Finite difference Hermite WENO schemes for the Hamilton-Jacobi equations[1]

Feng Zheng[2], Chi-Wang Shu[3] and Jianxian Qiu[4]

Abstract

In this paper, a new type of finite difference Hermite weighted essentially non-oscillatory (HWENO) schemes are constructed for solving Hamilton-Jacobi (HJ) equations. Point values of both the solution and its first derivatives are used in the HWENO reconstruction and evolved via time advancing. While the evolution of the solution is still through the classical numerical fluxes to ensure convergence to weak solutions, the evolution of the first derivatives of the solution is through a simple dimension-by-dimension non-conservative procedure to gain efficiency. The main advantages of this new scheme include its compactness in the spatial field and its simplicity in the reconstructions. Extensive numerical experiments in one and two dimensional cases are performed to verify the accuracy, high resolution and efficiency of this new scheme.

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

AMS(MOS) subject classification: 65M60, 35L65

[1]The research of F. Zheng and J. Qiu is partially supported by NSFC grant 1571290 and NSAF grant U1630247. The research of C.-W. Shu is partially supported by ARO grant W911NF-15-1-0226 and NSF grant DMS-1418750.

[2]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: fzbz200808-31@163.com.

[3]Division of Applied Mathematics, Brown University, Providence, RI 02912, USA. E-mail: shu@dam.brown.edu

[4]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.

1 Introduction

In this paper, we design a new finite difference Hermite weighted essentially non-oscillatory (HWENO) method to solve the following Hamilton Jacobi (HJ) equation

$$\phi_t + H(\nabla_{\mathbf{x}}\phi) = 0, \quad \phi(\mathbf{x}, 0) = \phi_0(\mathbf{x}) \quad \mathbf{x} \in \Omega \subset \mathbb{R}^d$$

where $\nabla_{\mathbf{x}}\phi = (\phi_{x_1}, \phi_{x_2}, ..., \phi_{x_n})^T$. We consider only up to two dimensions for simplicity and hence use x and y instead of x_1 and x_2 in the sequel.

The HJ equations come from many applications, from control theory and geometric optics, to image processing and level set method and so on. It is well known that solutions of the nonlinear HJ equation are typically continuous. However, the derivatives of the solutions could be discontinuous even though the initial condition $\phi_0(\mathbf{x})$ is smooth enough. The (weak) solution may not be unique unless suitable assumptions (viscosity solutions) are made, see, e.g. [4].

It is well known that HJ equations are closely related to conservation laws, so many successful numerical methods for solving conservation laws can be easily adapted to solve HJ equations. Hu and Shu [6] proposed a discontinuous Galerkin (DG) method to solve the HJ equation. Li and Shu [10] reinterpreted and simplified the two dimensional method of Hu and Shu. Zhu and Qiu [22, 23] used HWENO method for the HJ equations on both structured and unstructured meshes. Tao and Qiu [17] made use of the central HWENO method to solve the HJ equation successfully. As most of the finite volume methods resort to solving the conservation laws for the derivatives of the solution that seems to be not direct, Cheng and Shu [2] designed DG methods to directly solve the HJ equation. Later, Yan and Osher [18] proposed a local DG (LDG) method to solve the HJ equation directly. In [3] Cheng and Wang improved the work in [2] by utilizing the Roe speed and entropy fix at the cell interface, and based on [3], Zheng and Qiu [20] developed HWENO schemes to directly solve the HJ equations.

Comparing to the finite volume or DG methods, which evolve cell averages or a complete polynomial, the finite difference methods, which evolve only point values, may be easier to implement

and more efficient in multi-dimensions. Now, let us briefly review the early works. Crandall and Lion [5] introduced first order monotone schemes which can converge to the viscosity solution. Osher and Shu [11] presented higher order finite difference ENO schemes, and Jiang and Peng [7] developed higher order finite difference weighted ENO (WENO) schemes. Qiu et al. [14, 12] put forward Hermite WENO (HWENO) schemes for HJ equations with Runge-Kutta or Lax-Wendroff time discretization. Finite difference schemes on unstructured meshes have also been designed. Abgrall [1] proposed first order monotone schemes on triangular meshes. Lafon and Osher [9] designed second order schemes. Higher order WENO schemes were developed by Zhang and Shu [19], and higher order HWENO schemes were proposed by Zhu and Qiu [21]. For a detailed review of high order HJ equations about both the finite difference and finite volume method, we recommend [15].

In this paper, following the methods developed in [13, 14, 12], both ϕ and its first derivative (e.g. ϕ_x in one dimension) at the grid points are used in the HWENO method to reconstruct point values of the derivatives, and both of them are evolved by time marching. Comparing with the classical WENO method developed by Jiang and Shu in [8], the HWENO method requires extra work and storage but it is much more compact with the same order of accuracy. Comparing with DG methods, the HWENO method could achieve high order accuracy by adaptive stencils and hence could maintain the essentially non-oscillatory property. Notice that the HJ equations cannot be written in a "conservation form", hence it would seem more natural to use finite difference methods based on point values instead of finite volume methods based on cell averages. In this way, we could avoid the costly multi-dimensional reconstructions and could use dimension-by-dimension interpolations, thereby reducing the computational cost and improving efficiency. It is important to use numerical fluxes (monotone Hamiltonians) to evolve the solution itself for HJ equations to ensure convergence to weak solutions. Since the derivatives of the HJ solution satisfy conservation laws, previous HWENO schemes tend to use conservative approximations to evolve them as well, hence causing complications and extra computational cost in multi-dimensional reconstruction. In this paper, we design a new HWENO scheme which evolves the solution itself through classical numerical

fluxes to ensure convergence to weak solutions, and evolves the derivatives of the solution in a non-conservative fashion, thereby allowing efficient dimension-by-dimension interpolation resulting in a gain of simplicity and efficiency. This strategy is similar to solvers for conservation laws, such as discontinuous Galerkin methods or Hermite-type methods, which evolve cell averages in a conservative fashion but evolves other degrees of freedom (such as slopes or point values at cell interfaces) in a non-conservative fashion, without affecting convergence to weak solutions. Extensive numerical experiments are performed to illustrate the good performance of our schemes.

This paper is organized as follows. In Section 2, we describe the detailed steps about the construction and implementation of the finite difference HWENO schemes in both one and two dimensions for the HJ equations. In Section 3, we present extensive numerical results to verify the accuracy, stability and resolution of our method. Finally, a conclusion is given in Section 4.

2 The numerical method for the Hamilton-Jacobi equations

In this section, we will give the framework of the schemes first and then the detailed steps of the HWENO reconstructions for both one and two dimensional Hamilton-Jacobi equations.

2.1 One dimensional Hamilton-Jacobi equations

The Hamilton-Jacobi equations in the one dimensional case can be written as:

$$\phi_t + H(\phi_x) = 0 \qquad \phi(x,0) = \phi_0(x) \qquad x \in [a,b]. \tag{2.1}$$

For simplicity, we consider a uniform mesh that is defined as $a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$. However, this assumption is not needed for our schemes. We denote $\phi_j = \phi(x_j, t)$ as the numerical approximation to the viscosity solution and $u_j = \phi_x(x_j, t)$ as the numerical approximation to its first derivative. Then, by taking the spatial derivative on both sides of (2.1), we obtain the following system of equations:

$$\begin{cases} \dfrac{d\phi_j}{dt} = - H(\phi_x)|_{x=x_j} \\ \dfrac{du_j}{dt} = - H_1(u_j)u_x|_{x=x_j} \end{cases} \tag{2.2}$$

where the $H_1(u) = \frac{\partial H}{\partial u}$. We replace $H(\phi_x)|_{x=x_j}$ with a suitable monotone numerical flux, denoted by $\widehat{H}(u_j^-, u_j^+)$, and we will use the simple local Lax-Friedrichs flux as an example in the sequel. For stability, we also split $H_1(u_j)$ into a positive part and a negative part locally, denoted by $H_1^+{}_j$ and $H_1^-{}_j$ respectively, and approximate u_x by upwind methods. Then, we have the following scheme:

$$\begin{cases} \dfrac{d\phi_j}{dt} = -\widehat{H}(u_j^-, u_j^+) \\ \dfrac{du_j}{dt} = -H_1^+{}_j u_x^-{}_j - H_1^-{}_j u_x^+{}_j \end{cases} \tag{2.3}$$

where

$$\widehat{H}(u_j^-, u_j^+) = H(\frac{u_j^- + u_j^+}{2}) - \frac{1}{2}\alpha(u_j^+ - u_j^-) \tag{2.4}$$

with $\alpha = \max_{u\in D} |H_1(u)|, \quad D = [\min(u_j^-, u_j^+), \max(u_j^-, u_j^+)]$, and

$$H_1^-{}_j = \frac{1}{2}(H_1(\frac{u_j^- + u_j^+}{2}) - |H_1(\frac{u_j^- + u_j^+}{2})|), \quad H_1^+{}_j = \frac{1}{2}(H_1(\frac{u_j^- + u_j^+}{2}) + |H_1(\frac{u_j^- + u_j^+}{2})|). \tag{2.5}$$

$u_j^\pm$ and $u_x^\pm{}_j$, the left and right limits of the point values of $u(x_j, t)$ and $u_x(x_j, t)$, will be approximated by the HWENO method which we will describe in detail in the next subsection. Notice that, even though we already have u_j as part of our numerical solution, we would still want to obtain upwind-biased approximations $u_j^\pm$ through the point values of ϕ and u nearby (of course without using u_j itself) in order to use the monotone flux to ensure stability and convergence to weak solutions.

