High Order Finite Difference Hermite WENO Fixed-Point Fast Sweeping Method for Static Hamilton-Jacobi Equations

Yupeng Ren[1], Yulong Xing[2,*] and Jianxian Qiu[3]

[1] *School of Mathematical Sciences, Xiamen University, Xiamen, Fujian 361005, P.R. China.*
[2] *Department of Mathematics, The Ohio State University, Columbus, OH 43210, USA.*
[3] *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.*

Abstract. In this paper, we combine the nonlinear HWENO reconstruction in [43] and the fixed-point iteration with Gauss-Seidel fast sweeping strategy, to solve the static Hamilton-Jacobi equations in a novel HWENO framework recently developed in [22]. The proposed HWENO frameworks enjoys several advantages. First, compared with the traditional HWENO framework, the proposed methods do not need to introduce additional auxiliary equations to update the derivatives of the unknown function ϕ. They are now computed from the current value of ϕ and the previous spatial derivatives of ϕ. This approach saves the computational storage and CPU time, which greatly improves the computational efficiency of the traditional HWENO scheme. In addition, compared with the traditional WENO method, reconstruction stencil of the HWENO methods becomes more compact, their boundary treatment is simpler, and the numerical errors are smaller on the same mesh. Second, the fixed-point fast sweeping method is used to update the numerical approximation. It is an explicit method and does not involve the inverse operation of nonlinear Hamiltonian, therefore any Hamilton-Jacobi equations with complex Hamiltonian can be solved easily. It also resolves some known issues, including that the iterative number is very sensitive to the parameter ε used in the nonlinear weights, as observed in previous studies. Finally, to further reduce the computational cost, a hybrid strategy is also presented. Extensive numerical experiments are performed on two-dimensional problems, which demonstrate the good performance of the proposed fixed-point fast sweeping HWENO methods.

*Corresponding author. *Email addresses:* `ypren@stu.xmu.edu.cn` (Y. Ren), `xing.205@osu.edu` (Y. Xing), `jxqiu@xmu.edu.cn` (J. Qiu)

1 Introduction

The static Hamilton-Jacobi (HJ) equations often appear in many application fields, for instance in optimal control, computer vision, differential game, geometric optics, image processing and so on [7,31]. The general static HJ equations have the form

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

where Ω is the computational domain in $\mathbb{R}^d$, $\phi(\mathbf{x})$ is the unknown function in Ω, 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 $\Gamma\subset\Omega$. Eikonal equation is a prototype example of the static HJ equations, taking the form

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

where $f(\mathbf{x})>0$. It can be derived from Maxwell's electromagnetic equations and provides a link between physical optics and geometric optics.

In general, the global C^1 solution does not exist for the time-dependent nonlinear HJ equations, even if the initial condition is sufficiently 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 [3].

There are mainly two types of numerical methods to solve the static HJ equations. The first one is to solve the following equation

$$\phi_t+H(\nabla\phi)=f(\mathbf{x}), \quad \mathbf{x}\in\mathbb{R}^d,$$

with pseudo-time iteration. The equation is evolved in time [25] until the numerical solution converges to a steady state. However, such method requires a very large number of iterations to obtain the convergence of the solution in the entire domain, with the main reason being the finite speed of propagation, and the restrictive CFL time step requirement for stability. The second popular algorithm is to treat the problem as a stationary boundary value problem, so that the fast marching method (FMM) [5,24,28] or the fast sweeping method (FSM) [9,17,27,37] can be applied. 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 proposed in [1] by Boué and Dupuis when solving a deterministic control problem with quadratic running cost using Markov chain approximation. Later, Zhao [37] studied the FSM for the Eikonal equation, and demonstrate the efficiency and effectiveness of the method. Since then, many high order FSM have been proposed to solving static HJ equations. In the framework

of finite difference methods, Zhang et al. [36] proposed the third order weighted essentially non-oscillatory (WENO) FSM scheme [8], and its fifth order extension was studied in [32]. High order accurate boundary treatments that are consistent with high order FSM, including Richardson extrapolation and Lax-Wendroff type procedure, have been developed for the inflow boundary conditions in [6, 32]. In the setting of finite element methods, some researchers have combined high order FSM with the finite element discontinuous Galerkin (DG) method to solve Eikonal equation in [12, 29, 34], and their numerical performance were shown to be effective and robust. For the time-dependent HJ equations, there have also been a lot of research work on studying high-order numerical methods, see [8, 11, 15, 25, 41] and the references therein.

Inspired by these successful methods, Zhang et al. [35] proposed to embed FSM techniques into a time-marching scheme to solve static HJ equations, by introducing the fixed-point iterative schemes. These methods take the explicit form and do not have to involve any inverse operation of the nonlinear Hamiltonian. Hence, they are very flexible and can be used in solving very general static HJ equations with complex Hamiltonian. The fixed-point fast sweeping method has also been extended to solve the steady state hyperbolic conservation laws in [2, 13, 30].

Recently, high order Hermite WENO (HWENO) methods [14, 20] have gained more attention in solving hyperbolic partial differential equations. Both the WENO and HWENO methods can achieve the high order accuracy and preserve the essentially non-oscillatory property, but the HWENO scheme uses the Hermite interpolation in reconstructing polynomials, that involves both the unknown variable ϕ and its first order spatial derivative (or first moment). Therefore, the HWENO reconstruction stencil becomes more compact and their boundary treatment is much simpler, although more storage and some additional work are needed to evaluate the spatial derivatives. The HWENO scheme was first proposed in the construction of a suitable limiter for the DG method [18, 19] due to its compact stencil, and was later used to solve the time-dependent HJ equation in [20]. Numerical results demonstrate that the HWENO scheme has smaller errors than the traditional WENO method with the same mesh and order. The HWENO scheme was also extended to solve the hyperbolic conservation law in the finite difference framework [14]. Since then, many HWENO schemes have been developed to solve hyperbolic conservation laws [26, 33, 38] and time-dependent HJ equations on structured [39] and triangular meshes [40, 42]. In addition, it was observed in [23] that the finite volume HWENO scheme enjoys the asymptotic preserving property, when applied to the steady-state discrete ordinates (S_N) transport equation. Recently, a new HWENO scheme (denoted by HWENO-ZZQ) was proposed by Zhu et al. in [43] for solving the time-dependent HJ equations. The main difference between the standard HWENO and HWENO-ZZQ methods lies in the stencil used in the reconstruction procedure. The HWENO-ZZQ reconstruction involves one big stencil and two small stencils, and their linear weights can be any positivity number summing up to 1. In addition to the simplicity, this method is shown in [43] to yield small errors with the same high order accuracy in the smooth areas, and maintain sharp transitions and non-oscillatory

property near discontinuity.

In this paper, built upon the high order HWENO-ZZQ schemes and different time discretization schemes, we design a class of fixed-point methods with Gauss-Seidel fast sweeping strategy for efficiently solving the static HJ equations. It is worth mentioning that we are no longer using the traditional HWENO framework, namely, using one equation to update ϕ and several additional auxiliary equations to update its derivatives. Instead, we use the new approach presented in our recent work [22], to use only one equation to update ϕ. The derivatives of ϕ will be obtained from applying the HWENO-ZZQ reconstruction on the updated values of ϕ and the previous values of the derivatives. This approach saves the computational storage and CPU costs, which improves the computational efficiency of the traditional HWENO scheme. We will present four different sweeping methods to discrete the temporal derivatives, which include the forward Euler (FE), Runge-Kutta (RK) time-marching methods, as well as these two methods combined with Gauss-Seidel fast sweeping technique. Extensive numerical examples suggest that the FE time discretization with Gauss-Seidel fast sweeping strategy is the most effective method for solving the static HJ equations. In addition, a hybrid strategy which combines both linear and HWENO-ZZQ reconstruction is also proposed and numerically validated to offer additional savings in computational time.

It was numerically observed in [22,32] that, in some test cases, the convergence of the WENO/HWENO FSM scheme is very sensitive to the parameter ε used in the nonlinear weights of HWENO reconstruction procedure. In order to address this issue, they adjust the parameter ε according to the mesh size, so that the scheme can converge quickly and give the expected high order. However, the manual adjustment of ε will impact the application of the proposed scheme as the best choice of ε cannot be known a priori and may be problem and mesh dependent. An interesting observation is that the proposed HWENO fixed-point fast sweeping methods resolve this issue, and all the numerical examples work well with a fixed parameter ε. In addition, previous work [21,22] suggested that the HWENO FSM does not converge to machine epsilon when the Godunov flux is used for some numerical examples, and requires extra work to address them. The numerical examples demonstrate that the HWENO fixed-point fast sweeping methods proposed in this paper work well for these tests.

The rest of the paper is organized as follows. In Section 2, we present HWENO fixed-point fast sweeping methods and the hybrid strategy. The numerical tests are presented in Section 3 to show the effectiveness and efficiency of our schemes. Conclusion remarks are provided in Section 4.

2   High order fixed-point fast sweeping methods

In this section, we present high order HWENO fixed-point fast sweeping methods to solve the static HJ equations. For ease of presentation, we consider the following two-

dimensional static HJ equation

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

Suppose the computational domain Ω is discretized into uniform 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 Ω_h. Δx and Δy stand for the grid sizes in the x and y directions, respectively. We denote the numerical solution at the grid point (x_i,y_j) by $\phi_{i,j}$. In addition, we introduce the notations $u=\phi_x(x,y)$ and $v=\phi_y(x,y)$ as the first order partial derivatives of ϕ with respect to the variables x and y, respectively.

2.1   Numerical Hamiltonian

The numerical approximation of the Hamiltonian in (2.1) is given by a monotone numerical Hamiltonian $\widehat{H}$:

$$H(\phi_x,\phi_y)\big|_{ij}\approx\widehat{H}(\phi_x^-,\phi_x^+,\phi_y^-,\phi_y^+)_{ij}. \tag{2.2}$$

Such numerical Hamiltonian takes inputs $\phi_x^{\pm}$ and $\phi_y^{\pm}$ at the corresponding grid point, which will be reconstructed from its neighboring point values using the high order reconstruction procedure as detailed in the next subsection.

Two types of numerical Hamiltonian are often considered for Hamilton-Jacobi equations. For general static HJ equations, we adopt the Lax-Friedrichs (LF) numerical Hamiltonian [4]

$$\widehat{H}_{i,j}^{LF}=H\left(\frac{u_{i,j}^-+u_{i,j}^+}{2},\frac{v_{i,j}^-+v_{i,j}^+}{2}\right)-\frac{1}{2}\alpha(u_{i,j}^+-u_{i,j}^-)-\frac{1}{2}\beta(v_{i,j}^+-v_{i,j}^-), \tag{2.3}$$

where

$$\alpha=\max_{u,v}|H_1(u,v)|, \quad \beta=\max_{u,v}|H_2(u,v)|, \tag{2.4}$$

and $H_\ell(u,v)(\ell=1,2)$ denotes the partial derivative of H with respect to the ℓ-th argument.

The other commonly used numerical Hamiltonian is the Godunov numerical Hamiltonian, often employed in the approximation of the Eikonal equation. As discussed in [15], the Godunov numerical Hamiltonian for general Hamiltonian $H(u,v)$ takes the form

$$\widehat{H}^G(u^-,u^+,v^-,v^+)=ext_{u\in I(u^-,u^+)}ext_{v\in I(v^-,v^+)}H(u,v), \tag{2.5}$$

where $I(a,b)=[\min(a,b),\max(a,b)]$ and the function ext is defined by

$$ext_{u\in I(a,b)}=\begin{cases} \min\limits_{a\le u\le b}, & \text{if } a\le b, \\ \max\limits_{b\le u\le a}, & \text{if } a\ge b. \end{cases}$$

For the two-dimensional Eikonal equation

$$
\begin{cases}
\sqrt{\phi_x^2 + \phi_y^2} = f(x,y), & (x,y) \in \Omega, \\
\phi(x,y) = g(x,y), & (x,y) \in \Gamma \subset \Omega,
\end{cases}
\tag{2.6}
$$

the Godunov numerical Hamiltonian (2.5) reduces to

$$
\widehat{H}^G(u^-,u^+,v^-,v^+) = \sqrt{\max\{(u^-)^+,(u^+)^-\}^2 + \max\{(v^-)^+,(v^+)^-\}^2}, \tag{2.7}
$$

where $x^+ = \max(x,0)$ and $x^- = -\min(x,0)$.

2.2   The review of HWENO-ZZQ reconstruction

In this subsection, we will review the finite difference HWENO-ZZQ reconstruction recently developed in [43]. 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 [43] for the details. In this paper, we focus on the fifth order finite difference HWENO-ZZQ reconstruction.

