compute the smoothness indicators $\beta_l, l = 0, 1, \dots, 4$, which measures the smoothness of the polynomials p_l on the target cell Δ_0 . The smaller the indicators are, the smoother the polynomials are. We use a similar definition as in [11,38]:

$$\beta_k = \sum_{|l| \geq 2} \int_{\Delta_0} |\Delta_0|^{l-2} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} p_k(x, y) \right)^2 dx dy, \quad k = 0, 1, \dots, 4$$

where $l = (l_1, l_2)$ and $|l| = l_1 + l_2$.

Step 5. We set the positive linear weights as $\gamma_k = 0.2$ for $k = 0, 1, \dots, 4$. Then we can rewrite $p_0(x, y)$ as follows:

$$p_0(x, y) = \gamma_0 \left(\frac{1}{\gamma_0} p_0(x, y) - \sum_{k=1}^4 \frac{\gamma_k}{\gamma_0} p_k(x, y) \right) + \sum_{k=1}^4 \gamma_k p_k(x, y).$$

Step 6. We compute the nonlinear weights defined as follows:

$$w_k = \frac{\bar{w}_k}{\sum_l \bar{w}_l}, \quad \bar{w}_l = \gamma_l \left(1 + \frac{\tau}{\varepsilon + \beta_l} \right)^2, \quad l = 0, 1, \dots, 4,$$

where the parameter τ is defined as follows:

$$\tau = \sum_{k=1}^4 (\beta_0 - \beta_k)^2$$

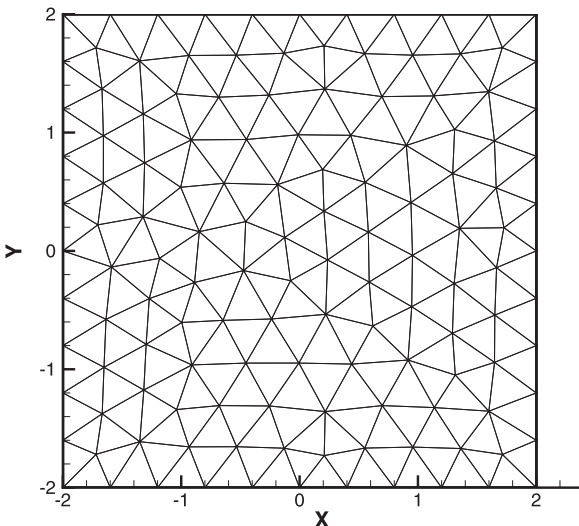


Fig. 3. The sample mesh with the number of nodes $N = 134$.

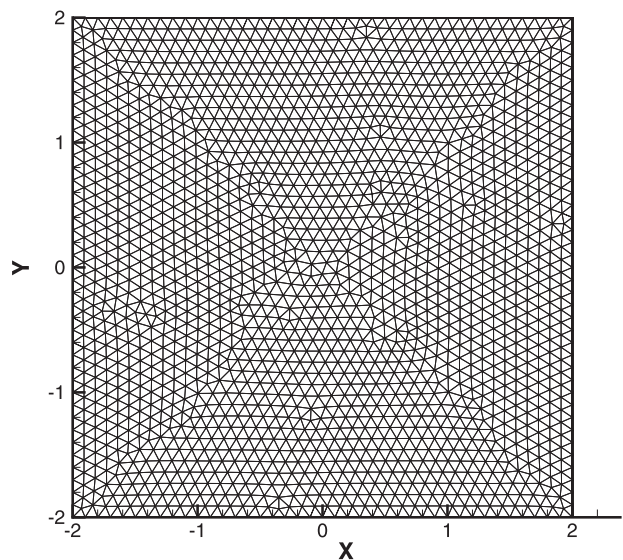


Fig. 4. The sample mesh for Examples 4 and 5 with the number of nodes $N = 1876$.

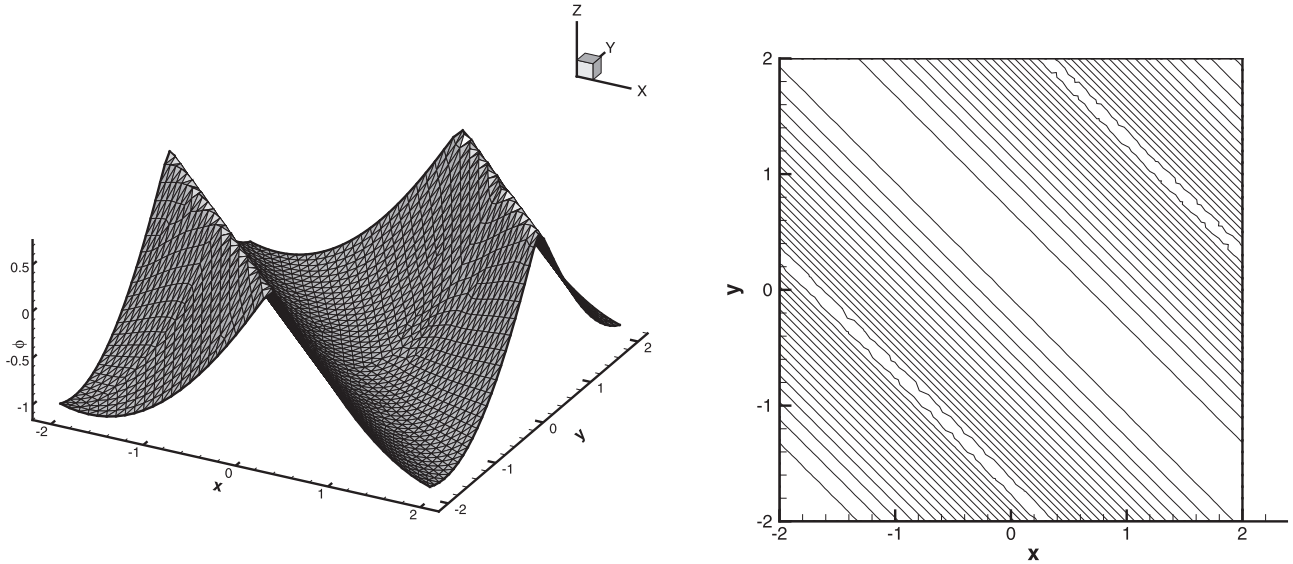


Fig. 5. Burgers equation. $T = 1.5/\pi^2$. Left: surface of the solution; right: contour plot of the solution.

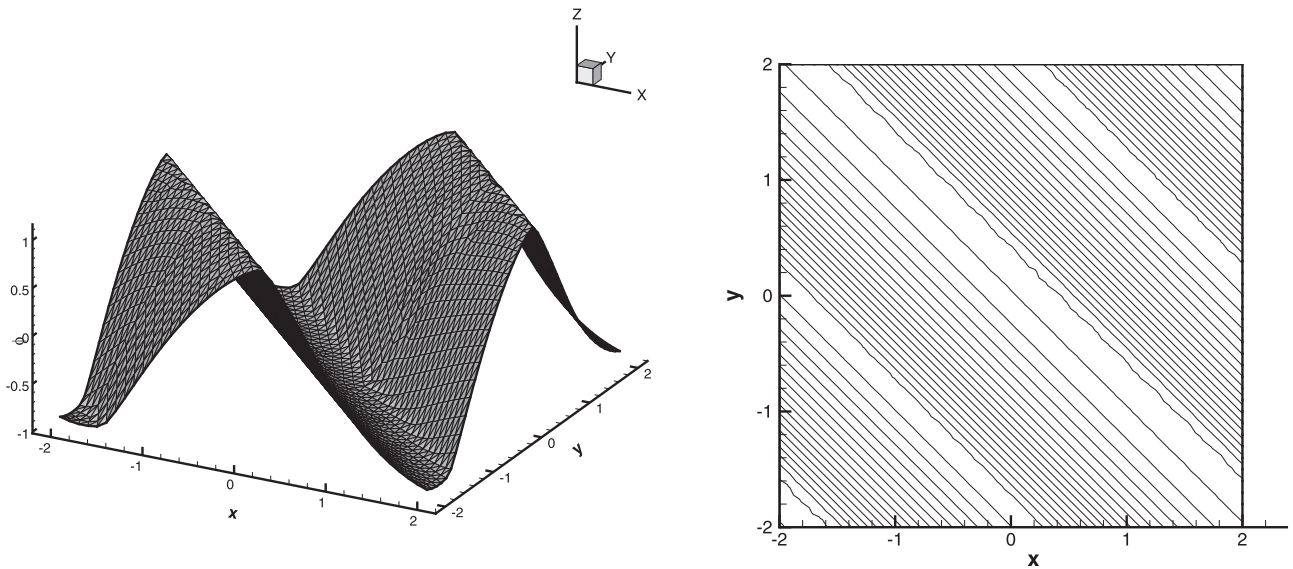


Fig. 6. Nonlinear equation. $T = 1.5/\pi^2$. Left: surface of the solution; right: contour plot of the solution.

