Journal of Scientific Computing (2020) 83:54
https://doi.org/10.1007/s10915-020-01228-7

A Hybrid Finite Difference WENO-ZQ Fast Sweeping Method for Static Hamilton–Jacobi Equations

Yupeng Ren[1] · **Tao Xiong**[2] · **Jianxian Qiu**[2]

Received: 29 June 2019 / Revised: 25 March 2020 / Accepted: 4 May 2020

Abstract
In this paper, we propose to combine a new fifth order finite difference weighted essentially non-oscillatory (WENO) scheme with high order fast sweeping methods, for directly solving static Hamilton–Jacobi equations. This is motivated by the work in Xiong et al. (J Sci Comput 45(1–3):514–536, 2010), where a fifth order fast sweeping method base on the classical finite difference WENO scheme is developed. Numerical results in Xiong et al. (2010) show that the iterative numbers of the scheme for some cases are very sensitive to the parameter ϵ, which is used to avoid the denominator to be 0 in the nonlinear weights. Here we propose to use the new fifth order finite difference WENO-ZQ scheme, which was recently developed in Zhu and Qiu (J Comput Phys 318:110–121, 2016), to alleviate this problem. Besides, to save computational cost from WENO reconstructions, a hybrid finite difference linear and WENO scheme is used, which works more robustly. Numerical experiments will be performed to demonstrate the good performance of the new proposed approach.

Keywords Hybrid scheme · Finite difference · WENO · Fast sweeping method · Static Hamilton–Jacobi equation

Mathematics Subject Classification 65M60 · 35L65

The research is partly supported by NSAF Grant U1630247, Science Challenge Project, No. TZ2016002, NSFC Grant 11971025, NSF Grant of Fujian Province 2019J06002 and Sino-German Research Group Project, No. GZ. 1465.

✉ Jianxian Qiu
jxqiu@xmu.edu.cn

Yupeng Ren
ypren@stu.xmu.edu.cn

Tao Xiong
txiong@xmu.edu.cn

[1] School of Mathematical Sciences, Xiamen University, Xiamen 361005, Fujian, People's Republic of China

[2] School of Mathematical Sciences and Fujian Provincial Key Laboratory of Mathematical Modeling and High-Performance Scientific Computing, Xiamen University, Xiamen 361005, Fujian, People's Republic of China

Published online: 06 June 2020

1 Introduction

In this paper, we consider the following static Hamilton–Jacobi (HJ) equation

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

where Ω is the computational domain in $\mathbb{R}^d$, the function $g(\mathbf{x})$ is the boundary condition on the subset $\Gamma \subset \Omega$, the Hamiltonian H is a nonlinear Lipschitz continuous function. HJ equations have many applications, such as in optimal control, computer vision, differential game and geometric optics [6,27]. Among them, the Eikonal equation is a special class and plays an extremely important role, which can be described as

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

where $f(\mathbf{x})$ is a positive function.

The boundary value problem (BVP) (1.1) can be solved by classical methods from characteristics in the phase space. Although the characteristics may never intersect in phase space, their projection into physical space may intersect so that the solution is not unique in physical space [30]. In [2] Crandall and Lions introduced the concept of viscosity solutions, and physically relevant solution can be defined for such first order nonlinear equations (1.1).

There are mainly two classes of numerical methods for solving static HJ equations. The first one is to solve the time-dependent problem

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

with pseudo-time iterations. The equation is first discretized in time by, e.g., a total variation diminishing (TVD) Runge-Kutta time discretization [21], and then is evolved in time until the numerical solution converges. However, such a method requires too many time steps for the convergence of the solution on the entire domain, due to the finite speed of propagation. CFL time step restriction is also needed for stability. The other is to solve the stationary BVP directly, such as the fast marching method (FMM) [4,20,24] and the fast sweeping method (FSM) [9,10,23,31,32]. As compared to FMM, FSM can be designed to be arbitrarily high order and it becomes an important approach. In [1], Boué and Dupuis first proposed FSM to solve a deterministic control problem with quadratic running cost using Markov chain approximation. Then in [32], it was reformulated by using a monotone upwind scheme for solving the Eikonal equation to get the distance function. In [31], Zhao introduced a systematic way for solving the Eikonal equations on a rectangular mesh. Based on this approach, later many high order extensions have been done. In [30], FSM has been coupled with third order finite difference WENO-JP scheme [7] to solve static HJ equations, and it has been extended to fifth order in [28]. High order accurate boundary treatments have been proposed with Richardson extrapolation and Lax–Wendroff type procedure for inflow boundary conditions in [5,28], which are consistent with high order finite difference WENO FSM. In [19], Serna and Qian proposed an effective stopping criterion for high order FSM. FSM has also been designed to high order by using discontinuous Galerkin (DG) finite element method [13,14,26,29].