- Reconstruction of $(\phi_x)_{i,j}^-$ from the 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

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

where $u = \phi_x(x,y)$, and $u_{i,j}$ is the high order approximation of $u(x,y)$ at point (x_i, y_j). The values of their first-order derivative at $x = x_i$ can be evaluated as

$$
(\phi_x)_{i,j}^{-,1} = (p_1^-)'|_{x_i} = \frac{\phi_{i-2,j} + 18\phi_{i-1,j} - 9\phi_{i,j} - 10\phi_{i+1,j} + 9\Delta x u_{i-1,j} + 3\Delta x u_{i+1,j}}{-18\Delta x}; \tag{2.8a}
$$
$$
(\phi_x)_{i,j}^{-,2} = (p_2^-)'|_{x_i} = \frac{\phi_{i-2,j} - 4\phi_{i-1,j} + 3\phi_{i,j}}{2\Delta x}; \tag{2.8b}
$$
$$
(\phi_x)_{i,j}^{-,3} = (p_3^-)'|_{x_i} = \frac{-\phi_{i-1,j} + \phi_{i+1,j}}{2\Delta x}. \tag{2.8c}
$$

- Reconstruction of $(\phi_x)_{i,j}^+$ from the 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

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

where $u_{i,j}$ is the high order approximation of $u(x,y)$ at point (x_i,y_j). The values of their first-order derivative at $x=x_i$ can be evaluated as

$$(\phi_x)_{i,j}^{+,1} = (p_1^+)'|_{x_i} = \frac{10\phi_{i-1,j}+9\phi_{i,j}-18\phi_{i+1,j}-\phi_{i+2,j}+3\Delta x u_{i-1,j}+9\Delta x u_{i+1,j}}{-18\Delta x}; \quad (2.9a)$$

$$(\phi_x)_{i,j}^{+,2} = (p_2^+)'|_{x_i} = \frac{\phi_{i+1,j}-\phi_{i-1,j}}{2\Delta x}; \quad (2.9b)$$

$$(\phi_x)_{i,j}^{+,3} = (p_3^+)'|_{x_i} = \frac{-3\phi_{i,j}+4\phi_{i+1,j}-\phi_{i+2,j}}{2\Delta x}. \quad (2.9c)$$

In the nonlinear HWENO reconstructions, $(\phi_x)_{i,j}^{\pm}$ are computed as a convex combination of these three corresponding values [10, 43]

$$(\phi_x)_{i,j}^{\pm} = \omega_1^{\pm}\left(\frac{1}{\gamma_1}(\phi_x)_{i,j}^{\pm,1} - \frac{\gamma_2}{\gamma_1}(\phi_x)_{i,j}^{\pm,2} - \frac{\gamma_3}{\gamma_1}(\phi_x)_{i,j}^{\pm,3}\right) + \omega_2^{\pm}(\phi_x)_{i,j}^{\pm,2} + \omega_3^{\pm}(\phi_x)_{i,j}^{\pm,3}, \quad (2.10)$$

where the parameters ω_n ($n=1,2,3$) are called the nonlinear weights. The parameters γ_n could be any positive constants satisfying $\gamma_1+\gamma_2+\gamma_3=1$. The nonlinear weights ω_n can be computed from

$$\omega_n^{\pm} = \frac{\overline{\omega}_n^{\pm}}{\sum_{l=1}^3 \overline{\omega}_l^{\pm}}, \quad \overline{\omega}_n^{\pm} = \gamma_n\left(1+\frac{\tau^{\pm}}{\varepsilon+\beta_n^{\pm}}\right), \quad n=1,2,3, \quad (2.11)$$

where ε is a small positive number to avoid the denominator from becoming 0, and

$$\tau^{\pm} = \left(\frac{|\beta_1^{\pm}-\beta_2^{\pm}|+|\beta_1^{\pm}-\beta_3^{\pm}|}{2}\right)^2,$$

with $\beta_n^{\pm}$ being the so-called smoothness indicators

$$\beta_n^{\pm} = \sum_{\alpha=2}^{r}\int_{x_{i-\frac{1}{2}}}^{x_{i+\frac{1}{2}}}\Delta x^{2\alpha-3}\left(\frac{d^{\alpha}p_n^{\pm}(x)}{dx^{\alpha}}\right)^2 dx, \quad n=1,2,3,$$

measuring the smoothness of the derivative functions of $p_n^{\pm}(x)$ near the target point x_i. The parameter r is the degree of polynomial $p_n^{\pm}(x)$, and here we set $r=5$ for $n=1$, and $r=2$ for $n=2,3$, respectively.

2.3 HWENO fixed-point fast sweeping methods

After the spatial HWENO discretization, the model (2.1) reduces to a large nonlinear system of algebraic equations, and its size is determined by the number of spatial grid points. There are different approaches available to solve this nonlinear system. Here, we propose to use the fixed-point fast sweeping methods, and discuss four different sweeping methods below.

Forward Euler Jacobi method: We could view the static HJ equation as the steady state version of the time-dependent HJ equation. For the time-dependent problem, the simple FE method could be used for the temporal discretization, and the resulting explicit time marching scheme can be written as follows

$$\phi_{i,j}^{n+1} = \phi_{i,j}^n + \Delta t \left[f_{i,j} - \widehat{H}(\phi_x^-, \phi_x^+, \phi_y^-, \phi_y^+)_{ij}^n \right], \tag{2.12}$$

where the LF numerical Hamiltonian (2.3) or Godunov numerical Hamiltonian (2.7) is used, and the time step size is given by

$$\Delta t = \gamma \left(\frac{1}{\frac{\alpha}{\Delta x} + \frac{\beta}{\Delta y}} \right),$$

with γ being the CFL number, and α, β defined in (2.4). Obviously, $\alpha = \beta = 1$ for the Eikonal equation (2.6). Here $\phi_x^\pm, \phi_y^\pm$ are obtained through the high order HWENO reconstruction procedure based on the values of ϕ, u, v. We introduce the following operator

$$L\left(\{\phi^n\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^n; \{\phi^n\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^n \right) := f_{i,j} - \widehat{H}(\phi_x^-, \phi_x^+, \phi_y^-, \phi_y^+)_{ij}^n$$

to represent this dependence. From the perspective of iterative schemes, the scheme (2.12) can be considered as a forward Euler Jacobi (FE-Jacobi) type fixed-point iterative scheme

$$\phi_{i,j}^{new} = \phi_{i,j}^{old} + \frac{\gamma}{\frac{\alpha}{\Delta x} + \frac{\beta}{\Delta y}} L\left(\{\phi^{old}\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^{old}; \{\phi^{old}\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^{old} \right), \tag{2.13}$$

where $\phi_{i,j}^{new}$ denotes the updated numerical approximations of ϕ at the grid point (x_i, y_j), $\phi_{i,j}^{old}$ denotes the previous value of ϕ, and $u_{i,j}^{old}$, $v_{i,j}^{old}$ denote the previous values of u and v at the same grid point.

The values of ϕ, u and v are used in the HWENO-ZZQ reconstruction. In the traditional HWENO framework, one needs to take the spatial derivatives of (2.1), to obtain two new equations involving u and v, which will be used to update u^{new} and v^{new}. In our recent work [22], we presented a novel HWENO method for static HJ equations to simplify this procedure, which will be used in this paper. Compared with traditional HWENO framework, the new method does not involve any additional auxiliary equations, and is more computationally efficient. The main idea is to reuse the step of reconstructing $\phi_x^\pm$ and $\phi_y^\pm$ from ϕ, u and v, which was already available during the HWENO-ZZQ reconstruction procedure. More specifically, we use the updated ϕ^{new} (computed by (2.13) with suitable numerical Hamiltonian) and the previous spatial derivatives 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} as

$$u_{i,j}^{new} = \begin{cases} (\phi_x)_{i,j}^-, & \text{if } (\phi_x)_{i,j}^\pm > 0, \\ (\phi_x)_{i,j}^+, & \text{if } (\phi_x)_{i,j}^\pm < 0, \\ u_{i,j}^{old}, & \text{otherwise,} \end{cases} \qquad v_{i,j}^{new} = \begin{cases} (\phi_y)_{i,j}^-, & \text{if } (\phi_y)_{i,j}^\pm > 0, \\ (\phi_y)_{i,j}^+, & \text{if } (\phi_y)_{i,j}^\pm < 0, \\ v_{i,j}^{old}, & \text{otherwise.} \end{cases} \tag{2.14}$$

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. In the traditional HWENO framework, u^{new} and v^{new} are solved from two additional equations, which will cost extra storage requirement and CPU time. Here, we follow the Approach 1 in [22] to compute them. We would like to remark that this approach can only be applied to static problems, and is not applicable to general time-dependent problems. More discussion of this approach can be found in [22].

This finishes the description of the FE-Jacobi iterative scheme. The pseudo code of this method is presented in Algorithm 1.

Algorithm 1 FE-Jacobi scheme

---

   for $i = 1$ **to** N_x, $j = 1$ **to** N_y **do**

$$\phi_{i,j}^{new} = \phi_{i,j}^{old} + \gamma \left(\frac{1}{\frac{\alpha}{\Delta x} + \frac{\beta}{\Delta y}} \right) L \left(\{\phi^{old}\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^{old}; \{\phi^{old}\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^{old} \right),$$

   end for
   for $i = 1$ **to** N_x, $j = 1$ **to** N_y **do**
      Update $u_{i,j}^{new}$ and $v_{i,j}^{new}$ by (2.14) from $\phi_{i,j}^{new}$, $u_{i,j}^{old}$ and $v_{i,j}^{old}$.
   end for

Forward Euler fast sweeping method: While the simple FE-Jacobi method converges for most of the problems, it suffers from linear stability problems when coupled with high-order spatial discretization, and requires a small CFL number γ, hence a lot of iteration steps, to converge. In order to accelerate its convergence under the simple explicit framework of fixed-point iteration (2.13), the Gauss-Seidel (GS) sweeping strategy could be applied to scheme (2.13). According to the GS philosophy, the newest available numerical values of ϕ are always used in the interpolation stencils as long as they are available, and the GS iterative scheme takes the form

$$\phi_{i,j}^{new} = \phi_{i,j}^{old} + \frac{\gamma}{\frac{\alpha}{\Delta x} + \frac{\beta}{\Delta y}} L \left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^* \right), \tag{2.15}$$

where $\phi_{i,j}^*$ represents the most up-to-date point values of ϕ at the point (x_i, y_j). Here, we further propose to combine the GS iteration with the fast sweeping idea, and proceed the sweeping in the following four alternating directions repeatedly

$$\begin{aligned}
&(1) \quad i = 1 : N_x, \ j = 1 : N_y; \quad &(2) \quad i = N_x : 1, \ j = 1 : N_y; \\
&(3) \quad i = N_x : 1, \ j = N_y : 1; \quad &(4) \quad i = 1 : N_x, \ j = N_y : 1,
\end{aligned}$$

which leads to the FE type fixed-point fast sweeping method (FE-FSM). This method allows a larger CFL number and can reduce the number of iterations significantly. As observed in the FSM applied to solve the static HJ equation, this strategy utilized the directions of characteristic lines, and leads to an acceleration of the convergence speed significantly, as observed in our numerical experiment section. The pseudo codes for FE-FSM are presented in Algorithm 2.

Algorithm 2 FE-FSM

---

   for $i = 1$ **to** N_x, $j = 1$ **to** N_y **do**

$$\phi_{i,j}^{new} = \phi_{i,j}^{old} + \gamma \left(\frac{1}{\frac{\alpha}{\Delta x} + \frac{\beta}{\Delta y}} \right) L \left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^* \right),$$

   update $u_{i,j}^{new}$ and $v_{i,j}^{new}$ by (2.14) from $\phi_{i,j}^{new}$, $u_{i,j}^{old}$ and $v_{i,j}^{old}$.

   end for

Repeat the above process in the other three sweeping directions.

