# HIGH ORDER FINITE DIFFERENCE HERMITE WENO FAST SWEEPING METHODS FOR STATIC HAMILTON-JACOBI EQUATIONS*

Yupeng Ren

*School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, China*
*Email: ypren@stu.xmu.edu.cn*

Yulong Xing

*Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA*
*Email: xing.205@osu.edu*

Jianxian Qiu[1]

*School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling*
*and High-Performance Scientific Computing, Xiamen University, Xiamen, Fujian 361005, China*
*Email: jxqiu@xmu.edu.cn*

## Abstract

In this paper, we propose a novel Hermite weighted essentially non-oscillatory (HWENO) fast sweeping method to solve the static Hamilton-Jacobi equations efficiently. During the HWENO reconstruction procedure, the proposed method is built upon a new finite difference fifth order HWENO scheme involving one big stencil and two small stencils. However, one major novelty and difference from the traditional HWENO framework lies in the fact that, we do not need to introduce and solve any additional equations to update the derivatives of the unknown function $\phi$. Instead, we use the current $\phi$ and the old spatial derivative of $\phi$ to update them. The traditional HWENO fast sweeping method is also introduced in this paper for comparison, where additional equations governing the spatial derivatives of $\phi$ are introduced. The novel HWENO fast sweeping methods are shown to yield great savings in computational time, which improves the computational efficiency of the traditional HWENO scheme. In addition, a hybrid strategy is also introduced to further reduce computational costs. Extensive numerical experiments are provided to validate the accuracy and efficiency of the proposed approaches.

*Mathematics subject classification:* 65M60, 35L65.
*Key words:* Finite difference, Hermite methods, Weighted essentially non-oscillatory method, Fast sweeping method, Static Hamilton-Jacobi equations, Eikonal equation.

## 1. Introduction

In this paper, we design and validate high order accurate and efficient Hermite weighted essentially non-oscillatory (HWENO) fast sweeping methods for solving the static Hamilton-Jacobi (HJ) equation

$$\begin{cases} H(\nabla\phi, \mathbf{x}) = 0, & \mathbf{x} \in \Omega \setminus \Gamma, \\ \phi(\mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in \Gamma \subset \Omega, \end{cases} \tag{1.1}$$

where $\Omega$ is the computational domain in $\mathbb{R}^d$, $\phi(\mathbf{x})$ is the unknown function in $\Omega$, the Hamiltonian $H$ is a nonlinear Lipschitz continuous function depending on $\nabla\phi$ and $\mathbf{x}$, and the boundary condition is given by $\phi(\mathbf{x}) = g(\mathbf{x})$ on the subset $\Gamma \subset \Omega$. One important example to be considered is the Eikonal equation, taking the form of

$$\begin{cases} |\nabla\phi| = f(\mathbf{x}), & \mathbf{x} \in \Omega \setminus \Gamma, \\ \phi(\mathbf{x}) = g(\mathbf{x}), & \mathbf{x} \in \Gamma \subset \Omega, \end{cases} \tag{1.2}$$

where $f(\mathbf{x}) > 0$.

The HJ equations have extensive applications in many different fields, for instance in optimal control, computer vision, differential game and geometric optics, image processing and so on [5,28]. It is well-known that global $C^1$ solution does not exist for time-dependent HJ equations in the generic situation, even if the initial condition is amply smooth. Singularities in the form of discontinuities would appear in the derivatives of the unknown function, hence it is necessary to define a "weak solution" for the HJ equations. The viscosity solutions of the HJ equations were first introduced by Crandall and Lions in [2].

One popular way to numerically solve the static HJ equations is to treat the problem as a stationary boundary value problem, such as the fast marching method (FMM) [3, 21, 26] and the fast sweeping method (FSM) [7,8,15,25,33,34] can be applied. Compared with FMM, FSM can be constructed to be high order accurate, and becomes a class of popular and effective methods for solving static HJ equations nowadays. The FSM was first introduced in [1] by Boué and Dupuis, to solve a deterministic control problem with quadratic running cost using Markov chain approximation. Later, Zhao [33] applied the FSM to solve the Eikonal equations. Since then, many high order FSM have been developed to solve static HJ equations. In the framework of finite difference methods, Zhang *et al.* [32] combined the third order finite difference WENO-JP scheme [6] with FSM, and Xiong *et al.* [29] studied fifth order WENO-JP FSM scheme. High order accurate boundary treatments (i.e., Richardson extrapolation and Lax-Wendroff type procedure), which are consistent with high order FSM, have been developed for the inflow boundary conditions in [4,29]. In [16], a competent stopping criterion was recommended for high order FSM. In addition, high order FSM was also investigated in the framework of discontinuous Galerkin (DG) finite element method to solve Eikonal equation, and their numerical performance was shown to be effective and robust [10,12,27,31].

In additional to finite difference WENO and DG methods, high order HWENO methods [11, 17–19] have recently gained many attention in solving hyperbolic conservation laws. Both the classical WENO and HWENO methods can achieve the high order accuracy and preserve the essentially non-oscillatory property. The main difference lies in the fact that the HWENO scheme uses the Hermite reconstruction, that involves both the unknown variable $\phi$ and its first order spatial derivative or first moment in the reconstruction. As a result, the reconstruction stencil becomes more compact, although more storage and some additional work are needed to evaluate the spatial derivatives. The HWENO scheme was first proposed during the construction of a suitable limiter for the DG method [17,18], since it is more compact than the standard WENO scheme. In [19], the HWENO scheme was first used to solve the time-dependent HJ equation, and achieved very good numerical results. Compared with the WENO scheme, its boundary treatment is simpler and the numerical error is smaller with the same mesh, as shown in [19]. The HWENO scheme was later extended to solve the hyperbolic conservation law in the finite difference framework [11], where the same advantages can be observed. Since then, a series of HWENO schemes [24,30,35,36] have been investigated to solve hyperbolic

conservation laws and time-dependent HJ equations under the framework of finite difference or finite volume methods. Recently, a new HWENO scheme (denoted by HWENO-ZZQ) was developed by Zhu *et al.* in [39] for time dependent HJ equations. Compared with the existing HWENO methods, the new HWENO-ZZQ reconstruction uses one big stencil and two small stencils. Also, during the reconstruction procedure, one only needs to apply the complicated HWENO reconstruction when updating the function values, and can use the simple high order linear reconstruction when updating the derivatives. As shown in [39], the new HWENO-ZZQ scheme can obtain small errors with the same high order accuracy in the smooth areas, and maintain sharp transitions and non-oscillatory property near discontinuity.

In this work, we propose to combine the HWENO-ZZQ method with the fast sweeping idea, to provide an efficient solver for the static HJ equations. Two approaches to design finite difference HWENO FSM for the static HJ equations will be presented. The first novel approach is unique to static HJ equation, and addresses the potential concern of HWENO method where additional equations governing the spatial derivatives of $\phi$ are introduced and increase computational costs. Here, we propose to use just one equation to update $\phi$, and then use the current $\phi$ and the old spatial derivatives of $\phi$ to update the spatial derivatives. This is very different from the traditional HWENO framework. By designing the algorithm in this way, there is no need to introduce additional equations for the spatial derivatives, which is commonly employed in the HWENO methods. This will lead to great savings in computational time, which improves the computational efficiency of the traditional HWENO scheme. For comparison, we also presented the standard HWENO fast sweeping method in this paper, where one equation is computed to update $\phi$, and the other one (or two) equation is computed to update the spatial derivative(s) of $\phi$. This can be viewed as a straightforward extension of HWENO-ZZQ method [39] in the framework of FSM. Finally, we also introduced a hybrid strategy which leads to further saving in the computational resources. Extensive numerical results are provided to demonstrate the performance of the proposed methods. Especially, one can observe that HWENO method yields smaller numerical error than WENO method after they converge on the same mesh size.

The rest of the paper is organized as follows. In Section 2, we present two novel HWENO fast sweeping methods for static HJ equations. The numerical tests are presented to demonstrate the effectiveness and efficiency of our schemes in Section 3. A hybrid strategy, together with some numerical results, is presented in Section 4. Conclusion remarks are given in Section 5.

## 2. HWENO FSMs for the Static HJ Equations

In this section, we present two types of HWENO FSMs to efficiently solve the static HJ equations. The flowchart of these two algorithms and their numerical implementation will be provided. A quick review of finite difference WENO FSM will also be provided.

### 2.1. Review of high order WENO FSM

In this subsection, we briefly review the high order WENO FSM to solve the static HJ equations [29, 32], where WENO reconstruction was used to approximate the first order derivatives appeared in the numerical Hamiltonian. For further details on this subject, we refer to [29, 32].

For ease of presentation, we only consider the following two dimensional static HJ equation

$$\begin{cases} H(\phi_x, \phi_y) = f(x,y), & (x,y) \in \Omega \setminus \Gamma, \\ \phi(x,y) = g(x,y), & (x,y) \in \Gamma \subset \Omega. \end{cases} \tag{2.1}$$

Suppose the computational domain $\Omega$ is discretized into the rectangular meshes

$$\Omega_h = \{(x_i, y_j), 1 \le i \le N_x, 1 \le j \le N_y\}$$

with $(x_i, y_j)$ being a grid point in $\Omega_h$. We denote the numerical solution at the grid point $(x_i, y_j)$ by $\phi_{i,j}$. $\Delta x$ and $\Delta y$ stand for the grid sizes in the $x$ and $y$ directions, respectively, and we assume $\Delta x = \Delta y = h$ for simplicity. The numerical approximation of (2.1) is given by

$$\begin{cases} \widehat{H}(\phi_x^-, \phi_x^+, \phi_y^-, \phi_y^+)_{ij} = f_{ij}, & (x_i, y_j) \in \Omega_h \setminus \Gamma_h, \\ \phi_{ij} = g_{ij}, & (x_i, y_j) \in \Gamma_h \subset \Omega_h, \end{cases} \tag{2.2}$$

where $\widehat{H}$ denotes a monotone numerical Hamiltonian which approximates the Hamiltonian $H$. Such numerical Hamiltonian takes inputs $\phi_x^\pm$ and $\phi_y^\pm$ at the corresponding grid point, which needs to be reconstructed from its neighboring point values using the high order WENO procedure.

Two types of numerical Hamiltonian are often considered in the literature. For general static HJ equation, we adopt the Lax-Friedrichs (LF) numerical Hamiltonian [13]

$$\widehat{H}_{i,j}^{LF} = H\left(\frac{u_{i,j}^- + u_{i,j}^+}{2}, \frac{v_{i,j}^- + v_{i,j}^+}{2}\right) - \frac{1}{2}\alpha(u_{i,j}^+ - u_{i,j}^-) - \frac{1}{2}\beta(v_{i,j}^+ - v_{i,j}^-),$$

where

$$\alpha = \max_{u,v} |H_1(u,v)|, \quad \beta = \max_{u,v} |H_2(u,v)|. \tag{2.3}$$

Here $H_\ell(u,v)\,(\ell = 1, 2)$ denotes the partial derivative of $H$ with respect to the $\ell$-th argument. The updating procedure of the LF FSM for static HJ equations can be written as [7,32]

$$\phi_{i,j}^{new} = \left(\frac{h}{\alpha + \beta}\right)\left[ f_{i,j} - H\left(\frac{(\phi_x)_{i,j}^+ + (\phi_x)_{i,j}^-}{2}, \frac{(\phi_y)_{i,j}^+ + (\phi_y)_{i,j}^-}{2}\right) \right.$$
$$\left. + \alpha\frac{(\phi_x)_{i,j}^+ - (\phi_x)_{i,j}^-}{2} + \beta\frac{(\phi_y)_{i,j}^+ - (\phi_y)_{i,j}^-}{2}\right] + \phi_{i,j}^{old}. \tag{2.4}$$

Here $\phi_{i,j}^{new}$ denotes the updated numerical approximations of $\phi$ at the grid point $(x_i, y_j)$ and $\phi_{i,j}^{old}$ denotes the previous value of $\phi$ at the same grid point.

The other commonly used numerical Hamiltonian is the Godunov numerical Hamiltonian, often employed in the approximation of the Eikonal equation (1.2). Again, we consider the two dimensional version

$$\begin{cases} \sqrt{\phi_x^2 + \phi_y^2} = f(x,y), & (x,y) \in \Omega, \\ \phi(x,y) = g(x,y), & (x,y) \in \Gamma \subset \Omega, \end{cases} \tag{2.5}$$

and utilize the following Godunov numerical Hamiltonian to approximate it on uniform meshes [32,33]

$$\left[\left(\frac{\phi_{i,j}^{new} - \phi_{i,j}^{x\,\min}}{h}\right)^+\right]^2 + \left[\left(\frac{\phi_{i,j}^{new} - \phi_{i,j}^{y\,\min}}{h}\right)^+\right]^2 = f_{i,j}^2, \quad x^+ = \begin{cases} x, & x > 0, \\ 0, & x < 0, \end{cases} \tag{2.6}$$

where

$$\phi_{i,j}^{x\,\min} = \min\left(\phi_{i,j}^{old} - h(\phi_x)_{i,j}^-, \phi_{i,j}^{old} + h(\phi_x)_{i,j}^+\right), \tag{2.7a}$$

$$\phi_{i,j}^{y\,\min} = \min\left(\phi_{i,j}^{old} - h(\phi_y)_{i,j}^-, \phi_{i,j}^{old} + h(\phi_y)_{i,j}^+\right). \tag{2.7b}$$

After obtaining $\phi_{i,j}^{x\,\min}$ and $\phi_{i,j}^{y\,\min}$ using the above formulas, $\phi_{i,j}^{new}$ can be computed as

$$\phi_{i,j}^{new} = \begin{cases} \min\left(\phi_{i,j}^{x\,\min}, \phi_{i,j}^{y\,\min}\right) + f_{i,j}h, & \text{if } \left|\phi_{i,j}^{x\,\min} - \phi_{i,j}^{y\,\min}\right| \geq f_{i,j}h, \\ \dfrac{1}{2}\left(\phi_{i,j}^{x\,\min} + \phi_{i,j}^{y\,\min} + \left(2f_{i,j}^2 h^2 - (\phi_{i,j}^{x\,\min} - \phi_{i,j}^{y\,\min})^2\right)^{1/2}\right), & \text{otherwise.} \end{cases} \tag{2.8}$$

A systematic method for solving the Eikonal equations by first order FSM is developed in [33]. The essential idea of the FSM is to adopt nonlinear upwind difference and Gauss-Seidel (GS) iterations with alternating sweeping ordering. The FSM follows the causality along characteristics, namely, all characteristic lines are classified as finite groups according to their directions, and each GS iteration with a specific sweep order covers a set of characteristic lines. We can refer to [32, 33] for more details including the flowchart of FSM.