In this work, we try to propose another fifth order finite difference WENO FSM. This is from observation that in the formal fifth order FSM based on the classical finite difference WENO-JP scheme [28], numerical results show that for some cases, the iterative numbers of the scheme are very sensitive to the small parameter ϵ, which is used to avoid the denominator becoming 0 in the nonlinear weights for the WENO reconstruction. The reason might be due to that for some problems with point sources (e.g. Examples 6 and 7 in Sect. 3), the solution becomes singular since the characteristics would intersect at these points. The finite difference WENO-JP reconstruction with equal sub-stencils would switch among all its sub-stencils. This switching would make the convergence error stop at some error level higher than the stopping criteria. The iteration may either converge very slowly or even not converge. So in the work [28], they adjust the parameter ϵ according to the mesh sizes to make the scheme converge quickly and get the desired high order. The artificial adjusting of the parameter ϵ in the scheme would greatly limit the scheme in real applications, since the most appropriate ϵ is not known beforehand.

Here we propose to use the new simple finite difference WENO-ZQ scheme recently developed by Zhu and Qiu [35]. This WENO-ZQ scheme is based on a combination of a large stencil and two small stencils. The large stencil has the same stencil and keeps the fifth order accuracy as the original linear scheme, while the two small stencils are used to achieve essentially non-oscillatory solutions under the WENO mechanism. For this scheme, we can freely choose the positive linear weights only with their summation to be 1. It can be easily extended to high dimensions. Besides, as compared to the classical WENO-JP scheme [7,8], it has less numerical truncation errors. Later in [34], it has been extended to solve time-dependent HJ equations in one and two dimensions. A finite volume WENO-ZQ scheme for hyperbolic conservation laws in multi-dimensions was designed in [36]. In [15], Lin et al. have proposed a high order residual distribution conservative finite difference WENO-ZQ scheme for solving steady state conservation laws. In [33], Zhao et al. have designed a hybrid WENO-ZQ scheme for solving hyperbolic conservation laws. Except using the new WENO-ZQ scheme, we also employ a hybrid approach. Namely, only linear reconstruction, instead of WENO reconstruction, is used when the numerical solution is monotone on its big stencil. This hybrid approach can not only save more computational cost, but also make the scheme more robust, as the dependence on the small parameter ϵ is further reduced.

The rest of the paper is organized as follows. In Sect. 2, we introduce FSM with the finite difference WENO-ZQ reconstruction, followed by the hybrid approach. In Sect. 3, numerical examples are performed to demonstrate the effectiveness and efficiency of our proposed scheme. Concluding remarks are given in Sect. 4.

2 The Finite Difference WENO-ZQ FSM

In this section, we will describe the high order FSM for directly solving the static HJ equations [28,30]. The fifth order finite difference WENO-ZQ scheme will be used to reconstruct the first order derivatives appeared in the numerical Hamiltonian. A flowchart for the full algorithm will be summarized and hybrid linear and WENO implementation is detailed.