Runge-Kutta Jacobi method: We only considered the FE time discretization scheme so far. In the temporal discretization of differential equations, the high order RK methods are frequently used. Combining them with the Jacobi iteration, we have the third order RK Jacobi (RK-Jacobi) type fixed-point method of the form

$$\phi_{i,j}^{(1)} = \phi_{i,j}^{old} + \Delta t L \left(\{\phi^{old}\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^{old}; \{\phi^{old}\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^{old} \right), \tag{2.16a}$$

$$\phi_{i,j}^{(2)} = \frac{3}{4}\phi_{i,j}^{old} + \frac{1}{4}\phi_{i,j}^{(1)} + \frac{1}{4}\Delta t L \left(\{\phi^{(1)}\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^{(1)}; \{\phi^{(1)}\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^{(1)} \right), \tag{2.16b}$$

$$\phi_{i,j}^{new} = \frac{1}{3}\phi_{i,j}^{old} + \frac{2}{3}\phi_{i,j}^{(2)} + \frac{2}{3}\Delta t L \left(\{\phi^{(2)}\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^{(2)}; \{\phi^{(2)}\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^{(2)} \right). \tag{2.16c}$$

Runge-Kutta fast sweeping method: Similar to FE-FSM, by combining the RK fixed-point method and Gauss-Seidel fast sweeping strategy, we can obtain the third order RK type fixed-point fast sweeping method (RK-FSM), which takes the form

$$\phi_{i,j}^{(1)} = \phi_{i,j}^{old} + \Delta t L \left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^* \right), \tag{2.17a}$$

$$\phi_{i,j}^{(2)} = \phi_{i,j}^{(1)} + \frac{1}{4}\Delta t L \left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^* \right), \tag{2.17b}$$

$$\phi_{i,j}^{new} = \phi_{i,j}^{(2)} + \frac{2}{3}\Delta t L \left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^* \right). \tag{2.17c}$$

Note that this method contains three sub-iterations, and again, each iteration proceeds the sweeping in four alternating directions repeatedly. In the implementation, the sweeping directions of these three sub-iterations are the same in each iteration step. The pseudo codes of RK-Jacobi and RK-FSM are given in Algorithm 3 and Algorithm 4, respectively.

Algorithm 3 RK-Jacobi scheme

---

   for $i = 1$ **to** N_x, $j = 1$ **to** N_y **do**

$$\phi_{i,j}^{(1)} = \phi_{i,j}^{old} + \gamma \left(\frac{1}{\frac{\alpha}{\Delta x} + \frac{\beta}{\Delta y}} \right) L \left(\{\phi^{old}\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^{old}; \{\phi^{old}\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^{old} \right),$$

   end for

   for $i = 1$ **to** N_x, $j = 1$ **to** N_y **do**

   update $u_{i,j}^{(1)}$ and $v_{i,j}^{(1)}$ from $\phi_{i,j}^{(1)}$, $u_{i,j}^{old}$, $v_{i,j}^{old}$ by (2.14).

end for
for $i=1$ **to** N_x, $j=1$ **to** N_y **do**

$$\phi_{i,j}^{(2)} = \tfrac{3}{4}\phi_{i,j}^{old} + \tfrac{1}{4}\phi_{i,j}^{(1)} + \tfrac{1}{4}\gamma\left(\frac{1}{\frac{\alpha}{\Delta_x}+\frac{\beta}{\Delta_y}}\right) L\left(\{\phi^{(1)}\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^{(1)}; \{\phi^{(1)}\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^{(1)}\right),$$

end for
for $i=1$ **to** N_x, $j=1$ **to** N_y **do**
  update $u_{i,j}^{(2)}$ and $v_{i,j}^{(2)}$ from $\phi_{i,j}^{(2)}$, $u_{i,j}^{(1)}$, $v_{i,j}^{(1)}$ by (2.14).
end for
for $i=1$ **to** N_x, $j=1$ **to** N_y **do**

$$\phi_{i,j}^{new} = \tfrac{1}{3}\phi_{i,j}^{old} + \tfrac{2}{3}\phi_{i,j}^{(2)} + \tfrac{2}{3}\gamma\left(\frac{1}{\frac{\alpha}{\Delta_x}+\frac{\beta}{\Delta_y}}\right) L\left(\{\phi^{(2)}\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^{(2)}; \{\phi^{(2)}\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^{(2)}\right),$$

end for
for $i=1$ **to** N_x, $j=1$ **to** N_y **do**
  update $u_{i,j}^{new}$ and $v_{i,j}^{new}$ from $\phi_{i,j}^{new}$, $u_{i,j}^{(2)}$, $v_{i,j}^{(2)}$ by (2.14).
end for

Algorithm 4 RK-FSM

---

for $i=1$ **to** N_x, $j=1$ **to** N_y **do**

$$\phi_{i,j}^{(1)} = \phi_{i,j}^{old} + \gamma\left(\frac{1}{\frac{\alpha}{\Delta_x}+\frac{\beta}{\Delta_y}}\right) L\left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^*\right),$$

  update $u_{i,j}^{(1)}$ and $v_{i,j}^{(1)}$ from $\phi_{i,j}^{(1)}$, $u_{i,j}^{old}$, $v_{i,j}^{old}$ by (2.14).
end for
for $i=1$ **to** N_x, $j=1$ **to** N_y **do**

$$\phi_{i,j}^{(2)} = \phi_{i,j}^{(1)} + \tfrac{1}{4}\gamma\left(\frac{1}{\frac{\alpha}{\Delta_x}+\frac{\beta}{\Delta_y}}\right) L\left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^*\right),$$

  update $u_{i,j}^{(2)}$ and $v_{i,j}^{(2)}$ from $\phi_{i,j}^{(2)}$, $u_{i,j}^{(1)}$, $v_{i,j}^{(1)}$ by (2.14).
end for
for $i=1$ **to** N_x, $j=1$ **to** N_y **do**

$$\phi_{i,j}^{new} = \phi_{i,j}^{(2)} + \tfrac{2}{3}\gamma\left(\frac{1}{\frac{\alpha}{\Delta_x}+\frac{\beta}{\Delta_y}}\right) L\left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^*\right),$$

  update $u_{i,j}^{new}$ and $v_{i,j}^{new}$ from $\phi_{i,j}^{new}$, $u_{i,j}^{(2)}$, $v_{i,j}^{(2)}$ by (2.14).
end for
Repeat the above process in the other three sweeping directions.

Remark 2.1. There are several different forms of the RK-FSM in addition to (2.17), since we could mix the usage of ϕ^{old}, $\phi^{(1)}$ and $\phi^{(2)}$ for this steady state problem. Numerically, we also tested other RK type fixed-point fast sweeping methods, including the following

RK3 fixed-point fast sweeping scheme

$$\phi_{i,j}^{(1)} = \phi_{i,j}^{old} + \Delta t L\left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^*\right),\tag{2.18a}$$

$$\phi_{i,j}^{(2)} = \frac{3}{4}\phi_{i,j}^{old} + \frac{1}{4}\phi_{i,j}^{(1)} + \frac{1}{4}\Delta t L\left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^*\right),\tag{2.18b}$$

$$\phi_{i,j}^{new} = \frac{1}{3}\phi_{i,j}^{old} + \frac{2}{3}\phi_{i,j}^{(2)} + \frac{2}{3}\Delta t L\left(\{\phi^*\}_{i-2,j}^{i+2,j}, u_{i\pm1,j}^*; \{\phi^*\}_{i,j-2}^{i,j+2}, v_{i,j\pm1}^*\right).\tag{2.18c}$$

The numerical experiments suggest that the scheme in (2.17) yields the fastest convergence among all these RK type fixed-point fast sweeping schemes.

2.4 The flowchart of four HWENO fixed-point sweeping schemes

We have proposed four HWENO fixed-point sweeping approaches to solve the static HJ equations. Here we will summarize the detailed procedure of these approaches, and provide a flowchart for them. We start by labeling the computational nodal points $\{(x_i, y_j)\}$ into several categories as in [21]:

Category I: For points on the boundary Γ, the exact values are assigned for these points.

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

Category III: For points near the Γ (whose distances to Γ are less than or equal to $2h$), the numerical boundary treatment from [6, 32] could be used (i.e., Richardson extrapolation for a single point or a set of isolated points, and Lax-Wendroff type procedure for continuous Γ). Since our main focus is on HWENO fixed-point sweeping method itself in this paper, the exact solutions are used on these points.

Category IV: All the remaining points, which will be updated by fixed-point sweeping methods.

Note that only *Category IV* points need to be updated by following numerical method. We now summarize our flowchart of these methods as follows:

Step 1. *Initialization*: The numerical solution from the first order fast sweeping method [37] is taken as the initial guess of ϕ. The forward or backward difference of the resulting ϕ is used as the initial guess of u and v.

Step 2. *Update ϕ^{new}*: We can choose one of these four schemes (2.13), (2.15), (2.16) and (2.17) to update $\phi_{i,j}^{new}$ at the grid point (x_i, y_j). The pseudo codes of these schemes have been given in Section 2.3. The values at ghost points will be updated by the third order extrapolations in the iterative methods.

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

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

2.5   Comments and remarks

We need to emphasize that the local solver of fixed-point sweeping methods derived from different time marching schemes with either Jacobi or Gauss-Seidel fast sweeping strategy, are given in the explicit form and do not involve solving nonlinear equations. In principle, the approaches can be applied to any general static HJ equations with complicated Hamiltonian. If the Godunov Hamiltonian is used, the numerical method is different from the fast sweeping methods discussed in [9, 21, 22, 36, 37], where the methods are implicit and need to solve nonlinear equations. However, it can be shown that the FE-FSM with LF Hamiltonian are equivalent to the fast sweeping method with LF Hamiltonian in [21, 22], if the same high order reconstruction is used.

Remark 2.2. In Remark 2.1 of the recent work [22] to study HWENO FSM, we observed that such method does not converge to machine epsilon when the Godunov flux is used for some examples. To fix this issue, it was proposed there to update the solution by

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

which may increase the number of iteration steps. The numerical examples suggest that there is no need to introduce any other parameters for HWENO fixed-point sweeping methods proposed in this paper.

Remark 2.3. It was numerically observed in [22,32] that, in some test cases (e.g. Examples 3.6 and 3.7 in Section 3), the convergence of the HWENO FSM scheme is very sensitive to the parameter ε used in the nonlinear weights of HWENO reconstruction procedure. In order to address this issue, the parameter ε was adjusted according to the mesh size, so that the scheme can converge quickly and provide the expected high order. However, the manual adjustment of ε will impact the application of the proposed scheme as the best choice of ε cannot be known a priori and may be problem and mesh dependent. An interesting observation is that the proposed HWENO fixed-point sweeping method resolves this issue, and all the numerical examples work well with a fixed parameter ε.

2.6   Hybrid strategy

In the HWENO reconstruction procedure, compared with simple linear reconstruction, the evaluation of the smoothness indicators occupies most of the additional computational cost. Here we explore a hybrid strategy to combine the linear and HWENO method, which is similar to the hybrid fast sweeping WENO method studied in [21, 22]. In this paper, we directly follow the approach in [22], and apply the fifth order linear reconstruction if the numerical solution is monotonic on the big stencil S_0 or $\widetilde{S}_0$, and apply the HWENO-ZZQ reconstruction in other cases.

In order to describe the fixed-point fast sweeping method with hybrid strategy, we separate the points $\{(x_i, y_j)\}$ in *Category IV* (defined in Section 2.4) into the following two subcategories, which will be handled slightly different in the hybrid method.

Category IV.1: For points whose distances to *Category III* are less than or equal to $2h$ (excluding those in *Category I*).

Category IV.2: All the remaining points in Category IV.

The specific flowchart of fixed-point sweeping method with hybrid strategy is similar to that in Section 2.4, except the **Step 2**. During the **Step 2**, for the points in *Category IV.1*: the nonlinear HWENO-ZZQ reconstruction (2.10) is applied to evaluate $(\phi_x)_{i,j}^{\pm}$. For the points in *Category IV.2*, the hybrid strategy is applied, by using either the linear or nonlinear reconstruction based on the following criteria:

$$(\phi_x)_{i,j}^{\pm} = \begin{cases} \text{(2.8a) or (2.9a)}, & \text{if } \{u_{i,j}\} \text{ have the same sign on } S_0 \text{ or } \widetilde{S}_0, \\ \text{(2.10)}, & \text{otherwise.} \end{cases} \tag{2.19}$$

The similar procedure for $(\phi_y)_{i,j}^{\pm}$ is used. The rest of the algorithm is the same as that in Section 2.4.

3   Numerical results

In this section, we will present extensive numerical examples by testing the proposed fifth order finite difference HWENO fixed-point sweeping methods on the Eikonal equation and general static HJ equations in two dimensions. We will compare the numerical results of four fixed-point sweeping methods with/without the hybrid strategy, and list their errors, convergence rates and the numbers of iterations. In all the numerical examples, ε in (2.11) is taken as 10^{-6}, and linear weights are taken as $\gamma_1 = 0.98$ and $\gamma_2 = \gamma_3 = 0.01$. The total number of grid points is assumed to be $N_x = N_y = N$, unless otherwise stated. We use "iter" to denote the number of iterations in all the tables. Note that one iteration means that all point values are updated once. Therefore, one GS fast sweeping iteration involves the sweeping in four alternating directions and would count as four iterations. All the computations are implemented by using MATLAB 2020a on ThinkPad computer with 1.80 GHz Intel Core i7 processor and 16GB RAM.

