High order finite difference Hermite WENO schemes for the Hamilton-Jacobi equations on unstructured meshes

Feng Zheng\textsuperscript{1}, Chi-Wang Shu\textsuperscript{2} and Jianxian Qiu\textsuperscript{3}

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 [29] 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.

Keywords: HWENO method; Hamilton-Jacobi equation; finite difference method; unstructured mesh.

\textsuperscript{1}School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, P. R. China. E-mail: fzbz20080831@163.com.

\textsuperscript{2}Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: shu@dam.brown.edu. Research supported by ARO grant W911NF-16-1-0103 and NSF grant DMS-1719410.

\textsuperscript{3}School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling & High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, P. R. China. E-mail: jxqiu@xmu.edu.cn. Research supported by NSFC grant 11571290 and NSAF grant U1630247.
1 Introduction

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

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

$$\phi(x, y, 0) = \phi_0(x, y)$$  \hspace{1cm} (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 [4], we can obtain the unique weak solution for the HJ equations.

As the 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 [5] 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 [21] designed high order essentially nonoscillatory (ENO) schemes for solving the HJ equations. Jiang and Peng [7] generalized them to weighted ENO (WENO) schemes, with tremendous success. The WENO schemes in [7] have become standard choices for solving HJ equations on structured meshes. Hermite WENO (HWENO) schemes and related methods have been developed in [24, 22, 36, 31] 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 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 [6] designed the first DG scheme of this type. Later, Li and Shu [14] reinterpreted and simplified the DG scheme in [6]. Li and Yakovlev [15] proposed central DG schemes to solve the HJ equations. Liu and Pollack [17] suggested the alternative evolution DG (AEDG) schemes. Zhu and Qiu [34] and Zheng and Qiu [30] 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 [16] also developed a monotone and convergent scheme based on the weak form of the viscid HJ equation. Lafon and Osher [10] proposed high order ENO methods on triangular meshes. Zhang and Shu [29] and Levy et al. [11] developed high order WENO and central WENO schemes on triangular meshes respectively. Zhu and Qiu [35, 33, 32] 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 [26].

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 [18, 8]. 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 [23, 24] 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 [25], it may not always fix the stability problem when the negative linear weight is very large [19]. 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 [36], with similar ideas in earlier work [3, 12, 13]. 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.

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 equation (1.1) solved on a domain $\Omega$, which is partitioned into non-overlapping triangles denoted by $\Delta_l, l = 1 \ldots N$. We define $|\Delta_l|$ and $(x_l, y_l)$ as the area and the barycenter of the triangle $\Delta_l$ respectively. For every node $i$, we define the angular sectors which share the same node $i$ as $T_0, \cdots, T_{k_i}$ in the anticlockwise order. $\overrightarrow{n_{l+\frac{1}{2}}}$ is the unit vector of the half-line $D_{l+\frac{1}{2}} = T_l \bigcap T_{l+1}$, and $\theta_l$ is the inner angle of the sector $T_l, 0 \leq l \leq k_i$. See Figure 1.
We define $\phi_i$ 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 equation (1.1), we obtain the following system for the approximation:

$$
\begin{align*}
\frac{d\phi_i}{dt} &= -H(\nabla \phi_i) \\
\frac{du_i}{dt} &= -H_1(\nabla \phi_i)u_{xi} - H_2(\nabla \phi_i)v_{xi} \\
\frac{dv_i}{dt} &= -H_1(\nabla \phi_i)v_{yi} - H_2(\nabla \phi_i)v_{yi}
\end{align*}
$$

(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:

$$
\begin{align*}
\frac{d\phi_i}{dt} &= -H(\nabla \phi_i) \\
\frac{du_i}{dt} &= -H_1(\nabla \phi_i)u_{xi} + H_2(\nabla \phi_i)v_{yi} \\
\frac{dv_i}{dt} &= -H_1(\nabla \phi_i)v_{xi} + H_2(\nabla \phi_i)v_{yi}
\end{align*}
$$

(2.2)

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

$$
\begin{align*}
\frac{d\phi_i}{dt} &= \hat{H}_i \\
\frac{du_i}{dt} &= -H_1(\nabla \hat{\phi}_i)u_{xi} + H_2(\nabla \hat{\phi}_i)v_{yi} \\
\frac{dv_i}{dt} &= -H_1(\nabla \hat{\phi}_i)v_{xi} + H_2(\nabla \hat{\phi}_i)v_{yi}
\end{align*}
$$