We observe that either $H_1^-{}_j$ or $H_1^+{}_j$ is equal to zero. In practice, we need to compute $u_x^-{}_j$ only when $H_1^+{}_j$ is not equal to zero, and compute $u_x^+{}_j$ only when $H_1^-{}_j$ is not equal to zero. This will reduce the computational cost because only one of the $u_x^-{}_j$ and $u_x^+{}_j$ needs to be computed.

After the spatial discretization, we can rewrite the scheme as $\mathrm{U}_t = \mathcal{L}(\mathrm{U})$, where $\mathcal{L}$ denotes the operator of the spatial discretization, and then use the third-order total variation diminishing (TVD) Runge-Kutta time discretization [16] to solve the semi-discrete form (2.3):

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

2.2 HWENO reconstruction in one dimension

In this subsection, we will describe the HWENO reconstruction procedure for $u_j^{\pm} = \phi_{x\,j}^{\pm}$ and $u_{x\,j}^{\pm}$.

Step 1. Reconstruction of $\phi_{x\,j}^{-}$ by the HWENO method from the point values $\{\phi_i, u_i\}$

1.1. Given the small stencils $S_0 = \{x_{j-2}, x_{j-1}, x_j\}$, $S_1 = \{x_{j-1}, x_j, x_{j+1}\}$, $S_2 = \{x_{j-2}, x_{j-1}, x_j, x_{j+1}\}$, and the big stencil $\mathcal{T} = \{S_0, S_1, S_2\}$, we construct Hermite cubic polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$, and a fifth-degree polynomial $q(x)$ such that:

$$
\begin{aligned}
&p_0(x_{j+i}) = \phi_{j+i}, \quad i = -2, -1, 0, && p_0'(x_{j-1}) = u_{j-1} \\
&p_1(x_{j+i}) = \phi_{j+i}, \quad i = -1, 0, 1, && p_1'(x_{j+1}) = u_{j+1} \\
&p_2(x_{j+i}) = \phi_{j+i}, \quad i = -2, -1, 0, 1, && \\
&q(x_{j+i}) = \phi_{j+i}, \quad i = -2, -1, 0, 1, && q'(x_{j\pm1}) = u_{j\pm1}
\end{aligned}
$$

In fact, we only need the derivative values of these polynomials at the point $x = x_j$, which have the following expressions:

$$
\begin{aligned}
p_{0x}(x_j) &= -\frac{\phi_{j-2} + 4\phi_{j-1} - 5\phi_i + 4u_{j-1}\Delta x}{2\Delta x} \\
p_{1x}(x_j) &= -\frac{\phi_{j-1} + 4\phi_j - 5\phi_{j+1} + 2u_{j+1}\Delta x}{4\Delta x} \\
p_{2x}(x_j) &= \frac{\phi_{j-2} - 6\phi_{j-1} + 3\phi_j + 2\phi_{j+1}}{6\Delta x} \\
q_x(x_j) &= -\frac{\phi_{j-2} + 18\phi_{j-1} - 9\phi_j - 10\phi_{j+1} + 9u_{j-1}\Delta x + 3u_{j+1}\Delta x}{18\Delta x}
\end{aligned}
$$

1.2. For each small stencil $S_m, m = 0, 1, 2$, we compute the smoothness indicator, which measures the smoothness of the polynomials in each stencil: the smaller the indicator is, the smoother the polynomial is in the stencil. We use the formula similar to [8] to figure out the indicators, denoted by $\beta_m, m = 0, 1, 2$:

$$
\beta_m = \sum_{l=2}^{3} \Delta x^{2l-1} \left(\frac{\partial^l}{\partial x^l} p_m(x_j)\right)^2 \qquad m = 0, 1, 2 \tag{2.7}
$$

1.3. We compute the linear weights, denoted by $\gamma_m, m = 0, 1, 2$, satisfying:

$$q_x(x_j) = \sum_{m=0}^{2} \gamma_m p_{mx}(x_j)$$

for all the point values of $\{\phi_i\}$ and $\{u_i\}$ in the big stencil $\mathcal{T}$, which leads to

$$\gamma_0 = \frac{1}{4}, \quad \gamma_1 = \frac{1}{3}, \quad \gamma_2 = \frac{5}{12}$$

1.4. We compute the nonlinear weights based on the linear weights and the smoothness indicators by:

$$\omega_m = \frac{\bar{\omega}_m}{\sum_{k=0}^{2} \bar{\omega}_k}, \qquad \bar{\omega}_m = \frac{\gamma_m}{(\beta_m + \varepsilon)^2}, \qquad m = 0, 1, 2 \tag{2.8}$$

where ε is a small positive number to avoid the denominator becoming zero. In our numerical tests, we use $\varepsilon = 10^{-6}$. The final HWENO approximation expression is:

$$\phi^-_{x\,j} = \sum_{m=0}^{2} \omega_m p_{mx}(x_j).$$

The reconstruction of u_j^+ is mirror symmetric with respect to x_j of the above procedure.

Step 2. Reconstruction of $u^-_{x\,j}$ by the HWENO method from the point values $\{\phi_i, u_i\}$

2.1. Given the small stencils $S_0 = \{x_{j-2}, x_{j-1}, x_j\}$, $S_1 = \{x_{j-1}, x_j, x_{j+1}\}$, $S_2 = \{x_{j-2}, x_{j-1}, x_j, x_{j+1}\}$, and the big stencil $\mathcal{T} = \{S_0, S_1, S_2\}$, we construct Hermite quartic polynomials $p_0(x)$, $p_1(x)$, $p_2(x)$, and a sixth-degree polynomial $q(x)$ such that:

$$\begin{aligned}
&p_0(x_{j+i}) = \phi_{j+i}, \quad i = -2, -1, 0, && p_0'(x_{j+i}) = u_{j+i}, \quad i = -1, 0 \\
&p_1(x_{j+i}) = \phi_{j+i}, \quad i = -1, 0, 1, && p_1'(x_{j+i}) = u_{j+i}, \quad i = 0, 1 \\
&p_2(x_{j+i}) = \phi_{j+i}, \quad i = -2, -1, 0, 1, && p_1'(x_j) = u_j \\
&q(x_{j+i}) = \phi_{j+i}, \quad i = -2, -1, 0, 1, && q'(x_{j+i}) = u_{j+1}, \quad i = -1, 0, 1
\end{aligned}$$

Again, we only need the second order derivative values of these polynomials at the point x_j, which

have the following expressions:

$$p_{0xx}(x_j) = \frac{10u_j\Delta x + 8u_{j-1}\Delta x + \phi_{j-2} + 16\phi_{j-1} - 17\phi_j}{2(\Delta x)^2}$$
$$p_{1xx}(x_j) = \frac{-4u_j\Delta x - 2u_{j+1}\Delta x + \phi_{j-1} - 8\phi_j + 7\phi_{j+1}}{2(\Delta x)^2}$$
$$p_{2xx}(x_j) = \frac{6u_j\Delta x - \phi_{j-2} + 12\phi_{j-1} - 15\phi_j + 4\phi_{j+1}}{6(\Delta x)^2}$$
$$q_{xx}(x_j) = \frac{18u_j\Delta x + 18u_{j-1}\Delta x - 6u_{j+1}\Delta x + \phi_{j-2} + 54\phi_{j-1} - 81\phi_j + 26\phi_{j+1}}{18(\Delta x)^2}$$

2.2. For each small stencil $S_m, m = 0, 1, 2$, we compute the smoothness indicators respectively:

$$\beta_m = \sum_{l=3}^{4} \Delta x^{2l-1} (\frac{\partial^l}{\partial x^l} p_m(x_j))^2, \qquad m = 0, 1, 2 \tag{2.9}$$

2.3. We compute the linear weights, denoted by $\gamma_m, m = 0, 1, 2$, satisfying:

$$q_{xx} = \sum_{m=0}^{2} \gamma_m p_{mxx}(x_j)$$

for all the point values of $\{\phi_i\}$ and $\{u_i\}$ in the big stencil $\mathcal{T}$, which leads to

$$\gamma_0 = \frac{1}{4}, \quad \gamma_1 = \frac{1}{3}, \quad \gamma_2 = \frac{5}{12}$$

2.4. We compute the nonlinear weights ω_m as in (2.8), the final HWENO approximation expression is:

$$u^-_{x\,j} = \sum_{m=0}^{2} \omega_m p_{mxx}(x_j).$$

The reconstruction of $u^+_{x\,j}$ is mirror symmetric with respect to x_j of the above procedure.