## 2.2. A novel HWENO FSM

Before introducing the new method, we start by briefly reviewing the traditional HWENO framework. We denote $u = \phi_x(x, y)$ and $v = \phi_y(x, y)$ as the first order partial derivatives of $\phi$ with respect to the variables $x$ and $y$, respectively. By taking spatial derivatives on both sides of (2.1), we obtain the following system of equations:

$$\begin{cases} H(\phi_x, \phi_y) = f(x, y), \\ H_1(u, v)u_x + H_2(u, v)u_y = f_x, \\ H_1(u, v)v_x + H_2(u, v)v_y = f_y, \end{cases} \tag{2.9}$$

where $H_1(u, v) = \partial H / \partial u$, $H_2(u, v) = \partial H / \partial v$, and $v_x = u_y$ is used in the derivation.

The first equation can be solved by the FSM (2.4) or (2.8), combined with the HWENO-ZZQ reconstruction to be discussed in Section 2.4, which involves $\phi, u$ and $v$ simultaneously. The auxiliary variables $u$ and $v$ are usually updated by solving these two equations arising from the derivative of the HJ equation, which is a common approach in the traditional HWENO method for time-dependent problem [19, 22, 36].

The targeting HJ equation in this paper is a steady state problem, and iterative method is used to update our approximation of $\phi$. During each iteration, we have already applied the HWENO-ZZQ reconstruction procedure to evaluate $\phi_x^\pm$ and $\phi_y^\pm$ when solving the first equation of (2.9). Such information could be reused to generate our updated numerical approximation of $u$ and $v$, and there is no need to re-evaluate them from solving these two additional equations. In other words, we use $\phi^{new}$ (computed by (2.4) or (2.8)) and $u^{old}$, $v^{old}$ to reconstruct $\phi_x^\pm$ and $\phi_y^\pm$ by HWENO-ZZQ reconstruction, and then define $u^{new}$ and $v^{new}$ from the following formulas:

$$u_{i,j}^{new} = \begin{cases} (\phi_x)_{i,j}^-, & \text{if } (\phi_x)_{i,j}^\pm > 0, \\ (\phi_x)_{i,j}^+, & \text{if } (\phi_x)_{i,j}^\pm < 0, \\ u_{i,j}^{old}, & \text{otherwise,} \end{cases} \qquad v_{i,j}^{new} = \begin{cases} (\phi_y)_{i,j}^-, & \text{if } (\phi_y)_{i,j}^\pm > 0, \\ (\phi_y)_{i,j}^+, & \text{if } (\phi_y)_{i,j}^\pm < 0, \\ v_{i,j}^{old}, & \text{otherwise.} \end{cases} \tag{2.10}$$

Here $u_{i,j}^{new}$ and $v_{i,j}^{new}$ denote the updated numerical approximations of $u$ and $v$ at the grid point $(x_i, y_j)$, respectively. We use $u_{i,j}^{old}$ and $v_{i,j}^{old}$ to denote the previous value of $u$ and $v$ at the same grid point. The HWENO-ZZQ reconstruction, to approximate the derivatives $\phi_x$, and $\phi_y$ at the grid point $(x_i, y_j)$ with high order accuracy, will be discussed in Section 2.4.

The definition of $u^{new}$ and $v^{new}$ in (2.10) comes from the fast sweeping method, that is, the solution of Eq. (2.1) is increasing along the characteristic lines [15, 32], and the information always comes from the upwind direction. Therefore we use this formulation to directly define the derivative values. Numerical tests in Section 3 also confirm that this simplified method is both robust and effective.

We would like to comment that this approach is different from the standard WENO method, and cannot be used in time-dependent problems. In the HWENO procedure to reconstruct the derivatives $\phi_x^{\pm}$ and $\phi_y^{\pm}$ (hence, $u^{new}$, $v^{new}$), the information of $\phi^{new}$, $u^{old}$ and $v^{old}$ are all used. Since this is an iterative method, $u^{old}$ and $v^{old}$ would also be "good" approximations of the exact derivatives. For time-dependent problems, $\phi^{new}$ would be the approximation at the next time step $t^n + \Delta t$, while $u^{old}$ and $v^{old}$ approximates the derivatives at the current time step $t^n$, hence such reconstruction cannot be applied.

### 2.3. The traditional HWENO FSM

In this subsection, we present how to solve the static HJ equation when the traditional HWENO framework is used. The first equation in (2.9) can be solved by the FSM (2.4) or (2.8), combined with the HWENO-ZZQ reconstruction. Here we describe how to solve the last two auxiliary equations to obtain the updated values of $u_{i,j}$ and $v_{i,j}$. These equations are approximated by the following scheme:

$$\begin{cases} \widetilde{H}_{i,j} = (f_x)_{i,j}, \\ \widetilde{\widetilde{H}}_{i,j} = (f_y)_{i,j}, \end{cases} \tag{2.11}$$

where the $\widetilde{H}_{i,j}$ and $\widetilde{\widetilde{H}}_{i,j}$ are the numerical flux defined as

$$\begin{aligned}
\widetilde{H}_{i,j} &= H_1\left(\frac{u_{i,j}^- + u_{i,j}^+}{2}, \frac{v_{i,j}^- + v_{i,j}^+}{2}\right) \frac{u_{x\;ij}^+ + u_{x\;ij}^-}{2} \\
&\quad + H_2\left(\frac{u_{i,j}^- + u_{i,j}^+}{2}, \frac{v_{i,j}^- + v_{i,j}^+}{2}\right) \frac{u_{y\;ij}^+ + u_{y\;ij}^-}{2} \\
&\quad - \frac{1}{2}\alpha\left(u_{x\;ij}^+ - u_{x\;ij}^-\right) - \frac{1}{2}\beta\left(u_{y\;ij}^+ - u_{y\;ij}^-\right), \\
\widetilde{\widetilde{H}}_{i,j} &= H_1\left(\frac{u_{i,j}^- + u_{i,j}^+}{2}, \frac{v_{i,j}^- + v_{i,j}^+}{2}\right) \frac{v_{x\;ij}^+ + v_{x\;ij}^-}{2} \\
&\quad + H_2\left(\frac{u_{i,j}^- + u_{i,j}^+}{2}, \frac{v_{i,j}^- + v_{i,j}^+}{2}\right) \frac{v_{y\;ij}^+ + v_{y\;ij}^-}{2} \\
&\quad - \frac{1}{2}\alpha\left(v_{x\;ij}^+ - v_{x\;ij}^-\right) - \frac{1}{2}\beta\left(v_{y\;ij}^+ - v_{y\;ij}^-\right)
\end{aligned} \tag{2.12}$$

with $\alpha$ and $\beta$ given in (2.3).

The iteration scheme for updating $u$ will be discussed below and the procedure for $v$ is exactly the same. We start by discussing the simpler first order case, where $u^{\pm}$ are simply the

backward and forward difference approximations. Hence, the first equation of (2.11) can be rewritten as

$$\widetilde{H}_{i,j} = H_1 \left( \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}, \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h} \right) \frac{u_{i+1,j} - u_{i-1,j}}{2h}$$
$$+ H_2 \left( \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}, \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h} \right) \frac{u_{i,j+1} - u_{i,j-1}}{2h}$$
$$- \frac{1}{2h} \alpha \left( u_{i+1,j} - 2u_{i,j} + u_{i-1,j} \right) - \frac{1}{2h} \beta \left( u_{i,j+1} - 2u_{i,j} + u_{i,j-1} \right) = (f_x)_{i,j}. \qquad (2.13)$$

Therefore, one can solve for the first order approximation of $u_{i,j}$ with the following expression:

$$u_{i,j} = \frac{h}{\alpha + \beta} \left[ (f_x)_{i,j} - H_1 \left( \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}, \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h} \right) \frac{u_{i+1,j} - u_{i-1,j}}{2h} \right.$$
$$- H_2 \left( \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}, \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h} \right) \frac{u_{i,j+1} - u_{i,j-1}}{2h}$$
$$\left. + \frac{1}{2h} \alpha \left( u_{i+1,j} + u_{i-1,j} \right) + \frac{1}{2h} \beta \left( u_{i,j+1} + u_{i,j-1} \right) \right]. \qquad (2.14)$$

To obtain the high order iterative scheme, we replace $u_{i+1,j}, u_{i-1,j}, u_{i,j+1}$ and $u_{i,j-1}$ with $u_{i,j} + h(u_x)_{i,j}^+, u_{i,j} - h(u_x)_{i,j}^-, u_{i,j} + h(u_y)_{i,j}^+$ and $u_{i,j} - h(u_y)_{i,j}^-$, respectively (see [32]), where $(u_x)_{i,j}^{\pm}$ and $(u_y)_{i,j}^{\pm}$ are high order approximation of the partial derivatives of $u$. Similarly, one can apply this idea to $\phi_{i+1,j}, \phi_{i-1,j}, \phi_{i,j+1}$ and $\phi_{i,j-1}$ as well, and the resulting high order schemes can be rewritten as

$$u_{i,j}^{new} = \frac{1}{\alpha/h + \beta/h} \left[ (f_x)_{i,j} - H_1 \left( \frac{(\phi_x^+)_{i,j} + (\phi_x^-)_{i,j}}{2}, \frac{(\phi_y^+)_{i,j} + (\phi_y^-)_{i,j}}{2} \right) \frac{(u_x^+)_{i,j} + (u_x^-)_{i,j}}{2} \right.$$
$$- H_2 \left( \frac{(\phi_x^+)_{i,j} + (\phi_x^-)_{i,j}}{2}, \frac{(\phi_y^+)_{i,j} + (\phi_y^-)_{i,j}}{2} \right) \frac{(u_y^+)_{i,j} + (u_y^-)_{i,j}}{2}$$
$$\left. + \frac{1}{2} \alpha \left( (u_x^+)_{i,j} - (u_x^-)_{i,j} \right) + \frac{1}{2} \beta \left( (u_y^+)_{i,j} - (u_y^-)_{i,j} \right) \right] + u_{i,j}^{old}. \quad (2.15)$$

The high order scheme to solve $v$ can be obtained in the similar way, and takes the form of

$$v_{i,j}^{new} = \frac{1}{\alpha/h + \beta/h} \left[ (f_y)_{i,j} - H_1 \left( \frac{(\phi_x^+)_{i,j} + (\phi_x^-)_{i,j}}{2}, \frac{(\phi_y^+)_{i,j} + (\phi_y^-)_{i,j}}{2} \right) \frac{(v_x^+)_{i,j} + (v_x^-)_{i,j}}{2} \right.$$
$$- H_2 \left( \frac{(\phi_x^+)_{i,j} + (\phi_x^-)_{i,j}}{2}, \frac{(\phi_y^+)_{i,j} + (\phi_y^-)_{i,j}}{2} \right) \frac{(v_y^+)_{i,j} + (v_y^-)_{i,j}}{2}$$
$$\left. + \frac{1}{2} \alpha \left( (v_x^+)_{i,j} - (v_x^-)_{i,j} \right) + \frac{1}{2} \beta \left( (v_y^+)_{i,j} - (v_y^-)_{i,j} \right) \right] + v_{i,j}^{old}. \quad (2.16)$$

The HWENO-ZZQ reconstruction will be used to approximate the derivatives $\phi_x, \phi_y, u_x, u_y, v_x$ and $v_y$ at the grid point $(x_i, y_j)$ with high order accuracy.

## 2.4. HWENO-ZZQ reconstruction

In this subsection, the new finite difference HWENO-ZZQ reconstruction recently proposed in [39] will be briefly reviewed. To save space, we only illustrate the reconstruction of $(\phi_x)_{i,j}^{\pm}$

along $x$-direction here. The approximation of $(\phi_y)_{i,j}^{\pm}$ along $y$-direction can be obtained similarly, and we refer to [39] for more details.