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

$$\phi(x,y) = \cos\left(\pi + \frac{\pi}{2}x\right) + \cos\left(\pi + \frac{\pi}{2}y\right).$$
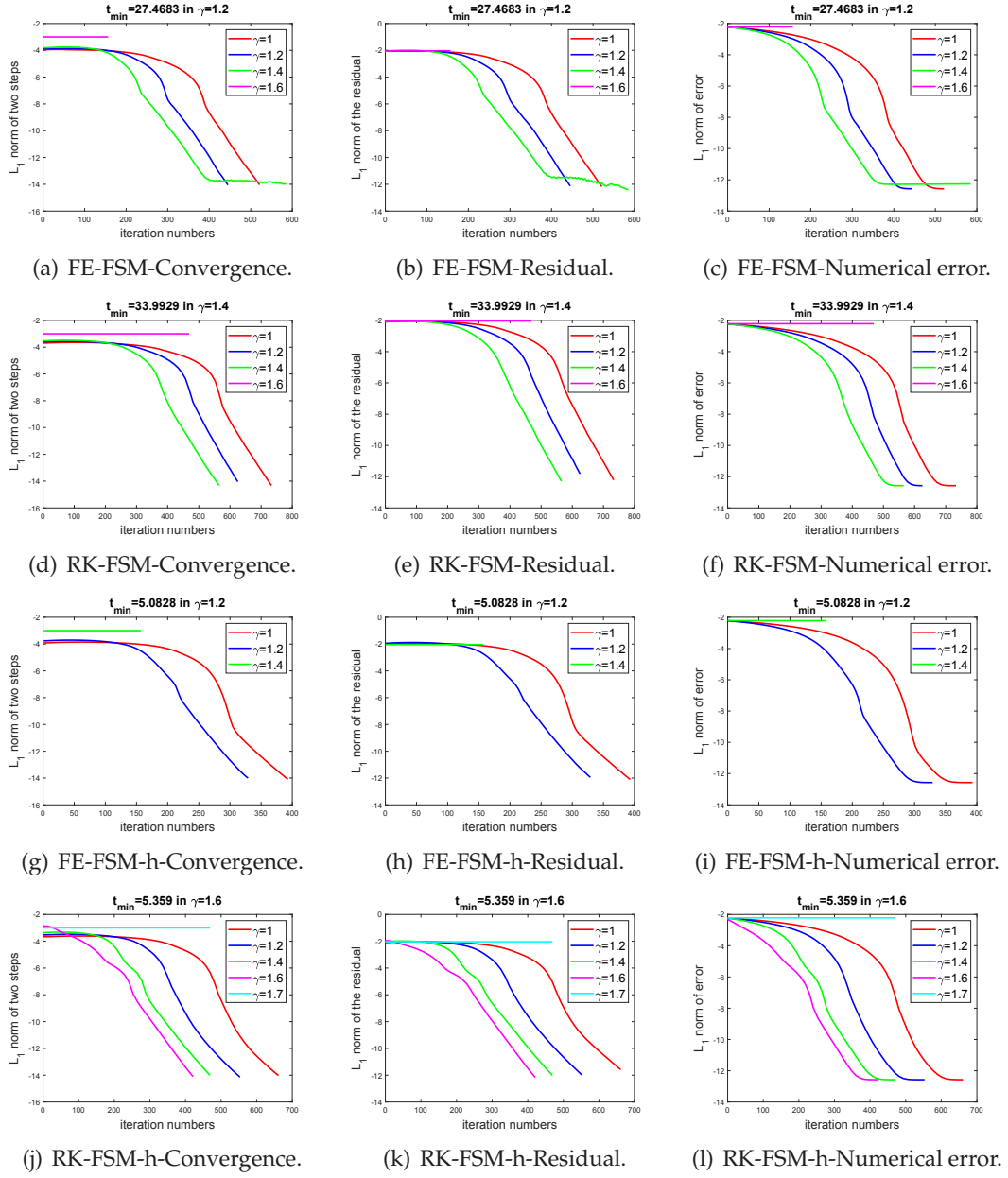
The Godunov numerical Hamiltonian (2.7) is used in this example. Table 1 lists the numerical results of four fixed-point sweeping methods, including the numerical errors,

Table 1: Example 3.1. Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence

	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	3.65e-06	-	2.97e-05	-	1690	1.8799	3.62e-06	-	2.93e-05	-	1690	0.73858
80	5.75e-08	5.99	1.07e-06	4.78	2113	9.4906	5.74e-08	5.97	1.07e-06	4.76	2113	2.5132
160	2.91e-10	7.62	1.16e-08	6.52	3281	58.4108	-	-	-	-	-	-
320	-	-	-	-	-	-	-	-	-	-	-	-
	FE-FSM $\gamma=1$						FE-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	3.65e-06	-	2.97e-05	-	240	0.29083	3.62e-06	-	2.93e-05	-	240	0.12409
80	5.75e-08	5.99	1.07e-06	4.78	272	1.0191	5.74e-08	5.97	1.07e-06	4.76	272	0.35377
160	2.91e-10	7.62	1.16e-08	6.53	348	5.2703	2.91e-10	7.62	1.16e-08	6.53	332	1.1835
320	2.66e-13	10.09	9.62e-12	10.23	524	32.0309	2.65e-13	10.09	1.03e-11	10.12	396	6.8875
	RK-Jacobi $\gamma=1$						RK-Jacobi $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	3.65e-06	-	2.97e-05	-	399	0.68148	3.62e-06	-	2.93e-05	-	399	0.19815
80	5.75e-08	5.99	1.07e-06	4.78	471	1.9727	5.74e-08	5.97	1.07e-06	4.76	471	0.57089
160	2.91e-10	7.62	1.15e-08	6.54	780	12.9281	2.91e-10	7.62	1.15e-08	6.54	780	3.5232
320	3.04e-13	9.90	8.34e-11	7.11	1413	102.4285	3.04e-13	9.90	8.34e-11	7.11	1413	27.8141
	RK-FSM $\gamma=1$						RK-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	3.65e-06	-	2.97e-05	-	360	0.61658	3.62e-06	-	2.93e-05	-	360	0.35017
80	5.75e-08	5.99	1.07e-06	4.78	420	1.7162	5.74e-08	5.97	1.07e-06	4.76	420	0.73419
160	2.91e-10	7.62	1.16e-08	6.53	516	8.4034	2.91e-10	7.62	1.16e-08	6.53	516	2.1677
320	2.65e-13	10.09	9.33e-12	10.28	744	46.2991	2.65e-13	10.09	1.10e-11	10.03	672	11.431

convergence order, number of iterations and CPU time. The numerical results without hybrid strategy are reported on the left side of Table 1, and those with the hybrid strategy are reported on the right side. The "-" in the error column of the table represents the "not convergent" case, namely, with the iteration numbers increasing, the error $\delta=||\phi^{new}-\phi^{old}||_{L_1}$ either stays at the level of approximately 10^{-10} and fails to reach 10^{-14}, or increases until $\gg 1$.

We can observe that the FE-Jacobi method requires the smaller CFL number of value 0.1, but still fails to converge on the refined mesh. The reason is the that FE time discretization coupled with high-order linear spatial discretization suffers from linear stability problems. However, when the third order RK time discretization is used, the RK-Jacobi scheme can take a larger CFL number of value 1. Also, the number of iterations of RK-Jacobi method is smaller than FE-Jacobi method on the same mesh. The fast sweeping technique (in the FE-FSM and RK-FSM) can improve the convergence of the Jacobi scheme, as observed in Table 1. In addition, when the fast sweeping technique is used, the FE-FSM can now use a larger CFL number than FE-Jacobi scheme. On the same re-

(a) FE-FSM-Convergence.   (b) FE-FSM-Residual.   (c) FE-FSM-Numerical error.

(d) RK-FSM-Convergence.   (e) RK-FSM-Residual.   (f) RK-FSM-Numerical error.

(g) FE-FSM-h-Convergence.   (h) FE-FSM-h-Residual.   (i) FE-FSM-h-Numerical error.

(j) RK-FSM-h-Convergence.   (k) RK-FSM-h-Residual.   (l) RK-FSM-h-Numerical error.

Figure 1: Example 3.1. Study on different γ's.

fined mesh, we observe that the RK-FSM only takes about 50% CPU time of the RK-Jacobi scheme. Furthermore, the FE-FSM costs even less CPU time than RK-FSM. The numerical results after the hybrid strategy is applied can be seen on the right side of Table 1, which suggests that the hybrid strategy can save 70%−80% of the CPU time on the re-
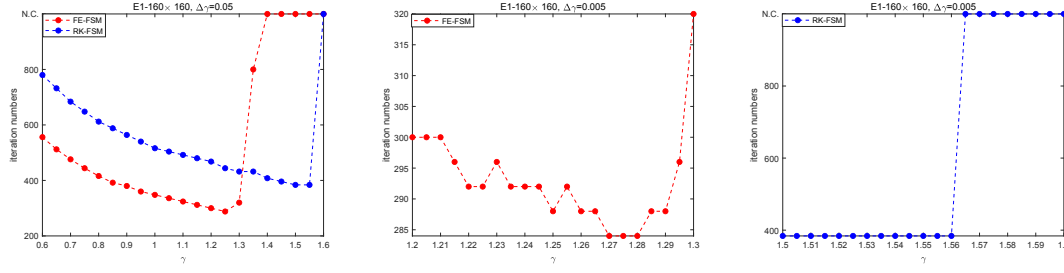
Figure 2: Example 3.1, the iteration numbers vs γ of FS-FSM and RK-FSM on mesh $N=160$. "N.C." represents the "not convergent" case: with the iteration numbers increasing, the error $\delta=||\phi^{new}-\phi^{old}||_{L_1}$ either stays at the level of approximately 10^{-10} and fails to reach 10^{-14}, or increases until $\gg 1$. Left: two schemes, $\gamma\in[0.6,1.6]$ with $\Delta\gamma=0.05$; Middle: FE-FSM in the zoom-in region of $\gamma\in[1.2,1.3]$ with $\Delta\gamma=0.005$; Right: RK-FSM in the zoom-in region of $\gamma\in[1.5,1.6]$ with $\Delta\gamma=0.005$.

fined mesh. In summary, the FE-FSM performs the best out of these four methods, and this observation is consistent with that in [30, 35]. The hybrid strategy can further reduce the computational cost.

Finally, we want to remark that, for the three schemes other than the FE-Jacobi scheme, the CFL number can be taken to be greater than 1. Fig. 1 shows that the convergence history of the FE-FSM and RK-FSM with larger CFL numbers on mesh $N=320$, reporting the errors between two consecutive iteration steps, the residual and numerical errors. Here, "FE-FSM-h" represents FE-FSM with hybrid strategy, and similarly for "RK-FSM-h". We have listed the optimal CFL number and CPU time of each scheme in the title of each sub-figure. For example, "$t_{min}=24.7683$ (unit: second) in $\gamma=1.2$" on first line of Fig. 1 represents the optimal CFL number for FE-FSM is 1.2, because it requires the least CPU time $t_{min}=24.7683$ among all the choices of γ that lead to convergent results. We also observe that the RK-FSM tends to admit a larger CFL number than the FE-FSM, however the RK-FSM still costs more CPU time than the FE-FSM.

In order to better observe the optimal parameter γ of FE-FSM and RK-FSM, we fix the mesh size $N=160$ and plot the iteration numbers when taking different γ in Fig. 2. The term "N.C." refers to the "not convergent" case, as defined above. From the figure, we can see that the optimal "γ" under the two schemes are $\gamma=1.28$ for the FE-FSM scheme, and $\gamma=1.56$ for the RK-FSM scheme, respectively. Under their respective optimal γ, the FE-FSM scheme still converges faster than the RK-FSM.

We have also tried with the case when the mesh sizes in x and y directions are not equal, i.e. $\Delta x\neq\Delta y$. The numerical results are shown in Table 2, and the similar conclusion can be concluded.