(2.3)
Here, $\hat{H}_i$ 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:

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

where $\beta_{l+\frac{1}{2}} = \tan \left( \frac{\theta_l}{2} \right) + \tan \left( \frac{\theta_{l+1}}{2} \right)$, $\alpha = \max \left\{ \max_{A \leq u \leq B} |H_1(u, v)|, \max_{C \leq v \leq D} |H_2(u, v)| \right\}$. Here $\nabla \phi_l$ is the numerical approximation to $\nabla \phi$ at node $i$ in sector $T_l$. $[A, B]$ is the range of the value $\phi_{x_l}$, and $[C, D]$ is the range of the value $\phi_{y_l}$, 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 $H_1 u_{x_l} + H_2 u_{y_l}$ in a similar way:

$$H_1 u_{x_l} + H_2 u_{y_l} = H_1 \left( \sum_{l=0}^{k_i} \frac{\theta_l(\nabla \phi_l)}{2\pi} \right) + H_2 \left( \sum_{l=0}^{k_i} \frac{\theta_l(u_{x_l})}{2\pi} \right) - \frac{\alpha}{\pi} \sum_{l=0}^{k_i} \beta_{l+\frac{1}{2}} \left( \frac{\nabla u_l + \nabla u_{l+1}}{2} \right) \cdot \vec{n}_{l+\frac{1}{2}} \quad (2.5)$$

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

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 [27] 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 $\nabla \phi$, $\nabla u$ and $\nabla v$, 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 [36, 32] to construct fourth order HWENO approximation to the nodes 1, 2, 3 of the target cell $\Delta_0$, as shown in Figure 2, for $\nabla \phi$, $\nabla u$ and $\nabla v$.

![Figure 2: The nodes used for the big stencil](image)

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 $\Delta_0$. 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

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

(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 $S_0 = \{1, 2, \cdots, 12\}$ as shown in Figure 2, we would like to obtain the fourth degree polynomial $p_0(x, y)$ such that

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

and

$$p_0 = \arg \min \left( \sum_l (p(x_l, y_l) - \phi_l)^2 + |\Delta_0| \sum_l (\frac{\partial}{\partial x} p(x_l, y_l) - u_l)^2 + |\Delta_0| \sum_l (\frac{\partial}{\partial y} p(x_l, y_l) - v_l)^2 \right)$$  \hspace{1cm} (2.9)

where $l = 4, 5, \cdots, 12$, 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:

$$\mathbf{a} = \arg \min_{\mathbf{x}} ||\mathbf{Bx} - \mathbf{f}||_2$$  \hspace{1cm} s.t.  \hspace{1cm} \mathbf{Ax} = \mathbf{b}$$  \hspace{1cm} (2.10)

where the $\mathbf{A}$ is a $3 \times 15$ matrix coming from the coefficients of the equation (2.8), $\mathbf{b}$ is a $3 \times 1$ matrix coming from right hand side of the equation (2.8), $\mathbf{B}$ is a $27 \times 15$ matrix coming from the equation (2.9) and $\mathbf{f}$ is $27 \times 1$ matrix coming from the information of $\phi_l, \sqrt{|\Delta_0|}u_l, \sqrt{|\Delta_0|}v_l$ in the equation (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:

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

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

$$||\mathbf{Bx} - \mathbf{f}||_2^2 = (\mathbf{Bx} - \mathbf{f})^T (\mathbf{Bx} - \mathbf{f})$$  \hspace{1cm} (2.11)

$$= \mathbf{x}^T \mathbf{B}^T \mathbf{B} \mathbf{x} - 2\mathbf{f}^T \mathbf{B} \mathbf{x} + \mathbf{f}^T \mathbf{f}$$

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 \( B \) and \( B(i, j) \) refers to the elements of \( B \).