- Reconstruction of $(\phi_x)_{i,j}^{-}$ from upwind information:
  Take a big stencil $S_0 = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$ and two small stencils $S_1 = \{x_{i-2}, x_{i-1}, x_i\}$, $S_2 = \{x_{i-1}, x_i, , x_{i+1}\}$, we compose a Hermite quintic polynomial $p_1^{-}(x)$, and two quadratic polynomials $p_2^{-}(x)$, $p_3^{-}(x)$ satisfying

$$p_1^{-}(x_k) = \phi_{k,j}, \quad k = i-2, \ldots, i+1, \quad \text{and} \quad (p_1^{-})'|_{x_k} = u_{k,j}, \quad k = i-1, i+1,$$
$$p_2^{-}(x_k) = \phi_{k,j}, \quad k = i-2, i-1, i, \quad \text{and} \quad p_3^{-}(x_k) = \phi_{k,j}, \quad k = i-1, i, i+1.$$

The values of their first-order derivative at $x = x_i$ can be evaluated as

$$(\phi_x)_{i,j}^{-,1} = (p_1^{-})'|_{x_i}$$
$$= \frac{\phi_{i-2,j} + 18\phi_{i-1,j} - 9\phi_{i,j} - 10\phi_{i+1,j} + 9hu_{i-1,j} + 3hu_{i+1,j}}{-18h}, \tag{2.17a}$$

$$(\phi_x)_{i,j}^{-,2} = (p_2^{-})'|_{x_i} = \frac{\phi_{i-2,j} - 4\phi_{i-1,j} + 3\phi_{i,j}}{2h}, \tag{2.17b}$$

$$(\phi_x)_{i,j}^{-,3} = (p_3^{-})'|_{x_i} = \frac{-\phi_{i-1,j} + \phi_{i+1,j}}{2h}. \tag{2.17c}$$

- Reconstruction of $(\phi_x)_{i,j}^{+}$ from downwind information:
  Take a big stencil $\widetilde{S}_0 = \{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$ and two small stencils $\widetilde{S}_1 = \{x_{i-1}, x_i, x_{i+1}\}$, $\widetilde{S}_2 = \{x_i, x_{i+1}, , x_{i+2}\}$, we compose a Hermite quintic polynomial $p_1^{+}(x)$, and two quadratic polynomials $p_2^{+}(x)$, $p_3^{+}(x)$ such that

$$p_1^{+}(x_k) = \phi_{k,j}, \quad k = i-1, \ldots, i+2, \quad \text{and} \quad (p_1^{+})'|_{x_k} = u_{k,j}, \quad k = i-1, i+1,$$
$$p_2^{+}(x_k) = \phi_{k,j}, \quad k = i-1, i, i+1, \quad \text{and} \quad p_3^{+}(x_k) = \phi_{k,j}, \quad k = i, i+1, i+2.$$

The values of their first-order derivative at $x = x_i$ can be evaluated as

$$(\phi_x)_{i,j}^{+,1} = (p_1^{+})'|_{x_i}$$
$$= \frac{10\phi_{i-1,j} + 9\phi_{i,j} - 18\phi_{i+1,j} - \phi_{i+2,j} + 3hu_{i-1,j} + 9hu_{i+1,j}}{-18h}, \tag{2.18a}$$

$$(\phi_x)_{i,j}^{+,2} = (p_2^{+})'|_{x_i} = \frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}, \tag{2.18b}$$

$$(\phi_x)_{i,j}^{+,3} = (p_3^{+})'|_{x_i} = \frac{-3\phi_{i,j} + 4\phi_{i+1,j} - \phi_{i+2,j}}{2h}. \tag{2.18c}$$

In the nonlinear HWENO reconstructions, $(\phi_x)_{i,j}^{\pm}$ are computed as a convex combination of these three corresponding values [9, 37, 38]

$$(\phi_x)_{i,j}^{\pm} = \omega_1^{\pm} \left( \frac{1}{\gamma_1} (\phi_x)_{i,j}^{\pm,1} - \frac{\gamma_2}{\gamma_1} (\phi_x)_{i,j}^{\pm,2} - \frac{\gamma_3}{\gamma_1} (\phi_x)_{i,j}^{\pm,3} \right) + \omega_2^{\pm} (\phi_x)_{i,j}^{\pm,2} + \omega_3^{\pm} (\phi_x)_{i,j}^{\pm,3}, \tag{2.19}$$

where the parameters $\omega_n$ ($n = 1, 2, 3$) and $\gamma_n$ ($n = 1, 2, 3$) are called the nonlinear weights and linear weights, respectively. The parameters $\gamma_n$ can be any positive constants that satisfy $\gamma_1 + \gamma_2 + \gamma_3 = 1$, and $\omega_n$ can be computed from

$$\omega_n^{\pm} = \frac{\overline{\omega}_n^{\pm}}{\sum_{l=1}^{3} \overline{\omega}_l^{\pm}}, \quad \overline{\omega}_n^{\pm} = \gamma_n \left( 1 + \frac{\tau^{\pm}}{\epsilon + \beta_n^{\pm}} \right), \quad n = 1, 2, 3, \tag{2.20}$$

where $\epsilon$ is a small positive number to avoid the denominator becoming 0, and for $n = 1, 2, 3$,

$$\tau^{\pm} = \left( \frac{|\beta_1^{\pm} - \beta_2^{\pm}| + |\beta_1^{\pm} - \beta_3^{\pm}|}{2} \right)^2, \quad \beta_n^{\pm} = \sum_{\alpha=2}^{r} \int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}} h^{2\alpha-3} \left( \frac{d^{\alpha} p_n^{\pm}(x)}{dx^{\alpha}} \right)^2 dx,$$

where $\beta_n^{\pm}$ are the so-called smoothness indicators, which measure how smooth the first-order derivative functions of $p_n^{\pm}(x)$ are near the target point $x_i$, and $r = 5$ for $n = 1$, and $r = 2$ for $n = 2, 3$, respectively.

The approximation of $(u_x^{\pm})_{i,j}$ is based on the high order linear reconstructions in x-direction, instead of the HWENO reconstructions. Given two big spatial stencils $Q = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}\}$ and $\widetilde{Q} = \{x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$, then we can construct two Hermite seventh order polynomials $q^{\pm}(x)$ such that

$$q^{-}(x_k) = \phi_{k,j}, \quad k = i-2, \dots, i+1, \quad \text{and} \quad (q^{-})'|_{x_k} = u_{k,j}, \quad k = i-1, i, i+1,$$
$$q^{+}(x_k) = \phi_{k,j}, \quad k = i-1, \dots, i+2, \quad \text{and} \quad (q^{+})'|_{x_k} = u_{k,j}, \quad k = i-1, i, i+1.$$

One can evaluate their second-order derivatives at $x = x_i$ as

$$(u_x^{-})_{i,j} = (q^{-})''|_{x_i} = \frac{\phi_{i-2,j} + 54\phi_{i-1,j} - 81\phi_{i,j} + 26\phi_{i+1,j} + 18hu_{i-1,j} + 18hu_{i,j} - 6hu_{i+1,j}}{18h^2},$$
$$(u_x^{+})_{i,j} = (q^{+})''|_{x_i} = \frac{26\phi_{i-1,j} - 81\phi_{i,j} + 54\phi_{i+1,j} + \phi_{i+2,j} + 6hu_{i-1,j} - 18hu_{i,j} - 18hu_{i+1,j}}{18h^2}.$$

The approximation of $(v_y)_{i,j}^{\pm}$ along $y$-direction can be obtained in a similar way and is skipped here. The mixed derivative $u_y$ and $v_x$ can be evaluated using central difference easily, since they play smaller role on the spurious oscillations according to [19]. Therefore, one can use the fourth order central approximations in the $x$- and $y$-directions, and obtain

$$(u_y)_{i,j} \approx \frac{-u_{i,j+2} + 8u_{i,j+1} - 8u_{i,j-1} + u_{i,j-2}}{12h},$$
$$(v_x)_{i,j} \approx \frac{-v_{i+2,j} + 8v_{i+1,j} - 8v_{i-1,j} + v_{i-2,j}}{12h}.$$

We then set $u_y^{+} = u_y^{-} = u_y$, and $v_x^{+} = v_x^{-} = v_x$ at the point $(x_i, y_j)$. This finishes the description of the HWENO-ZZQ reconstruction.

### 2.5. The flowchart of both approaches

We have discussed two approaches to solve the static HJ equations. For simplicity, let us denote the novel HWENO FSM in Section 2.2 by Approach 1, and the traditional HWENO FSM in Section 2.3 by Approach 2. Next we will summarize the detailed procedure of these two approaches, and provide a flowchart for them. We start by labelling the points $\{(x_i, y_j)\}$ into several categories as in [20]:

*Category I*: For points on the boundary $\Gamma$, the exact values are assigned for these points.

*Category II*: For ghost points (exterior of the boundary), we use the high order extrapolation to compute their numerical solution $\phi_{i,j}$.

*Category III*: For points near the $\Gamma$ (whose distances to $\Gamma$ are less than or equal to $2h$). The numerical boundary treatment from [23, 29] is used (i.e., Richardson extrapolation for a single point or a set of isolated points, while Lax-Wendroff type procedure for continuous $\Gamma$).

*Category IV*: All remaining points. which will be updated by FSM.

Note that only *Category IV* points need to be updated by following sweepings. We now summarize our flowchart for two approaches as follows:

**Step 1**. *Initialization*: The numerical solution from the first order fast sweeping method [33] is taken as the initial guess of $\phi$. The forward or backward difference of this $\phi$ is used as the initial guess of $u$ and $v$.

**Step 2**. *Gauss-Seidel iteration*. We solve the discretized nonlinear system (2.9) by GS iterations with four alternating direction sweepings

$$(1) \quad i = 1 : N_x, \ j = 1 : N_y, \quad (2) \quad i = N_x : 1, \ j = 1 : N_y,$$

$$(3) \quad i = N_x : 1, \ j = N_y : 1, \quad (4) \quad i = 1 : N_x, \ j = N_y : 1.$$

In this step, we first compute the $\phi_{i,j}^{new}$ by (2.4) or (2.8), then

- for Approach 1, $\phi_{i,j}^{new} \, u_{i,j}^{old}$ and $v_{i,j}^{old}$ are used to reconstruct $u_{i,j}^{new}$ and $v_{i,j}^{new}$ following (2.10),

- for Approach 2, $u_{i,j}^{new}$ and $v_{i,j}^{new}$ are evaluated by (2.15)-(2.16) in each sweeping direction. Note that we need to apply linear reconstruction to obtain $u_x, u_y, v_x$ and $v_y$ first, as explained in Section 2.4.

The values at ghost points will be updated by high order extrapolations in both approaches.

**Step 3**. *Convergence*: In general, the iteration will stop if, for two consecutive iteration steps,

$$\delta = \|\phi^{new} - \phi^{old}\|_{L_1} < 10^{-14}.$$

## 2.6. Comments and remarks

At the end of this section, we would like to present some comments and remarks about the proposed algorithms. Approach 1 needs only one equation to update $\phi$, while its derivative approximations are obtained by applying the HWENO-ZZQ reconstruction on the updated $\phi$ and old $u$, $v$. As a comparison, Approach 2 adopted the traditional HWENO idea, namely, one equation to update original variable $\phi$, and two auxiliary equations to update the derivative values. The major difference between Approach 1 and Approach 2 is that the latter involves extra work for solving the auxiliary equations, more specifically, we need to store the values of $u_x^{\pm}, v_y^{\pm}, u_y$, and $v_x$ in Approach 2, and use them to update the $u^{new}$ and $v^{new}$. Therefore, Approach 1 is simpler and could save computational time, especially when the mesh is refined.

**Remark 2.1.** For the Eikonal equation, the numerical tests [20, 29] show that if the Godunov flux (2.6) is used to solve the first equation, both approaches will not converge to machine epsilon, i.e. $\delta$ will not decrease to $10^{-14}$, especially for the examples with singularities. Therefore, following the idea in [40], we propose to update the solution by

$$\phi^{new} = \omega\phi^{new} + (1 - \omega)\phi^{old}, \quad 0 < \omega < 1.$$

This fix is shown to yield good convergence, although it may slightly increase the number of iterations. Numerically, one observes that $\omega = 0.7$ or $0.8$ is the optimal choice. When the LF numerical Hamiltonian is considered, it is not necessary to take $\omega < 1$.

**Remark 2.2.** In Approach 2, we use (2.15) and (2.16) to update the auxiliary variables $u$ and $v$, respectively. We could increase the values of $\alpha$ and $\beta$ appropriately to improve the convergence speed. Numerically, we observe that doubling their values can reduce the iteration numbers.

# 3. Numerical Examples

In this section, we will present extensive numerical examples by testing the proposed fifth order finite difference HWENO FSM for the Eikonal equations and general static HJ equations in two dimensions. In all the numerical examples, $\epsilon$ in (2.20) is taken as $10^{-6}$ unless otherwise specified. We use "iter" to indicates the number of iterations (noting that one iteration includes four alternating sweepings) in all the tables. The total number of grid points is assumed to be $N_x = N_y = N$. We take $\omega = 0.7$ for Approach 1 and $\omega = 0.8$ for Approach 2 in Example 1-6. While for Example 7 P-wave, we take $\omega = 1.2$ for both Approach 1 and 2, and $\omega = 0.9$ for both Approach 1 and 2 in Example 7 SV-wave.

We will compare the numerical results of these two approaches with results of the classical WENO-JP FSM [6, 29], and list their errors, convergence rates and the numbers of iterations. One could also choose the WENO-ZQ FSM instead of WENO-JP FSM for comparison and these two methods yield similar errors and number of iterations.

As shown in the numerical experiments, the numerical error of HWENO method (after it converges) is smaller than that of WENO method. Between these two HWENO methods, Approach 1 is more efficient than Approach 2 (which is the traditional HWENO method).