2.3 Two dimensional Hamilton-Jacobi equations

The Hamilton-Jacobi equation in the two dimensional case is written as

$$\phi_t + H(\phi_x, \phi_y) = 0 \qquad \phi(x, y, 0) = \phi_0(x, y) \qquad (x, y) \in [a, b] \times [c, d] \tag{2.10}$$

For simplicity, we also assume the computational domain has been uniformly meshed as $a = x_0 < x_1 < \cdots < x_{N-1} < x_N = b$ and $c = y_0 < y_1 < \cdots < y_{N-1} < y_N = d$. Furthermore, we define $\phi_{ij} = \phi(x_i, y_j, t)$ as the numerical approximation to the viscosity solution, and $u_{ij} = \phi_x(x_i, y_j, t)$

and $v_{ij} = \phi_y(x_i, y_j, t)$ as the numerical approximations to its first order partial derivatives with respect to the variables x and y, respectively. Then, by taking spatial derivatives on both sides of (2.10), we have the following system of equations:

$$\begin{cases} \frac{d\phi_{ij}}{dt} = - H(\phi_x, \phi_y) \\ \frac{du_{ij}}{dt} = - H_1(u,v)u_x - H_2(u,v)v_x \\ \frac{dv_{ij}}{dt} = - H_1(u,v)u_y - H_2(u,v)v_y \end{cases} \tag{2.11}$$

where the $H_1(u) = \frac{\partial H}{\partial u}$ and $H_2(u) = \frac{\partial H}{\partial v}$. As u_y is equal to v_x in the smooth case, we can also rewrite the system of equations as the following:

$$\begin{cases} \frac{d\phi_{ij}}{dt} = - H(\phi_x, \phi_y) \\ \frac{du_{ij}}{dt} = - H_1(u,v)u_x - H_2(u,v)u_y \\ \frac{dv_{ij}}{dt} = - H_1(u,v)v_x - H_2(u,v)v_y \end{cases} \tag{2.12}$$

In the same way as before, we replace $H(\phi_x, \phi_y)$ with a monotone numerical flux, and split $H_1(u,v)$ and $H_2(u,v)$ into a positive part and a negative part respectively. Then we discretize (2.12) into the following

$$\begin{cases} \frac{d\phi_{ij}}{dt} = - \widehat{H}(u_{ij}^-, u_{ij}^+, v_{ij}^-, v_{ij}^+) \\ \frac{du_{ij}}{dt} = - H_1^+{}_{ij} u_x^-{}_{ij} - H_1^-{}_{ij} u_x^+{}_{ij} - \overline{H_2}_{ij} u_{y_{ij}} \\ \frac{dv_{ij}}{dt} = - \overline{H_1}_{ij} v_{xij} - H_2^+{}_{ij} v_y^-{}_{ij} - H_2^-{}_{ij} v_y^+{}_{ij} \end{cases} \tag{2.13}$$

where $\widehat{H}(u_{ij}^-, u_{ij}^+, v_{ij}^-, v_{ij}^+)$ refers to a two-dimensional monotone flux, such as the simple local Lax-Friedrichs flux defined as

$$\widehat{H}(u_{ij}^-, u_{ij}^+, v_{ij}^-, v_{ij}^+) = H(\frac{u_{ij}^- + u_{ij}^+}{2}, \frac{v_{ij}^- + v_{ij}^+}{2}) - \frac{1}{2}\alpha(u_{ij}^+ - u_{ij}^-) - \frac{1}{2}\beta(v_{ij}^+ - v_{ij}^-),$$

where $\alpha = \max_{u \in D, v \in E} |H_1(u,v)|$ and $\beta = \max_{u \in D, v \in E} |H_2(u,v)|$. With regard to α, we take D as a local region and E as a global region, namely $D = [\min(u_{ij}^-, u_{ij}^+), \max(u_{ij}^-, u_{ij}^+)]$ and $E = [\min(v^-, v^+), \max(v^-, v^+)]|_{[c,d]}$. We compute the coefficient β similarly, except that we then take D globally and E locally. This way of computing the local Lax-Friedrichs flux is needed to ensure

monotonicity, see [11]. $\overline{H_1}_{ij}$, $\overline{H_2}_{ij}$ are the average values of H_1 and H_2 at (x_i, y_j) defined as the following:

$$\overline{H_1}_{ij} = H_1(\frac{u_{ij}^- + u_{ij}^+}{2}, \frac{v_{ij}^- + v_{ij}^+}{2}), \quad \overline{H_2}_{ij} = H_2(\frac{u_{ij}^- + u_{ij}^+}{2}, \frac{v_{ij}^- + v_{ij}^+}{2}),$$

and H_1^-, H_1^+, H_2^-, H_2^+ as the negative and positive parts of H_1 and H_2 respectively, defined as follows:

$$H_1^-{}_{ij} = \frac{1}{2}(\overline{H_1}_{ij} - |\overline{H_1}_{ij}|), \quad H_1^+{}_{ij} = \frac{1}{2}(\overline{H_1}_{ij} + |\overline{H_1}_{ij}|),$$
$$H_2^-{}_{ij} = \frac{1}{2}(\overline{H_2}_{ij} - |\overline{H_2}_{ij}|), \quad H_2^+{}_{ij} = \frac{1}{2}(\overline{H_2}_{ij} + |\overline{H_2}_{ij}|).$$

$\phi_x^\pm{}_{ij}$, $u_x^\pm{}_{ij}$ and $v_x^\pm{}_{ij}$ are the left and right limits of the point values $u(x_i, y_j, t)$, $u_x(x_i, y_j, t)$ and $v_x(x_i, y_j, t)$ with respect to the variable x, and $\phi_y^\pm{}_{ij}$, $u_y^\pm{}_{ij}$ and $v_y^\pm{}_{ij}$ are the left and right limits of the point values $v(x_i, y_j, t)$, $u_y(x_i, y_j, t)$ and $v_y(x_i, y_j, t)$ with respect to the variable y. The values of $\phi_x^\pm{}_{ij}$, $\phi_y^\pm{}_{ij}$, $u_x^\pm{}_{ij}$ and $v_y^\pm{}_{ij}$ are reconstructed with the one dimensional method in each direction with the other direction fixed. As to the mixed derivatives v_x, u_y, we simply use the fourth order central approximations in the x and y directions, for there is reason to believe the mixed derivatives play lesser role on spurious oscillations according to [14]:

$$v_{xij} = \frac{-v_{i+2,j} + 8v_{i+1,j} - 8v_{i-1,j} + v_{i-2,j}}{12\Delta x},$$
$$u_{y_{ij}} = \frac{-u_{i,j+2} + 8u_{i,j+1} - 8u_{i,j-1} + u_{i,j-2}}{12\Delta y}.$$

In practice, we only need to compute either $u_x^-{}_{ij}$ or $u_x^+{}_{ij}$ but not both. $v_y^-{}_{ij}$ and $v_y^+{}_{ij}$ can be computed similarly.

Then, we can rewrite the scheme (2.13) as $\mathrm{U}_t = \mathcal{L}(\mathrm{U})$, where $\mathcal{L}$ denotes the operator of spatial discretization, and use the third-order total variation diminishing (TVD) Runge-Kutta time discretization (2.6) to solve the semi-discrete form (2.13).

3 Numerical results

In this section, we provide numerical experiments for the fifth order HWENO method in one and two dimensional cases. In all the accuracy tests, we set $\Delta t = 0.6\Delta x^{\frac{5}{3}}/\alpha$ in the one dimensional case

and $\Delta t = \frac{1}{\frac{\alpha}{0.6\Delta x^{\frac{5}{3}}}+\frac{\beta}{0.6\Delta y^{\frac{5}{3}}}}$ in the two dimensional case, in order to guarantee that spatial numerical errors dominate. In other tests, we simply take $\Delta t = 0.6\Delta x/\alpha$ and $\Delta t = \frac{1}{\frac{\alpha}{0.6\Delta x}+\frac{\beta}{0.6\Delta y}}$ in the one and two dimensional cases respectively, unless otherwise indicated. The HWENO method for HJ equations with conservative approximations to the derivative variables (HWENO-C), developed by Qiu and Shu [14], using the same mesh partition, time steps and local Lax-Friedrichs flux, is used for comparison in some of the numerical examples.

In Table 1, we provide a CPU time comparison between the method developed in this paper which uses non-conservative approximation for the derivative variables (HWENO-NC) and the HWENO-C method for accuracy tests in Examples 3.1, 3.2, 3.3 and 3.4 below. The total CPU time for $N = 10, 20, 40, 80, 160$ and 320 cells is recorded. We observe that the HWENO-NC method spends less CPU time than the HWENO-C method. The computations are performed on a Dell Vostro 2420 with 4GB RAM.

Table 1: CPU time (in seconds) for the HWENO-NC and HWENO-C schemes.

	Example 3.1	Example 3.2	Example 3.3	Example 3.4
HWENO-NC	2.2152	0.4680	0.1092	246.9964
HWENO-C	3.5568	0.6708	0.1248	268.7273

Example 3.1: We solve the following linear scalar equation:

$$\phi_t + \phi_x = 0, \qquad 0 \le x \le 2$$

with the initial condition $\phi(x, 0) = \sin(\pi x)$, and periodic boundary condition. We compute the solution up to $t = 2$, the numerical errors and numerical orders of accuracy for the HWENO-NC method are shown in Table 2. We can see that the scheme achieves or exceeds fifth order accuracy, and it actually yields smaller errors than the HWENO-C method.