By requiring \( L_x(x, \lambda) = 0 \) and \( L_\lambda(x, \lambda) = 0 \), we have the following linear system:

\[
\begin{cases}
  cB^T Bx - A^T \lambda = cB^T f \\
  Ax = b
\end{cases}
\]  

(2.12)

We can rewrite the linear system in the matrix form:

\[
\begin{pmatrix}
  cB^T B & -A^T \\
  -A & 0
\end{pmatrix}
\begin{pmatrix}
  x \\
  \lambda
\end{pmatrix}
= \begin{pmatrix}
  cB^T f \\
  -b
\end{pmatrix}
\]  

(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 \( p_k(x, y) \)

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

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 \( \beta_l, l = 0, 1, \cdots, 4 \), which measures the smoothness of the polynomials \( p_l \) on the target cell \( \Delta_0 \). The smaller the indicators are, the smoother the polynomials are. We use a similar definition as in [7, 29]:

\[
\beta_k = \sum_{|l| \geq 2} |\Delta_0||l|^{-2}(\frac{\partial^{2l}}{\partial x^{l_1} \partial y^{l_2}}p_k(x, y)) |dxdy|, \quad k = 0, 1, \cdots, 4
\]

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 \( k = 0, 1, \ldots, 4 \). 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:

\[ w_k = \frac{\sum_l w_l}{w_l} = \gamma_l \left( 1 + \frac{\tau}{\varepsilon + \beta_l} \right)^2, \quad l = 0, 1, \ldots, 4, \]

where the parameter \( \tau \) is defined as follows:

\[ \tau = \sum_{k=1}^{4} (\beta_1 - \beta_k)^2 \]

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 \( \beta \)'s in smooth regions:

\[ \beta_0 = \left( \sum_{|l|=2} \left( \frac{\partial^{|l|}}{\partial x^l \partial y^l} p_0(x, y) \right) \bigg|_{(x_0, y_0)}^2 \right) |\Delta_0|(1 + O(|\Delta_0|)) \]

and

\[ \beta_k = \left( \sum_{|l|=2} \left( \frac{\partial^{|l|}}{\partial x^l \partial y^l} p_k(x, y) \right) \bigg|_{(x_0, y_0)}^2 \right) |\Delta_0|(1 + O(\sqrt{|\Delta_0|})), \quad k = 1, 2, 3, 4. \]

Also by Taylor expansions, we have

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

which leads to

\[ w_l = \gamma_l + O(|\Delta_0|^2), \quad l = 0, 1, \ldots, 4. \]
Therefore, we have

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

\[
= (\gamma_0 + w_0 - \gamma_0) \left( \frac{1}{\gamma_0} \nabla p_0(x, y) - \sum_{k=1}^{4} \frac{\gamma_k}{\gamma_0} \nabla p_k(x, y) - \nabla \phi(x, y) \right)
\]

\[
+ \sum_{k=1}^{4} (\gamma_k + w_k - \gamma_k) (\nabla p_k(x, y) - \nabla \phi(x, y))
\]

\[
= \nabla p_0(x, y) - \nabla \phi(x, y) + (w_0 - \gamma_0) \left( \frac{1}{\gamma_0} \nabla p_1(x, y) - \sum_{k=1}^{4} \frac{\gamma_k}{\gamma_0} \nabla p_k(x, y) - \nabla \phi(x, y) \right)
\]

\[
+ \sum_{k=1}^{4} (w_k - \gamma_k) (\nabla p_k(x, y) - \nabla \phi(x, y))
\]

\[
= O(|\Delta_0|^2) + O(|\Delta_0|^2)O(|\Delta_0|) + O(|\Delta_0|^2)O(|\Delta_0|)
\]

\[
= O(|\Delta_0|^2)
\]

(2.14)

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 \( \phi \) at the three nodes 1, 2, 3 of the target cell \( \Delta_0 \) 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 Figure 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 $\nabla u$ and $\nabla v$

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

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

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

(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 $S_0 = \{1, 2, \cdots, 12\}$ as shown in Figure 2, we would like to obtain the fifth degree polynomial $p_0(x, y)$ such that

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

$$p_0_x(x_l, y_l) = u_l \quad l = 1, 2, 3$$

$$p_0_y(x_l, y_l) = v_l \quad l = 1, 2, 3$$

(2.16)

and

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

(2.17)

where $l = 4, 5, \cdots, 12$, 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 $\nabla u$ and $\nabla v$.

**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 $p_k(x, y)$

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

such that

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

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 $\nabla u$ and $\nabla v$.