**Example 3.1.** We solve the Eikonal equation with

$$f(x, y) = \frac{\pi}{2} \sqrt{\sin^2 \left( \pi + \frac{\pi}{2} x \right) + \sin^2 \left( \pi + \frac{\pi}{2} y \right)},$$

on the computational domain $[-1, 1]^2$, with the inflow boundary $\Gamma = (0, 0)$. The exact solution is given by

$$\phi(x, y) = \cos \left( \pi + \frac{\pi}{2} x \right) + \cos \left( \pi + \frac{\pi}{2} y \right).$$

The Godunov numerical Hamiltonian (2.6) is used. The picture of numerical solution by Approach 1 on mesh $N = 160$ is presented in Fig. 3.1. The numerical errors and orders of convergence for these different schemes are provided in Table 3.1 for comparison. We can see that the errors of Approach 1 and 2 are smaller than that of the WENO-JP scheme on the same mesh size, although they have slightly more iteration numbers when the mesh is refined. We
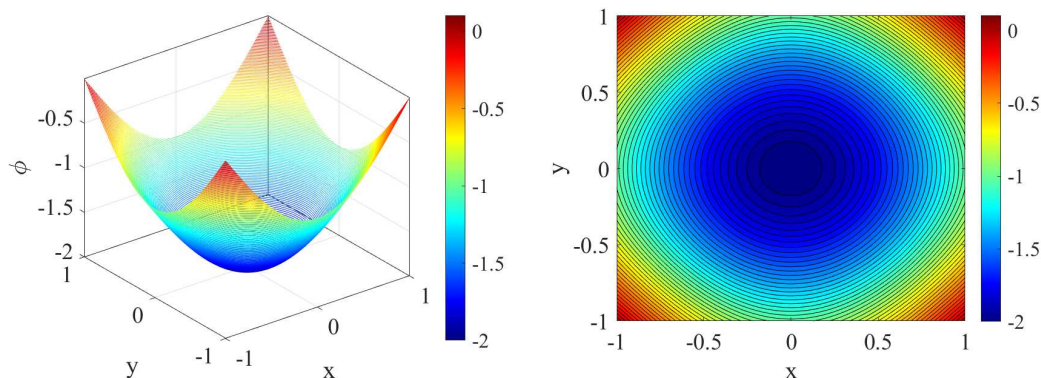


Fig. 3.1. Example 3.1. The numerical solution by Approach 1 on mesh $N = 160$. Left: the 3D plot of numerical solution $\phi$. Right: the contour plot for $\phi$.
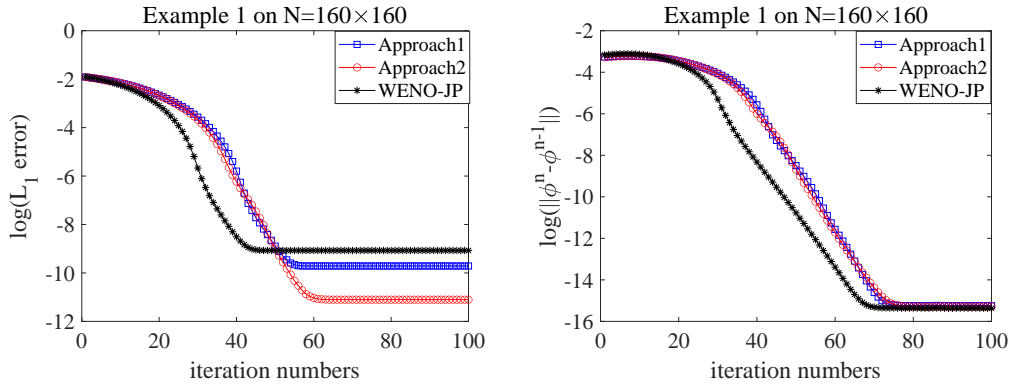
Fig. 3.2. Three methods solving Example 3.1. Left: number of iterations vs $\log(L_1\ error)$. Right: number of iterations vs $\log(||\phi^n - \phi^{n-1}||_{L_1})$.

fix the mesh to be $N = 160$, and run all three methods with 100 iterations. The history of their numerical errors is shown in Fig. 3.2, from which we can clearly observe that Approach 2 yields the smallest error (around $10^{-11}$), while the error of WENO-JP scheme stays at the level of $10^{-9}$. The error of Approach 1 stays between them. Although the error of WENO-JP scheme reaches $10^{-9}$ with the least number of iterations, its error will not decay further and stays at that level. We also plot the history of the quantity $\delta = ||\phi^{new} - \phi^{old}||_{L_1}$ for comparison, where we can observe that all three methods converge to $\delta = 10^{-15}$ after the 80th iteration.

We provide the comparison of these numerical results in Fig. 4.1, which also includes the results with hybrid strategy to be presented in next section. The left subfigure shows the mesh vs. CPU time of three methods on different meshes, from which one can observe that Approach 1 enjoys more savings in computational time than Approach 2 with the refinement of mesh. Also, the second and third subfigures are consistent with the numerical results in Table 3.1.

Table 3.1: Example 3.1. Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter |
|---|---|---|---|---|---|
| 40 | 3.21e-06 | - | 2.61e-05 | - | 40 |
| 80 | 3.80e-08 | 6.40 | 7.13e-07 | 5.19 | 46 |
| 160 | 1.92e-10 | 7.62 | 7.68e-09 | 6.53 | 69 |
| 320 | 2.18e-13 | 9.78 | 7.28e-12 | 10.00 | 111 |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 2.38e-07 | - | 4.09e-06 | - | 41 |
| 80 | 8.21e-10 | 8.18 | 3.10e-08 | 7.03 | 50 |
| 160 | 6.28e-12 | 7.02 | 2.02e-11 | 11.05 | 70 |
| 320 | 2.75e-13 | 4.51 | 8.73e-12 | 1.21 | 111 |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 1.54e-05 | - | 9.83e-05 | - | 47 |
| 80 | 1.13e-07 | 7.09 | 1.60e-06 | 5.94 | 51 |
| 160 | 8.45e-10 | 7.06 | 2.24e-08 | 6.15 | 63 |
| 320 | 3.01e-12 | 8.13 | 3.29e-11 | 9.40 | 80 |

**Example 3.2.** We solve the Eikonal equation with $f(x, y) = 1$. The computational domain is set as $[-1, 1]^2$, and the inflow boundary $\Gamma$ is the circle with center at $(0, 0)$ and radius 0.5, that is

$$\Gamma = \left\{ (x, y) \mid x^2 + y^2 = \frac{1}{4} \right\}.$$

The boundary condition is given as $\phi(x, y) = 0$ on $\Gamma$. The exact solution is a distance function to the circle $\Gamma$, and it has a singularity at the center of the circle (due to the intersection of characteristic lines).

The Godunov numerical Hamiltonian is used, and the numerical errors are measured in the box $[-0.9, 0.9]^2$ and outside the box $[-0.15, 0.15]^2$, which aims to remove the influence of singularity and outflow boundary treatment. The picture of numerical solution by Approach 1 is presented in Fig. 3.3. The numerical errors and orders of convergence are listed in Table 3.2. Again, we can observe that the errors of the Approach 1 and 2 are smaller than that of the WENO-JP schemes. Moreover, all these three methods achieve the designed fifth order accuracy on this test example. The number of iterations of WENO-JP scheme is slightly smaller than the other two methods. We also fix the mesh to be $N = 160$, and run all three methods with 100 iterations. The history of their numerical errors is shown in Fig. 3.4, from which we can clearly observe that Approach 2 yields the smallest error, WENO-JP method produces the largest error, and the error of Approach 1 stays between them. We plot the history of the quantity $\delta$ for comparison, where we can observe that all three methods converge to $\delta = 10^{-16}$ after the 70th iteration.

Again, we provide the comparison of these numerical results in Fig. 4.3, and the left subfigure shows the mesh vs. CPU time of three methods on different meshes, from which one can observe that Approach 1 enjoys more savings in computational time than Approach 2 with the refinement of mesh.

Table 3.2: Example 3.2. Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter |
|---|---|---|---|---|---|
| 40 | 6.06e-07 | - | 2.34e-05 | - | 37 |
| 80 | 1.17e-08 | 5.69 | 1.74e-06 | 3.74 | 44 |
| 160 | 8.72e-11 | 7.07 | 2.20e-08 | 6.30 | 55 |
| 320 | 1.78e-12 | 5.60 | 1.30e-10 | 7.40 | 81 |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 1.92e-07 | - | 1.09e-05 | - | 33 |
| 80 | 1.66e-09 | 6.85 | 1.87e-07 | 5.85 | 38 |
| 160 | 5.45e-11 | 4.92 | 3.67e-09 | 5.67 | 51 |
| 320 | 2.01e-12 | 4.75 | 1.23e-10 | 4.90 | 73 |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 5.18e-07 | - | 3.66e-05 | - | 35 |
| 80 | 3.84e-08 | 3.75 | 3.40e-06 | 3.42 | 35 |
| 160 | 7.52e-10 | 5.67 | 2.72e-07 | 3.64 | 44 |
| 320 | 1.65e-11 | 5.50 | 1.09e-09 | 7.95 | 59 |
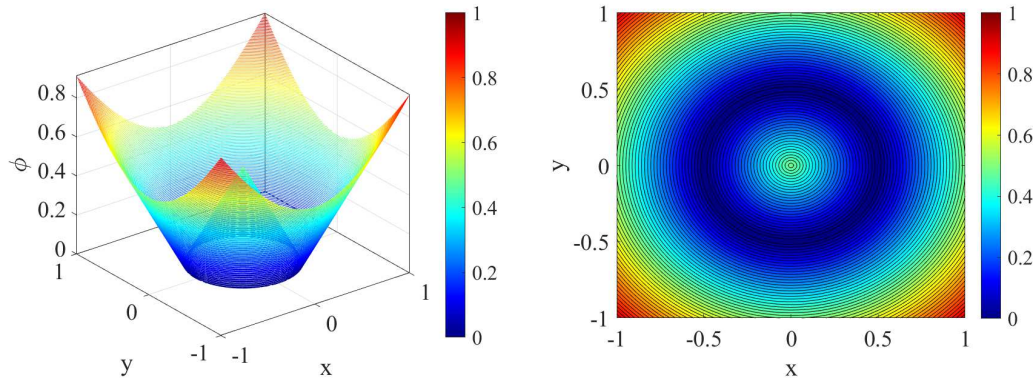
Fig. 3.3. Example 3.2. The numerical solution by Approach 1 on mesh $N = 160$. Left: the 3D plot of numerical solution $\phi$. Right: the contour plot for $\phi$.
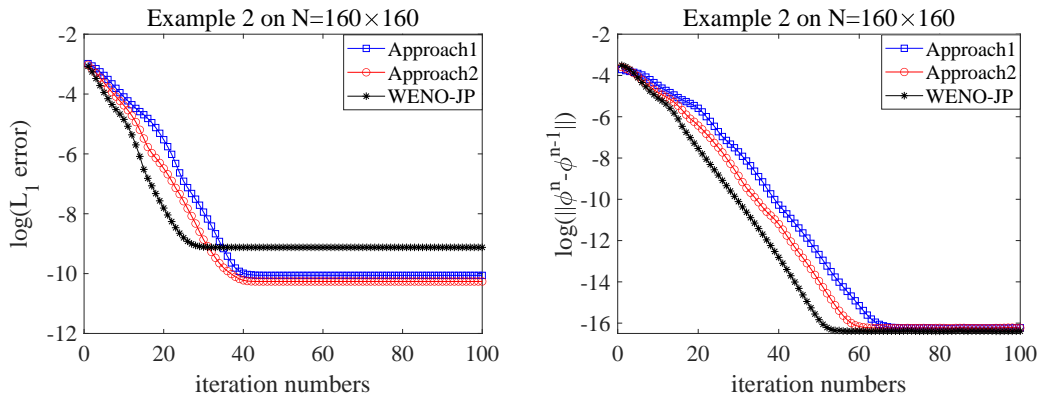


Fig. 3.4. Three methods solving Example 3.2. Left: number of iterations vs $\log(L_1\ error)$. Right: number of iterations vs $\log(\|\phi^n - \phi^{n-1}\|_{L_1})$.

**Example 3.3.** We solve the Eikonal equation with $f(x, y) = 1$. The computational domain is set as $[-3, 3]^2$, and the inflow boundary $\Gamma$ consists of two circles of equal radius 0.5 with the centers located at $(-1, 0)$ and $(\sqrt{1.5}, 0)$, respectively, that is

$$\Gamma = \left\{ (x, y) \mid (x+1)^2 + y^2 = \frac{1}{4} \quad or \quad (x - \sqrt{1.5})^2 + y^2 = \frac{1}{4} \right\}.$$

The exact solution is a distance function to the inflow boundary $\Gamma$, containing the singularities at the center of each circle and the line $x = 0.5(\sqrt{1.5} - 1)$ that is of equal distance to two circle centers.

Again, the Godunov numerical Hamiltonian is used. We measure the numerical errors within the box of $[-2.85, 2.85]^2$, which also excludes the boxes $[-1.15, -0.85] \times [-0.15, 0.15]$, $[\sqrt{1.5} - 0.15, \sqrt{1.5} + 0.15] \times [-0.15, 0.15]$ and $[\sqrt{0.375} - 0.65, \sqrt{0.375} - 0.35] \times [-2.85, 2.85]$. These excluded boxes contain two centers of $\Gamma$ and the singular line.

Fig. 3.5 shows that the numerical solution by Approach 1 on mesh $N = 160$. The numerical errors and orders of convergence are shown in Table 3.3. All three methods achieved the designed high order accuracy. Although WENO-JP method needs the least number of iterations, its

Table 3.3: Example 3.3. Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter |
|---|---|---|---|---|---|
| 80 | 7.13e-07 | - | 1.04e-04 | - | 58 |
| 160 | 1.09e-07 | 2.70 | 9.45e-06 | 3.47 | 55 |
| 320 | 2.97e-10 | 8.52 | 3.86e-07 | 4.61 | 85 |
| 640 | 6.82e-12 | 5.44 | 3.22e-9 | 6.90 | 139 |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter |
| 80 | 1.34e-06 | - | 8.57e-05 | - | 38 |
| 160 | 2.04e-07 | 2.72 | 1.86e-05 | 2.20 | 55 |
| 320 | 7.53e-10 | 8.08 | 4.45e-07 | 5.38 | 93 |
| 640 | 6.26e-12 | 6.91 | 7.86e-10 | 9.14 | 150 |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter |
| 80 | 1.12e-06 | - | 1.22e-04 | - | 55 |
| 160 | 1.19e-07 | 3.22 | 6.73e-06 | 4.18 | 45 |
| 320 | 2.19e-09 | 5.76 | 4.30e-07 | 4.02 | 52 |
| 640 | 5.90e-11 | 5.21 | 8.87e-09 | 5.59 | 78 |
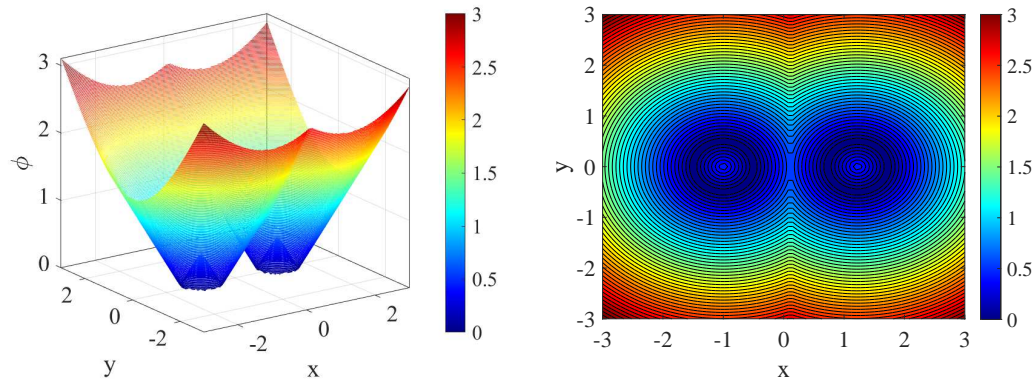


Fig. 3.5. Example 3.3. The numerical solution by Approach 1 on mesh $N = 160$. Left: the 3D plot of numerical solution $\phi$. Right: the contour plot for $\phi$.

numerical errors are also the largest among all three methods. On the other hand, Approach 1 and Approach 2 yield similar numerical errors and the number of iterations

**Example 3.4.** We solve the Eikonal equation with $f(x, y) = 1$. The computational domain is set as $[-1, 1]^2$, and the inflow boundary is given by $\Gamma = (0, 0)$. The exact solution for this problem is a distance function to $\Gamma$, and it contains a singularity at $\Gamma$.