Here, we take $\varepsilon = 10^{-6}$ in order for avoiding the denominator to become zero.

Step 7. The final HWENO approximation is given by:

$$p(x, y) = w_0 \left(\frac{1}{\gamma_0} p_0(x, y) - \sum_{k=1}^4 \frac{\gamma_k}{\gamma_0} p_k(x, y) \right) + \sum_{k=1}^4 w_k p_k(x, y).$$

Remark 1. We would like to verify that, in smooth regions, the approximation to $\nabla\phi$ can achieve fourth order accuracy. First, by Taylor expansions, we have the following estimate for the β 's in smooth regions:

$$\beta_0 = \left(\sum_{|l|=2} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} p_0(x, y) \Big|_{(x_0, y_0)} \right)^2 \right) |\Delta_0| (1 + O(|\Delta_0|))$$

and

$$\beta_k = \left(\sum_{|l|=2} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} p_k(x, y) \Big|_{(x_0, y_0)} \right)^2 \right) |\Delta_0| (1 + O(\sqrt{|\Delta_0|})),$$

$k = 1, 2, 3, 4.$

Also by Taylor expansions, we have

$$\frac{\tau}{\varepsilon + \beta_l} = O(|\Delta_0|^2), \quad l = 0, 1, \dots, 4$$

which leads to

$$w_l = \gamma_l + O(|\Delta_0|^2), \quad l = 0, 1, \dots, 4.$$

Therefore, we have

$$\begin{aligned} & \nabla p(x, y) - \nabla\phi(x, y) \\ &= w_0 \left(\frac{1}{\gamma_0} \nabla p_0(x, y) - \sum_{k=1}^4 \frac{\gamma_k}{\gamma_0} \nabla p_k(x, y) \right) \\ & \quad + \sum_{k=1}^4 w_k \nabla p_k(x, y) - \nabla\phi(x, y) \\ &= (\gamma_0 + w_0 - \gamma_0) \left(\frac{1}{\gamma_0} \nabla p_0(x, y) - \sum_{k=1}^4 \frac{\gamma_k}{\gamma_0} \nabla p_k(x, y) - \nabla\phi(x, y) \right) \end{aligned}$$

$$\begin{aligned}
 & + \sum_{k=1}^4 (\gamma_k + w_k - \gamma_k) (\nabla p_k(x, y) - \nabla \phi(x, y)) \\
 & = \nabla p_0(x, y) - \nabla \phi(x, y) + (w_0 - \gamma_0) \left(\frac{1}{\gamma_0} \nabla p_1(x, y) \right. \\
 & \quad \left. - \sum_{k=1}^4 \frac{\gamma_k}{\gamma_0} \nabla p_k(x, y) - \nabla \phi(x, y) \right) \\
 & + \sum_{k=1}^4 (w_k - \gamma_k) (\nabla p_k(x, y) - \nabla \phi(x, y)) \\
 & = O(|\Delta_0|^2) + O(|\Delta_0|^2)O(|\Delta_0|) + O(|\Delta_0|^2)O(|\Delta_0|) \\
 & = O(|\Delta_0|^2) \tag{2.14}
 \end{aligned}$$

which verifies fourth order accuracy for $\nabla \phi$.

Remark 2. It is well known that, for general triangular meshes, if we use exactly 15 pieces of information to determine the fourth degree polynomial, the interpolation problem may not be well defined, as the linear system could be ill-conditioned or even singular. It is therefore more prudent to use more than 15 pieces of information and the least square procedure to determine the fourth degree polynomial. Also, it appears to be necessary to require the polynomial to interpolate the function ϕ at the three nodes 1,2,3 of the target cell Δ_0 in order to ensure stability. We do observe linear instability in our numerical experiments when we do not require exact interpolation at these three nodes and treat them the same way as the other conditions through the least square procedure.

Remark 3. In fact, the nodes used in the stencils may not be different. For example, the node 8 and node 9 as shown in Fig. 2 may be the same in some triangulations. Because of the many slacks in the least square procedure, our method still works in such situation.

2.2.2. Fourth order HWENO approximation for ∇u and ∇v

The procedure to obtain fourth order HWENO approximations for ∇u and ∇v is similar to the one described in the previous section for approximating $\nabla \phi$.

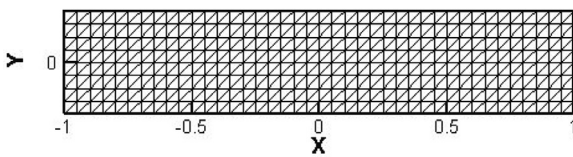


Fig. 7. The uniform mesh for Example 6 with the number of nodes $N = 369$.

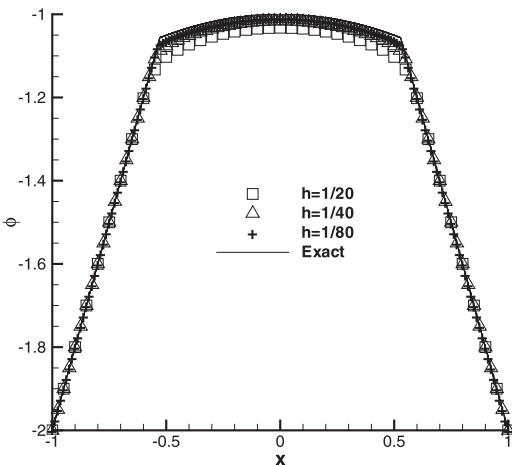


Fig. 8. Convergence study of the nonconvex and nonconcave case.

Step 1. In order to get a fourth order approximation to ∇u , we would need to construct a fifth degree polynomial $p_0(x, y)$ as

$$p_0(x, y) = \sum_{j=0}^5 \sum_{s+r=j} a_{rj} \xi^s \eta^r \tag{2.15}$$

It has 21 degrees of freedom, so we would need to use at least 21 conditions at the nodes.

Step 2. Given the same big stencil $S_0 = \{1, 2, \dots, 12\}$ as shown in Fig. 2, we would like to obtain the fifth degree polynomial $p_0(x, y)$ such that

$$\begin{aligned}
 p_0(x_l, y_l) &= \phi_l \quad l = 1, 2, 3 \\
 p_{0x}(x_l, y_l) &= u_l \quad l = 1, 2, 3 \\
 p_{0y}(x_l, y_l) &= v_l \quad l = 1, 2, 3
 \end{aligned} \tag{2.16}$$

and

$$\begin{aligned}
 p_0 = \operatorname{argmin} & \left(\sum_l (p(x_l, y_l) - \phi_l)^2 + |\Delta_0| \sum_l \left(\frac{\partial}{\partial x} p(x_l, y_l) - u_l \right)^2 \right. \\
 & \left. + |\Delta_0| \sum_l \left(\frac{\partial}{\partial y} p(x_l, y_l) - v_l \right)^2 \right) \tag{2.17}
 \end{aligned}$$

where $l = 4, 5, \dots, 12$, and the minimum is taken over all polynomials p of degree at most 5.

Again, we can rewrite the above problem into a matrix form as before, facilitating its implementation. We skip the details here to save space. The fifth degree polynomial p_0 thus obtained would have the following properties: $p_{0xx}(x, y)$ would be a fourth order approximation to $\phi_{xx}(x, y)$, $p_{0xy}(x, y)$ would be a fourth order approximation to $\phi_{xy}(x, y)$, and $p_{0yy}(x, y)$ would be a fourth order approximation to $\phi_{yy}(x, y)$. We have therefore obtained fourth order approximations to ∇u and ∇v .