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 Γ is the circle with center at $(0,0)$ and radius 0.5, that is

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

Table 2: Example 3.1. Comparison of the four methods with $\Delta x \neq \Delta y$: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

	FE-Jacobi $\gamma = 0.1$						FE-Jacobi $\gamma = 0.1$ with hybrid strategy					
$N_x \times N_y$	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40×32	1.04e-05	-	6.24e-05	-	1712	1.6030	1.02e-05	-	6.08e-05	-	1711	0.72914
80×64	1.62e-07	6.01	2.24e-06	4.80	2053	6.5593	1.62e-07	5.98	2.23e-06	4.76	2055	1.9648
160×128	1.37e-09	6.88	4.45e-08	5.65	2999	40.1175	-	-	-	-	-	-
320×256	-	-	-	-	-	-	-	-	-	-	-	-
	FE-FSM $\gamma = 1$						FE-FSM $\gamma = 1$ with hybrid strategy					
$N_x \times N_y$	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40×32	1.04e-05	-	6.24e-05	-	260	0.43989	1.02e-05	-	6.08e-05	-	256	0.13353
80×64	1.62e-07	6.01	2.24e-06	4.80	288	1.3961	1.62e-07	5.98	2.23e-06	4.76	288	0.29079
160×128	1.37e-09	6.88	4.44e-08	5.65	344	6.0251	1.37e-09	6.88	4.44e-08	5.65	344	0.99301
320×256	2.17e-12	9.29	1.29e-10	8.42	484	35.521	2.17e-12	9.29	1.27e-10	8.44	420	5.4661
	RK-Jacobi $\gamma = 1$						RK-Jacobi $\gamma = 1$ with hybrid strategy					
$N_x \times N_y$	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40×32	1.04e-05	-	6.24e-05	-	429	0.52569	1.02e-05	-	6.08e-05	-	429	0.34221
80×64	1.62e-07	6.01	2.24e-06	4.80	492	1.7184	1.62e-07	5.98	2.23e-06	4.76	492	0.80352
160×128	1.37e-09	6.88	4.44e-08	5.65	720	9.1995	1.37e-09	6.88	4.44e-08	5.65	720	2.1971
320×256	2.19e-12	9.28	1.21e-10	8.51	1293	74.9266	2.19e-12	9.28	1.21e-10	8.51	1293	16.29
	RK-FSM $\gamma = 1$						RK-FSM $\gamma = 1$ with hybrid strategy					
$N_x \times N_y$	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40×32	1.04e-05	-	6.24e-05	-	372	0.44122	1.02e-05	-	6.08e-05	-	372	0.18977
80×64	1.62e-07	6.01	2.24e-06	4.80	408	1.4815	1.62e-07	5.98	2.23e-06	4.76	408	0.38255
160×128	1.37e-09	6.88	4.44e-08	5.65	504	7.7328	1.37e-09	6.88	4.44e-08	5.65	504	1.6295
320×256	2.17e-12	9.29	1.27e-10	8.44	696	49.7756	2.17e-12	9.29	1.27e-10	8.44	672	9.0935

The boundary condition is given as $\phi(x,y) = 0$ on Γ. The exact solution is a distance function to the circle Γ, 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 surface and contour of numerical solution by FE-FSM are shown in Fig. 3.

The numerical results without the hybrid strategy are reported on the left side of Table 3, and those with the hybrid strategy are reported on the right side. Again, the FE-Jacobi iteration requires a smaller CFL number of 0.1 due to the linear instability, and fails to converge on the refined mesh. For the other three methods, a larger CFL number can be taken. Again, we can observe that the fast sweeping technique can improve the convergence of the Jacobi scheme. On the same refined mesh, it can be seen that the RK-FSM only takes about 50% CPU time of the RK-Jacobi scheme. Furthermore, the FE-FSM costs even less CPU time than RK-FSM. The numerical results obtained with the hybrid strategy can be seen on the right side of Table 3, which indicates that the hybrid strategy

Table 3: Example 3.2. Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	6.09e-07	-	2.33e-05	-	1121	0.9706	6.09e-07	-	2.33e-05	-	1106	0.8406
80	1.16e-08	5.70	1.74e-06	3.74	1553	5.3702	1.16e-08	5.70	1.74e-06	3.74	1541	3.1265
160	8.70e-11	7.06	2.20e-08	6.30	2260	32.6669	1.04e-10	6.80	3.21e-08	5.76	2207	15.6215
320	-	-	-	-	-	-	-	-	-	-	-	-
	FE-FSM $\gamma=1$						FE-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	6.75e-07	-	5.41e-05	-	160	0.1337	6.04e-07	-	2.33e-05	-	280	0.2301
80	1.16e-08	5.85	1.74e-06	4.95	208	0.6374	1.15e-08	5.70	1.74e-06	3.74	280	0.46745
160	8.76e-11	7.05	2.21e-08	6.30	244	3.1495	9.98e-11	6.85	4.55e-08	5.26	248	1.2264
320	1.78e-12	5.61	1.30e-10	7.40	324	19.7232	1.78e-12	5.80	1.30e-10	8.44	336	5.6339
	RK-Jacobi $\gamma=1$						RK-Jacobi $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	9.00e-07	-	6.26e-05	-	282	0.3819	8.95e-07	-	6.25e-05	-	282	0.4601
80	1.16e-08	6.26	1.74e-06	5.16	363	1.2983	1.15e-08	6.27	1.74e-06	5.16	369	1.0393
160	8.76e-11	7.05	2.21e-08	6.30	513	7.0839	8.77e-11	7.03	2.21e-08	6.30	516	3.4253
320	1.80e-12	5.60	1.30e-10	7.40	903	54.8592	1.80e-12	5.60	1.30e-10	7.40	903	26.9378
	RK-FSM $\gamma=1$						RK-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	5.97e-07	-	2.33e-05	-	252	0.2615	5.96e-07	-	2.33e-05	-	252	0.1610
80	1.41e-08	5.40	4.14e-06	2.49	312	0.9659	1.17e-08	5.66	1.74e-06	3.74	312	0.5905
160	8.72e-11	7.33	2.20e-08	7.55	372	4.8890	9.28e-11	6.98	3.42e-08	5.67	372	1.8319
320	1.78e-12	5.61	1.30e-10	7.39	480	27.7620	1.78e-12	5.70	1.30e-10	8.03	480	7.6022

Figure 3: Example 3.2. The numerical solution by FE-FSM on mesh $N=80$. Left: the 3D plot of numerical solution ϕ; Right: the contour plot for ϕ.

can save about 50%−75% CPU time on the refined mesh.

Again, we want to remark that, for the three schemes other than the FE-Jacobi scheme, the CFL number can be taken to be greater than 1. Fig. 4 shows that the convergence

(a) FE-FSM-Convergence. (b) FE-FSM-Residual. (c) FE-FSM-Numerical error.

(d) RK-FSM-Convergence. (e) RK-FSM-Residual. (f) RK-FSM-Numerical error.

(g) FE-FSM-h-Convergence. (h) FE-FSM-h-Residual. (i) FE-FSM-h-Numerical error.

(j) RK-FSM-h-Convergence. (k) RK-FSM-h-Residual. (l) RK-FSM-h-Numerical error.

Figure 4: Example 3.2. Study on different γ's.

history of the FE-FSM and RK-FSM with larger CFL numbers on mesh $N=320$, reporting the errors between two consecutive iteration steps, the residual and numerical errors. Similar to Example 3.1, we have presented the optimal CFL number and CPU time of each scheme in the title of each sub-figure. With larger CFL number, fewer iterations

are needed for convergence. We observe that the RK-FSM tends to admit a larger CFL number than the FE-FSM, but it still under performs in terms of the computational time.

In summary, the FE-FSM demonstrates to yield the best performance out of these four methods, which is consistent with the observation from Example 3.1. Also, the hybrid strategy can further reduce the computational cost.

Example 3.3. In this test case, we solve the Eikonal equation with $f(x,y)=1$. The computational domain is set as $[-3,3]^2$, and the inflow boundary Γ consists of two circles of equal radius 0.5 with the centers located at $(-1,0)$ and $(\sqrt{1.5},0)$, respectively, which leads to

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

The exact solution is a distance function to the inflow boundary Γ, 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 Γ and the singular line.

Fig. 5 shows that the numerical solution by FE-FSM. The numerical results with hybrid strategy and without hybrid strategy are reported on the Table 4. Again, We can observe that the FE-Jacobi method requires the smaller CFL number of value 0.1, but still fails to converge on the refined mesh. The other three schemes can take a larger CFL number and the fast sweeping technique can improve the convergence of the Jacobi scheme. On the same refined mesh, we can see that the RK-FSM only takes about 50% CPU time of the RK-Jacobi scheme. Furthermore, the FE-FSM costs even less CPU time than RK-FSM. When the hybrid strategy is used, all four schemes can save about 75% CPU time on the refined mesh.

In addition, we want to remark that, for the three schemes other than the FE-Jacobi scheme, the CFL number can be taken to be greater than 1. With larger CFL number,
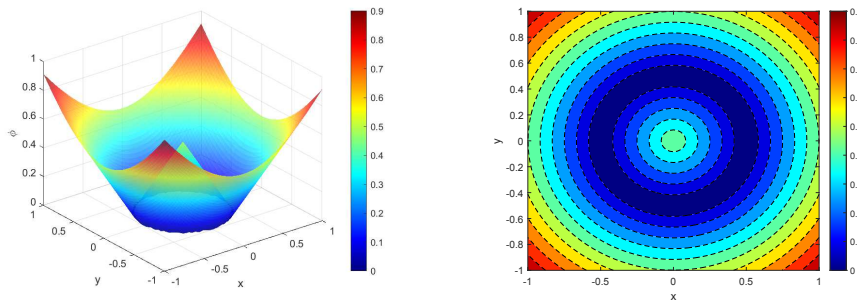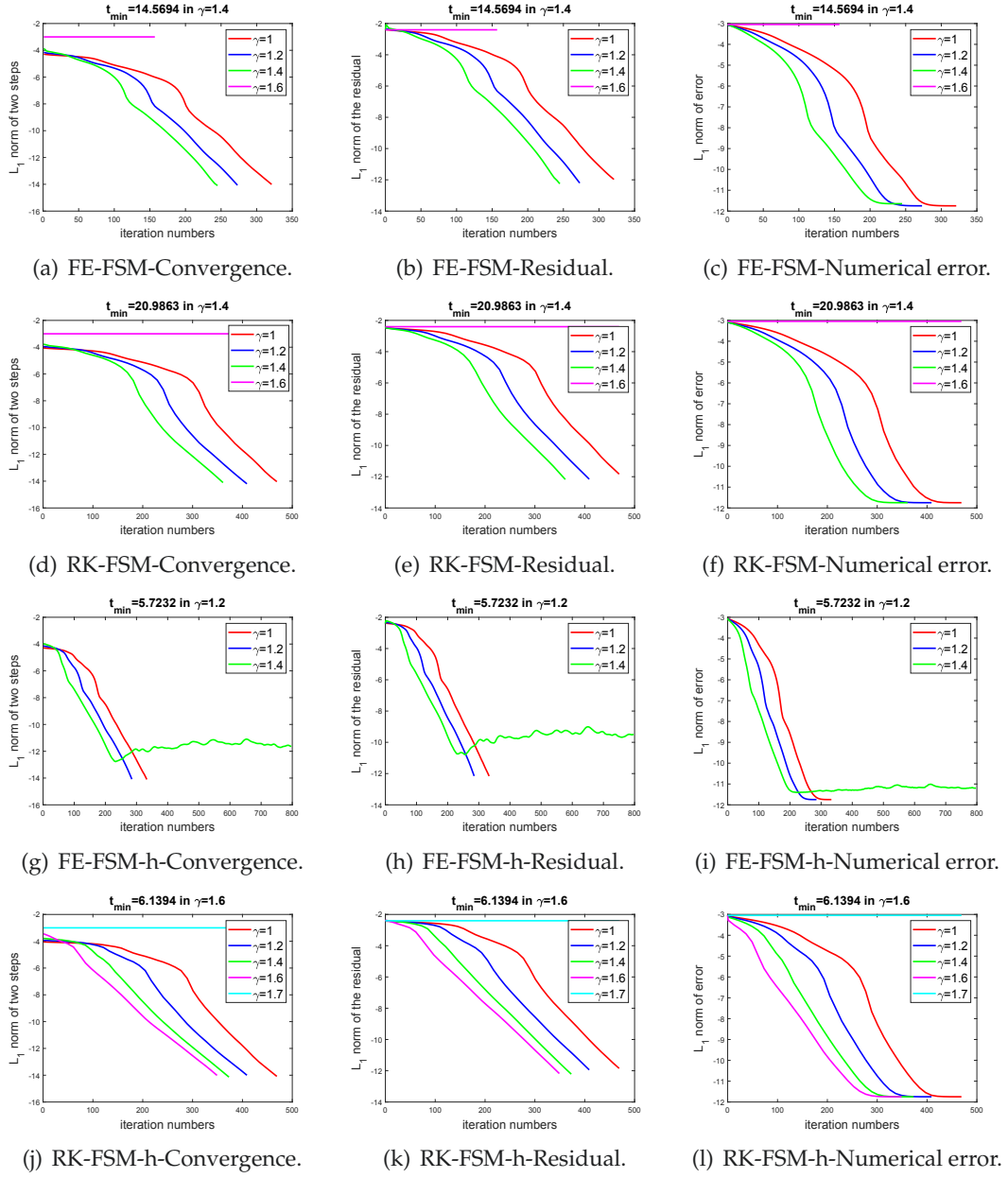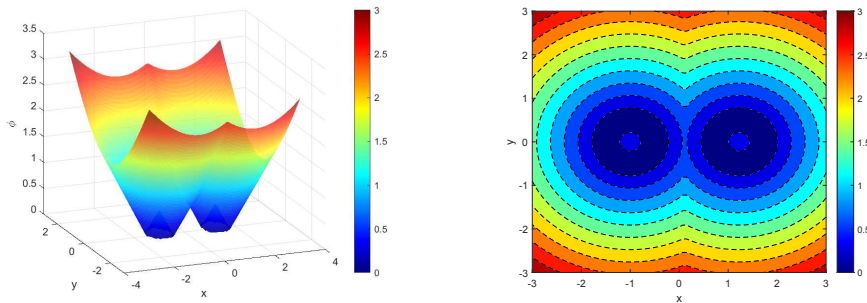

Figure 5: Example 3.3. The numerical solution by FE-FSM on mesh $N=80$. Left: the 3D plot of numerical solution ϕ; Right: the contour plot for ϕ.