The Godunov numerical Hamiltonian is used. Due to the singularity, we follow the setup in [29], and pre-assign the exact solution in a small box with length 0.3 around the source point. Numerical errors and orders are listed in Table 3.4. Again, we can observe that fifth order accuracy can be obtained for all schemes, and Approach 1 and 2 yield smaller numerical errors than the WENO-JP scheme.

Table 3.4: Example 3.4. Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter |
|---|---|---|---|---|---|
| 40 | 3.11e-07 | - | 4.60e-06 | - | 41 |
| 80 | 6.95e-09 | 5.48 | 1.62e-07 | 4.82 | 47 |
| 160 | 1.19e-10 | 5.85 | 1.78e-09 | 6.50 | 62 |
| 320 | 3.26e-12 | 5.20 | 8.67e-12 | 7.68 | 95 |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 8.42e-08 | - | 7.14e-07 | - | 32 |
| 80 | 2.88e-09 | 4.87 | 1.14e-08 | 5.96 | 40 |
| 160 | 1.04e-10 | 4.78 | 2.55e-10 | 5.47 | 53 |
| 320 | 3.57e-12 | 4.87 | 9.84e-12 | 4.70 | 77 |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 6.17e-06 | - | 9.31e-05 | - | 35 |
| 80 | 4.85e-07 | 5.03 | 7.34e-06 | 3.66 | 40 |
| 160 | 8.92e-09 | 5.76 | 2.05e-07 | 5.16 | 50 |
| 320 | 2.16e-10 | 5.36 | 2.68e-09 | 6.25 | 66 |

**Example 3.5.** We solve the Eikonal equation with $f(x, y) = 1$. The computational domain is set as $[-2, 2]^2$, and the inflow boundary $\Gamma$ is a sector of three quarters of the circle centered at $(0,0)$ with radius 0.5, closed with the $x$-axis and $y$-axis in the first quadrant, which can be described as

$$\Gamma = \left\{ (x,y) : \sqrt{x^2 + y^2} = 0.5, \text{ if } x < 0, y < 0 \right\} \cup$$
$$\left\{ (x,0) : 0 \leq x \leq 0.5 \right\} \cup \left\{ (0,y) : 0 \leq y \leq 0.5 \right\}.$$

The exact solution is still the distance function to $\Gamma$. Singularities appear at the two corners in $\Gamma$, which give rise to both shock and rarefaction wave in the solution.

The Godunov numerical Hamiltonian is used. We measure the errors in smooth regions inside the box of $[-1.9, 1.9]^2$ with $x \leq 0$ or $y \leq 0$, and outside the box $[-0.5, 0.5]^2$. The picture of numerical solution is presented in Fig. 3.6. Numerical errors and orders of convergence are
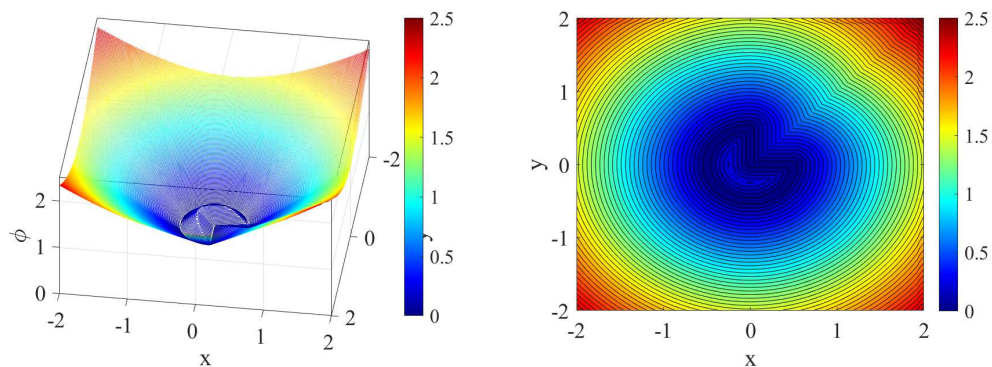


Fig. 3.6. Example 3.5. The numerical solution by Approach 1 on mesh $N = 160$. Left: the 3D plot of numerical solution $\phi$. Right: the contour plot for $\phi$.

Table 3.5: Example 3.5. Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter |
|---|---|---|---|---|---|
| 40 | 1.12e-06 | - | 1.61e-05 | - | 71 |
| 80 | 4.28e-08 | 4.71 | 1.08e-06 | 3.89 | 59 |
| 160 | 1.05e-09 | 5.33 | 3.93e-08 | 4.78 | 75 |
| 320 | 2.11e-11 | 5.64 | 4.44e-10 | 6.46 | 104 |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 8.84e-07 | - | 1.52e-05 | - | 33 |
| 80 | 3.04e-08 | 4.86 | 9.49e-07 | 4.00 | 41 |
| 160 | 1.07e-09 | 4.82 | 4.83e-08 | 4.29 | 65 |
| 320 | 2.66e-11 | 5.33 | 1.21e-09 | 5.31 | 108 |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 3.46e-06 | - | 2.68e-05 | - | 35 |
| 80 | 2.41e-07 | 3.84 | 1.10e-06 | 4.60 | 44 |
| 160 | 8.43e-09 | 4.84 | 4.83e-08 | 4.51 | 58 |
| 320 | 1.67e-10 | 5.65 | 9.94e-10 | 5.60 | 74 |

listed in Table 3.5. Again, the fifth order accuracy can be obtained in the smooth regions for all three methods, and Approach 1 and Approach 2 yield smaller numerical errors on the same mesh size. We also fix the mesh to be $N = 160$, and run all three methods with 100 iterations. The history of their numerical errors is shown in Fig. 3.7, from which we can clearly observe that Approach 1 yields the smallest error this time. WENO-JP method produces the largest error, and the error of Approach 2 stays between them. We plot the history of the quantity $\delta$ for comparison, where we can observe that all three methods converge to $\delta = 10^{-16}$.
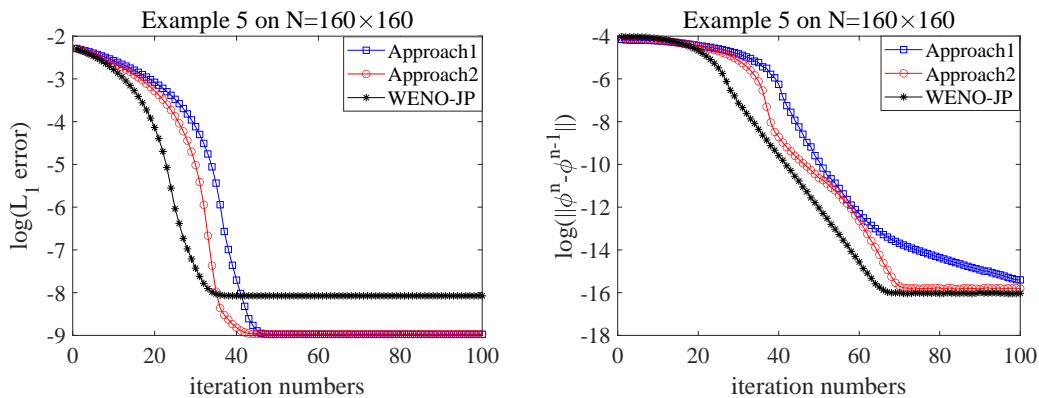


Fig. 3.7. Three methods solving Example 3.5. Left: number of iterations vs $\log(L_1\ error)$. Right: number of iterations vs $\log(||\phi^n - \phi^{n-1}||_{L_1})$.

**Example 3.6.** We solve the Eikonal equation with

$$f(x,y) = 2\pi\sqrt{[\cos(2\pi x)\sin(2\pi y)]^2 + [\sin(2\pi x)\cos(2\pi y)]^2}.$$

The computational domain is set as $\Omega = [0,1]^2$, and the inflow boundary condition is given

by $\Gamma = \{(\frac{1}{4}, \frac{1}{4}), (\frac{3}{4}, \frac{3}{4}), (\frac{1}{4}, \frac{3}{4}), (\frac{3}{4}, \frac{1}{4}), (\frac{1}{2}, \frac{1}{2})\}$, consisting of five isolated points. $\phi(x, y) = 0$ is prescribed at the boundary of the unit square. The exact solution of this problem is the shape function [32]. Two cases are considered here, based on different boundary conditions:

*Case (a):*

$$g\left(\frac{1}{4}, \frac{1}{4}\right) = g\left(\frac{3}{4}, \frac{3}{4}\right) = 1, \quad g\left(\frac{1}{4}, \frac{3}{4}\right) = g\left(\frac{3}{4}, \frac{1}{4}\right) = -1, \quad g\left(\frac{1}{2}, \frac{1}{2}\right) = 0$$

with the exact solution being

$$\phi(x, y) = \sin(2\pi x)\sin(2\pi y).$$

*Case (b):*

$$g\left(\frac{1}{4}, \frac{1}{4}\right) = g\left(\frac{3}{4}, \frac{3}{4}\right) = g\left(\frac{1}{4}, \frac{3}{4}\right) = g\left(\frac{3}{4}, \frac{1}{4}\right) = 1, \quad g\left(\frac{1}{2}, \frac{1}{2}\right) = 2$$

with the exact solution being

$$\phi(x, y) = \begin{cases} \max\left(|\sin(2\pi x)\sin(2\pi y)|, 1 + \cos(2\pi x)\cos(2\pi y)\right), & \text{if} \quad |x + y - 1| < 1/2 \\ \qquad\qquad\qquad\qquad \text{and } |x - y| < 1/2, \\ |\sin(2\pi x)\sin(2\pi y)|, & \text{otherwise}, \end{cases}$$

which is not smooth.

Due to the singularity of these point sources, the exact solutions are placed in a small box with a length $2h$ around these isolated points in both test cases. The Godunov numerical Hamiltonian is used in this test.

For *Case a*, the Fig. 3.8 shows that numerical solution. The numerical errors and orders of convergence of three methods are listed in Table 3.6. We can see that the fifth order accuracy can be obtained, and the errors of the Approach 1 and 2 are smaller than that of the WENO-JP schemes. For this example, we would comment that, the number of iterations depends on the parameter $\epsilon$ in (2.20) in order to achieve the desired fifth order. We also fix the mesh to be $N = 160$, and run all three methods with 100 iterations. The history of their numerical errors is
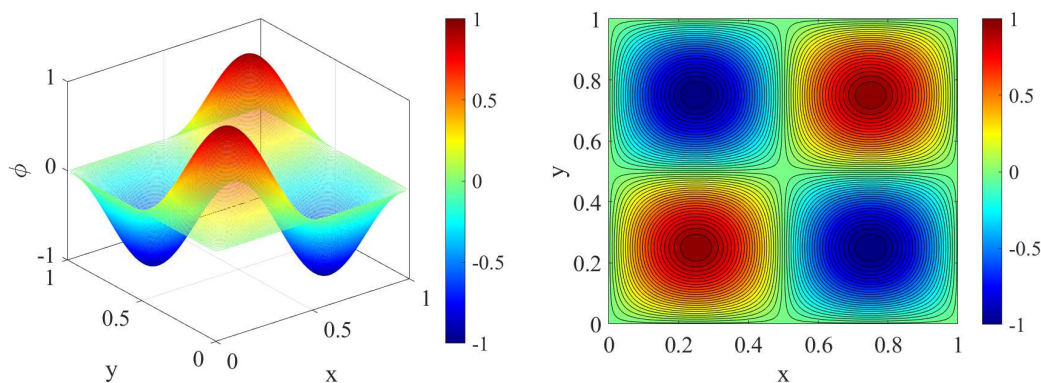


Fig. 3.8. Example 3.6 (Case a). The numerical solution by Approach 1 on mesh $N = 160$. Left: the 3D plot of numerical solution $\phi$. Right: the contour plot for $\phi$.
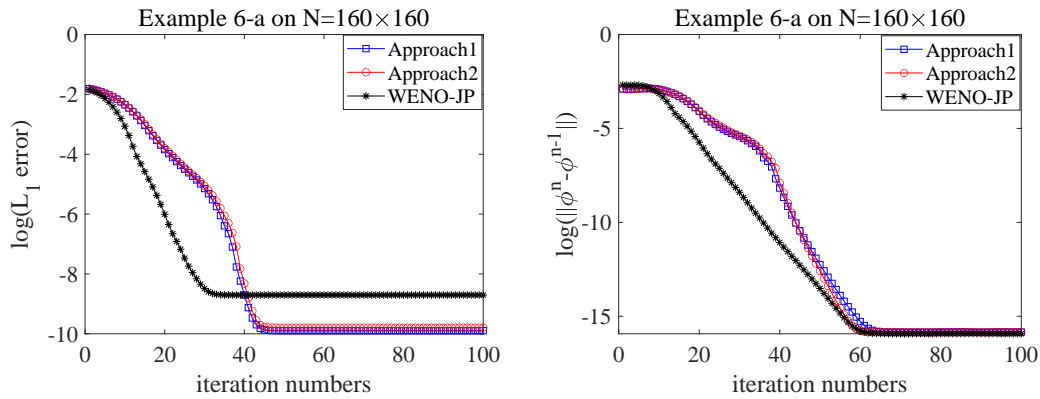
Fig. 3.9. Three methods solving Example 3.6 (Case a). Left: number of iterations vs $\log(L_1\ error)$; Right: number of iterations vs $\log(\|\phi^n - \phi^{n-1}\|_{L_1})$.

shown in Fig. 3.9, from which we can clearly observe that Approach 1 and Approach 2 yield the smallest error, and WENO-JP method produces the largest error. We plot the history of the quantity $\delta$ for comparison, where we can observe that all three methods converge to $\delta = 10^{-16}$ after the 65th iteration.