Step 3. We also need to construct four third degree polynomials depending on four different small stencils in order to form the HWENO approximation. We select these small stencils as $S_1 = \{1, 2, 3, 4, 5, 6\}$, $S_2 = \{1, 2, 3, 4, 7, 8\}$, $S_3 = \{1, 2, 3, 5, 9, 10\}$ and $S_4 = \{1, 2, 3, 6, 11, 12\}$. We would like to construct the least square polynomials $p_k(x, y)$

$$p_k(x, y) = \sum_{j=0}^3 \sum_{s+r=j} a_{rj}^k \xi^s \eta^r, \quad k = 1, 2, 3, 4$$

such that

$$p_k(x_{l_1}, y_{l_1}) = \phi_{l_1}$$

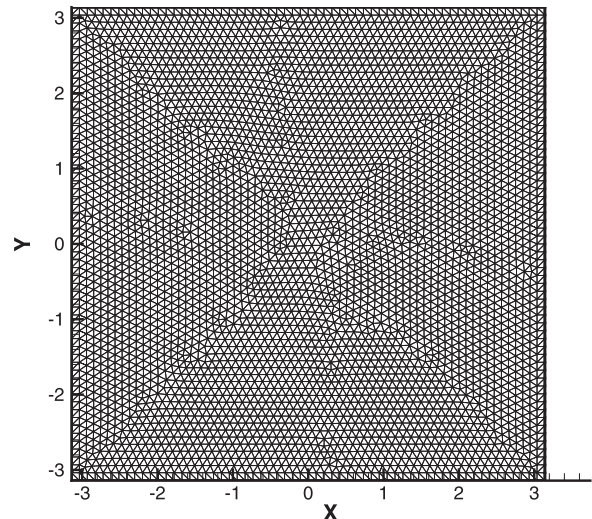


Fig. 9. The mesh for Example 7 with the number of nodes $N = 4138$.

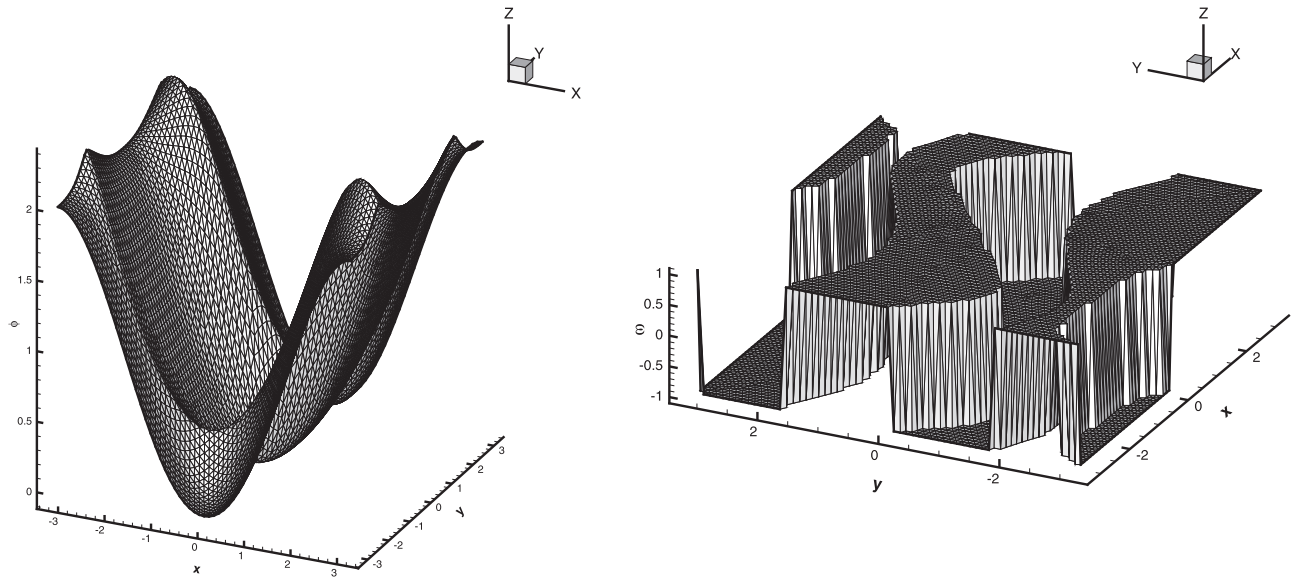


Fig. 10. Control problem. Left: surface of the solution; Right: the optimal control $\omega = \text{sign}(\phi_y)$.

and

$$p_k = \text{argmin} \left(\sum_{l_2} (p(x_{l_2}, y_{l_2}) - \phi_{l_2})^2 + |\Delta_0| \sum_{l_1} \left(\frac{\partial}{\partial x} p(x_{l_1}, y_{l_1}) - u_{l_1} \right)^2 + |\Delta_0| \sum_{l_1} \left(\frac{\partial}{\partial y} p(x_{l_1}, y_{l_1}) - v_{l_1} \right)^2 \right)$$

where $l_1 = 1, 2, 3$ and $l_2 \in S_k \setminus \{1, 2, 3\}$. The minimum is taken over all the polynomials p_k of degree at most 3. Then, $p_{kxx}(x, y)$ is a second order approximation to $\phi_{xx}(x, y)$, $p_{kxy}(x, y)$ is a second order approximation to $\phi_{xy}(x, y)$ and $p_{kyy}(x, y)$ is a second order approximation to $\phi_{yy}(x, y)$, hence we have obtained four second order approximations to ∇u and ∇v .

Step 4. We compute the smoothness indicators $\beta_l, l = 0, 1, \dots, 4$ as follows:

$$\beta_k = \sum_{|l| \geq 3} \int_{\Delta_0} |\Delta_0|^{||l|-3} \left(\frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} p_k(x, y) \right)^2 dx dy, \quad k = 0, 1, \dots, 4$$

where $l = (l_1, l_2)$ and $|l| = l_1 + l_2$.

The remaining **Step 5** through **Step 7** are identical to the ones described in the previous subsection for approximating ϕ . Once the final fifth degree HWENO approximation polynomial $p(x, y)$ is obtained in Step 7, we can take its second derivatives to obtain approximations to ∇u and ∇v . By Taylor expansions, we can verify that the HWENO approximations are indeed fourth order accurate in smooth regions.

3. Numerical results

In this section, we present numerical experiments using the fourth order HWENO method on triangular meshes. The time step is taken as

$$\Delta t = \frac{1}{2} \frac{\sqrt{|\Delta|_{\min}}}{\alpha}$$

where $|\Delta|_{\min} = \min_i |\Delta_i|$, α is the parameter in the LF type monotone Hamiltonian. The only exception is for the accuracy test which needs smaller time step to guarantee that the spatial error dominates. For the sake of evaluating the performance of different choices of the linear weights, we set three different types of linear weights for the accuracy tests: (1) $\gamma_0 = \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 =$

0.2; (2) $\gamma_0 = 0.96$ and $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.01$; (3) $\gamma_0 = 0.04$ and $\gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.24$. We also present the results of the linear scheme, which use the linear weights instead of the nonlinear weights, to make the comparison with our HWENO schemes. All the numerical tests are nondimensionalized.

Example 1. We solve the two dimensional linear scalar equation

$$\phi_t + \phi_x + \phi_y = 0, \quad -2 \leq x, y \leq 2$$

with the initial datum $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$ and periodic boundary condition. A sample mesh with boundary triangle size $h = 0.4$ is shown in Fig. 3. We compute the result up to $t = 2$ to test the accuracy of ϕ and $\nabla \phi$ of both the linear scheme and the HWENO schemes. The errors and numerical orders of accuracy are shown in Tables 1 and 2, where HWENO(i) refers to the (i)th choice of the linear weights indicated above, for $i = 1, 2, 3$. We can see that the HWENO schemes can achieve its designed order of accuracy, at least in L_1 and L_2 norms. We also notice that the different choices of the linear weights do not change the results significantly. We list the CPU time of these schemes as well, see Table 3.