**Step 4.** We compute the smoothness indicators $\beta_{k}, l = 0, 1, \cdots, 4$ as follows:

$$\beta_{k} = \sum_{|l| \geq 3} \int_{\Delta_{0}} |\Delta_{0}|^{-3} \left( \frac{\partial^{|l|}}{\partial x^{l_1} \partial y^{l_2}} p_{k}(x, y) \right)^{2} dxdy, \quad k = 0, 1, \cdots, 4$$

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 $\phi$. 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 $\nabla u$ and $\nabla v$. 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

$$\Delta t = \frac{1}{2} \sqrt{\frac{\Delta_{\text{min}}}{\alpha}}$$
where $|\Delta|_{\min} = \min_i |\Delta_i|$, $\alpha$ 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. In accuracy test cases, we also present the results of the linear schemes, which use the linear weights instead of the nonlinear weights, to make the comparison with our HWENO schemes.

![Figure 3: The sample mesh with the number of nodes N=134](image)

**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 $\phi(x, y, 0) = -\cos\left(\frac{x}{2}(x + y)\right)$ and periodic boundary condition. A sample mesh with boundary triangle size $h = 0.4$ is shown in Figure 3. We compute the result up to $t = 2$ to test the accuracy of $\phi$ and $\nabla \phi$ of both the linear scheme and the HWENO scheme. The errors and numerical orders of accuracy are shown in Table 1 and Table 2. We can see that the HWENO scheme can achieve its designed order of accuracy, at least in $L_1$ and $L_2$ norms.

**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 \( \phi(x, y, 0) = -\cos\left(\frac{x}{2}(x + y)\right) \) and periodic boundary condition. The coarsest mesh with \( h = 0.4 \) is shown in Figure 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 scheme are tested in this case and the errors and the orders of accuracy of \( \phi \) and \( \nabla \phi \) are listed in Table 3 and Table 4. We can see that the HWENO scheme can reach its designed order of accuracy.
Table 3: \( \phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 = 0, \phi(x, y, 0) = -\cos(\frac{\pi}{2}(x + y)) \). Periodic boundary conditions.

\[
\begin{array}{c|c|c|c|c|c|c}
N & L_{\infty} \text{ error} & \text{ order} & L_2 \text{ error} & \text{ order} & L_1 \text{ error} & \text{ order} \\
\hline
\text{Linear scheme: } \phi & & & & & & \\
134 & 2.42E-02 & 5.59E-03 & 3.55E-03 &  &  & \\
492 & 2.60E-03 & 3.22 & 6.01E-04 & 3.22 & 2.84E-04 & 3.65 &  \\
1876 & 1.87E-04 & 3.80 & 3.30E-05 & 4.19 & 1.26E-05 & 4.50 &  \\
7337 & 1.39E-05 & 3.75 & 1.73E-06 & 4.25 & 6.10E-07 & 4.36 &  \\
29204 & 4.96E-07 & 4.80 & 6.40E-08 & 4.76 & 2.37E-08 & 4.69 &  \\
\hline
\text{HWENO scheme: } \phi & & & & & & \\
134 & 1.07E-01 & 4.09E-02 & 3.11E-02 &  &  & \\
492 & 1.70E-02 & 2.65 & 5.78E-03 & 2.82 & 3.75E-03 & 3.05 &  \\
1876 & 4.09E-04 & 5.38 & 9.83E-05 & 5.88 & 5.74E-05 & 6.03 &  \\
7337 & 1.18E-05 & 5.11 & 1.70E-06 & 5.85 & 7.99E-07 & 6.17 &  \\
29204 & 4.80E-07 & 4.62 & 6.41E-08 & 4.73 & 2.46E-08 & 5.02 &  \\
\end{array}
\]

Table 4: \( \phi_t + \frac{1}{2}(\phi_x + \phi_y + 1)^2 = 0, \phi(x, y, 0) = -\cos(\frac{\pi x}{2}(x + y)) \). Periodic boundary conditions.