For *Case b*, we modified the convergence criteria to $\delta < 10^{-12}$ since the solution is not smooth. The numerical errors and orders of convergence are listed in Table 3.7. Due to the non-smoothness of the exact solution, we can only achieve second order accuracy.

**Remark 3.1.** Similar to the observations in [20, 29], we would like to comment that if we take a fixed $\epsilon$, e.g., $\epsilon = 10^{-3}$ or $\epsilon = 10^{-6}$, the three methods may either lose order or even blow up during the mesh refinement.

Table 3.6: Example 3.6 (Case a). Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter | $\epsilon$ |
|---|---|---|---|---|---|---|
| 40 | 1.91e-07 | - | 2.05e-06 | - | 46 | $10^{-2}$ |
| 80 | 3.96e-09 | 5.59 | 1.59e-08 | 7.01 | 45 | $10^{-3}$ |
| 160 | 1.27e-10 | 4.96 | 5.33e-10 | 4.90 | 56 | $10^{-4}$ |
| 320 | 4.19e-12 | 4.92 | 1.73e-11 | 4.94 | 89 | $10^{-5}$ |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter | $\epsilon$ |
| 40 | 2.58e-07 | - | 3.25e-06 | - | 41 | $10^{-2}$ |
| 80 | 4.97e-09 | 5.69 | 2.46e-08 | 7.04 | 45 | $10^{-3}$ |
| 160 | 1.60e-10 | 4.95 | 5.23e-10 | 5.56 | 54 | $10^{-4}$ |
| 320 | 5.38e-12 | 4.89 | 1.51e-11 | 5.11 | 91 | $10^{-5}$ |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter | $\epsilon$ |
| 40 | 4.61e-07 | - | 2.10e-06 | - | 36 | $10^{-2}$ |
| 80 | 4.83e-08 | 3.25 | 1.93e-07 | 3.44 | 37 | $10^{-3}$ |
| 160 | 1.97e-09 | 4.61 | 7.18e-09 | 4.75 | 52 | $10^{-4}$ |
| 320 | 4.12e-11 | 5.58 | 1.50e-10 | 5.58 | 71 | $10^{-4}$ |

Table 3.7: Example 3.6 (Case b). Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter | $\epsilon$ |
|---|---|---|---|---|---|---|
| 40 | 4.71e-04 | - | 4.05e-03 | - | 34 | $10^{-2}$ |
| 80 | 1.43e-04 | 1.71 | 1.23e-03 | 1.71 | 42 | $10^{-3}$ |
| 160 | 3.16e-05 | 2.18 | 3.02e-04 | 2.02 | 56 | $10^{-4}$ |
| 320 | 6.75e-06 | 2.22 | 7.30e-05 | 2.04 | 92 | $10^{-5}$ |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter | $\epsilon$ |
| 40 | 6.48e-04 | - | 4.43e-03 | - | 38 | $10^{-2}$ |
| 80 | 1.63e-04 | 1.98 | 1.31e-03 | 1.75 | 43 | $10^{-3}$ |
| 160 | 3.42e-05 | 2.25 | 3.22e-04 | 2.02 | 60 | $10^{-4}$ |
| 320 | 8.22e-06 | 2.06 | 6.15e-05 | 2.38 | 100 | $10^{-5}$ |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter | $\epsilon$ |
| 40 | 1.75e-04 | - | 2.11e-03 | - | 34 | $10^{-2}$ |
| 80 | 1.14e-04 | 0.61 | 1.12e-03 | 0.91 | 38 | $10^{-3}$ |
| 160 | 3.78e-05 | 1.60 | 3.18e-04 | 1.81 | 49 | $10^{-4}$ |
| 320 | 1.05e-05 | 1.83 | 8.79e-05 | 1.85 | 68 | $10^{-5}$ |

**Example 3.7.** The travel-time problem in elastic wave propagation is considered in this example. The quasi-P and the quasi-SV slowness surfaces are defined as follows [14]:

$$c_1\phi_x^4 + c_2\phi_x^2\phi_y^2 + c_3\phi_y^4 + c_4\phi_x^2 + c_5\phi_y^2 + 1 = 0,$$

where

$$c_1 = a_{11}a_{44}, \quad c_2 = a_{11}a_{33} + a_{44}^2 - (a_{13} + a_{44})^2,$$
$$c_3 = a_{33}a_{44}, \quad c_4 = -(a_{11} + a_{44}), \quad c_5 = -(a_{33} + a_{44}),$$

in which $a_{i,j}$ are given elastic parameters. The quasi-P wave Eikonal equation is

$$\sqrt{-\frac{1}{2}\left(c_4\phi_x^2 + c_5\phi_y^2\right) + \sqrt{\frac{1}{4}\left(c_4\phi_x^2 + c_5\phi_y^2\right)^2 - \left(c_1\phi_x^4 + c_2\phi_x^2\phi_y^2 + c_3\phi_y^4\right)}} = 1,$$

which is a convex HJ equation, and the elastic parameters are taken to be

$$a_{11} = 15.0638, \quad a_{33} = 10.8373, \quad a_{13} = 1.6381, \quad a_{44} = 3.1258.$$

The corresponding quasi-SV wave Eikonal equation is given by

$$\sqrt{-\frac{1}{2}\left(c_4\phi_x^2 + c_5\phi_y^2\right) - \sqrt{\frac{1}{4}\left(c_4\phi_x^2 + c_5\phi_y^2\right)^2 - \left(c_1\phi_x^4 + c_2\phi_x^2\phi_y^2 + c_3\phi_y^4\right)}} = 1,$$

which is a nonconvex HJ equation, and the elastic parameters are taken to be

$$a_{11} = 15.90, \quad a_{33} = 6.21, \quad a_{13} = 4.82, \quad a_{44} = 4.00.$$

The computational domain is set as $\Omega = [-1, 1]^2$, and the inflow boundary is given by $\Gamma = (0, 0)$. Exact values are assigned in a small box with length 0.3 around the source point. Because these

Table 3.8: Example 3.7 P-wave. Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter |
|---|---|---|---|---|---|
| 40 | 4.78e-06 | - | 3.95e-05 | - | 41 |
| 80 | 2.07e-07 | 4.52 | 2.25e-06 | 4.13 | 44 |
| 160 | 7.05e-09 | 4.87 | 8.09e-08 | 4.80 | 56 |
| 320 | 2.29e-10 | 4.93 | 2.61e-09 | 4.95 | 77 |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 5.77e-06 | - | 4.35e-05 | - | 34 |
| 80 | 2.27e-07 | 4.66 | 2.30e-06 | 4.23 | 42 |
| 160 | 7.42e-09 | 4.93 | 7.44e-08 | 4.95 | 54 |
| 320 | 2.40e-10 | 4.94 | 2.24e-09 | 5.05 | 78 |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter |
| 40 | 2.56e-05 | - | 2.67e-04 | - | 37 |
| 80 | 1.64e-06 | 3.96 | 2.05e-05 | 3.70 | 44 |
| 160 | 5.46e-08 | 4.90 | 6.85e-07 | 4.90 | 56 |
| 320 | 1.51e-09 | 5.17 | 1.44e-08 | 5.57 | 83 |

Table 3.9: Example 3.7 SV-wave. Comparison of the three methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Approach 1 | $L_1$ error | order | $L_\infty$ error | order | iter |
|---|---|---|---|---|---|
| 80 | 9.63e-07 | - | 1.60e-05 | - | 71 |
| 160 | 1.95e-08 | 5.61 | 8.97e-07 | 4.15 | 92 |
| 320 | 6.37e-11 | 8.26 | 9.12e-09 | 6.61 | 112 |
| 640 | 5.60e-13 | 6.82 | 1.65e-11 | 9.10 | 190 |
| Approach 2 | $L_1$ error | order | $L_\infty$ error | order | iter |
| 80 | 1.76e-06 | - | 2.39e-05 | - | 43 |
| 160 | 5.03e-09 | 7.64 | 3.91e-07 | 5.93 | 62 |
| 320 | 2.85e-11 | 7.46 | 1.03e-09 | 8.56 | 95 |
| 640 | 7.88e-13 | 5.17 | 5.95e-11 | 4.11 | 170 |
| WENO-JP FSM | $L_1$ error | order | $L_\infty$ error | order | iter |
| 80 | 1.28e-06 | - | 2.02e-05 | - | 50 |
| 160 | 1.99e-08 | 6.01 | 8.26e-07 | 4.61 | 70 |
| 320 | 1.80e-10 | 6.79 | 1.15e-08 | 6.16 | 108 |
| 640 | 4.11e-12 | 5.45 | 1.16e-10 | 6.62 | 181 |

Hamiltonians are pretty complicated, we use the Lax-Friedrich numerical Hamiltonian for both equations. In addition, since we only know the numerical solution of $\phi$, the "exact solution" of $u$ and $v$ on *Category I* will be obtained by fifth order WENO-JP reconstruction.

For quasi-P wave equation, the numerical errors and orders of convergence are presented in Table 3.8 for three methods. We can observe that all three methods achieve the fifth order
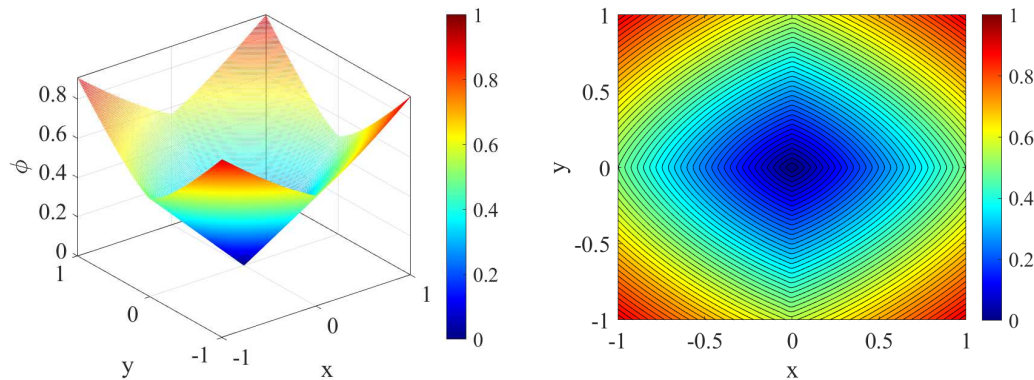
Fig. 3.10. Example 3.7 SV-wave. The numerical solution by Approach 1 on mesh $N = 160$. Left: the 3D plot of numerical solution $\phi$. Right: the contour plot for $\phi$.

accuracy as we expected. Moreover, the numbers of iterations required by three methods are basically the same, while Approach 1 and Approach 2 produce smaller numerical errors. The CPU time are plotted in Fig. 4.7, which shows that WENO-JP costs the least CPU time, but it has the largest error. And, Approach 1 and Approach 2 have the same error, but Approach 1 costs less CPU time than Approach 2. We provide the comparison of these numerical results in Fig. 4.7, and the left subfigure shows the mesh vs. CPU time of three methods on different meshes, from which one can observe that Approach 1 enjoys more savings in computational time than Approach 2 with the refinement of mesh.

For the quasi-SV wave equation, we set the $\delta < 10^{-9}$ for Approach 2 on mesh $N = 80, 160, 320$, otherwise the Approach 2 will not convergence. The errors are measured in the region away from the singular lines of $x = 0$ and $y = 0$. The Fig. 3.10 shows that picture of numerical solution for SV-wave. The numerical errors and orders of convergence are shown in Table 3.9, from which we observe that the designed fifth order accuracy is again achieved. Approach 1 and Approach 2 yield smaller numerical errors for this test case, and in the same time, Approach 2 requires slightly less numbers of iterations when compared with WENO-JP method.

## 4. Hybrid Strategy

In the HWENO reconstruction procedure, the evaluation of the smoothness indicators occupies most of the extra computational costs when compared with the simple linear reconstruction. In this section, we explore a hybrid method which combines both linear and HWENO method, similar to the hybrid WENO fast sweeping method studied in [20]. The main idea is to adopt the fifth order linear reconstruction on the big stencil $S_0$ or $\tilde{S}_0$ if the numerical solution is monotone, that is $u_{i,j}$ do not change sign in either $S_0$ or $\tilde{S}_0$. Otherwise, the HWENO reconstruction is used.

In Section 2.5, the points $\{(x_i, y_j)\}$ were classified into two categories. Here, we further separate the *Category IV* into the following two subcategories, which will be handled slightly differently in the hybrid method.

*Category IV.1*: For points whose distances to *Category III* are less than or equal to $2h$ (excluding *Category I*). These points will be updated by FSM.