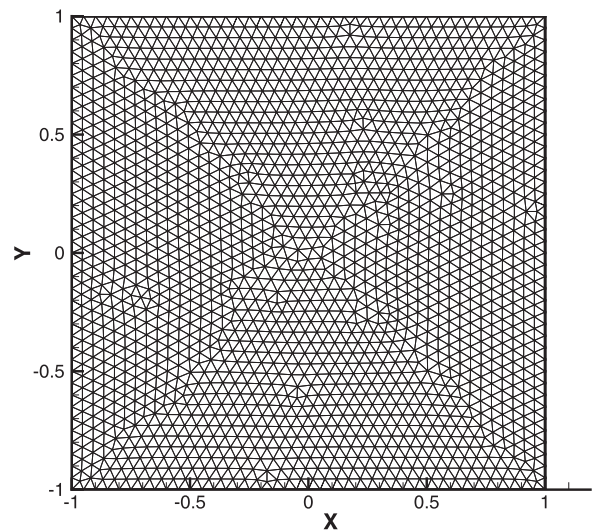


Fig. 11. The mesh for Example 8 with the number of nodes $N = 1876$.

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

N	Linear: ϕ				HWENO(1): ϕ			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	3.60E-02		1.76E-02		4.93E-01		3.15E-01	
492	1.78E-03	4.34	8.57E-04	4.36	7.02E-02	2.81	2.97E-02	3.40
1876	1.01E-04	4.15	4.84E-05	4.15	1.32E-03	5.73	2.67E-04	6.80
7337	6.18E-06	4.03	2.98E-06	4.02	2.29E-05	5.85	4.19E-06	6.00
29204	3.57E-07	4.11	1.81E-07	4.04	5.32E-07	5.43	1.86E-07	4.49

N	HWENO(2): ϕ				HWENO(3): ϕ			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	1.82E-01		1.14E-01		5.05E-01		3.23E-01	
492	6.53E-03	4.80	1.91E-03	5.90	7.51E-02	2.75	3.30E-02	3.29
1876	1.71E-04	5.25	5.46E-05	5.13	1.51E-03	5.64	3.16E-04	6.71
7337	6.93E-06	4.63	3.02E-06	4.18	2.64E-05	5.84	4.57E-06	6.11
29204	3.61E-07	4.26	1.82E-07	4.06	5.68E-07	5.54	1.87E-07	4.61

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

N	Linear: $\nabla\phi$				HWENO(1): $\nabla\phi$			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	6.47E-02		2.75E-02		8.67E-01		4.41E-01	
492	3.70E-03	4.13	1.43E-03	4.27	2.15E-01	2.01	1.12E-01	1.97
1876	2.38E-04	3.96	7.93E-05	4.17	9.87E-03	4.44	1.81E-03	5.95
7337	1.76E-05	3.76	5.03E-06	3.98	2.22E-04	5.47	2.68E-05	6.08
29204	1.43E-06	3.62	3.03E-07	4.05	4.59E-06	5.60	4.99E-07	5.75

N	HWENO(2): $\nabla\phi$				HWENO(3): $\nabla\phi$			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	3.98E-01		1.49E-01		8.84E-01		4.53E-01	
492	2.55E-02	3.96	7.03E-03	4.40	2.26E-01	1.97	1.17E-01	1.96
1876	7.19E-04	5.15	1.45E-04	5.60	1.16E-02	4.29	2.16E-03	5.75
7337	2.52E-05	4.83	5.67E-06	4.68	2.65E-04	5.44	3.14E-05	6.11
29204	1.43E-06	4.14	3.08E-07	4.20	5.30E-06	5.65	5.43E-07	5.85

Table 3CPU time (in seconds) for $\phi_t + \phi_x + \phi_y = 0$, $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$. Periodic boundary conditions. $t = 2$.

N	134	492	1876	7337	29204
Linear scheme	0.72	8.17	78.82	914.32	9039.87
Average time for three HWENO schemes	1.03	11.75	113.20	1328.17	13211.49

Example 2. We solve the two dimensional Burgers equation

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

with the initial datum $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$ and periodic boundary condition. The coarsest mesh with $h = 0.4$ is shown in Fig. 3. We compute the result up to $t = \frac{0.5}{\pi^2}$. At this time, the solution is still smooth. Both the linear scheme and the HWENO schemes are tested in this case and the errors and the orders of accuracy of ϕ and $\nabla\phi$ are listed in Tables 4 and 5. We can see that the HWENO schemes can reach its designed order of accuracy. We also list the CPU time of these schemes, see Table 6.

Example 3. We solve the two dimensional nonlinear equation

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

with the initial datum $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$ and periodic boundary condition. The coarsest mesh with $h = 0.4$ is shown in Fig. 3. We compute the result up to $t = \frac{0.5}{\pi^2}$. At this time, the solution is still smooth. Again, we test the accuracy of ϕ and $\nabla\phi$ of both the linear scheme and the HWENO scheme. From Tables 7 and 8, we can see that the HWENO schemes can reach the expected order of accuracy. We also list the CPU time of these schemes, see Table 9.

For the following examples, we will use the first choice of the linear weights, namely $\gamma_0 = \gamma_1 = \gamma_2 = \gamma_3 = \gamma_4 = 0.2$.

Example 4. We solve the two dimensional Burgers equation

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

with the initial datum $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$ and periodic boundary condition, and compute the result up to $t = 1.5/\pi^2$. At this time, the solution is not smooth any more. We compute the HWENO scheme with the mesh shown in Fig. 4 and plot the results in Fig. 5. From the figure, we can see that the HWENO scheme can achieve good resolution in this case.

Example 5. We solve the two dimensional equation

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

with the initial datum $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$ and periodic boundary condition, see [25]. The mesh is shown in Fig. 4. We compute the result up to $t = 1.5/\pi^2$ when the solution is not smooth any more. From Fig. 6, we can see that the scheme can achieve high resolution in this example.

Example 6. We solve the problem

$$\phi_t + \frac{1}{4}(\phi_x^2 - 1)(\phi_x^2 - 4) = 0$$

Table 4

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

N	Linear: ϕ				HWENO(1): ϕ			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	2.42E-02		3.55E-03		1.07E-01		3.11E-02	
492	2.60E-03	3.22	2.84E-04	3.65	1.70E-02	2.65	3.75E-03	3.05
1876	1.87E-04	3.80	1.26E-05	4.50	4.09E-04	5.38	5.74E-05	6.03
7337	1.39E-05	3.75	6.10E-07	4.36	1.18E-05	5.11	7.99E-07	6.17
29204	4.96E-07	4.80	2.37E-08	4.69	4.80E-07	4.62	2.46E-08	5.02

N	HWENO(2): ϕ				HWENO(3): ϕ			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	5.75E-02		1.42E-02		1.10E-01		3.23E-02	
492	4.74E-03	3.60	4.87E-04	4.86	1.83E-02	2.59	4.15E-03	2.96
1876	1.81E-04	4.71	1.17E-05	5.37	4.74E-04	5.27	6.76E-05	5.94
7337	1.36E-05	3.73	5.97E-07	4.30	1.16E-05	5.36	8.67E-07	6.29
29204	4.96E-07	4.78	2.36E-08	4.66	4.77E-07	4.60	2.51E-08	5.11

Table 5

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

N	Linear: $\nabla\phi$				HWENO(1): $\nabla\phi$			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	7.87E-02		1.44E-02		1.95E-01		5.52E-02	
492	1.47E-02	2.42	1.65E-03	3.12	6.35E-02	1.62	9.32E-03	2.57
1876	1.54E-03	3.25	9.69E-05	4.09	6.00E-03	3.40	5.58E-04	4.06
7337	1.95E-04	2.98	5.70E-06	4.09	7.70E-04	2.96	1.95E-05	4.84
29,204	1.19E-05	4.04	2.22E-07	4.68	1.69E-05	5.51	3.01E-07	6.02

N	HWENO(2): $\nabla\phi$				HWENO(3): $\nabla\phi$			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	1.48E-01		3.15E-02		1.96E-01		5.66E-02	
492	2.68E-02	2.47	2.58E-03	3.61	6.68E-02	1.55	1.01E-02	2.48
1876	5.30E-03	2.34	1.78E-04	3.86	6.02E-03	3.47	6.17E-04	4.04
7337	3.02E-04	4.13	6.58E-06	4.76	7.91E-04	2.93	2.17E-05	4.83
29204	1.19E-05	4.67	2.24E-07	4.88	1.81E-05	5.45	3.21E-07	6.08

Table 6

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

N	134	492	1876	7337	29,204
Linear scheme	0.10	0.86	8.34	93.08	938.36
Average time for HWENO schemes	0.13	1.12	10.88	122.01	1093.53

with the initial datum $\phi(x, y, 0) = -2|x|$, see [38]. The periodic boundary condition is applied in the y -direction. We solve the problem in the domain $[-1, 1] \times [-0.2, 0.2]$ with the sample mesh shown in Fig. 7. This is a demanding test case, many schemes can not obtain satisfactory results, some of them may even fail to converge to the correct viscosity solution. We compute the results up to $t = 1$ with $h = \frac{1}{20}, \frac{1}{40}, \frac{1}{80}$, and plot the solution along the cut line $y = 0$. From Fig. 8, we can see that the HWENO scheme can converge to the correct viscosity solution with mesh refinement.