Example 3.2: We solve the Burgers equation:

$$\phi_t + \frac{1}{2}(\phi_x + 1)^2 = 0, \qquad -1 \le x \le 1$$

with the initial condition $\phi(x, 0) = -\cos(\pi x)$, and periodic boundary condition. We compute the

Table 2: $\phi_t + \phi_x = 0, \phi(x,0) = \sin(\pi x)$. Periodic boundary conditions. $t = 2$

	HWENO-NC				HWENO-C			
N	L_∞ error	order	L_1 error	order	L_∞ error	order	L_1 error	order
10	2.58E-02		1.06E-02		3.49E-02		2.58E-02	
20	9.42E-04	4.77	4.40E-04	4.59	1.76E-03	4.31	1.16E-03	4.48
40	2.75E-05	5.10	1.56E-05	4.81	7.28E-05	4.59	4.49E-05	4.69
80	8.34E-07	5.05	4.95E-07	4.98	2.48E-06	4.87	1.54E-06	4.86
160	1.94E-08	5.43	1.02E-08	5.61	5.99E-08	5.37	3.62E-08	5.41
320	1.68E-10	6.86	8.66E-11	6.88	5.41E-10	6.79	3.22E-10	6.82

solution up to $t = \frac{0.5}{\pi^2}$. At this time, the solution is still smooth. The numerical results are shown in Table 3. Again, we can see that the scheme can reach its designed order of accuracy, and actually yields smaller errors than the HWENO-C method for the more refined meshes.

Table 3: $\phi_t + \frac{1}{2}(\phi_x + 1)^2 = 0, \phi(x,0) = -\cos(\pi x)$. Periodic boundary conditions. $t = 0.5/\pi^2$

	HWENO-NC				HWENO-C			
N	L_∞ error	order	L_1 error	order	L_∞ error	order	L_1 error	order
10	2.20E-03		8.04E-04		1.76E-03		6.95E-04	
20	1.64E-04	3.75	3.24E-05	4.63	1.15E-04	3.94	2.74E-05	4.66
40	1.27E-05	3.69	1.88E-06	4.11	1.50E-05	2.94	1.81E-06	3.92
80	5.10E-07	4.63	5.95E-08	4.98	6.25E-07	4.58	6.63E-08	4.77
160	1.70E-08	4.91	1.05E-09	5.82	2.11E-08	4.89	1.34E-09	5.63
320	2.87E-10	5.88	1.93E-11	5.76	4.08E-10	5.69	2.96E-11	5.50

Example 3.3: We solve the nonlinear scalar equation:

$$\phi_t - \cos(\phi_x + 1) = 0, \qquad -1 < x < 1$$

with the initial condition $\phi(x,0) = -\cos(\pi x)$, and periodic boundary condition. When $t = \frac{0.5}{\pi^2}$, the solution is still smooth. The numerical results are shown in Table 4. Again, we observe that the scheme can achieve its designed accuracy, and actually yields smaller errors than the HWENO-C method for the more refined meshes.

Example 3.4: We solve the two dimensional Burgers equation

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

with the initial data $\phi(x,y,0) = -\cos(\frac{\pi}{2}(x+y))$ and periodic boundary condition. We compute the result up to $t = 0.5/\pi^2$ and the solution is still smooth at that time. Again, from Table 5, we

Table 4: $\phi_t - \cos(\phi_x + 1) = 0, \phi(x, 0) = -\cos(\pi x)$. Periodic boundary conditions. $t = 0.5/\pi^2$

	HWENO-NC				HWENO-C			
N	L_∞ error	order	L_1 error	order	L_∞ error	order	L_1 error	order
10	1.54E-03		7.38E-04		1.91E-03		8.74E-04	
20	1.68E-04	3.20	2.96E-05	4.64	2.12E-04	3.17	4.62E-05	4.24
40	1.64E-05	3.35	1.96E-06	3.92	2.34E-05	3.18	2.66E-06	4.12
80	9.62E-07	4.10	7.24E-08	4.76	1.31E-06	4.16	1.08E-07	4.62
160	5.49E-08	4.13	2.64E-09	4.78	7.19E-08	4.19	4.06E-09	4.73
320	1.40E-09	5.29	7.60E-11	5.12	2.07E-09	5.12	1.27E-10	5.00

can see that the scheme achieves its designed order. The errors we obtain are comparable with the one obtained by the HWENO-C method.

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. $t = 0.5/\pi^2$

	HWENO-NC				HWENO-C			
N	L_∞ error	order	L_1 error	order	L_∞ error	order	L_1 error	order
10× 10	1.78E-03		8.31E-04		1.83E-03		8.96E-04	
20× 20	3.64E-04	2.29	5.49E-05	3.92	4.19E-04	2.13	6.10E-05	3.88
40× 40	3.21E-05	3.51	2.97E-06	4.21	3.62E-05	3.54	3.16E-06	4.27
80× 80	1.36E-06	4.56	1.06E-07	4.81	1.46E-06	4.63	1.11E-07	4.84
160× 160	4.51E-08	4.91	3.17E-09	5.06	4.64E-08	4.98	2.61E-09	5.41
320× 320	9.86E-10	5.51	8.71E-11	5.19	8.74E-10	5.73	4.99E-11	5.71

Example 3.5: We solve the linear equation:

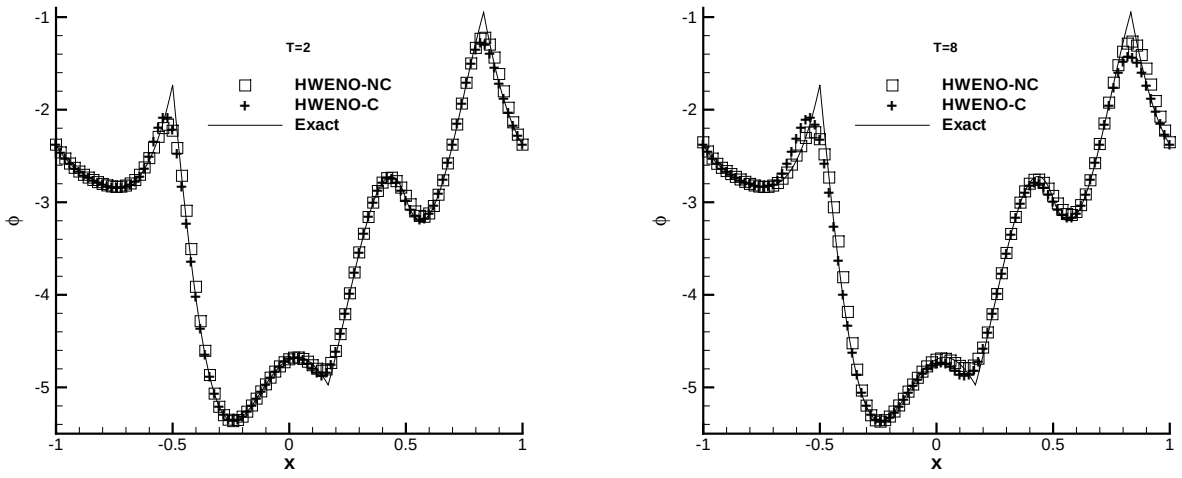
$$\phi_t + \phi_x = 0$$

with the initial condition $\phi(x, 0) = \phi_0(x - 0.5)$ together with the periodic boundary condition, where

$$\phi_0(x) = -(\frac{\sqrt{3}}{2} + \frac{9}{2} + \frac{2\pi}{3})(x + 1) + \begin{cases} 2\cos(\frac{3\pi x^2}{2}) - \sqrt{3} & -1 \le x < -\frac{1}{3}, \\ \frac{3}{2} + 3\cos(2\pi x) & -\frac{1}{3} \le x < 0, \\ \frac{15}{2} - 3\cos(2\pi x) & 0 \le x < \frac{1}{3}, \\ \frac{28 + 4\pi + \cos(3\pi x)}{3} + 6\pi x(x - 1) & \frac{1}{3} \le x < 1. \end{cases}$$

We plot the results at $t = 2.0$ and $t = 8.0$ in Figure 1(a) and Figure 1(b) respectively, and observe

that our scheme can reach comparable resolution for the corner singularity with the one obtained by the HWENO-C method.