Table 4.1: All examples. Comparison of the two hybrid methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

| Test | N | $A1 - L_1$ | order | $A1 - L_\infty$ | order | iter | $A2 - L_1$ | order | $A2 - L_\infty$ | order | iter |
|------|-----|----------|-------|---------------|-------|------|----------|-------|---------------|-------|------|
| 1 | 40 | 2.51e-06 | - | 1.94e-05 | - | 39 | 1.81e-07 | - | 1.35e-06 | - | 33 |
| | 80 | 3.80e-08 | 6.04 | 7.13e-07 | 4.76 | 46 | 1.33e-09 | 7.08 | 5.25e-08 | 4.68 | 41 |
| | 160 | 1.94e-10 | 7.60 | 7.69e-09 | 6.53 | 58 | 9.82e-12 | 7.08 | 1.62e-11 | 11.65 | 51 |
| | 320 | 2.49e-13 | 9.60 | 8.37e-12 | 9.84 | 76 | 3.04e-13 | 5.01 | 2.17e-12 | 2.90 | 65 |
| 2 | 40 | 6.04e-07 | - | 2.34e-05 | - | 37 | 1.92e-07 | - | 1.09e-05 | - | 25 |
| | 80 | 1.17e-08 | 5.68 | 1.74e-06 | 3.74 | 44 | 1.65e-09 | 6.85 | 1.87e-07 | 5.85 | 31 |
| | 160 | 8.74e-11 | 7.06 | 2.21e-08 | 6.30 | 57 | 5.45e-11 | 4.92 | 3.67e-09 | 5.67 | 42 |
| | 320 | 1.78e-12 | 5.61 | 1.30e-10 | 7.40 | 83 | 2.01e-12 | 4.75 | 1.23e-10 | 4.89 | 61 |
| 3 | 80 | 7.14e-07 | - | 1.04e-04 | - | 59 | 1.57e-06 | - | 8.57e-05 | - | 35 |
| | 160 | 1.94e-07 | 1.87 | 1.81e-05 | 2.53 | 55 | 3.19e-07 | 2.30 | 4.26e-05 | 1.00 | 47 |
| | 320 | 3.14e-10 | 9.27 | 3.86e-07 | 5.55 | 67 | 9.77e-10 | 8.35 | 6.55e-07 | 6.02 | 69 |
| | 640 | 6.41e-12 | 5.61 | 1.25e-09 | 8.26 | 121 | 6.44e-12 | 7.24 | 7.85e-10 | 9.70 | 117 |
| 4 | 40 | 3.10e-07 | - | 4.60e-06 | - | 42 | 8.42e-08 | - | 7.14e-07 | - | 30 |
| | 80 | 6.95e-09 | 5.48 | 1.62e-07 | 4.82 | 49 | 2.88e-09 | 4.87 | 1.14e-08 | 5.96 | 36 |
| | 160 | 1.19e-10 | 5.85 | 1.78e-09 | 6.51 | 64 | 1.04e-10 | 4.78 | 2.55e-10 | 5.47 | 48 |
| | 320 | 3.26e-12 | 5.20 | 8.67e-12 | 7.68 | 88 | 3.57e-12 | 4.87 | 1.80e-11 | 3.82 | 73 |
| 5 | 40 | 1.12e-06 | - | 1.61e-05 | - | 84 | 8.84e-07 | - | 1.52e-05 | - | 32 |
| | 80 | 4.28e-08 | 4.71 | 1.08e-06 | 3.89 | 60 | 4.11e-08 | 4.42 | 1.11e-06 | 3.77 | 42 |
| | 160 | 1.05e-09 | 5.33 | 3.93e-08 | 4.78 | 74 | 1.41e-09 | 4.85 | 5.95e-08 | 4.22 | 66 |
| | 320 | 2.11e-11 | 5.64 | 4.44e-10 | 6.46 | 92 | 3.17e-11 | 5.47 | 1.55e-09 | 5.26 | 105 |
| 6a | 40 | 1.71e-07 | - | 2.05e-06 | - | 47 | 2.26e-07 | - | 3.25e-06 | - | 44 |
| | 80 | 3.45e-09 | 5.63 | 1.57e-08 | 7.02 | 46 | 4.28e-09 | 5.72 | 2.46e-08 | 7.04 | 46 |
| | 160 | 1.20e-10 | 4.84 | 5.25e-10 | 4.90 | 52 | 1.52e-10 | 4.81 | 5.23e-10 | 5.56 | 51 |
| | 320 | 4.11e-12 | 4.87 | 1.72e-11 | 4.92 | 74 | 5.29e-12 | 4.84 | 1.39e-11 | 5.23 | 80 |
| 6b | 40 | 3.35e-04 | - | 2.79e-03 | - | 55 | 5.18e-04 | - | 4.05e-03 | - | 30 |
| | 80 | 9.15e-05 | 1.87 | 6.44e-04 | 2.11 | 35 | 8.45e-05 | 2.61 | 4.86e-04 | 3.05 | 34 |
| | 160 | 2.39e-05 | 1.93 | 1.67e-04 | 1.94 | 42 | 1.55e-05 | 2.44 | 1.28e-04 | 1.91 | 47 |
| | 320 | 4.28e-06 | 2.48 | 4.41e-05 | 1.92 | 71 | 4.56e-06 | 1.76 | 3.91e-05 | 1.71 | 80 |
| 7p | 40 | 4.78e-06 | - | 3.95e-05 | - | 46 | 5.77e-06 | - | 4.35e-05 | - | 33 |
| | 80 | 2.07e-07 | 4.52 | 2.25e-06 | 4.13 | 48 | 2.27e-07 | 4.66 | 2.30e-06 | 4.23 | 41 |
| | 160 | 7.05e-09 | 4.87 | 8.09e-08 | 4.80 | 56 | 7.42e-09 | 4.93 | 7.44e-08 | 4.95 | 51 |
| | 320 | 2.29e-10 | 4.93 | 2.61e-09 | 4.95 | 82 | 2.40e-10 | 4.94 | 2.24e-09 | 5.05 | 80 |
| 7sv | 80 | 8.61e-07 | - | 2.19e-05 | - | 68 | 8.58e-07 | - | 2.09e-05 | - | 35 |
| | 160 | 2.25e-08 | 5.25 | 1.44e-06 | 3.92 | 78 | 6.08e-09 | 7.14 | 5.04e-07 | 5.37 | 54 |
| | 320 | 7.85e-11 | 8.16 | 7.90e-09 | 7.51 | 113 | 2.90e-11 | 7.71 | 1.45e-09 | 8.44 | 126 |
| | 640 | 5.68e-13 | 7.11 | 1.84e-11 | 8.74 | 189 | 7.85e-13 | 5.20 | 2.57e-11 | 5.82 | 162 |

*Category IV.2*: All remaining points, which will also be updated by FSM.

Below, we present the flowchart for Approach 1, coupled with the hybrid strategy.

**Step 1**. *Initialization*: The numerical solution from the first order fast sweeping method [33] is taken as the initial guess of $\phi$. The forward or backward difference of this $\phi$ is used as the initial guess of $u$ and $v$.

**Step 2**. *Gauss-Seidel iteration*. We Solve the discretized nonlinear system by GS iterations with four alternating direction sweepings

$$(1) \quad i = 1 : N_x, \quad j = 1 : N_y, \quad (2) \quad i = N_x : 1, \quad j = 1 : N_y,$$

$$(3) \quad i = N_x : 1, \quad j = N_y : 1, \quad (4) \quad i = 1 : N_x, \quad j = N_y : 1.$$

During each sweeping, the updating strategy for the points in *Category IV* is outlined below. For the points in *Category IV.1*: the HWENO reconstruction (2.19) is applied to evaluate $(\phi_x)_{i,j}^{\pm}$, and the similar procedure for $(\phi_y)_{i,j}^{\pm}$ is used. For the points in *Category IV.2*, the hybrid strategy is applied here, and we define

$$(\phi_x)_{i,j}^{\pm} = \begin{cases} (2.17a) \text{ or } (2.18a), & \text{if } \{u_{i,j}\} \text{ have the same sign on } S_0 \text{ or } \widetilde{S_0}, \\ (2.19), & \text{otherwise.} \end{cases} \tag{4.1}$$

Similarly, one can evaluate $(\phi_y)_{i,j}^{\pm}$ along the $y$-direction. The rest of the algorithm is the same as before. $\phi_{i,j}^{new}$ is updated using (2.4) or (2.8), and $u_{i,j}^{new}$ and $v_{i,j}^{new}$ are evaluated by (2.10) in each sweeping direction. High order extrapolations are used at the ghost points.

**Step 3**. *Convergence*: In general, the iteration will stop if, for two consecutive iteration steps,

$$\delta = \|\phi^{new} - \phi^{old}\|_{L_1} < 10^{-14}.$$

The procedure of Approach 2 with hybrid strategy is similar, and is omitted here.

We expect these hybrid HWENO FSM to be more efficient. Next, some numerical results of the hybrid HWENO FSM will be presented. All the seven examples in Section 3 have been tested using the hybrid algorithms, and the comparison of their numerical performance with those of two HWENO methods in Section 2 are presented in Table 4.1. In Figs. 4.1-4.7, the comparison of their CPU time, number of iterations and $L_1$ numerical errors of some examples on various mesh size is provided. In the table, we denote the $L_1$ error and $L_\infty$ error in Approach 1 with hybrid strategy by "A1-$L_1$" and "A1-$L_\infty$", respectively. Similar notations
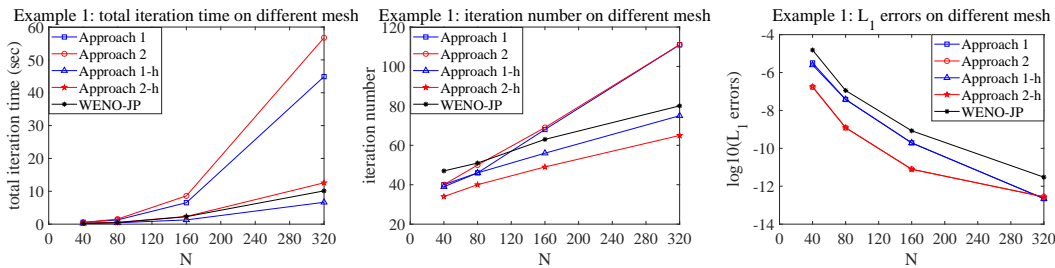


Fig. 4.1. Example 3.1 with Approach 1, 2, the hybrid Approach 1, 2 and WENO-JP. Left: mesh number $N$ vs CPU time. Middle: mesh number $N$ vs number of iterations. Right: mesh number $N$ vs $L_1$ error.
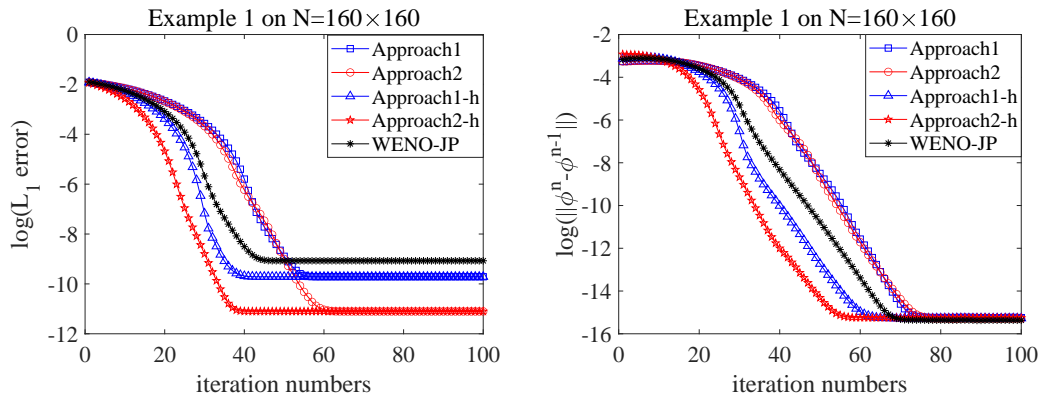
Fig. 4.2. Fifth methods solving Example 3.1. Left: number of iterations vs $\log(L_1\ error)$. Right: number of iterations vs $\log(\|\phi^n - \phi^{n-1}\|_{L_1})$.



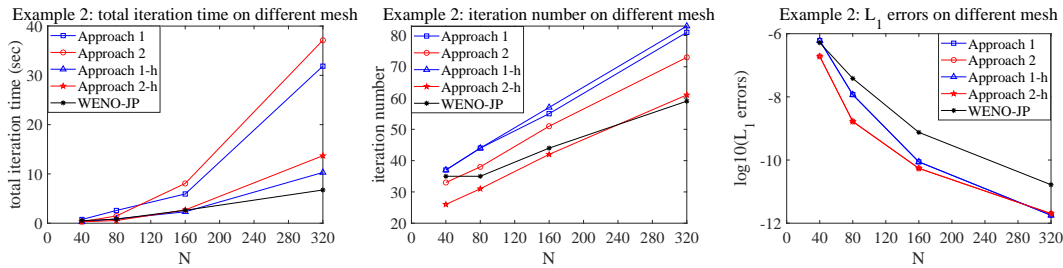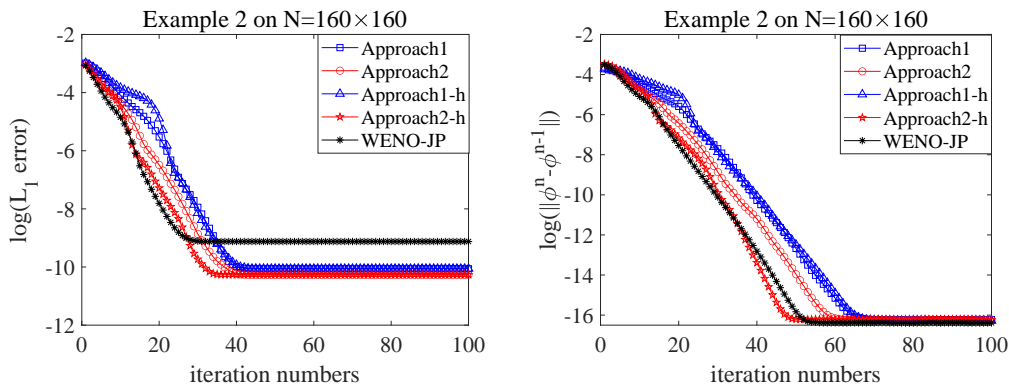Fig. 4.3. Example 3.2 with Approach 1, 2, the hybrid Approach 1, 2 and WENO-JP. Left: mesh number $N$ vs CPU time. Middle: mesh number $N$ vs number of iterations. Right: mesh number $N$ vs $L_1$ error.