Example 7. We solve a problem from optimal control:

$$\phi_t + \sin(y)\phi_x + (\sin(x) + \text{sign}(\phi_y))\phi_y - \frac{1}{2}\sin^2(y) + \cos(x) - 1 = 0, \\ -\pi < x, y < \pi$$

with $\phi(x, y, 0) = 0$ and periodic boundary conditions, see [25]. Notice that this is a HJ equation with a Hamiltonian which also depends on x and y :

$$\phi_t + H(\phi_x, \phi_y, x, y) = 0$$

The scheme in this case is the same

$$\begin{cases} \frac{d\phi_i}{dt} = -\widehat{H}_i \\ \frac{du_i}{dt} = -H_1 u_{x_i} + \widehat{H}_2 u_{y_i} - \widehat{H}_{x_i} \\ \frac{dv_i}{dt} = -H_1 v_{x_i} + \widehat{H}_2 v_{y_i} - \widehat{H}_{y_i} \end{cases} \quad (3.1)$$

in which the definition of \widehat{H}_i , $H_1 u_{x_i} + \widehat{H}_2 u_{y_i}$, $H_1 v_{x_i} + \widehat{H}_2 v_{y_i}$, \widehat{H}_x and \widehat{H}_y are similar as before, just adding x_i and y_i inside the Hamiltonians. The mesh is shown in Fig. 9. The solution at $t = 1$ is shown in Fig. 10, and we can see that our scheme can obtain good result for this example.

Example 8. We solve the two dimensional Riemann problem:

$$\phi_t + \sin(\phi_x + \phi_y) = 0, \quad -1 \leq x, y \leq 1$$

with $\phi(x, y, 0) = \pi(|y| - |x|)$, see [25]. The mesh is shown in Fig. 11. We compute the solution up to $t = 1$. The solution is shown in Fig. 12. Again, we observe our scheme can achieve good result.

Example 9. We solve the level set equation

$$\phi_t + \text{sign}(\phi_0) \left(\sqrt{\phi_x^2 + \phi_y^2} - 1 \right) = 0, \quad \frac{1}{2} < \sqrt{x^2 + y^2} < 1 \quad (3.2)$$

Table 7
 $\phi_t - \cos(\phi_x + \phi_y + 1) = 0$, $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$. Periodic boundary conditions. $t = 0.5/\pi^2$.

N	Linear: ϕ				HWENO(1): ϕ			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	5.77E-03		1.80E-03		2.47E-02		7.73E-03	
492	1.57E-03	1.88	2.52E-04	2.84	3.63E-03	2.77	8.14E-04	3.25
1876	3.61E-04	2.12	1.92E-05	3.71	3.77E-04	3.27	2.39E-05	5.09
7337	4.88E-05	2.89	1.55E-06	3.64	4.80E-05	2.97	1.47E-06	4.02
29204	2.94E-06	4.05	7.02E-08	4.46	2.86E-06	4.07	6.76E-08	4.44
116012	1.17E-07	4.65	2.96E-09	4.56	1.17E-07	4.61	2.96E-09	4.51

N	HWENO(2): ϕ				HWENO(3): ϕ			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	1.75E-02		3.55E-03		2.48E-02		8.05E-03	
492	1.59E-03	3.46	2.70E-04	3.72	3.77E-03	2.72	8.91E-04	3.18
1876	3.61E-04	2.14	1.91E-05	3.82	3.80E-04	3.31	2.53E-05	5.14
7337	4.86E-05	2.89	1.53E-06	3.64	4.79E-05	2.99	1.47E-06	4.11
29204	2.94E-06	4.05	7.00E-08	4.45	2.84E-06	4.08	6.73E-08	4.45
116,012	1.17E-07	4.65	2.96E-09	4.56	1.17E-07	4.60	2.96E-09	4.51

Table 8
 $\phi_t - \cos(\phi_x + \phi_y + 1) = 0$, $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$. Periodic boundary conditions. $t = 0.5/\pi^2$.

N	Linear: $\nabla\phi$				HWENO(1): $\nabla\phi$			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	4.79E-02		7.15E-03		9.19E-02		1.74E-02	
492	1.74E-02	1.46	1.81E-03	1.98	2.84E-02	1.70	2.80E-03	2.63
1876	4.42E-03	1.97	2.42E-04	2.91	6.90E-03	2.04	4.38E-04	2.68
7337	7.71E-04	2.52	2.80E-05	3.11	1.70E-03	2.02	5.51E-05	2.99
29,204	7.69E-05	3.33	1.48E-06	4.24	2.26E-04	2.91	2.95E-06	4.23
116,012	5.58E-06	3.78	6.10E-08	4.60	7.14E-06	4.99	7.70E-08	5.26

N	HWENO(2): $\nabla\phi$				HWENO(3): $\nabla\phi$			
	L_∞ error	Order	L_1 error	Order	L_∞ error	Order	L_1 error	Order
134	7.59E-02		9.75E-03		9.25E-02		1.77E-02	
492	2.25E-02	1.75	1.96E-03	2.32	2.87E-02	1.69	2.88E-03	2.62
1876	5.25E-03	2.10	3.13E-04	2.64	6.98E-03	2.04	4.53E-04	2.67
7337	1.10E-03	2.25	3.37E-05	3.22	1.75E-03	2.00	5.72E-05	2.99
29204	1.08E-04	3.34	1.60E-06	4.40	2.36E-04	2.88	3.12E-06	4.20
116,012	5.59E-06	4.28	6.18E-08	4.69	8.22E-06	4.85	8.02E-08	5.28

Table 9
 CPU time (in seconds) for $\phi_t - \cos(\phi_x + \phi_y + 1) = 0$, $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x+y))$. Periodic boundary conditions. $t = 0.5/\pi^2$.

N	134	492	1876	7337	29,204	116012
Linear scheme	0.03	0.31	2.44	24.41	246.24	2554.30
Average time for three HWENO schemes	0.03	0.38	3.00	29.65	300.55	2955.74

with the initial datum $\phi(x, y, 0) = \phi_0(x, y)$. This problem comes from Sussman et al. [33]. The solution has the same zero level set as the initial condition ϕ_0 , and the steady state solution is the distance function to that zero level curve. In this example, the exact solution is the distance function to the inner circle of the domain. It is difficult to use rectangular meshes for this problem. Instead we use the triangle mesh shown in Fig. 13 left. We compute the problem to reach a steady state solution, using the exact solution of the steady state as the boundary condition. The numerical solution is shown in Fig. 13 right. We can see that the scheme can obtain good result for this test.

Example 10. We solve the two dimensional eikonal equation

$$\phi_t + \sqrt{\phi_x^2 + \phi_y^2} + 1 = 0, \quad 0 \leq x, y < 1$$

with the initial datum $\phi(x, y, 0) = \frac{1}{4}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1) - 1$. This problem comes from Jin and Xin [13]. We compute the solution up to $t = 0.6$ on the mesh shown in Fig. 14. The solution is shown in Fig. 15. High resolutions are observed with our scheme.