We start with the discretization of the computational domain Ω. Suppose that a rectangular mesh Ω_h covers the computational domain Ω. Let (x_i, y_j) denote a grid point in Ω_h, that is $\Omega_h = \{(x_i, y_j), 1 \le i \le N_x, 1 \le j \le N_y\}$, and $\phi_{i,j}$ denotes the numerical solution at the grid point (x_i, y_j). $I_{i,j} = I_i \times J_j$, where $I_i = [x_i, x_{i+1}]$ and $J_j = [y_j, y_{j+1}]$, h_x and h_y denote uniform grid sizes in the x-direction and the y-direction, respectively. For simplicity, we take $h_x = h_y = h$ in this paper. Next we discretize the Hamiltonian H by a monotone numerical Hamiltonian $\widehat{H}$ [30]

$$
\begin{cases} \widehat{H}(\phi_x^-, \phi_x^+, \phi_y^-, \phi_y^+)_{ij} = f_{ij}, & (x_i, y_j) \in \Omega_h \setminus \Gamma_h, \\ \phi_{ij} = g_{ij}, & (x_i, y_j) \in \Gamma_h \subset \Omega_h. \end{cases} \tag{2.1}
$$

In (2.1) a local solver is needed based on a fast sweeping method, which reconstructs the values $\phi_x^\pm$ and $\phi_y^\pm$ at the standing mesh point, according to its neighboring values. There are two numerical Hamiltonians which will be presented in the next subsection. The Godunov numerical Hamiltonian is defined for solving convex Hamiltonians, especially for the Eikonal equation (1.2), fast convergence can be guaranteed. The other is the Lax–Friedrichs numerical Hamiltonian, which can handle more general Hamiltonians [9], but usually requires more iterative steps.

2.1 Godunov Hamiltonian for the Eikonal Equation

Let us consider the Eikonal equation in two dimensions

$$
\begin{cases} \sqrt{\phi_x^2 + \phi_y^2} = f(x, y), & (x, y) \in \Omega, \\ \phi(x, y) = g(x, y), & (x, y) \in \Gamma \subset \Omega. \end{cases} \tag{2.2}
$$

A Godunov numerical Hamiltonian to discrete (2.2) on uniform meshes is given as follows [30]

$$
\left[\left(\frac{\phi_{i,j}^{new} - \phi_{i,j}^{xmin}}{h}\right)^+\right]^2 + \left[\left(\frac{\phi_{i,j}^{new} - \phi_{i,j}^{ymin}}{h}\right)^+\right]^2 = f_{i,j}^2, \quad x^+ = \begin{cases} x, & x > 0, \\ 0, & x < 0, \end{cases} \tag{2.3}
$$

where

$$
\begin{cases} \phi_{i,j}^{xmin} = \min(\phi_{i,j}^{old} - h(\phi_x)_{i,j}^-, \phi_{i,j}^{old} + h(\phi_x)_{i,j}^+), \\ \phi_{i,j}^{ymin} = \min(\phi_{i,j}^{old} - h(\phi_y)_{i,j}^-, \phi_{i,j}^{old} + h(\phi_y)_{i,j}^+). \end{cases}
$$

Here $\phi_{i,j}^{new}$ denotes the to-be-updated numerical solution for ϕ at the grid point (x_i, y_j), and $\phi_{i,j}^{old}$ denotes the current available value for ϕ at the same grid point. $(\phi_x)_{i,j}^\pm$ and $(\phi_y)_{i,j}^\pm$ denote high order approximations for ϕ_x and ϕ_y at the grid point (x_i, y_j) from $\{\phi_{i,j}^{old}\}_{1 \le i \le N_x, 1 \le j \le N_y}$, respectively. In the following, we will omit the super index "old" and use $\phi_{i,j}$ instead of $\phi_{i,j}^{old}$ if without any confusion, for all (i, j). For example, for a first order Godunov type FSM, $(\phi_x)_{i,j}^\pm$ can be approximated by

$$
(\phi_x)_{i,j}^- = \frac{\phi_{i,j} - \phi_{i-1,j}}{h}, \quad (\phi_x)_{i,j}^+ = \frac{\phi_{i+1,j} - \phi_{i,j}}{h}, \tag{2.4}
$$

similarly for $(\phi_y)_{i,j}^\pm$ along the y-direction. For a fifth order approximation, it will be described in the next subsection. After obtaining $\phi_{i,j}^{xmin}$ and $\phi_{i,j}^{ymin}$, the new solution can be updated from

$$
\phi_{i,j}^{new} = \begin{cases} \min(\phi_{i,j}^{xmin}, \phi_{i,j}^{ymin}) + f_{i,j}h, & \text{if } |\phi_{i,j}^{xmin} - \phi_{i,j}^{ymin}| \ge f_{i,j}h, \\ \frac{1}{2}\left(\phi_{i,j}^{xmin} + \phi_{i,j}^{ymin} + (2f_{i,j}^2h^2 - (\phi_{i,j}^{xmin} - \phi_{i,j}^{ymin})^2)^{1/2}\right), & \text{otherwise.} \end{cases} \tag{2.5}
$$