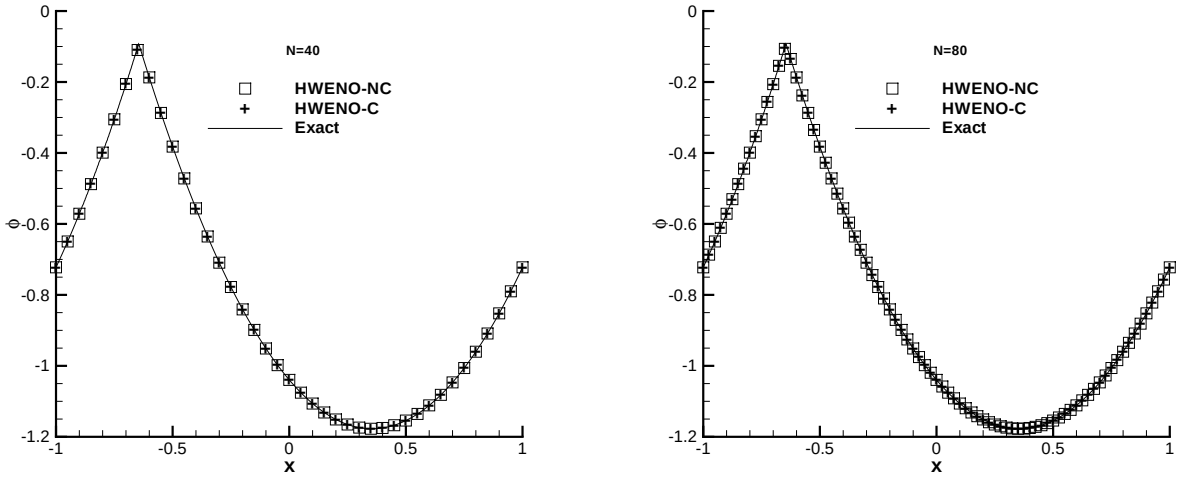
(a) (b)

Figure 1: One dimensional linear equation. $N = 100$ cells. (a) $t = 2.0$, (b) $t = 8.0$. Solid line: the exact solution; Square symbol: the HWENO-NC scheme; Plus symbol: the HWENO-C scheme.

Example 3.6: We solve one dimensional Burgers equation:

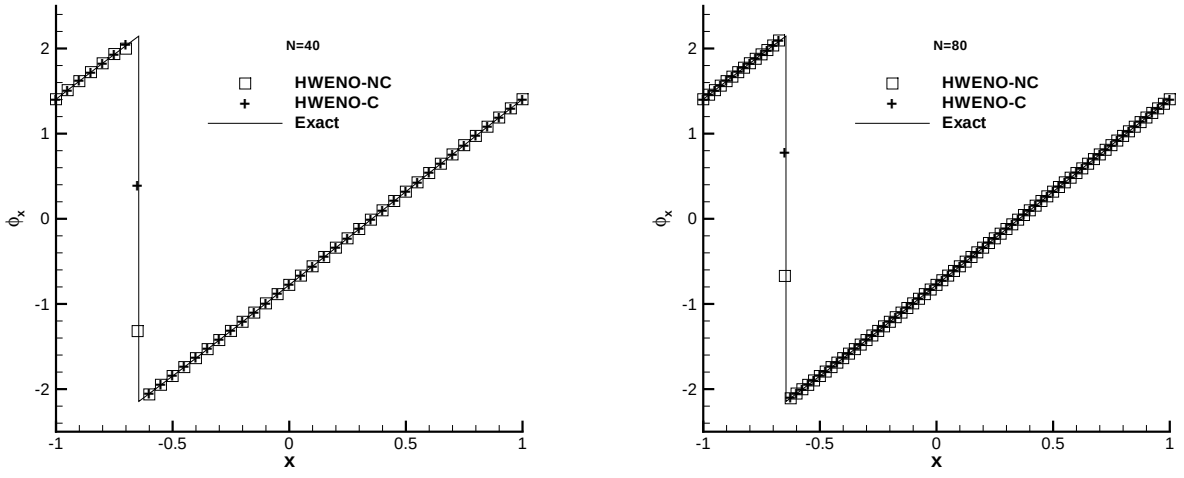
$$\phi_t + \frac{1}{2}(\phi_x + 1)^2 = 0 \qquad -1 \le x \le 1$$

with $\phi(x, 0) = -\cos(\pi x)$ and periodic boundary conditions. We compute the solution up to $t = 3.5/\pi^2$. At this time, the discontinuous derivative has already appeared in the solution. We show the numerical results with the meshes $N = 40$ and $N = 80$ in Figure 2(a) and Figure 2(b) respectively. We can see that both schemes give high resolution for this problem. We also plot the derivatives of schemes with the meshes $N = 40$ and $N = 80$ in Figure 3(a) and Figure 3(b) respectively, and we observe sharp resolution with correct shock location for both schemes. This verifies the claim that our non-conservative treatments of the solution derivatives does not affect convergence to weak solutions with correct shock speeds.



(a) (b)

Figure 2: One dimensional Burgers equation. $t = 3.5/\pi^2$. (a) $N = 40$ and (b) $N = 80$. Solid line: the exact solution; Square symbol: the HWENO-NC scheme; Plus symbol: the HWENO-C scheme.



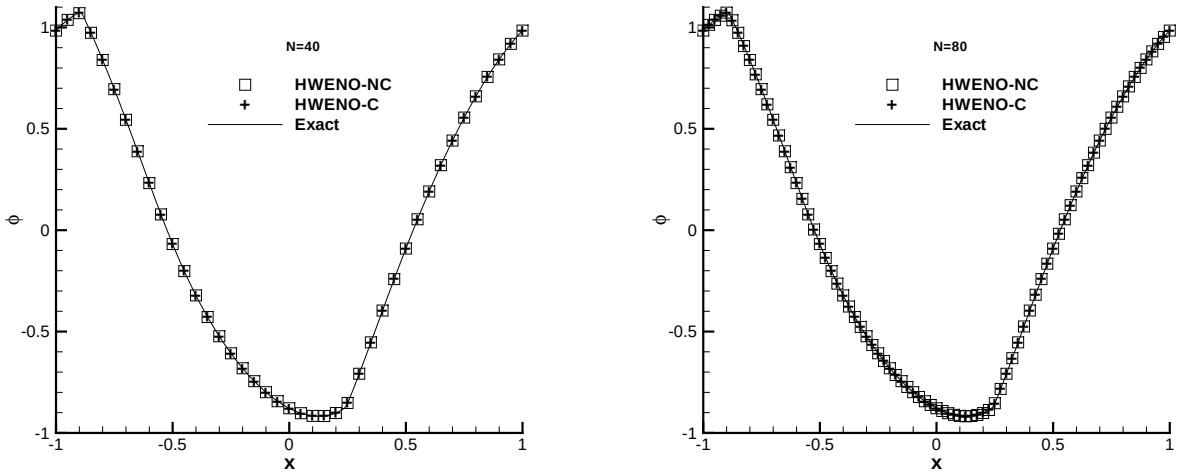
(a) (b)

Figure 3: The derivative of the one dimensional Burgers equation. $t = 3.5/\pi^2$. (a) $N = 40$ and (b) $N = 80$. Solid line: the exact solution; Square symbol: the HWENO-NC scheme; Plus symbol: the HWENO-C scheme.

Example 3.7: We solve the nonlinear equation with a non-convex / non-concave flux

$$\phi_t - \cos(\phi_x + 1) = 0$$

with the initial data $\phi(x, 0) = -\cos(\pi x)$ and periodic boundary conditions. This time, we compute the solution up to $t = 1.5/\pi^2$. We observe the results with $N = 40$ and $N = 80$, which are shown in Figure 4, and find that both methods give high resolution in this case. We also plot the derivatives of the schemes with the meshes $N = 40$ and $N = 80$ in Figure 5, and observe that both give correct shock location and good resolution.



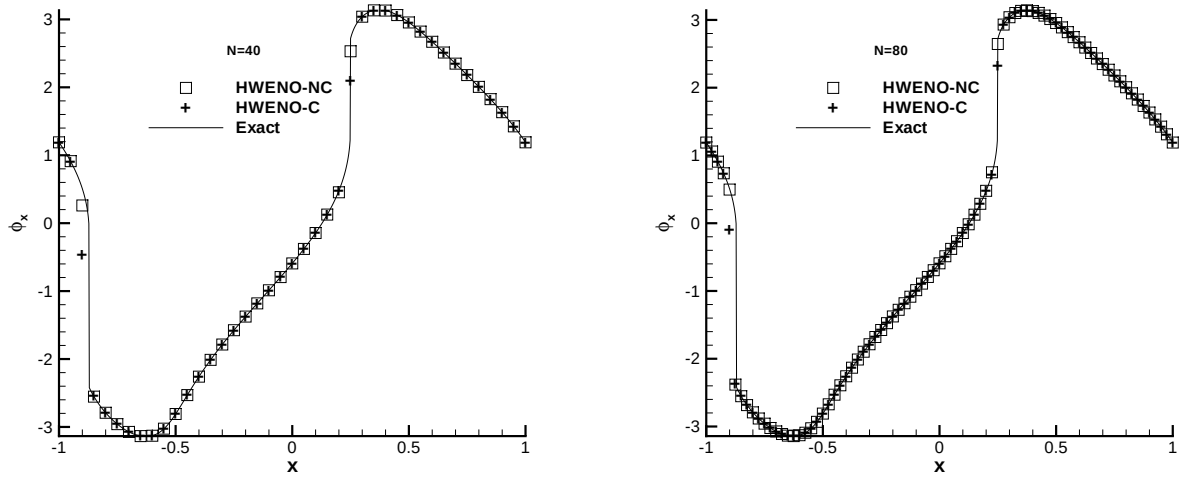
(a) (b)

Figure 4: $H(u) = -\cos(u + 1)$. $t = 1.5/\pi^2$. (a) $N = 40$ and (b) $N = 80$. Solid line: the exact solution; Square symbol: the HWENO-NC scheme; Plus symbol: the HWENO-C scheme.