\[
\begin{array}{c|c|c|c|c|c|c}
N & L_{\infty} \text{ error} & \text{ order} & L_2 \text{ error} & \text{ order} & L_1 \text{ error} & \text{ order} \\
\hline
\text{Linear scheme: } \nabla \phi & & & & & & \\
134 & 7.87E-02 & 2.51E-02 & 1.44E-02 &  &  & \\
492 & 1.47E-02 & 2.42 & 3.71E-03 & 2.76 & 1.65E-03 & 3.13 &  \\
1876 & 1.39E-03 & 3.40 & 2.58E-04 & 3.84 & 9.70E-05 & 4.09 &  \\
7337 & 1.95E-05 & 2.83 & 1.66E-05 & 3.96 & 5.70E-06 & 4.09 &  \\
29204 & 1.19E-05 & 4.03 & 6.40E-07 & 4.69 & 2.22E-07 & 4.68 &  \\
\hline
\text{HWENO scheme: } \nabla \phi & & & & & & \\
134 & 1.95E-01 & 7.83E-02 & 5.52E-02 &  &  & \\
492 & 6.35E-02 & 1.62 & 1.57E-02 & 2.32 & 9.32E-03 & 2.56 & \\
1876 & 5.90E-03 & 3.43 & 1.10E-03 & 3.84 & 5.59E-04 & 4.06 &  \\
7337 & 7.70E-04 & 2.94 & 7.04E-05 & 3.96 & 1.96E-05 & 4.84 &  \\
29204 & 1.69E-05 & 5.51 & 9.63E-07 & 6.19 & 3.02E-07 & 6.02 &  \\
\end{array}
\]

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 \( \phi(x, y, 0) = -\cos(\frac{\pi x}{2}(x + y)) \) and periodic boundary condition. The coarsest mesh with \( h = 0.4 \) is shown in Figure 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 \( \phi \) and \( \nabla \phi \) of both the linear scheme and
the HWENO scheme. From Table 5 and Table 6, we can see that the HWENO scheme can reach the expected order of accuracy.

Table 5: $\phi_t - \cos(\phi_x + \phi_y + 1) = 0$, $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x + y))$. Periodic boundary conditions. $t = 0.5/\pi^2$

<table>
<thead>
<tr>
<th>N</th>
<th>$L_\infty$ error</th>
<th>order</th>
<th>$L_2$ error</th>
<th>order</th>
<th>$L_1$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear scheme: $\phi$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>5.77E-03</td>
<td></td>
<td>2.14E-03</td>
<td></td>
<td>1.80E-03</td>
<td></td>
</tr>
<tr>
<td>492</td>
<td>1.57E-03</td>
<td>1.88</td>
<td>3.84E-04</td>
<td>2.48</td>
<td>2.52E-04</td>
<td>2.84</td>
</tr>
<tr>
<td>1876</td>
<td>3.61E-04</td>
<td>2.12</td>
<td>4.31E-05</td>
<td>3.16</td>
<td>1.92E-05</td>
<td>3.71</td>
</tr>
<tr>
<td>7337</td>
<td>4.88E-05</td>
<td>2.89</td>
<td>4.52E-06</td>
<td>3.25</td>
<td>1.55E-06</td>
<td>3.64</td>
</tr>
<tr>
<td>29204</td>
<td>2.94E-06</td>
<td>4.05</td>
<td>2.31E-07</td>
<td>4.29</td>
<td>7.02E-08</td>
<td>4.46</td>
</tr>
<tr>
<td>116012</td>
<td>1.17E-07</td>
<td>4.65</td>
<td>9.26E-09</td>
<td>4.64</td>
<td>2.96E-09</td>
<td>4.56</td>
</tr>
<tr>
<td>HWENO scheme: $\phi$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>2.47E-02</td>
<td></td>
<td>9.95E-03</td>
<td></td>
<td>7.73E-03</td>
<td></td>
</tr>
<tr>
<td>492</td>
<td>3.63E-03</td>
<td>2.77</td>
<td>1.17E-03</td>
<td>3.09</td>
<td>8.14E-04</td>
<td>3.25</td>
</tr>
<tr>
<td>1876</td>
<td>3.77E-04</td>
<td>3.27</td>
<td>4.63E-05</td>
<td>4.66</td>
<td>2.39E-05</td>
<td>5.09</td>
</tr>
<tr>
<td>7337</td>
<td>4.80E-05</td>
<td>2.97</td>
<td>4.31E-06</td>
<td>3.43</td>
<td>1.47E-06</td>
<td>4.02</td>
</tr>
<tr>
<td>29204</td>
<td>2.86E-06</td>
<td>4.07</td>
<td>2.21E-07</td>
<td>4.29</td>
<td>6.76E-08</td>
<td>4.44</td>
</tr>
<tr>
<td>116012</td>
<td>1.17E-07</td>
<td>4.61</td>
<td>9.20E-09</td>
<td>4.58</td>
<td>2.96E-09</td>
<td>4.51</td>
</tr>
</tbody>
</table>