2.2 The Fifth Order WENO-ZQ Reconstruction

In order to get a high order scheme, we need to approximate the derivative ϕ_x by $\phi_x^{\pm}$ at the grid point (x_i, y_j) with high order accuracy, from upwind and downwind reconstructions respectively. In [28], $\phi_x^{\pm}$ are reconstructed by the classical fifth order finite difference WENO-JP reconstruction [7]. Here we will adopt the new fifth order finite difference WENO-ZQ reconstruction developed in [34]. For simplicity, we only describe the reconstruction of $(\phi_x)^{\pm}_{i,j}$ along the x-direction from upwind and downwind information, while $(\phi_y)^{\pm}_{i,j}$ along the y-direction can be done similarly which is omitted here. For more details we refer to [34,35].

- Approximation of $(\phi_x)^{-}_{i,j}$ from upwind information:
  Given the big stencil $S_0 = \{x_{i-3}, x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}\}$ 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 construct a quartic polynomial $p_1^-(x)$, and two linear polynomials $p_2^-(x)$, $p_3^-(x)$, such that

$$\frac{1}{h}\int_{I_k} p_1^-(x)dx = \frac{\triangle_x^+\phi_{k,j}}{h}, \qquad k = i-3, \cdots, i+1, \tag{2.6}$$

$$\frac{1}{h}\int_{I_k} p_2^-(x)dx = \frac{\triangle_x^+\phi_{k,j}}{h}, \qquad k = i-2, i-1, \tag{2.7}$$

$$\frac{1}{h}\int_{I_k} p_3^-(x)dx = \frac{\triangle_x^+\phi_{k,j}}{h}, \qquad k = i-1, i, \tag{2.8}$$

where $\triangle_x^+\phi_{k,j} = \phi_{k+1,j} - \phi_{k,j}$. We assume $p_1^-(x)$, $p_2^-(x)$, and $p_3^-(x)$ have following expressions:

$$p_1^-(x) = a_1 + b_1\xi + c_1\xi^2 + d_1\xi^3 + e_1\xi^4,$$
$$p_2^-(x) = a_2 + b_2\xi, \quad p_3^-(x) = a_3 + b_3\xi,$$

where $\xi = \frac{x-x_i}{h}$. Substituting them into (2.6)-(2.8), we get

$$\begin{aligned}
p_1^-(x) &= \frac{1}{30}\frac{\triangle_x^+\phi_{i-3,j}}{h} - \frac{13}{60}\frac{\triangle_x^+\phi_{i-2,j}}{h} + \frac{47}{60}\frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{9}{20}\frac{\triangle_x^+\phi_{i,j}}{h} - \frac{1}{20}\frac{\triangle_x^+\phi_{i+1,j}}{h} \\
&\quad + \left(\frac{1}{12}\frac{\triangle_x^+\phi_{i-2,j}}{h} - \frac{5}{4}\frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{5}{4}\frac{\triangle_x^+\phi_{i,j}}{h} - \frac{1}{12}\frac{\triangle_x^+\phi_{i+1,j}}{h}\right)\frac{x-x_i}{h} \\
&\quad + \left(-\frac{1}{8}\frac{\triangle_x^+\phi_{i-3,j}}{h} + \frac{3}{4}\frac{\triangle_x^+\phi_{i-2,j}}{h} - \frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{1}{4}\frac{\triangle_x^+\phi_{i,j}}{h} + \frac{1}{8}\frac{\triangle_x^+\phi_{i+1,j}}{h}\right)\left(\frac{x-x_i}{h}\right)^2 \\
&\quad + \left(-\frac{1}{6}\frac{\triangle_x^+\phi_{i-2,j}}{h} + \frac{1}{2}\frac{\triangle_x^+\phi_{i-1,j}}{h} - \frac{1}{2}\frac{\triangle_x^+\phi_{i,j}}{h} + \frac{1}{6}\frac{\triangle_x^+\phi_{i+1,j}}{h}\right)\left(\frac{x-x_i}{h}\right)^3 \\
&\quad + \left(\frac{1}{24}\frac{\triangle_x^+\phi_{i-3,j}}{h} - \frac{1}{6}\frac{\triangle_x^+\phi_{i-2,j}}{h} + \frac{1}{4}\frac{\triangle_x^+\phi_{i-1,j}}{h} - \frac{1}{6}\frac{\triangle_x^+\phi_{i,j}}{h} + \frac{1}{24}\frac{\triangle_x^+\phi_{i+1,j}}{h}\right)\left(\frac{x-x_i}{h}\right)^4, \\
p_2^-(x) &= -\frac{1}{2}\frac{\triangle_x^+\phi_{i-2,j}}{h} + \frac{3}{2}\frac{\triangle_x^+\phi_{i-1,j}}{h} + \left(-\frac{\triangle_x^+\phi_{i-2,j}}{h} + \frac{\triangle_x^+\phi_{i-1,j}}{h}\right)\frac{x-x_i}{h}, \\
p_3^-(x) &= \frac{1}{2}\frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{1}{2}\frac{\triangle_x^+\phi_{i,j}}{h} + \left(-\frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{\triangle_x^+\phi_{i,j}}{h}\right)\frac{x-x_i}{h}.
\end{aligned}$$