Example 11. We solve

$$\phi_t - (1 - \varepsilon K)\sqrt{\phi_x^2 + \phi_y^2} + 1 = 0, \quad 0 \leq x, y < 1$$

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, with the initial datum $\phi(x, y, 0) = 1 - \frac{1}{4}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1)$ and periodic boundary condition is used. This problem comes from Osher and Sethian [24].

When $\varepsilon = 0$, we can treat the equation with the same method as before. When $\varepsilon \neq 0$, we can rewrite the equation as follows:

$$\phi_t + H(\phi_x, \phi_y, \phi_{xx}, \phi_{xy}, \phi_{yy}) = 0.$$

Then, we have

$$\begin{cases} \frac{d\phi_t}{dt} = -\widehat{H}(\phi_x, \phi_y, \phi_{xx}, \phi_{xy}, \phi_{yy}) \\ \frac{d\phi_x}{dt} = -H_1\widehat{u}_x + H_2u_y - H_3u_{xx} - H_4u_{xy} - H_5u_{yy} \\ \frac{d\phi_y}{dt} = -H_1v_x + H_2u_y - H_3v_{xx} - H_4u_{xy} - H_5u_{yy} \end{cases} \quad (3.3)$$

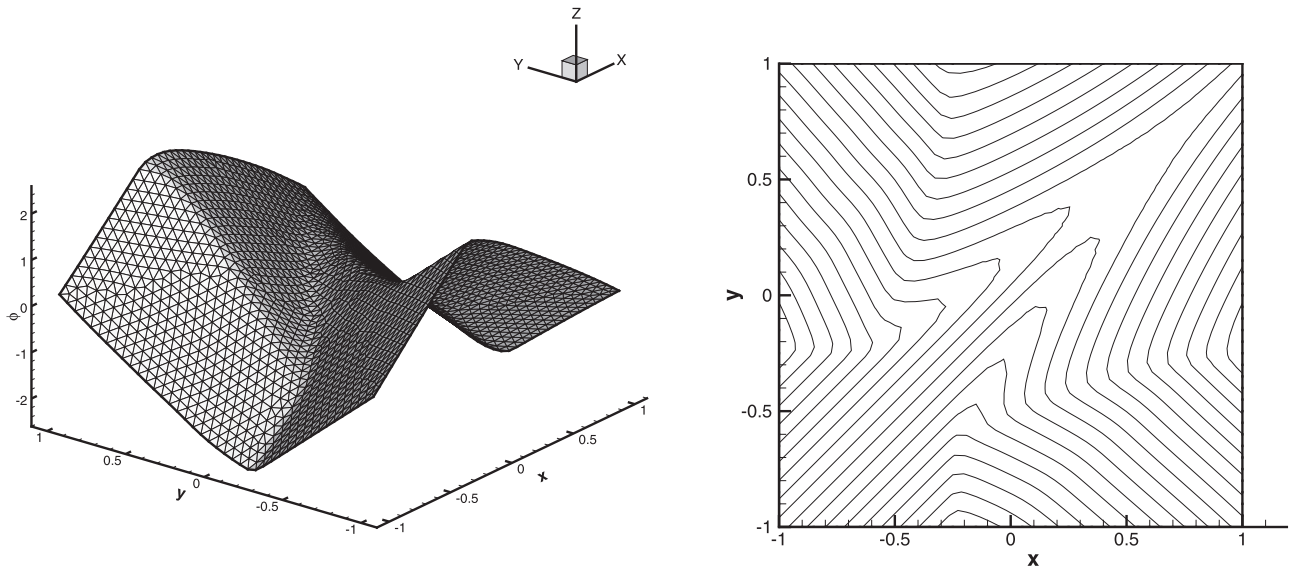


Fig. 12. Two dimensional Riemann problem with nonconvex and nonconcave flux. Left: surface of the solution; Right: the contour plot of the solution.

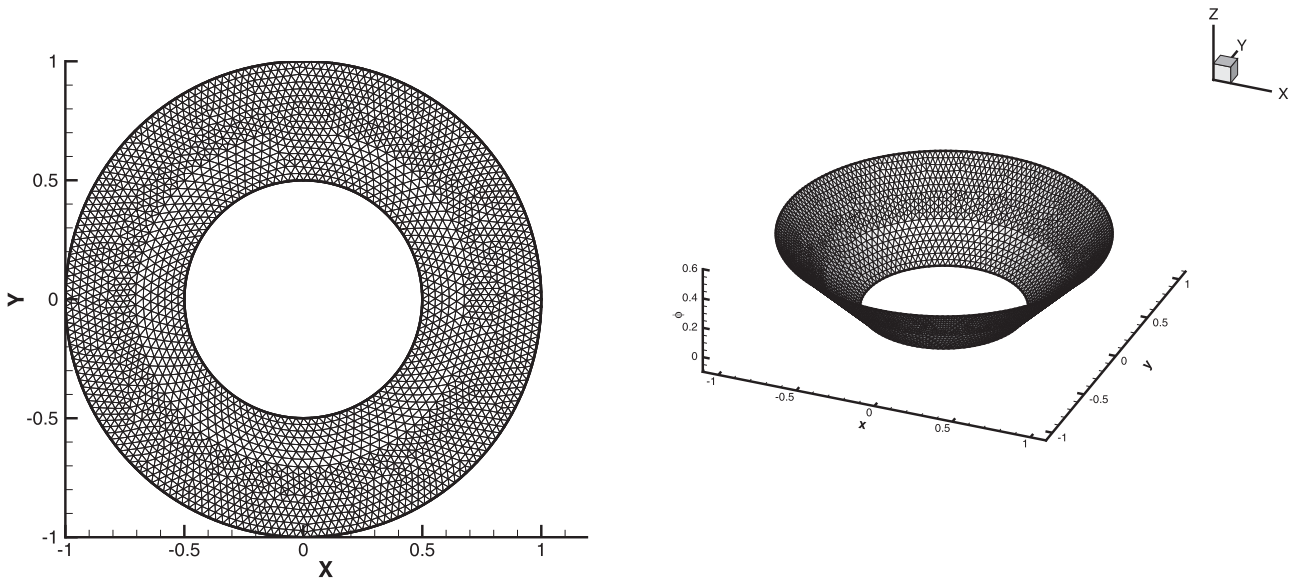


Fig. 13. The level set equation. Left: the sample of the mesh; Right: the surface of the solution.

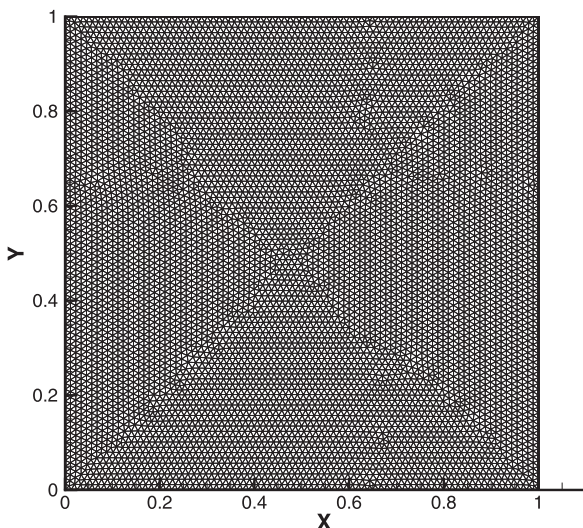


Fig. 14. The mesh for Examples 10 and 11 with the number of nodes $N = 7357$.