Example 3.8: We solve the problem

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

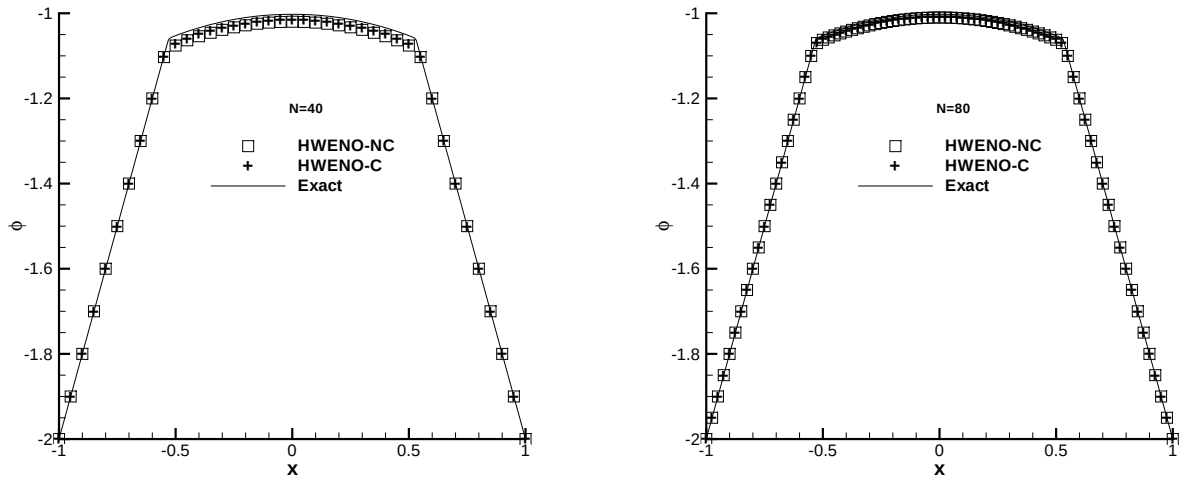
with the initial data $\phi(x, 0) = -2|x|$. As the derivative of $\phi(x, 0)$ is discontinuous, the initial value of u is undefined at $x = 0$ which is a grid point. We simply take $u(0, 0) = 0$ in our code, and observe that the value of $u(0, 0)$ makes little influence on the final result in our numerical experiment. We



(a) (b)

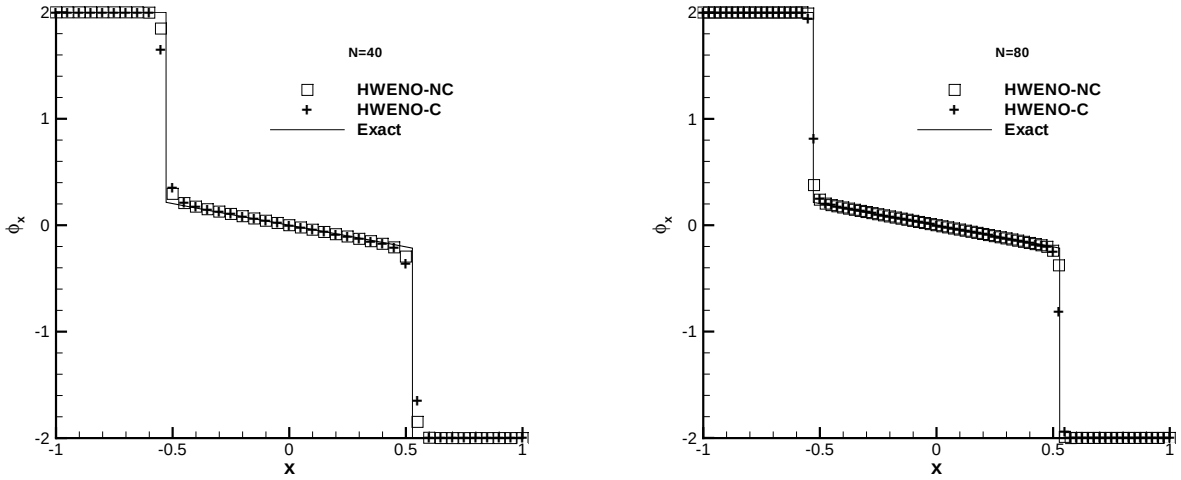
Figure 5: The derivative plot, $H(u) = -\cos(u+1)$. $t = 1.5/\pi^2$. (a) $N = 40$ and (b) $N = 80$. Solid line: the exact solution; Square symbol: the HWENO-NC scheme; Plus symbol: the HWENO-C scheme.

plot the results at $t = 1$ with $N = 40$ and $N = 80$ cells in Figure 6 and Figure 7, observe that both the function values and its derivatives give good results for both schemes.



(a) (b)

Figure 6: $H(u) = (1/4)(u^2-1)(u^2-4)$. $t = 1$. (a) $N = 40$ and (b) $N = 80$. Solid line: the exact solution; Square Symbol: the HWENO-NC scheme; Plus symbol: the HWENO-C scheme.



(a) (b)

Figure 7: The derivatives plot, $H(u) = (1/4)(u^2-1)(u^2-4)$. $t = 1$. (a) $N = 40$ and (b) $N = 80$. Solid line: the exact solution; Square Symbol: the HWENO-NC scheme; Plus symbol: the HWENO-C scheme.

Example 3.9: We solve the two dimensional Burgers equation

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

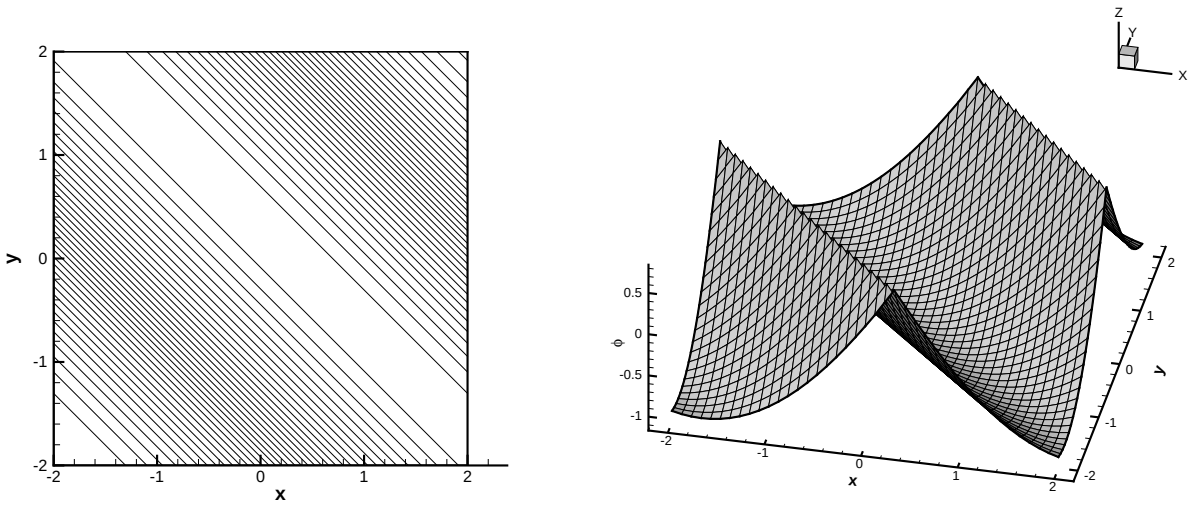
with the initial data $\phi(x, y, 0) = -\cos(\frac{\pi}{2}(x + y))$ and periodic boundary condition. We compute the result up to $t = 1.5/\pi^2$ and the derivative discontinuity has appeared in the solution. We plot the results with 40×40 cells in Figure 8 and observe high resolution in this example.