We only need the polynomial values at $x = x_i$, which are given as:

$$
\begin{aligned}
(\phi_x)_{i,j}^{-,1} &:= p_1^-(x_i) \\
&= \frac{1}{30}\frac{\triangle_x^+\phi_{i-3,j}}{h} - \frac{13}{60}\frac{\triangle_x^+\phi_{i-2,j}}{h} + \frac{47}{60}\frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{9}{20}\frac{\triangle_x^+\phi_{i,j}}{h} \\
&\quad - \frac{1}{20}\frac{\triangle_x^+\phi_{i+1,j}}{h},
\end{aligned} \tag{2.9}
$$

$$
(\phi_x)_{i,j}^{-,2} := p_2^-(x_i) = -\frac{1}{2}\frac{\triangle_x^+\phi_{i-2,j}}{h} + \frac{3}{2}\frac{\triangle_x^+\phi_{i-1,j}}{h}, \tag{2.10}
$$

$$
(\phi_x)_{i,j}^{-,3} := p_3^-(x_i) = \frac{1}{2}\frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{1}{2}\frac{\triangle_x^+\phi_{i,j}}{h}. \tag{2.11}
$$

- Approximation of $(\phi_x)_{i,j}^+$ from downwind information:
  Given the big stencil $\widetilde{S}_0 = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+2}, x_{i+3}\}$ 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 construct a quartic polynomial $p_1^+(x)$, and two linear polynomials $p_2^+(x)$, $p_3^+(x)$, such that

$$
\frac{1}{h}\int_{I_k} p_1^+(x)dx = \frac{\triangle_x^+\phi_{k,j}}{h}, \qquad k = i-2, \cdots, i+2, \tag{2.12}
$$

$$
\frac{1}{h}\int_{I_k} p_2^+(x)dx = \frac{\triangle_x^+\phi_{k,j}}{h}, \qquad k = i-1, i, \tag{2.13}
$$

$$
\frac{1}{h}\int_{I_k} p_3^+(x)dx = \frac{\triangle_x^+\phi_{k,j}}{h}, \qquad k = i, i+1. \tag{2.14}
$$

  $p_1^+(x)$, $p_2^+(x)$ and $p_3^+(x)$ are in mirror-symmetric with respect to $p_1^-(x)$, $p_2^-(x)$ and $p_3^-(x)$, correspondingly. The values $(\phi_x)_{i,j}^{+,n}$ $(n = 1, 2, 3)$ can be given directly as follows:

$$
\begin{aligned}
(\phi_x)_{i,j}^{+,1} &:= p_1^+(x_i) \\
&= -\frac{1}{20}\frac{\triangle_x^+\phi_{i-2,j}}{h} + \frac{9}{20}\frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{47}{60}\frac{\triangle_x^+\phi_{i,j}}{h} - \frac{13}{60}\frac{\triangle_x^+\phi_{i+1,j}}{h} \\
&\quad + \frac{1}{30}\frac{\triangle_x^+\phi_{i+2,j}}{h},
\end{aligned} \tag{2.15}
$$

$$
(\phi_x)_{i,j}^{+,2} := p_2^+(x_i) = \frac{1}{2}\frac{\triangle_x^+\phi_{i-1,j}}{h} + \frac{1}{2}\frac{\triangle_x^+\phi_{i,j}}{h}, \tag{2.16}
$$