Table 6: $\phi_t - \cos(\phi_x + \phi_y + 1) = 0$, $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x + y))$. Periodic boundary conditions. $t = 0.5/\pi^2$

<table>
<thead>
<tr>
<th>N</th>
<th>$L_\infty$ error</th>
<th>order</th>
<th>$L_2$ error</th>
<th>order</th>
<th>$L_1$ error</th>
<th>order</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear scheme: $\nabla \phi$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>4.46E-02</td>
<td></td>
<td>1.19E-02</td>
<td></td>
<td>7.16E-03</td>
<td></td>
</tr>
<tr>
<td>492</td>
<td>1.74E-02</td>
<td>1.36</td>
<td>3.81E-03</td>
<td>1.64</td>
<td>1.81E-03</td>
<td>1.98</td>
</tr>
<tr>
<td>1876</td>
<td>4.36E-03</td>
<td>2.00</td>
<td>6.52E-04</td>
<td>2.54</td>
<td>2.42E-04</td>
<td>2.91</td>
</tr>
<tr>
<td>7337</td>
<td>7.71E-04</td>
<td>2.50</td>
<td>9.31E-05</td>
<td>2.81</td>
<td>2.80E-05</td>
<td>3.11</td>
</tr>
<tr>
<td>29204</td>
<td>7.69E-05</td>
<td>3.33</td>
<td>5.83E-06</td>
<td>4.00</td>
<td>1.49E-06</td>
<td>4.24</td>
</tr>
<tr>
<td>116012</td>
<td>5.58E-06</td>
<td>3.78</td>
<td>2.42E-07</td>
<td>4.59</td>
<td>6.10E-08</td>
<td>4.61</td>
</tr>
<tr>
<td>HWENO scheme: $\nabla \phi$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>134</td>
<td>8.96E-02</td>
<td></td>
<td>2.49E-02</td>
<td></td>
<td>1.74E-02</td>
<td></td>
</tr>
<tr>
<td>492</td>
<td>2.30E-02</td>
<td>1.96</td>
<td>4.72E-03</td>
<td>2.40</td>
<td>2.80E-03</td>
<td>2.63</td>
</tr>
<tr>
<td>1876</td>
<td>6.84E-03</td>
<td>1.75</td>
<td>1.09E-03</td>
<td>2.11</td>
<td>4.39E-04</td>
<td>2.67</td>
</tr>
<tr>
<td>7337</td>
<td>1.70E-03</td>
<td>2.01</td>
<td>2.05E-04</td>
<td>2.41</td>
<td>5.51E-05</td>
<td>2.99</td>
</tr>
<tr>
<td>29204</td>
<td>2.26E-04</td>
<td>2.91</td>
<td>1.63E-05</td>
<td>3.65</td>
<td>2.95E-06</td>
<td>4.22</td>
</tr>
<tr>
<td>116012</td>
<td>7.12E-06</td>
<td>4.99</td>
<td>3.66E-07</td>
<td>5.48</td>
<td>7.70E-08</td>
<td>5.26</td>
</tr>
</tbody>
</table>

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 $\phi(x, y, 0) = -\cos\left(\frac{\pi}{2}(x + y)\right)$ 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 Figure 4 and plot the results in Figure 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 $\phi(x, y, 0) = -\cos\left(\frac{\pi}{2}(x + y)\right)$ and periodic boundary condition, see [21]. The mesh is shown in Figure 4. We compute the result up to $t = 1.5/\pi^2$ when the solution is not smooth any more. From Figure 6, we can see that the scheme can achieve high resolution in this example.

**Example 6.**
Figure 5: Burgers equation. $T = 1.5/\pi^2$. Left: surface of the solution; right: contour plot of the solution.

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

Figure 7: The uniform mesh for Example 6 with the number of nodes $N=369$. 
We solve the problem
\[ \phi_t + \frac{1}{4}(\phi_x^2 - 1)(\phi_x^2 - 4) = 0 \]
with the initial datum \( \phi(x,y,0) = -2|x| \), see [29]. 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 Figure 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 Figure 8, we can see that the HWENO scheme can converge to the correct viscosity solution with mesh refinement.

\[ \text{Figure 8: Convergence study of the nonconvex and nonconcave case.} \]