Example 3.10: 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. In this case, the equation can be denoted as $\phi_t + H(\phi_x, \phi_y, x, y) = 0$, and our scheme can be obtained through the following system of equations as before:

$$\begin{cases} \dfrac{d\phi_{ij}}{dt} = -H(\phi_x, \phi_y, x, y) \\ \dfrac{du_{ij}}{dt} = -H_1(u, v, x, y)u_x - H_2(u, v, x, y)u_y - H_x(u, v, x, y) \\ \dfrac{dv_{ij}}{dt} = -H_1(u, v, x, y)v_x - H_2(u, v, x, y)v_y - H_y(u, v, x, y) \end{cases}$$



(a) (b)

Figure 8: Two dimensional Burgers equation. $t = 1.5/\pi^2$ with the HWENO-NC schemes with 40×40 cells. Contours of the solution in Figure 8(a) and the surface of the solution in Figure 8(b)

Following (2.13), we can easily deal with the terms $H(\phi_x, \phi_y, x, y), H_1(u, v, x, y), H_2(u, v, x, y)$. The additional terms $H_x(u, v, x, y)$ and $H_y(u, v, x, y)$ can be approximated as

$$H_x(u, v, x, y) \mid_{x=x_i, y=y_j} \approx H_x(\frac{u_{ij}^- + u_{ij}^+}{2}, \frac{v_{ij}^- + v_{ij}^+}{2}, x_i, y_j)$$

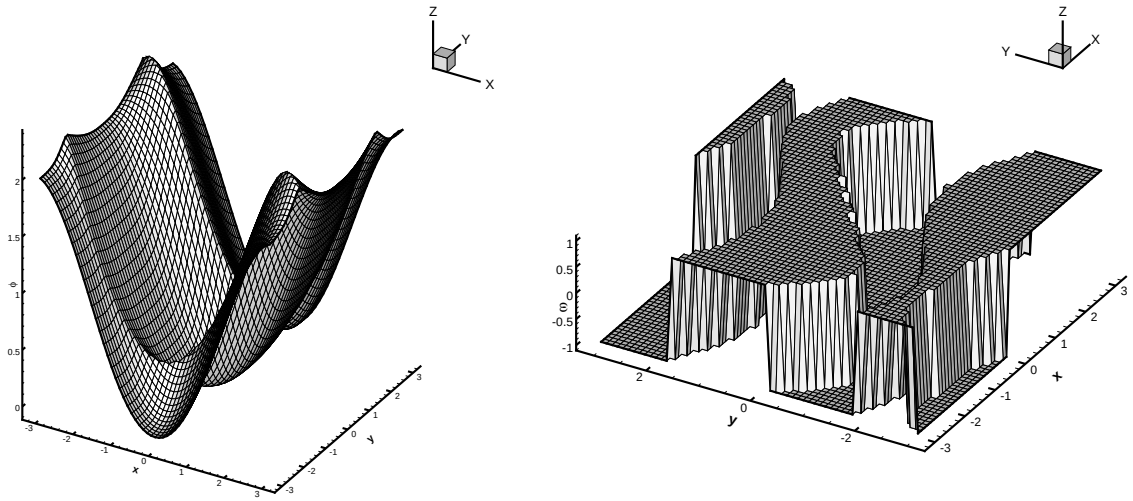
$$H_y(u, v, x, y) \mid_{x=x_i, y=y_j} \approx H_y(\frac{u_{ij}^- + u_{ij}^+}{2}, \frac{v_{ij}^- + v_{ij}^+}{2}, x_i, y_j)$$

The time step is still taken as $\Delta t = \frac{1}{\frac{\alpha}{0.6\Delta x} + \frac{\beta}{0.6\Delta y}}$. The solution and optimal control $\omega = \text{sign}(\phi)$ at $t = 1$ are plotted in Figure 9(a) and Figure 9(b), respectively. Again, we can observe that high resolution is achieved by our method.

Example 3.11: We solve the problem with another neither convex nor concave Hamiltonian

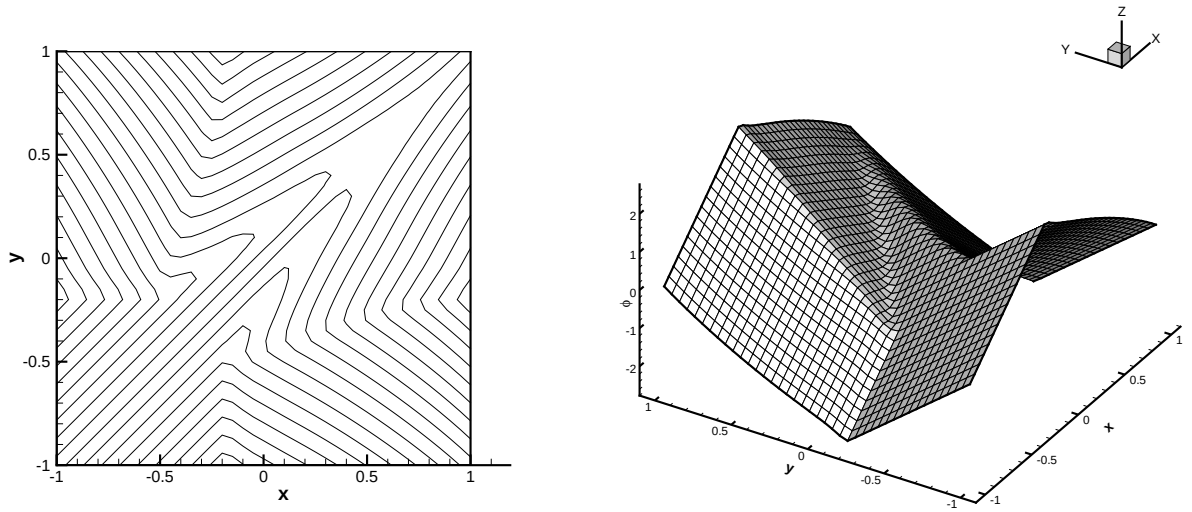
$$\phi_t + \sin(\phi_x + \phi_y) = 0, \qquad -1 < x, y < 1$$

with $\phi(x, y, 0) = \pi(|y| - |x|)$. In this case, $u(x, y, 0)$ is undefined along $x = 0$ and $v(x, y, 0)$ is undefined along $y = 0$, and we simply take $u(0, y, 0) = 0$ and $v(x, 0, 0) = 0$ respectively. We compute the solution up to $t = 1$. The solution is shown in Figure 10. Again, we observe our schemes can achieve high resolution.



(a) (b)

Figure 9: The optimal control problem. $t = 1$ with the HWENO-NC schemes with 60×60 cells. Surface of the solution in Figure 9(a) and of the optimal control $\omega = sign(\phi_y)$ in Figure 9(b)



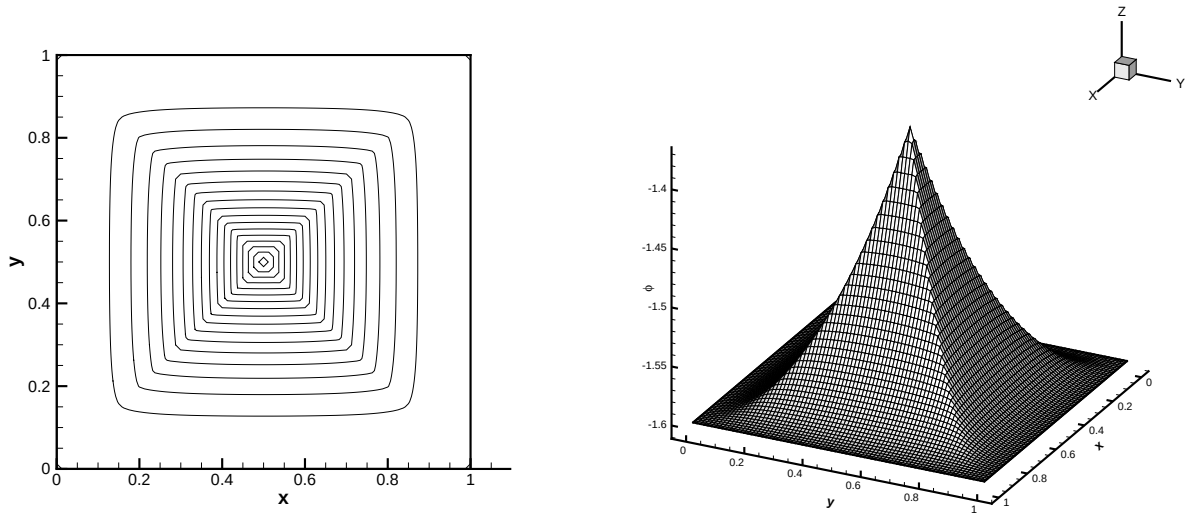
(a) (b)

Figure 10: Two dimensional equation with a neither convex nor concave Hamiltonian. $t = 1$ by the HWENO-NC schemes with 40×40 cells. Contours of the solution in Figure 10(a) and surface of the solution in Figure 10(b)

Example 3.12: We solve the two dimensional Eikonal equation

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

with the initial data $\phi(x, y, 0) = \frac{1}{4}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1) - 1$. We compute the solution up to $t = 0.6$. The solution is shown in Figure 11. High resolutions are observed with our scheme.