Fig. 4.4. Three methods solving Example 3.2. Left: number of iterations vs $\log(L_1\ error)$. Right: number of iterations vs $\log(\|\phi^n - \phi^{n-1}\|_{L_1})$.

are adopted for Approach 2 with hybrid strategy. In these figures, we denote Approach 1 with hybrid strategy by "Approach 1-h", and Approach 2 with hybrid strategy by "Approach 2-h". The "blue -□-" lines indicate Approach 1, the "blue -△-" lines indicate hybrid Approach 1, the "red -○-" lines indicate Approach 2, the "red -⋆-" lines indicate hybrid Approach 2, and the "black -∗-" lines indicate WENO-JP. The table and all of these figures demonstrate that the hybrid schemes cost much less CPU time, and converge with a smaller number of iterations when
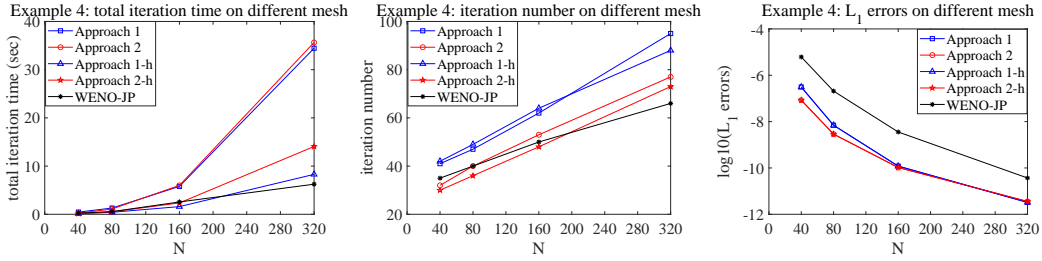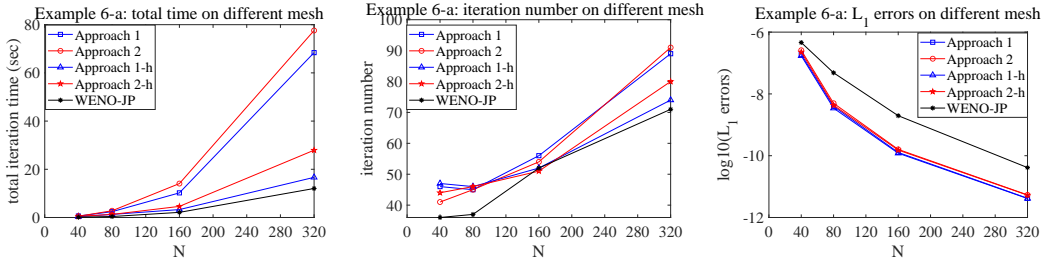
Fig. 4.5. Example 3.4 with Approach 1, 2, the hybrid Approach 1, 2 and WENO-JP. Left: mesh number $N$ vs CPU time. Middle: mesh number $N$ vs number of iterations. Right: mesh number $N$ vs $L_1$ error.
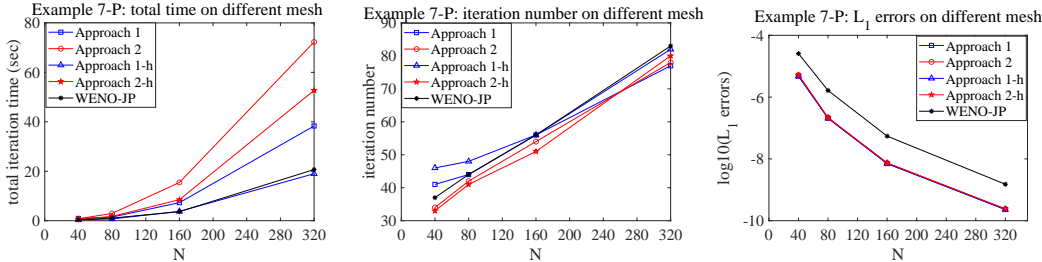


Fig. 4.6. Example 3.6 (Case a) with Approach 1, 2, the hybrid Approach 1, 2 and WENO-JP. Left: mesh number $N$ vs CPU time. Middle: mesh number $N$ vs number of iterations. Right: mesh number $N$ vs $L_1$ error.



Fig. 4.7. Example 3.7 P-wave with Approach 1, 2, the hybrid Approach 1, 2 and WENO-JP. Left: mesh number $N$ vs CPU time. Middle: mesh number $N$ vs number of iterations. Right: mesh number $N$ vs $L_1$ error.

the mesh is refined. In addition, for most of these examples, Approach 1 enjoys more savings in computational time than Approach 2. The computational time of the hybrid HWENO FSM is comparable to that of the WENO-JP scheme, yet the numerical error of WENO-JP scheme is the largest among all these five methods, when the same mesh size is considered.

**Remark 4.1.** At the end of this section, we propose a small trick to further reduce the computational cost, namely to freeze the nonlinear weights. Since the targeting problem is steady state and iterative method is used to solve it, the nonlinear weights change only slightly when the iteration is close to converge. Therefore, when $L_1$ error of the nonlinear weights between two iteration steps is less than the given threshold, we can fix these nonlinear weights in following iterations until convergence. Numerical tests demonstrate that the value of $\epsilon_1$ can be taken as $10^{-4}$ or $10^{-5}$, and this could effectively reduce the computational costs.

## 5. Conclusion Remark

In this work, we have combined the fifth order HWENO-ZZQ scheme with the fast sweeping idea, to design efficient algorithms for directly solving static Hamilton-Jacobi equations. The novel approach seeks to use the updated $\phi$ to directly approximate the spatial derivatives of $\phi$, and there is no need to introduce and solve additional equations. As a comparison, the second approach is based on the traditional HWENO idea, with additional equations governing the spatial derivatives of $\phi$. The first approach has great savings in computational time, which improves the computational efficiency of the traditional HWENO scheme. Extensive numerical experiments demonstrate that these two methods perform well numerically and lead to smaller numerical errors when compared with WENO methods. A hybrid strategy which combines both linear and HWENO-ZZQ reconstruction is also proposed and tested, which yields additional savings in computational time. Especially, the hybrid version of the proposed novel HWENO method enjoys more savings in computational time than the traditional HWENO approach.

## References

[1] M. Boué and P. Dupuis. Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control, *SIAM J. Numer. Anal.*, **36**:3 (1999), 667–695.

[2] M.G. Crandall and P.L. Lions, Viscosity solutions of Hamilton-Jacobi equations, *Trans. Amer. Math. Soc.*, **277**:1 (1983), 1–42.

[3] J. Helmsen, E. Puckett, P. Colella and M. Dorr, Two new methods for simulating photolithography development in 3D, *Proc. SPIE*, **27**:26 (1996), 253–262.

[4] L. Huang, C.W. Shu and M.P. Zhang, Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation, *J. Comput. Math.*, **26**:3 (2008), 336–346.

[5] L. Huang, S.C. Wong, M. Zhang, C.W. Shu and W.H.K. Lam, Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm, *Transport. Res. B*, **43**:1 (2009), 127–141.

[6] G.S. Jiang and D.P. Peng, Weighted ENO schemes for Hamilton-Jacobi equations, *SIAM J. Sci. Comput.*, **21**:6 (2000), 2126–2143.

[7] C.Y. Kao, S. Osher and J. Qian, Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations, *J. Comput. Phys.*, **196**:1 (2004), 367–391.

[8] C.-Y. Kao, S. Osher and Y.H. Tsai, Fast sweeping methods for static Hamilton-Jacobi equations, *SIAM J. Numer. Anal.*, **42**:6 (2005), 2612–2632.

[9] D. Levy, G. Puppo and G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, *ESAIM-Math. Model. Num.*, **33**:3 (1999), 547–571.

[10] F. Li, C.W. Shu, Y.T. Zhang and H. Zhao, Second order discontinuous Galerkin fast sweeping method for Eikonal equations, *J. Comput. Phys.*, **227**:17 (2008), 8191–8208.

[11] H. Liu and J. Qiu, Finite Difference Hermite WENO schemes for conservation laws, *J. Sci. Comput.*, **63**:2 (2015), 548–572.

[12] S. Luo, A uniformly second order fast sweeping method for Eikonal equations, *J. Comput. Phys.*, **241**:10 (2013), 104–117.

[13] S. Osher and C.W. Shu, High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations, *SIAM J. Numer. Anal.*, **28**:4 (1991), 907–922.

[14] J. Qian, L.T. Cheng and S. Osher, A level set based Eulerian approach for anisotropic wave propagations, *Wave Motion*, **37**:4 (2003), 365–379.

[15] J. Qian, Y.T. Zhang and H.K. Zhao, A fast sweeping method for static convex Hamilton-Jacobi equations, *J. Sci. Comput.*, **31**:1 (2007), 237–271.

[16] S. Serna and J. Qian, A stopping criterion for higher-order sweeping schemes for static Hamilton-Jacobi equations, *J. Comput. Math.*, **28**:4 (2010), 552–568.

[17] 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**:1 (2004), 115–135.

[18] J. Qiu and C.W. Shu, Hermite WENO schemes and their application as limiters for Runge-Kutta discontinuous Galerkin method II: two-dimensional case, *Comput. Fluids*, **34**:6 (2005), 642-663.

[19] J. Qiu and C.W. Shu, Hermite WENO schemes for Hamilton-Jacobi equations, *J. Comput. Phys.*, **204**:1 (2005), 82–99.

[20] Y. Ren, T. Xiong and J. Qiu, A hybrid finite difference WENO-ZQ fast sweeping method for static Hamilton-Jacobi equations, *J. Sci. Comput.*, **83** (2020), 54.

[21] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Nat. Acad. Sci.*, **93**:4 (1996), 1591–1595.

[22] C.W. Shu, High order numerical methods for time dependent Hamilton-Jacobi equations, IMS Lecture Notes Series, Vol 11: *Mathematics and Computation in Imaging Science and Information Processing*, World Scientific Publishing, Singapore, 2007, 47-91.

[23] S. Tan and C.W. Shu, Inverse Lax-Wendroff procedure for numerical boundary conditions of conservation laws, *J. Comput. Phys.*, **229**:21 (2010), 8144–8166.

[24] Z. Tao, F. Li and J. Qiu, High-order central Hermite WENO schemes on staggered meshes for hyperbolic conservation laws, *J. Comput. Phys.*, **281** (2015), 148–176.

[25] R. Tsai, L.T. Cheng, S. Osher and H.-K. Zhao, Fast sweeping algorithms for a class of Hamilton-Jacobi equations, *SIAM J. Numer. Anal.*, **41**:2 (2003), 673–694.

[26] J.N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, *IEEE T. Automat. Contr.*, **40**:9 (1995), 1528–1538.

[27] L. Wu and Y.T. Zhang, A third order fast sweeping method with linear computational complexity for Eikonal equations, *J. Sci. Comput.*, **62**:1 (2015), 198–229.

[28] Y. Xia, S.C. Wong, M. Zhang, C.W. Shu and W.H.K. Lam, An efficient discontinuous Galerkin method on triangular meshes for a pedestrian flow model, *Int. J. Numer. Meth. Eng.*, **76**:3 (2008), 337–350.

[29] T. Xiong, M.P. Zhang, Y.T. Zhang and C.W. Shu, Fast sweeping fifth order WENO scheme for static Hamilton-Jacobi equations with accurate boundary treatment, *J. Sci. Comput.*, **45**:1–3 (2010), 514–536.

[30] Y.H. Zahran and A.H. Abdalla, Seventh order Hermite WENO scheme for hyperbolic conservation laws, *Comput. Fluids*, **131** (2016), 66–80.

[31] Y.T. Zhang, S. Chen, F. Li, H.K. Zhao and C.W. Shu, Uniformly accurate discontinuous Galerkin fast sweeping methods for Eikonal equations, *SIAM J. Sci. Comput.*, **33**:4 (2011), 1873–1896.

[32] Y.T Zhang, H.K. Zhao and J. Qian, High Order fast sweeping methods for static Hamilton-Jacobi equations, *J. Sci. Comput.*, **29**:1 (2006), 25–56.

[33] H.K. Zhao, A fast sweeping method for Eikonal equations, *Math. Comput.*, **74**:250 (2005), 603–627.

[34] H.K. Zhao, S. Osher, B. Merriman and M. Kang, Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method, *Comput. Vis. Image Und.*, **80**:3 (2000), 295–314.

[35] Z. Zhao, Y. Chen and J. Qiu, A hybrid Hermite WENO method for hyperbolic conservation laws, *J. Comput. Phys.*, **405** (2020), 109175.

[36] F. Zheng, C.W. Shu and J. Qiu, Finite difference Hermite WENO schemes for the Hamilton-Jacobi equations, *J. Comput. Phys.*, **337** (2017), 27–41.

[37] J. Zhu and J. Qiu, A new fifth order finite difference WENO scheme for Hamilton-Jacobi equations, *Numer. Meth. Part. D. E.*, **33**:4 (2017), 1095–1113.

[38] J. Zhu and J. Qiu, A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws, *J. Comput. Phys.*, **318** (2016), 110–121.

[39] J. Zhu, F. Zheng and J. Qiu, New finite difference Hermite WENO schemes for Hamilton-Jacobi equations, *J. Sci. Comput.*, **83** (2020), 7.

[40] P. Zhu and S.Z. Zhou, Relaxation Lax-Friedrichs sweeping scheme for static Hamilton-Jacobi equations, *Numer. Algor.*, **54**:3 (2010), 325–342.