**Example 7.**

We solve a problem from optimal control:
\[ \phi_t + \sin(y)\phi_x + (\sin(x) + \text{sign}(\phi_y))\phi_y - \frac{1}{2}\sin^2(y) + \cos(x) - 1 = 0, \quad -\pi < x, y < \pi \]
with \( \phi(x,y,0) = 0 \) and periodic boundary conditions, see [21]. 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

\[
\begin{align*}
\frac{d\phi_i}{dt} &= -\hat{H}_i \\
\frac{du_i}{dt} &= -H_1 u_{xi} + H_2 u_{yi} - \overline{H}_{xi} \\
\frac{dv_i}{dt} &= -H_1 v_{xi} + H_2 v_{yi} - \overline{H}_{yi}
\end{align*}
\]

in which the definition of $\hat{H}_i$, $H_1 u_{xi} + H_2 u_{yi}$, $H_1 v_{xi} + H_2 v_{yi}$, $\overline{H}_x$ and $\overline{H}_y$ are similar as before, just adding $x_i$ and $y_i$ inside the Hamiltonians. The mesh is shown in Figure 9. The solution at $t = 1$ is shown in Figure 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 [21]. The mesh is shown in Figure 11. We compute the solution up to $t = 1$. The solution is shown in Figure 12. Again, we observe our scheme can achieve good result.

**Example 9.**

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

![Figure 9](image-url)
We solve the level set equation

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

with the initial datum $\phi(x, y, 0) = \phi_0(x, y)$. This problem comes from [28]. The solution has the same zero level set as the initial condition $\phi_0$, 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 Figure 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 Figure 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 [9]. We compute the solution up to $t = 0.6$ on the mesh shown in Figure 14. The solution is shown in Figure 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 $\varepsilon$ is a small constant, with the initial datum $\phi(x, y, 0) = 1 - \frac{1}{2}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1)$ and periodic boundary condition is used. This problem comes from [20].

When $\varepsilon = 0$, we can treat the equation with the same method as before. When $\varepsilon \neq 0$, we can
Figure 15: Two dimensional Eikonal equation, Left: the surface of the solution; Right: the contour plot of the solution.

rewrite the equation as follows:

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

Then, we have

\[
\begin{align*}
\frac{d\phi_i}{dt} &= -\tilde{H}(\phi_x, \phi_y, \phi_{xx}, \phi_{xy}, \phi_{yy}) \\
\frac{du_i}{dt} &= -H_1u_x + H_2u_y - \overline{H}_3u_{xx} - \overline{H}_4u_{xy} - \overline{H}_5u_{yy} \\
\frac{dv_i}{dt} &= -H_1v_x + H_3v_y - \overline{H}_3v_{xx} - \overline{H}_4v_{xy} - \overline{H}_5v_{yy}
\end{align*}
\tag{3.3}
\]

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

\[
\overline{H}_3 = H_3 \left( \sum_{l=0}^{k_i} \frac{\theta_l \phi_{xl}}{2\pi}, \sum_{l=0}^{k_i} \frac{\theta_l \phi_{yl}}{2\pi}, \sum_{l=0}^{k_i} \frac{\theta_l u_{xl}}{2\pi}, \sum_{l=0}^{k_i} \frac{\theta_l(u_{yl} + v_{xl})}{4\pi}, \sum_{l=0}^{k_i} \frac{\theta_l v_{yl}}{2\pi} \right)
\]

and define \( \overline{H}_4 \) and \( \overline{H}_5 \) 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:

\[ q_0(x_l, y_l) = u_l \quad l = 1, 2, 3 \]
\[ q_0 = \text{argmin} \left( \sum_l (p(x_l, y_l) - u_l)^2 \right) \]

where \( l = 4, 5, \cdots, 12 \), 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 approximations to the second derivatives \( v_{xx}, v_{xy}, v_{yy} \). The time step is taken as

\[
\Delta t = \frac{1}{\max \left\{ \frac{\alpha}{0.5 \sqrt{\Delta_{\text{min}}}}, \frac{\gamma_1}{0.25 \Delta_{\text{min}}}, \frac{\gamma_2}{0.25 \Delta_{\text{min}}}, \frac{\gamma_3}{0.25 \Delta_{\text{min}}} \right\}},
\]

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 Figure 14, and list the results of \( \varepsilon = 0 \) (pure convection) and \( \varepsilon = 0.1 \) in Figure 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


Figure 16: Propagating surface with 14392 cells. Left: $\varepsilon = 0$; Right: $\varepsilon = 0.1$.