(a) (b)

Figure 11: Eikonal equation. $t = 0.6$ by the HWENO-NC schemes with 60×60 cells. Contours of the solution in Figure 11(a) and surface of the solution in Figure 11(b)

Example 3.13: We solve

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

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 data $\phi(x, y, 0) = 1 - \frac{1}{4}(\cos(2\pi x) - 1)(\cos(2\pi y) - 1)$ and periodic boundary condition. When $\varepsilon \neq 0$, the equation can be denoted as $\phi_t + H(\phi_x, \phi_y, \phi_{xx}, \phi_{xy}, \phi_{yy}) =$

0. The system of equations to be approximated then becomes

$$\begin{cases} \dfrac{d\phi_{ij}}{dt} = -H(\phi_x, \phi_y, \phi_{xx}, \phi_{xy}, \phi_{yy}) \\ \dfrac{du_{ij}}{dt} = -H_{\phi_x}u_x - H_{\phi_y}u_y - H_{\phi_{xx}}u_{xx} - H_{\phi_{xy}}u_{xy} - H_{\phi_{yy}}u_{yy} \\ \dfrac{dv_{ij}}{dt} = -H_{\phi_x}v_x - H_{\phi_y}v_y - H_{\phi_{xx}}v_{xx} - H_{\phi_{xy}}v_{xy} - H_{\phi_{yy}}v_{yy} \end{cases}$$

The procedure to deal with the terms H, H_{ϕ_x} and H_{ϕ_y} are the same as before. Here, we simply use fourth order central differences in each direction with the other direction fixed to approximate the terms u_{xx}, u_{yy}:

$$u_{xx_{ij}} = -\frac{u_{i+2,j} - 16u_{i+1,j} + 30u_{i,j} - 16u_{i-1,j} + u_{i-2,j}}{12\Delta x^2}$$

$$u_{yy_{ij}} = -\frac{u_{i,j+2} - 16u_{i,j+1} + 30u_{i,j} - 16u_{i,j-1} + u_{i,j-2}}{12\Delta y^2}$$

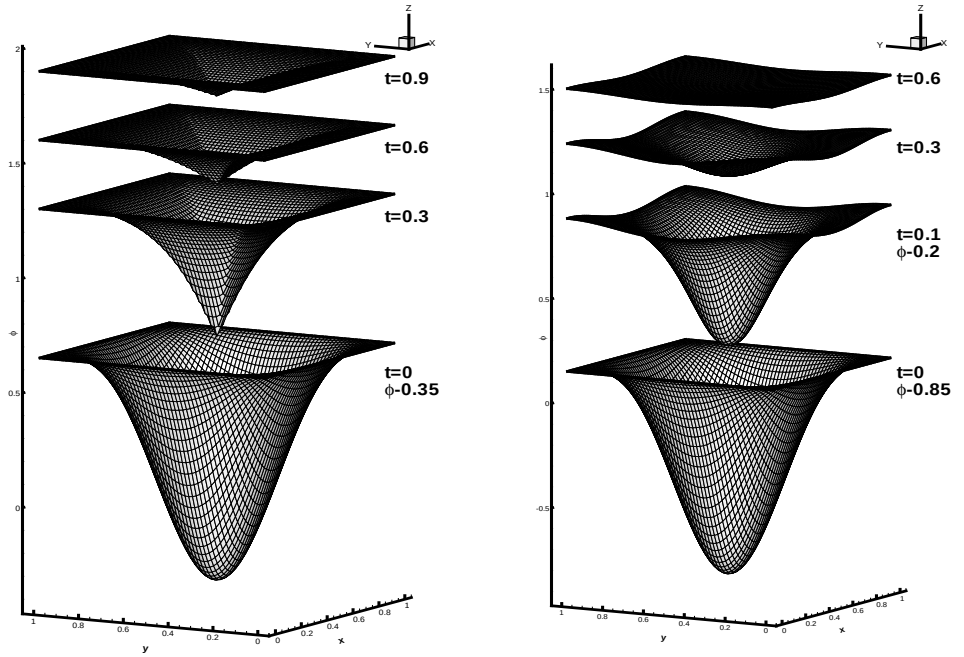
As to the term u_{xy}, we take the following $\mathbb{Q}_k$ polynomial approximation using the values $\{u_{kl}, k = i-2, \cdots, i+2, l = j-2, \cdots, j+2\}$:

$$\begin{aligned} u_{xy_{ij}} = -\frac{1}{144\Delta x\Delta y}(&u_{i-2,j-2} - 8u_{i-2,j-1} + 8u_{i-2,j+1} - u_{i-2,j+2} - 8u_{i-1,j-2} + 64u_{i-1,j-1} \\ &- 64u_{i-1,j+1} + 8u_{i-1,j+2} + 8u_{i+1,j-2} - 64u_{i+1,j-1} + 64u_{i+1,j+1} - 8u_{i+1,j+2} \\ &- u_{i+2,j-2} + 8u_{i+2,j-1} - 8u_{i+2,j+1} + u_{i+2,j+2}). \end{aligned}$$

The approximation to the terms v_{xx}, v_{xy}, v_{yy} can be obtained in a similar way. The time step is taken as $\Delta t = \frac{1}{\frac{\alpha}{0.6\Delta x} + \frac{\beta}{0.6\Delta y} + \frac{\gamma_1}{0.3\Delta x^2} + \frac{\gamma_2}{0.3\Delta x\Delta y} + \frac{\gamma_3}{0.3\Delta y^2}}$, where $\gamma_1 = \max_{\phi_{xx}} |H_{\phi_{xx}}|$, $\gamma_2 = \max_{\phi_{xy}} |H_{\phi_{xy}}|$, $\gamma_3 = \max_{\phi_{yy}} |H_{\phi_{yy}}|$ and α, β are defined the same as before. The results of $\varepsilon = 0$ (pure convection) and $\varepsilon = 0.1$ by the HWENO-NC method with 60×60 cells are presented in Figure 12(a) and Figure 12(b) respectively. 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.

4 Conclusion

In this paper, we present a high order scheme based on the finite difference framework for the Hamilton-Jacobi equations in one and two dimensions. The main advantage of this scheme is



(a) (b)

Figure 12: Propagating surface. 60×60 cells. Figure 12(a): $\varepsilon = 0$; Figure 12(b): $\varepsilon = 0.1$

its compactness and efficiency. Both the solution values and its first derivatives are used in the HWENO reconstruction and evolved via time marching. Extensive numerical experiments in one dimensional and two dimensional cases show that the scheme can achieve high order accuracy in the smooth region and can maintain high resolution when the derivative becomes discontinuous.

References

[1] R. Abgrall. Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes. *Commum. Pur. Appl. Math.*, 49:1339–1373, 1996.

[2] Y. Cheng and C.-W. Shu. A Discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations. *J. Comput. Phys.*, 223:398–415, 2007.

[3] Y. Cheng and Z. Wang. A new Discontinuous Galerkin finite element method for directly solving the Hamilton-Jacobi equations. *J. Comput. Phys.*, 268:134–153, 2014.

[4] M. Crandall and P. L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Trans. Americ. Math. Soc.*, 277:1–42, 1983.

[5] M. Crandall and P. L. Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Trans. Americ. Math. Soc.*, 43:1–19, 1984.

[6] C. Hu and C.-W. Shu. A Discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:666–690, 1999.

[7] G. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM J. Sci. Comput.*, 21:2126–2143, 1999.

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

[9] F. Lafon and S. Osher. High order two dimensional nonoscillatory methods for solving Hamilton-Jacobi scalar equations. *J. Comput. Phys.*, 123:235–253, 1996.

[10] F. Li and C.-W. Shu. Reinterpretation and simplied implementation of a Discontinuous Galerkin method for Hamilton-Jacobi equations. *Appl. Math. Lett.*, 18:1204–1209, 2005.

[11] S. Osher and C.-W. Shu. High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Numer. Anal*, 28:907–922, 1991.

[12] J. Qiu. Hermite WENO schemes with Lax-Wendroff type time discretizations for Hamilton-Jacobi equations. *J. Comput. Math.*, 25:131–144, 2007.

[13] J. Qiu and C.-W. Shu. Hermite WENO schemes and their application as limiters for Runge-Kutta Discontinuous Galerkin method: one-dimensional case. *J. Comput. Phys.*, 193:115–135, 2004.

[14] J. Qiu and C.-W. Shu. Hermite WENO schemes for Hamilton-Jacobi equations. *J. Comput. Phys.*, 204:82–99, 2005.

[15] C.-W. Shu. High order numerical methods for time dependent Hamilton-Jacobi equations. *in Mathematics and Computation in Imaging Science and Information Processing, S.S. Goh, A. Ron and Z. Shen, Editors, Lecture Notes Series, Institute for Mathematical Sciences, National University of Singapore, World Scientific Press*, 11:47–91, 2007.

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

[17] Z. Tao and J. Qiu. Dimension-by-dimension moment-based central Hermite WENO schemes for directly solving Hamilton-Jacobi equations. *Adv. Comput. Math.*, submitted.

[18] J. Yan and S. Osher. A local Discontinuous Galerkin method for directly solving Hamilton-Jacobi equations. *J. Comput. Phys.*, 230:232–244, 2011.

[19] Y. T. Zhang and C.-W. Shu. High order WENO schemes for Hamilton-Jacobi equations on triangular meshes. *SIAM J. Sci. Comput*, 24:1005–1030, 2003.

[20] F. Zheng and J. Qiu. Directly solving the Hamilton-Jacobi equations by Hermite WENO schemes. *J. Comput. Phys.*, 307:423–445, 2016.

[21] J. Zhu and J. Qiu. Hermite WENO schemes for Hamilton-Jacobi equations on unstructured meshes. *J. Comput. Phys.*, 254:76–92, 2013.

[22] J. Zhu and J. Qiu. Finite volume Hermite WENO schemes for solving the Hamilton-Jacobi equation. *Commun. Comput. Phys.*, 15:959–980, 2014.

[23] J. Zhu and J. Qiu. Finite volume Hermite WENO schemes for solving the Hamilton-Jacobi equations II: unstructured meshes. *Computers Math. Appl.*, 68:1137–1150, 2014.