Table 4: Example 3.3. Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
80	7.09e-07	-	1.17e-04	-	1884	7.6586	7.09e-07	-	1.17e-04	-	1867	3.3288
160	1.10e-07	2.68	9.25e-06	3.67	2859	49.5274	1.94e-07	1.86	1.81e-05	2.70	2852	14.9261
320	-	-	-	-	-	-	-	-	-	-	-	-
	FE-FSM $\gamma=1$						FE-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
80	7.02e-07	-	1.17e-04	-	236	0.88271	7.03e-07	-	1.17e-04	-	236	0.34559
160	1.10e-07	2.66	9.71e-06	3.59	292	4.599	2.01e-07	1.80	1.81e-05	2.69	288	1.3335
320	2.98e-10	8.53	3.86e-07	4.65	408	26.4953	3.18e-10	9.30	4.03e-07	5.48	368	6.1812
	RK-Jacobi $\gamma=1$						RK-Jacobi $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
80	7.09e-07	-	1.18e-04	-	426	1.6574	7.09e-07	-	1.18e-04	-	423	0.65345
160	1.19e-07	2.57	1.75e-05	2.74	681	11.6017	2.05e-07	1.78	1.81e-05	2.70	678	3.5579
320	3.54e-10	8.39	3.74e-07	5.55	1197	83.9635	3.86e-10	9.05	4.64e-07	5.28	1194	24.711
	RK-FSM $\gamma=1$						RK-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
80	7.10e-07	-	1.01e-04	-	360	1.4243	7.11e-07	-	1.01e-04	-	360	0.53576
160	1.09e-07	2.70	9.53e-06	3.41	480	9.2416	1.94e-07	1.87	1.81e-05	2.48	468	2.4039
320	3.01e-10	8.50	3.85e-07	4.63	648	41.7686	3.16e-10	9.26	3.84e-07	5.55	624	11.6644

fewer iterations are needed for convergence. This observation is exactly the same as the previous example, hence we will no longer report the convergence history with different CFL numbers to save space.

Example 3.4. Here we again consider 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 Γ, and it contains a singularity at Γ.

The Godunov numerical Hamiltonian is used. Due to the singularity, we follow the setup in [32], 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 5. The same behavior as in the previous examples can be observed, namely, a smaller CFL number is needed for FE-Jacobi method, and the fast sweeping technique can improve the convergence of the Jacobi scheme. Again, the FE-FSM performs the best out of these four methods, and the hybrid strategy can further reduce the computational cost.

Example 3.5. Consider the Eikonal equation with $f(x,y)=1$ on the computational domain $[-1,1]^2$. The inflow boundary Γ 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

Table 5: Example 3.4. Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

|	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	1.29e-07	-	4.17e-07	-	1354	1.4606	3.10e-07	-	4.60e-06	-	1281	0.85274
80	3.58e-09	5.17	1.07e-08	5.27	1910	7.7459	6.95e-09	5.48	1.62e-07	4.82	1765	2.8821
160	1.06e-10	5.06	2.85e-10	5.24	2707	44.3751	-	-	-	-	-	-
320	-	-	-	-	-	-	-	-	-	-	-	-
	FE-FSM $\gamma=1$						FE-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	3.10e-07	-	4.60e-06	-	192	0.19561	3.10e-07	-	4.60e-06	-	292	0.16633
80	6.95e-09	5.48	1.62e-07	4.82	240	0.91323	6.95e-09	5.48	1.62e-07	4.82	336	0.49803
160	1.19e-10	5.85	1.78e-09	6.51	296	4.466	1.19e-10	5.85	1.79e-09	6.50	340	1.5533
320	3.26e-12	5.20	8.67e-12	7.68	412	22.457	3.26e-12	5.20	8.67e-12	7.69	380	5.9912
	RK-Jacobi $\gamma=1$						RK-Jacobi $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	1.29e-07	-	4.27e-07	-	339	0.35881	3.10e-07	-	4.60e-06	-	345	0.23766
80	3.58e-09	5.17	1.07e-08	5.31	441	1.8265	6.95e-09	5.48	1.62e-07	4.82	444	0.75318
160	1.06e-10	5.06	2.85e-10	5.23	615	9.8505	1.19e-10	5.85	1.78e-09	6.51	621	3.3396
320	3.26e-12	5.03	8.54e-12	5.06	1047	69.828	3.26e-12	5.20	8.66e-12	7.68	1053	21.902
	RK-FSM $\gamma=1$						RK-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	3.10e-07	-	4.60e-06	-	300	0.30303	3.10e-07	-	4.60e-06	-	300	0.19219
80	6.95e-09	5.48	1.62e-07	4.82	372	1.4035	6.95e-09	5.48	1.62e-07	4.82	372	0.56225
160	1.19e-10	5.85	1.78e-09	6.51	468	7.0361	1.19e-10	5.85	1.78e-09	6.51	468	2.2295
320	3.26e-12	5.20	8.67e-12	7.68	612	32.526	3.26e-12	5.20	8.67e-12	7.68	624	10.4386

described as

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

The exact solution is still the distance function to Γ. Singularities appear at the two corners in Γ, 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 $x \le 0$ or $y \le 0$, and outside the box $[-0.5,0.5]^2$. The surface and contour of the numerical solution by FE-FSM are shown in Fig. 6. The numerical errors and orders of convergence are shown in Table 6. Again, we observe that the FE-Jacobi method requires the smaller CFL number of value 0.1, the other three schemes can take a lager CFL number, and the fast sweeping technique can improve the convergence of the Jacobi scheme. On the same refined mesh, we can see that the RK-FSM only takes about the 50% CPU time of the RK-Jacobi scheme. Furthermore, the FE-FSM costs even less CPU time than RK-FSM. The numerical results with the hybrid strategy can be

Table 6: Example 3.5. Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	1.12e-06	-	1.61e-05	-	1338	0.81886	1.12e-06	-	1.61e-05	-	1338	0.72088
80	4.28e-08	4.71	1.08e-06	3.89	1978	5.7302	4.28e-08	4.71	1.08e-06	3.89	1978	2.8225
160	1.05e-09	5.33	3.93e-08	4.78	3182	41.18	1.05e-09	5.33	3.93e-08	4.78	3183	14.5589
320	2.11e-11	5.64	4.44e-10	6.46	5530	323.1253	-	-	-	-	-	-
	FE-FSM $\gamma=1$						FE-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	1.12e-06	-	1.61e-05	-	236	0.13161	1.12e-06	-	1.61e-05	-	236	0.11094
80	4.28e-08	4.71	1.08e-06	3.89	264	0.72234	4.28e-08	4.71	1.08e-06	3.89	244	0.30641
160	1.05e-09	5.33	3.93e-08	4.78	296	3.5751	1.05e-09	5.33	3.93e-08	4.78	292	1.1575
320	2.11e-11	5.64	4.44e-10	6.46	444	25.457	2.11e-11	5.64	4.44e-10	6.46	396	6.0395
	RK-Jacobi $\gamma=1$						RK-Jacobi $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	1.12e-06	-	1.61e-05	-	327	0.23457	1.12e-06	-	1.61e-05	-	327	0.16019
80	4.28e-08	4.71	1.08e-06	3.89	438	1.3605	4.28e-08	4.710	1.08e-06	3.89	438	0.68153
160	1.05e-09	5.33	3.93e-08	4.78	732	10.2797	1.05e-09	5.33	3.93e-08	4.78	732	3.3886
320	2.11e-11	5.64	4.44e-10	6.46	1311	75.9122	2.11e-11	5.64	4.44e-10	6.46	1311	22.6901
	RK-FSM $\gamma=1$						RK-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	1.12e-06	-	1.61e-05	-	288	0.16363	1.12e-06	-	1.61e-05	-	300	0.13274
80	4.28e-08	4.71	1.08e-06	3.89	360	1.0679	4.28e-08	4.71	1.08e-06	3.89	372	0.44366
160	1.05e-09	5.33	3.93e-08	4.78	468	5.6948	1.05e-09	5.33	3.93e-08	4.78	480	2.0111
320	2.11e-11	5.64	4.44e-10	6.46	648	36.3658	2.11e-11	5.64	4.44e-10	6.46	624	8.9956
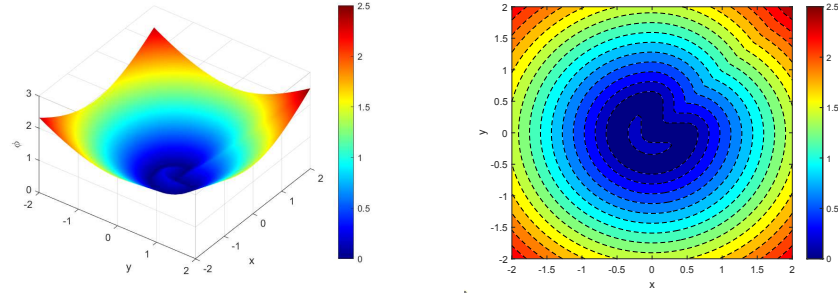
Figure 6: Example 3.5. The numerical solution by FE-FSM on mesh $N=80$. Left: the 3D plot of numerical solution ϕ; Right: the contour plot for ϕ.

seen on the right side of Table 6, which suggests that the hybrid strategy can save 60%-75% CPU time on refined mesh. As the previous examples, the FE-FSM is more efficient scheme for the example, and the hybrid strategy can further reduce the computational cost.

Example 3.6. In this test, 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 [36]. 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

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

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 the case (a), Fig. 7 shows the surface and contour of numerical solution by FE-FSM. The numerical errors and orders of convergence of four methods are listed in Table 7. We can observe that the fifth order accuracy can be obtained, and the same behavior as the previous examples can be observe for these four schemes. On the same refined mesh, we can see that the RK-FSM only takes about 50% CPU time of the RK-Jacobi scheme, furthermore, the FE-FSM costs even less CPU time than RK-FSM. In addition, when the fast sweeping technique is used, the FE-FSM can now use a larger CFL number than FE-Jacobi scheme. The numerical results after the hybrid strategy is used can be seen on the right side of Table 7, which suggests that the hybrid strategy can save 50%−75% of the CPU time on the refined mesh.