Here, H_3, H_4 and H_5 are the partial derivatives of H with respect to ϕ_{xx}, ϕ_{xy} and ϕ_{yy} , respectively. We take \bar{H}_3 as

$$\bar{H}_3 = H_3 \left(\frac{\sum_{l=0}^{k_i} \theta_l \phi_{x_l}}{2\pi}, \frac{\sum_{l=0}^{k_i} \theta_l \phi_{y_l}}{2\pi}, \frac{\sum_{l=0}^{k_i} \theta_l u_{x_l}}{2\pi}, \frac{\sum_{l=0}^{k_i} \theta_l (u_{y_l} + v_{x_l})}{4\pi}, \frac{\sum_{l=0}^{k_i} \theta_l v_{y_l}}{2\pi} \right)$$

and define \bar{H}_4 and \bar{H}_5 similarly. In order to obtain the approximation to the second derivatives u_{xx}, u_{xy}, u_{yy} , we simply find a third degree polynomial $q_0(x, y)$, such that:

$$q_0(x_l, y_l) = u_l \quad l = 1, 2, 3$$

and

$$q_0 = \operatorname{argmin} \left(\sum_l (p(x_l, y_l) - u_l)^2 \right)$$

where $l = 4, 5, \dots, 12$, and the minimum is taken over all the polynomials p of degree at most 3. Then, we take the second derivatives of the obtained polynomial q_0 as approximations to the second derivatives u_{xx}, u_{xy}, u_{yy} . In a similar way, we can obtain

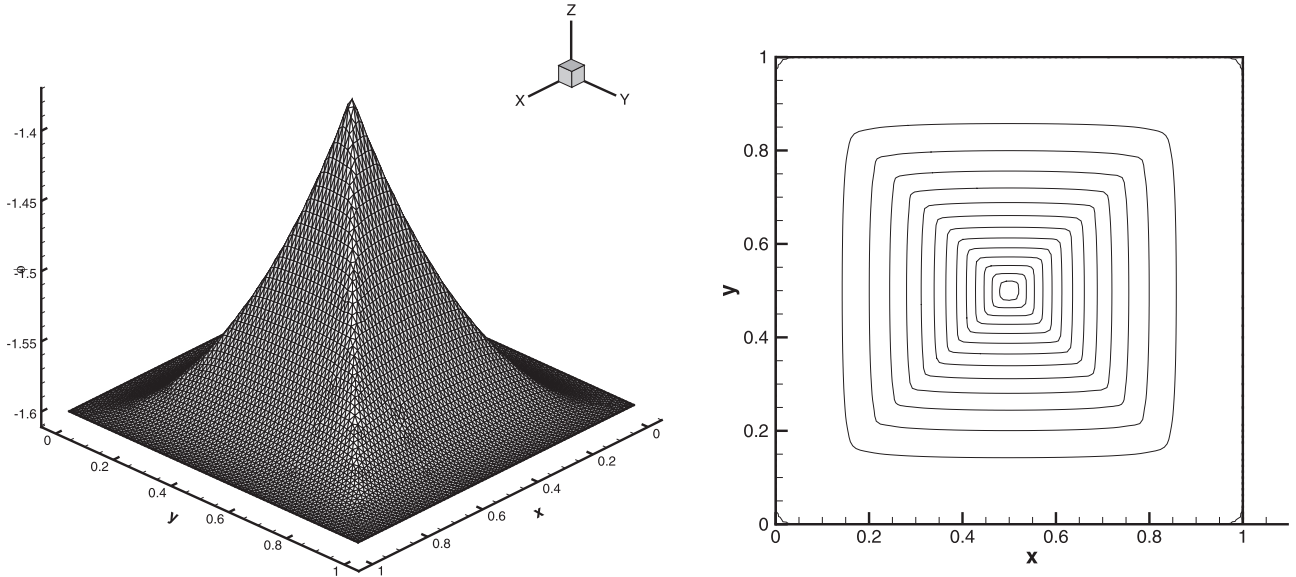


Fig. 15. Two dimensional Eikonal equation, Left: the surface of the solution; Right: the contour plot of the solution.

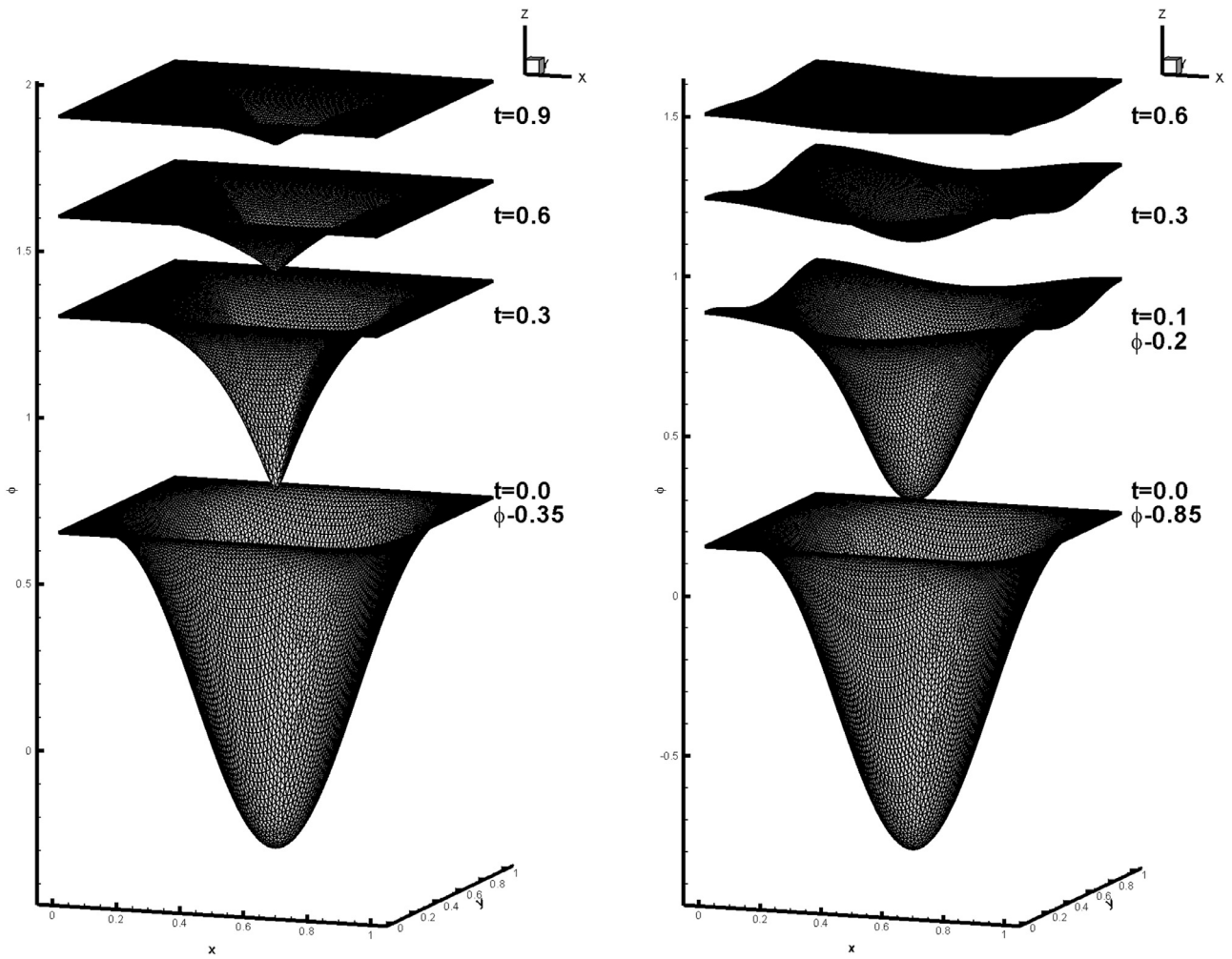


Fig. 16. Propagating surface with 14,392 cells. Left: $\epsilon = 0$; Right: $\epsilon = 0.1$.

approximations to the second derivatives v_{xx} , v_{xy} , v_{yy} . The time step is taken as

$$\Delta t = \frac{1}{\max \left\{ \frac{\alpha}{0.5\sqrt{|\Delta|_{\min}}}, \frac{\gamma_1}{0.25|\Delta|_{\min}}, \frac{\gamma_2}{0.25|\Delta|_{\min}}, \frac{\gamma_3}{0.25|\Delta|_{\min}} \right\}},$$

where $\gamma_1 = \max |H_3|$, $\gamma_2 = \max |H_4|$, $\gamma_3 = \max |H_5|$. We compute the solution on the mesh shown in Fig. 14, and list the results of $\varepsilon = 0$ (pure convection) and $\varepsilon = 0.1$ in Fig. 16. 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 details of the solution at later time. We can see that our scheme can obtain good result in this case.