In Fig. 9, we plot the iteration numbers for different γ on the fixed mesh $N = 160$. The term "N.C." refers to the "not convergent" case, as defined above. We can observe

Table 7: Example 3.6. Case (a). Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

|	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	7.46e-08	-	3.55e-07	-	1519	1.3384	7.46e-08	-	3.54e-07	-	1526	1.7981
80	3.41e-09	4.45	1.49e-08	4.56	1974	7.7447	3.41e-09	4.45	1.49e-08	4.56	1974	8.0321
160	-	-	-	-	-	-	-	-	-	-	-	-
320	-	-	-	-	-	-	-	-	-	-	-	-
	FE-FSM $\gamma=1$						FE-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	7.46e-08	-	3.54e-07	-	216	0.29623	7.46e-08	-	3.54e-07	-	216	0.15314
80	3.41e-09	4.45	1.49e-08	4.56	244	0.8919	3.41e-09	4.45	1.49e-08	4.56	244	0.53176
160	1.09e-10	4.96	4.67e-10	5.00	288	4.4203	1.09e-10	4.96	4.67e-10	5.00	288	2.1206
320	3.46e-12	4.98	1.45e-11	5.00	372	24.2263	3.46e-12	4.98	1.45e-11	5.00	372	9.6624
	RK-Jacobi $\gamma=1$						RK-Jacobi $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	7.46e-08	-	3.56e-07	-	417	0.34677	7.46e-08	-	3.56e-07	-	417	0.33492
80	3.41e-09	4.45	1.49e-08	4.57	501	1.9144	3.41e-09	4.45	1.49e-08	4.57	501	1.2738
160	1.09e-10	4.96	4.67e-10	5.00	771	12.7353	1.09e-10	4.96	4.67e-10	5.00	771	6.5294
320	3.49e-12	4.97	6.73e-11	2.79	1392	97.0302	3.49e-12	4.97	6.74e-11	2.79	1392	45.1382
	RK-FSM $\gamma=1$						RK-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	7.46e-08	-	3.55e-07	-	324	0.37401	7.46e-08	-	3.56e-07	-	324	0.20401
80	3.41e-09	4.45	1.49e-08	4.56	372	1.4448	3.41e-09	4.45	1.50e-08	4.57	372	0.66739
160	1.09e-10	4.96	4.67e-10	5.00	468	7.4771	1.09e-10	4.96	4.67e-10	5.00	468	2.5455
320	3.46e-12	4.98	1.45e-11	5.00	636	41.7842	3.47e-12	4.98	1.45e-11	5.00	636	11.4085
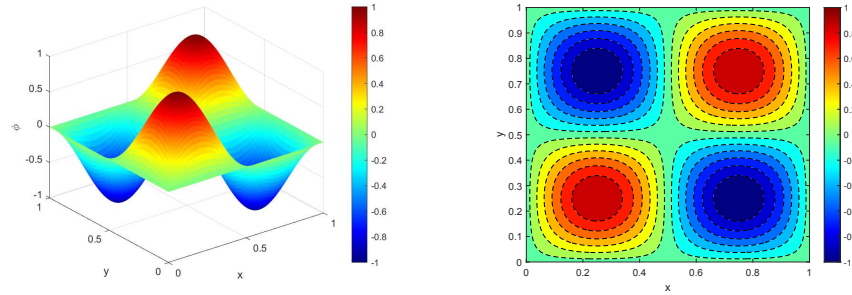
Figure 7: Example 3.6. Case (a). The numerical solution by FE-FSM on mesh $N=80$. Left: the 3D plot of numerical solution ϕ; Right: the contour plot for ϕ.

the optimal "$\gamma=1.44$" for both schemes. Under the optimal $\gamma=1.44$, the FE-FSM still converges faster than the RK-FSM.

For case (b), Fig. 8 shows the surface and contour of numerical solution by FE-FSM.

Table 8: Example 3.6. Case (b). Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	2.08e-04	-	1.39e-03	-	1528	1.325	2.13e-04	-	1.37e-03	-	1527	1.6133
80	6.69e-05	1.63	5.75e-04	1.27	1907	7.5108	6.70e-05	1.67	5.74e-04	1.26	1905	7.2127
160	1.82e-05	1.87	2.03e-04	1.50	2992	49.5382	1.82e-05	1.87	2.03e-04	1.49	2992	43.7957
320	-	-	-	-	-	-	-	-	-	-	-	-
	FE-FSM $\gamma=1$						FE-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	2.08e-04	-	1.41e-03	-	212	0.23206	2.14e-04	-	1.40e-03	-	212	0.19333
80	6.72e-05	1.63	5.79e-04	1.29	240	0.88731	6.73e-05	1.66	5.79e-04	1.27	240	0.56582
160	1.83e-05	1.87	2.04e-04	1.50	284	4.3167	1.83e-05	1.87	2.04e-04	1.50	284	1.8842
320	4.78e-06	1.93	6.51e-05	1.64	360	22.9094	4.78e-06	1.93	6.51e-05	1.64	360	8.9642
	RK-Jacobi $\gamma=1$						RK-Jacobi $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	2.08e-04	-	1.38e-03	-	402	0.38608	2.13e-04	-	1.37e-03	-	396	0.37579
80	6.69e-05	1.63	5.74e-04	1.27	456	1.7332	6.70e-05	1.67	5.75e-04	1.25	456	1.0565
160	1.82e-05	1.87	2.03e-04	1.49	756	12.0887	1.82e-05	1.87	2.03e-04	1.50	756	6.0166
320	4.76e-06	1.93	6.49e-05	1.64	1380	94.8398	4.76e-06	1.93	6.49e-05	1.64	1380	43.1701
	RK-FSM $\gamma=1$						RK-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	2.08e-04	-	1.41e-03	-	324	0.36728	2.14e-04	-	1.40e-03	-	324	0.22781
80	6.72e-05	1.63	5.79e-04	1.29	372	1.4231	6.73e-05	1.66	5.79e-04	1.27	372	0.59763
160	1.83e-05	1.87	2.04e-04	1.50	456	7.1213	1.83e-05	1.87	2.04e-04	1.50	456	2.1953
320	4.78e-06	1.93	6.51e-05	1.64	612	39.3192	4.78e-06	1.93	6.51e-05	1.64	612	10.2167
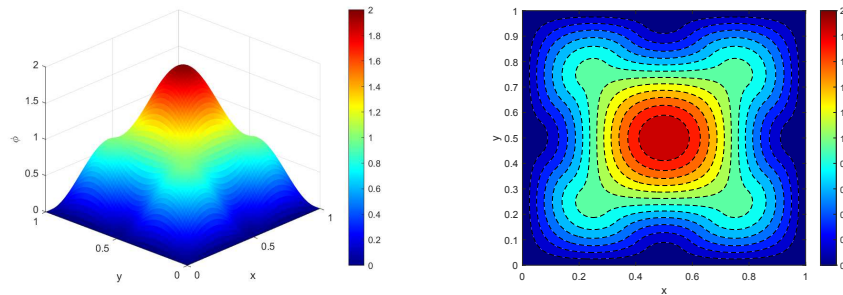
Figure 8: Example 3.6. Case (b). The numerical solution by FE-FSM on mesh $N=80$. Left: the 3D plot of numerical solution ϕ; Right: the contour plot for ϕ.

The numerical errors and orders of convergence are listed in Table 8. Due to the non-smoothness of the exact solution, we can only achieve second order accuracy. Again, the same behavior as the previous examples can be observe for these four methods.
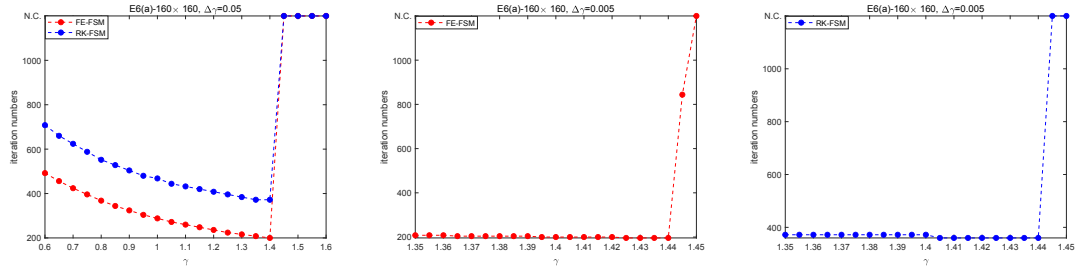
Figure 9: Example 3.6. Case (a). the iteration numbers vs γ of FS-FSM and RK-FSM on mesh $N=160$. "N.C." represents the "not convergent" case. Left: two schemes, $\gamma\in[0.6,1.6]$ with $\Delta\gamma=0.05$; Middle: FS-FSM in the zoom-in region of $\gamma\in[1.35,1.45]$ with $\Delta\gamma=0.005$; Right: RK-FSM in the zoom-in region of $\gamma\in[1.35,1.45]$ with $\Delta\gamma=0.005$.

Example 3.7. We solve the Eikonal equation with

$$f(x,y)=2\sqrt{y^2(1-x^2)^2+x^2(1-y^2)^2}.$$

The computational domain is $\Omega=[-1,1]^2$, and the inflow boundary is the whole outside boundary of the box $[-1,1]^2$, namely $\Gamma=\{(x,y)\,|\,|x|=1\text{ or }|y|=1\}$. The boundary condition $\phi(x,y)=0$ is prescribed on Γ, and an additional boundary condition $\phi(0,0)=1$ is also prescribed at the center of domain. The exact solutions is given by

$$\phi(x,y)=(1-x^2)(1-y^2).$$

The exact values are arranged in a small box with length $3h$ around the point $(0,0)$. The errors are measured on the whole domain.

Since the exact solution is a bi-quadratic polynomial, the proposed fifth order HWENO method can recover the exact solution, and the round-off errors can be observed, as shown in Table 9. We can also observe that the fast sweeping technique can improve the convergence of the Jacobi scheme. On the same refined mesh, the RK-FSM only takes about 40% CPU time of the RK-Jacobi scheme, and the FE-FSM costs even less CPU time than RK-FSM. The numerical results after the hybrid strategy is used can be seen on the right side of Table 9, which suggests that the hybrid strategy can save about 80% of the CPU time on the refined mesh.

Example 3.8. 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 [16]

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

where

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

Table 9: Example 3.7. Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

	FE-Jacobi $\gamma=0.1$				FE-Jacobi $\gamma=0.1$ with hybrid strategy			
N	L_1 error	L_∞ error	iter	time	L_1 error	L_∞ error	iter	time
40	1.05e-14	3.15e-13	1547	1.4301	1.05e-14	3.15e-13	1547	0.5632
80	9.90e-15	8.99e-13	1850	6.9849	9.73e-15	9.00e-13	1850	1.8203
160	9.85e-15	1.58e-12	3016	51.576	8.71e-15	1.42e-12	3019	10.1809
320	1.04e-14	4.06e-12	5559	428.72	9.51e-15	4.08e-12	5559	89.2658
	FE-FSM $\gamma=1$				FE-FSM $\gamma=1$ with hybrid strategy			
N	L_1 error	L_∞ error	iter	time	L_1 error	L_∞ error	iter	time
40	4.69e-16	8.32e-15	212	0.2448	3.49e-16	7.54e-15	212	0.1366
80	8.31e-16	3.48e-14	236	0.8822	3.73e-16	3.68e-14	236	0.3952
160	1.51e-15	1.21e-13	284	4.5492	4.01e-16	1.17e-13	284	0.8534
320	2.64e-15	1.10e-13	376	25.933	3.62e-16	1.16e-13	376	4.2301
	RK-Jacobi $\gamma=1$				RK-Jacobi $\gamma=1$ with hybrid strategy			
N	L_1 error	L_∞ error	iter	time	L_1 error	L_∞ error	iter	time
40	1.12e-15	4.96e-14	393	0.4107	1.00e-15	3.49e-14	402	0.2235
80	6.93e-15	6.47e-13	477	1.8100	7.96e-15	6.73e-13	474	0.64002
160	2.62e-15	1.21e-12	741	12.404	3.03e-15	1.22e-12	741	2.6417
320	3.37e-15	5.18e-12	1356	101.06	4.50e-15	5.20e-12	1356	20.7209
	RK-FSM $\gamma=1$				RK-FSM $\gamma=1$ with hybrid strategy			
N	L_1 error	L_∞ error	iter	time	L_1 error	L_∞ error	iter	time
40	4.32e-16	8.10e-15	324	0.3867	3.22e-16	8.54e-15	324	0.1814
80	8.51e-16	3.87e-14	360	1.3585	3.91e-16	4.09e-14	360	0.3997
160	1.72e-15	2.24e-13	444	7.1011	6.35e-16	2.18e-13	444	1.2264
320	2.76e-15	2.14e-13	612	41.072	5.05e-16	2.27e-13	612	6.576

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

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

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

$$\sqrt{-\frac{1}{2}(c_4\phi_x^2+c_5\phi_y^2)-\sqrt{\frac{1}{4}(c_4\phi_x^2+c_5\phi_y^2)^2-(c_1\phi_x^4+c_2\phi_x^2\phi_y^2+c_3\phi_y^4)}}=1,$$
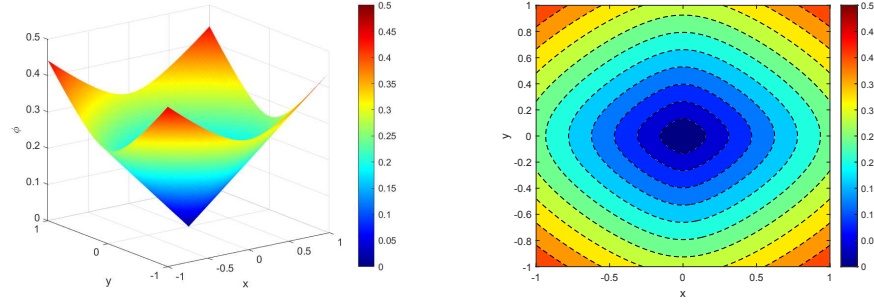
Figure 10: Example 3.8. P-wave. The numerical solution by FE-FSM on mesh $N = 80$. Left: the 3D plot of numerical solution ϕ; Right: the contour plot for ϕ.