4. Conclusion

In this paper, we design a fourth order finite difference HWENO scheme for the Hamilton–Jacobi equations on triangle meshes. The main advantage of this scheme is its compactness and efficiency. Extensive numerical experiments show that the scheme can maintain high order accuracy in the smooth case and can keep high resolution in the non-smooth case.

References

- [1] Abgrall R. Numerical discretization of the first-order Hamilton–Jacobi equation on triangular meshes. *Commun Pure Appl Math* 1996;49:1339–73.
- [2] Bryson S, Levy D. High-order semi-discrete central-upwind schemes for multi-dimensional Hamilton–Jacobi equations. *J Comput Phys* 2003;189:63–87.
- [3] Capdeville G. A central WENO scheme for solving hyperbolic conservation laws on non-uniform meshes. *J Comput Phys* 2008;227:2977–3014.
- [4] Christlieb A, Liu Y, Tang Q, Xu Z. Positivity-preserving finite difference weighted ENO schemes with constrained transport for ideal magnetohydrodynamic equations. *SIAM J Sci Comput* 2015;37:A1825–45.
- [5] Christlieb A, Xiao F, Yan J, Qi T. A high-order finite difference WENO scheme for ideal magnetohydrodynamics on curvilinear meshes. *SIAM J Sci Comput* 2018;40:A2631–66.
- [6] Crandall M, Lions PL. Viscosity solutions of Hamilton–Jacobi equations. *Trans Am Math Soc* 1983;277:1–42.
- [7] Crandall M, Lions PL. Two approximations of solutions of Hamilton–Jacobi equations. *Math Comput* 1984;43:1–19.
- [8] Gao Z, Don W. Mapped hybrid central-WENO finite difference scheme for detonation waves simulations. *J Sci Comput* 2013;55:351–71.
- [9] Hu C, Shu CW. A discontinuous Galerkin finite element method for Hamilton–Jacobi equations. *SIAM J Sci Comput* 1999;21:666–90.
- [10] Hu G, Li R, Tang T. A robust WENO type finite volume solver for steady euler equations on unstructured grids. *Commun Comput Phys* 2011;9:627–48.
- [11] Jiang G, Peng D. Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM J Sci Comput* 2000;21:2126–43.
- [12] Jiang G, Shu CW. Efficient implementation of weighted ENO schemes. *J Comput Phys* 1996;126:202–28.
- [13] Jin S, Xin Z. Numerical passage from systems of conservation laws to Hamilton–Jacobi equations, and relaxation schemes. *SIAM J Numer Anal* 1998;35:2385–404.
- [14] Lafon F, Osher S. High order two dimensional nonoscillatory methods for solving Hamilton–Jacobi scalar equations. *J Comput Phys* 1996;123:235–53.
- [15] Levy D, Nayak S, Shu CW, Zhang Y. Central WENO schemes for Hamilton–Jacobi equations on triangular meshes. *SIAM J Sci Comput* 2006;28:2229–47.
- [16] Levy D, Puppo G, Russo G. Central WENO schemes for hyperbolic systems of conservation laws. *ESAIM–Math Modell Numer Anal* 1999;33:547–71.
- [17] Levy D, Puppo G, Russo G. Compact central WENO schemes for multidimensional conservation laws. *SIAM J Sci Comput* 2000;22:656–72.
- [18] Li F, Shu CW. Reinterpretation and simplified implementation of a discontinuous Galerkin method for Hamilton–Jacobi equations. *Appl Math Lett* 2005;18:1204–9.
- [19] Li F, Yakovlev S. A central discontinuous Galerkin method for Hamilton–Jacobi equations. *J Sci Comput* 2010;45:404–28.
- [20] Li XG, Yan W, Chan CK. Numerical schemes for Hamilton–Jacobi equations on unstructured meshes. *Numerische Mathematik* 2003;94:315–31.
- [21] Liu H, Pollack M. Alternating evolution discontinuous Galerkin methods for Hamilton–Jacobi equations. *J Comput Phys* 2014;258:31–46.
- [22] Liu XD, Osher S, Chan T. Weighted essentially nonoscillatory schemes. *J Comput Phys* 1994;115:200–12.
- [23] Liu Y, Zhang Y. A robust reconstruction for unstructured WENO schemes. *J Sci Comput* 2013;54:603–21.
- [24] Osher S, Sethian J. Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. *J Comput Phys* 1988;79:12–49.
- [25] Osher S, Shu CW. High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations. *SIAM J Numer Anal* 1991;28:907–22.
- [26] Parna P, Meyer K, Falconer R. GPU driven finite difference WENO scheme for real time solution of the shallow water equations. *Comput Fluids* 2018;161:107–20.
- [27] Qiu J. Hermite WENO schemes with lax-wendroff type time discretizations for Hamilton–Jacobi equations. *J Comput Math* 2007;25:131–44.
- [28] Qiu J, Shu CW. Hermite WENO schemes and their application as limiters for Runge–Kutta discontinuous Galerkin method: one-dimensional case. *J Comput Phys* 2004;193:115–35.
- [29] Qiu J, Shu CW. Hermite WENO schemes for Hamilton–Jacobi equations. *J Comput Phys* 2005;204:82–99.
- [30] Shi J, Hu C, Shu CW. A technique of treating negative weights in WENO schemes. *J Comput Phys* 2002;175:108–27.
- [31] Shu CW. High order numerical methods for time dependent Hamilton–Jacobi equations. In: Goh SS, Ron A, Shen Z, editors. *Mathematics and Computation in Imaging Science and Information Processing*, vol. 11. Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore, World Scientific Press; 2007. p. 47–91.
- [32] Shu CW, Osher S. Efficient implementation of essentially non-oscillatory shock capturing schemes. *J Comput Phys* 1988;77:439–71.
- [33] Sussman M, Smereka P, Osher S. A level set approach for computing solution to incompressible two-phase flow. *J Comput Phys* 1994;114:146–59.
- [34] Wang C, Shu CW, Han W, Ning J. High resolution WENO simulation of 3d detonation waves. *Combust Flame* 2013;160:447–62.
- [35] Xing Y, Shu CW. High-order finite volume WENO schemes for the shallow water equations with dry states. *Adv Water Resour* 2011;34:1026–38.
- [36] Xiong T, Shu CW, Zhang M. WENO scheme with subcell resolution for computing nonconservative euler equations with applications to one-dimensional compressible two-medium flows. *J Sci Comput* 2012;53:222–47.
- [37] Zhang S, Jiang S, Shu CW. Improvement of convergence to steady state solutions of euler equations with the WENO schemes. *J Sci Comput* 2011;47:216–38.
- [38] Zhang Y, Shu CW. High-order WENO schemes for Hamilton–Jacobi equations on triangular meshes. *SIAM J Sci Comput* 2003;24:1005–30.
- [39] Zheng F, Qiu J. Directly solving the Hamilton–Jacobi equations by hermite WENO schemes. *J Comput Phys* 2016;307:423–45.
- [40] Zheng F, Shu CW, Qiu J. Finite difference hermite WENO schemes for the Hamilton–Jacobi equations. *J Comput Phys* 2017;337:27–41.
- [41] Zhu J, Qiu J. Hermite WENO schemes for Hamilton–Jacobi equations on unstructured meshes. *J Comput Phys* 2013;254:76–92.
- [42] Zhu J, Qiu J. Finite volume hermite WENO schemes for solving the Hamilton–Jacobi equation. *Commun Comput Phys* 2014;15:959–80.
- [43] Zhu J, Qiu J. Finite volume hermite WENO schemes for solving the Hamilton–Jacobi equations II: unstructured meshes. *Comput Math Appl* 2014;68:1137–50.
- [44] Zhu J, Qiu J. A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. *J Comput Phys* 2016;318:110–21.
- [45] Zhu J, Qiu J. A new type of high order WENO schemes for Hamilton–Jacobi equations on triangular meshes. *Commun Comput Phys*; 2019. To appear.
- [46] Zhu J, Qiu J, Liu T, Khoo B. RKDG methods with WENO type limiters and conservative interfacial procedure for one-dimensional compressible multi-medium flow simulations. *Appl Numer Math* 2011;61:554–80.