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 Hamiltonians are in complicated forms, we use the LF numerical Hamiltonian for both equations. In addition, since we only know the numerical solution of ϕ, the "exact solution" of u and v on *Category I* will be obtained by fifth order WENO reconstruction.

For the P-wave equation, the surface and contour of numerical solution by FE-FSM are shown in Fig. 10. The numerical errors and orders of convergence are presented in Table 10 for four methods. We can observe that the FE-Jacobi method requires the smaller CFL number of value 0.1. When the third order RK time discretization is used, the RK-Jacobi scheme can take a larger CFL number of value 1. Also, the number of iterations of RK-Jacobi method is smaller than FE-Jacobi method on the same mesh. The fast sweeping technique can improve the convergence of the Jacobi scheme. On the same refined mesh, we can see that the RK-FSM only takes about 50% CPU time of the RK-Jacobi scheme. Furthermore, the FE-FSM costs even less CPU time than RK-FSM. The numerical results after the hybrid strategy is used can be seen on the right side of Table 10, which suggests that the hybrid strategy can save 50% of the CPU time on the refined mesh.

For the SV-wave equation, Fig. 11 shows the surface and contour of numerical solution by FE-FSM. The numerical errors and orders of convergence are listed in Table 11 for four methods. Again, the FE-FSM performs the best out of these four methods, and the hybrid strategy can further reduce the computational cost.

4 Conclusion remark

In this paper, we design a fifth order HWENO fixed-point fast sweeping method for solving the static HJ equations, by combining the fixed-point iteration with the fast sweep-

Table 10: Example 3.8. P-wave. Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	4.77e-06	-	3.95e-05	-	1469	2.1388	4.77e-06	-	3.95e-05	-	1463	2.0531
80	2.07e-07	4.52	2.25e-06	4.13	1781	9.9557	2.07e-07	4.52	2.25e-06	4.13	1938	9.4128
160	7.05e-09	4.87	8.09e-08	4.79	2778	61.2665	7.05e-09	4.87	8.09e-08	4.79	2941	57.1844
320	2.29e-10	4.93	2.61e-09	4.95	4785	435.3249	2.29e-10	4.93	2.61e-09	4.95	4856	391.72
	FE-FSM $\gamma=1$						FE-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	4.77e-06	-	3.95e-05	-	184	0.24145	4.77e-06	-	3.95e-05	-	184	0.1427
80	2.07e-07	4.52	2.25e-06	4.13	216	1.1029	2.07e-07	4.52	2.25e-06	4.13	212	0.54758
160	7.05e-09	4.87	8.09e-08	4.79	272	5.6603	7.05e-09	4.87	8.09e-08	4.79	272	2.8627
320	2.29e-10	4.93	2.61e-09	4.95	376	30.9783	2.29e-10	4.93	2.61e-09	4.95	400	15.4856
	RK-Jacobi $\gamma=1$						RK-Jacobi $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	4.77e-06	-	3.95e-05	-	429	0.5894	4.78e-06	-	3.95e-05	-	429	0.42699
80	2.07e-07	4.52	2.25e-06	4.13	495	2.6246	2.07e-07	4.52	2.25e-06	4.13	498	1.7419
160	7.05e-09	4.87	8.09e-08	4.79	738	15.6503	7.05e-09	4.87	8.09e-08	4.79	723	7.9175
320	2.29e-10	4.93	2.61e-09	4.95	1296	112.4767	2.29e-10	4.93	2.61e-09	4.95	1296	56.0106
	RK-FSM $\gamma=1$						RK-FSM $\gamma=1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
40	4.77e-06	-	3.95e-05	-	300	0.3865	4.77e-06	-	3.95e-05	-	300	0.2366
80	2.07e-07	4.52	2.25e-06	4.13	336	1.6854	2.07e-07	4.52	2.25e-06	4.13	336	0.9660
160	7.05e-09	4.87	8.09e-08	4.79	420	8.638	7.05e-09	4.87	8.09e-08	4.79	420	4.2972
320	2.29e-10	4.93	2.61e-09	4.95	612	48.9773	2.29e-10	4.93	2.61e-09	4.95	612	24.5722
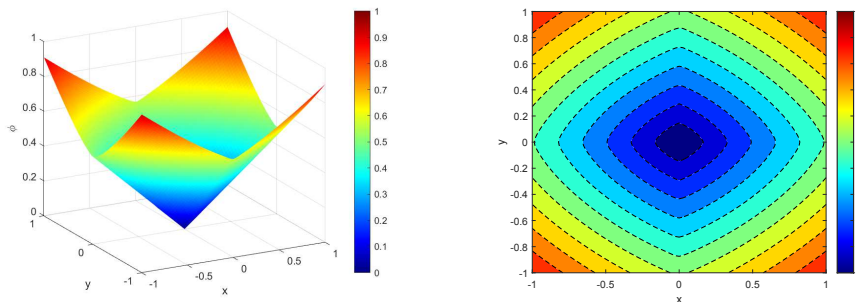
Figure 11: Example 3.8. SV-wave. The numerical solution by FE-FSM on mesh $N=80$. Left: the 3D plot of numerical solution ϕ; Right: the contour plot for ϕ.

ing strategy in a novel HWENO framework. The numerical study suggests that the fast sweeping technique can greatly improve the stability of the high-order spatial scheme. We presented a large number of numerical experiments to test four different methods,

Table 11: Example 3.8. SV-wave. Comparison of the four methods: The errors of the numerical solution, the accuracy obtained and the number of iterations for convergence.

	FE-Jacobi $\gamma=0.1$						FE-Jacobi $\gamma=0.1$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
80	1.00e-06	-	1.65e-05	-	1891	9.7947	1.01e-06	-	1.65e-05	-	1967	9.3438
160	2.58e-08	5.28	1.03e-06	4.00	2865	59.192	2.59e-08	5.28	1.03e-06	3.99	2974	55.317
320	1.08e-10	7.89	1.15e-08	6.48	4995	415.46	1.09e-10	7.89	1.15e-08	6.49	4999	384.21
	FE-FSM $\gamma=0.9$						FE-FSM $\gamma=0.9$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
80	9.63e-07	-	1.60e-05	-	284	1.5094	8.61e-07	-	2.19e-05	-	272	0.75179
160	1.95e-08	5.61	8.97e-07	4.15	368	7.6795	2.25e-08	5.25	1.44e-06	3.92	312	2.959
320	6.37e-11	8.26	9.12e-09	6.61	448	35.524	7.85e-11	8.16	7.90e-09	7.51	452	17.488
	RK-Jacobi $\gamma=0.9$						RK-Jacobi $\gamma=0.9$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
80	1.00e-06	-	1.65e-05	-	567	3.1826	9.27e-07	-Inf	1.60e-05	-	570	2.2798
160	2.58e-08	5.28	1.03e-06	4.00	855	18.276	2.191-08	5.40	9.80e-07	4.03	846	8.9459
320	1.08e-10	7.89	1.15e-08	6.48	1476	125.21	9.06e-11	7.91	1.12e-08	6.44	1467	60.973
	RK-FSM $\gamma=0.9$						RK-FSM $\gamma=0.9$ with hybrid strategy					
N	L_1 error	order	L_∞ error	order	iter	time	L_1 error	order	L_∞ error	order	iter	time
80	9.51e-07	-	1.50e-05	-	360	1.8518	9.17e-07	-	1.65e-05	-	552	1.7596
160	2.10e-08	5.49	9.78e-07	3.9473	480	9.5202	2.66e-08	5.10	1.21e-06	3.76	480	4.6605
320	6.60e-11	8.31	1.00e-08	6.604	720	57.929	8.52e-11	8.28	1.08e-08	6.79	720	27.38

including FE method and RK time marching method, and the methods combined with fast sweeping technique. Numerical results show that the FE time-marching method with fast sweeping technique is the most effective method to solve the static HJ equations. In addition, a hybrid strategy which combines both linear and HWENO reconstruction is also proposed and tested, which yields additional savings in computational time.

Acknowledgments

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 (1999), 667-695.

[2] S. Chen, Fixed-point fast sweeping WENO methods for steady state solution of scalar hyperbolic conservation laws, Int. J. Num. Anal. Mod., 11 (2014), 117-130.

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

[4] M.G. Crandall and P.-L. Lions, Two approximations of solutions of Hamilton-Jacobi equations, Math. Comput., 43 (1984), 1-19.

[5] J. Helmsen, E. Puckett, P. Colella and M. Dorr, Two new methods for simulating photolithography development in 3D, Proc. SPIE, 2726 (1996), 253-262.

[6] 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 (2008), 336-346.

[7] 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 (2009), 127-141.

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

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

[10] D. Levy, G. Puppo and G. Russo, Central WENO schemes for hyperbolic systems of conservation laws, M2AN Math. Model. Numer. Anal., 33 (1999), 547-571.

[11] Y. Li, J. Cheng, Y. Xia and C.-W. Shu, High order arbitrary Lagrangian-Eulerian finite difference WENO scheme for Hamilton-Jacobi equations, Commun. Comput. Phys., 26 (2019), 1530-1574.

[12] 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 (2008), 8191-8208.

[13] L. Li, J. Zhu and Y.-T. Zhang, Absolutely convergent fixed-point fast sweeping WENO methods for steady state of hyperbolic conservation laws, J. Comput. Phys., (2021), 110516.

[14] H. Liu and J. Qiu, Finite Difference Hermite WENO schemes for conservation laws, J. Sci. Comput., 63 (2015), 548-572.

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

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

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

[18] 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 (2004), 115-135.

[19] 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 (2005), 642-663.

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

[21] 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.

[22] Y. Ren, Y. Xing and J. Qiu, High order finite difference Hermite WENO fast sweeping methods for static Hamilton-Jacobi equations, arXiv preprint arXiv:2009.03494, (2020).

[23] Y. Ren, Y. Xing, D. Wang and J. Qiu, High order asymptotic preserving Hermite WENO fast sweeping method for the steady-state S_N transport equation, arXiv preprint

arXiv:2104.03480, (2021).

[24] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, Proc. Nat. Acad. Sci., 93 (1996), 1591-1595.

[25] 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.

[26] 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.

[27] 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 (2003), 673-694.

[28] J.N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, IEEE T. Automat. Contr., 40 (1995), 1528-1538.

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

[30] L. Wu, Y.-T. Zhang, S.H. Zhang and C.-W. Shu, High order fixed-point sweeping WENO methods for steady state of hyperbolic conservation laws and its convergence study, Commun. Comput. Phys., 20 (2005), 835-869.

[31] 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 (2008), 337-350.

[32] 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 (2010), 514-536.

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

[34] 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 (2011), 1873-1896.

[35] Y.-T. Zhang, H.-K. Zhao and S. Chen, Fixed-point iterative sweeping methods for static Hamilton-Jacobi equations, Methods Appl. Anal., 13 (2006), 299-320.

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

[37] H.-K. Zhao, A fast sweeping method for Eikonal equations, Math. Comput., 74 (2005), 603-627.

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

[39] 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.

[40] F. Zheng, C.-W. Shu and J. Qiu, High order finite difference Hermite WENO schemes for the Hamilton-Jacobi equations on unstructured meshes, Comput. Fluids, 183 (2019), 53-65.

[41] J. Zhu and J. Qiu, A New Type of High-Order WENO Schemes for Hamilton-Jacobi Equations on Triangular Meshes, Commun. Comput. Phys., 27 (2020), 897-920.

[42] J. Zhu and J. Qiu, Hermite WENO schemes for Hamilton-Jacobi equations on unstructured meshes, J. Comput. Phys., 254 (2013), 76-92.

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