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

Based on these values, in the WENO-ZQ reconstruction, $(\phi_x)_{i,j}^{\pm}$ are computed by a combination of them [11,12,34]

$$
(\phi_x)_{i,j}^{\pm} = \omega_1^{\pm}\left(\frac{1}{\gamma_1}(\phi_x)_{i,j}^{\pm,1} - \frac{\gamma_2}{\gamma_1}(\phi_x)_{i,j}^{\pm,2} - \frac{\gamma_3}{\gamma_1}(\phi_x)_{i,j}^{\pm,3}\right) + \omega_2^{\pm}(\phi_x)_{i,j}^{\pm,2} + \omega_3^{\pm}(\phi_x)_{i,j}^{\pm,3}, \tag{2.18}
$$

where the parameters $\omega_n^{\pm}$ $(n = 1, 2, 3)$ and γ_n $(n = 1, 2, 3)$ are called nonlinear weights and linear weights, respectively. The γ_n's can be any positive constants only if $\gamma_1 + \gamma_2 + \gamma_3 = 1$ and $\omega_n^{\pm}$'s are computed from

$$
\omega_n^{\pm} = \frac{\overline{\omega}_n^{\pm}}{\sum_{l=1}^{3}\overline{\omega}_l^{\pm}}, \qquad \overline{\omega}_n = \gamma_n\left(1 + \frac{\tau^{\pm}}{\epsilon + \beta_n^{\pm}}\right), \qquad n = 1, 2, 3, \tag{2.19}
$$

in which ϵ is a small positive number to avoid the denominator becoming 0, and

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

Here $\beta_n^{\pm}(n = 1, 2, 3)$ are called smoothness indicators. They can be computed from [34]

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

and

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

where $r = 4$ for $n = 1$, and $r = 1$ for $n = 2, 3$. The explicit expressions for the smoothness indicators β_n^- are given as follows

$$\begin{aligned}
\beta_1^- &= \frac{1}{144} \left(\frac{\triangle_x^+ \phi_{i-3,j}}{h} - 8 \frac{\triangle_x^+ \phi_{i-2,j}}{h} + 8 \frac{\triangle_x^+ \phi_{i,j}}{h} - \frac{\triangle_x^+ \phi_{i+1,j}}{h} \right)^2 \\
&\quad + \frac{781}{2880} \left(-\frac{\triangle_x^+ \phi_{i-3,j}}{h} + 2 \frac{\triangle_x^+ \phi_{i-2,j}}{h} - 2 \frac{\triangle_x^+ \phi_{i,j}}{h} + \frac{\triangle_x^+ \phi_{i+1,j}}{h} \right)^2 \\
&\quad + \frac{1421461}{1310400} \left(\frac{\triangle_x^+ \phi_{i-3,j}}{h} - 4 \frac{\triangle_x^+ \phi_{i-2,j}}{h} + 6 \frac{\triangle_x^+ \phi_{i-1,j}}{h} - 4 \frac{\triangle_x^+ \phi_{i,j}}{h} + \frac{\triangle_x^+ \phi_{i+1,j}}{h} \right)^2 \\
&\quad + \frac{1}{15600} \left(-11 \frac{\triangle_x^+ \phi_{i-3,j}}{h} + 174 \frac{\triangle_x^+ \phi_{i-2,j}}{h} - 326 \frac{\triangle_x^+ \phi_{i-1,j}}{h} + 174 \frac{\triangle_x^+ \phi_{i,j}}{h} - 11 \frac{\triangle_x^+ \phi_{i+1,j}}{h} \right)^2, \\
\beta_2^- &= \left(\frac{\triangle_x^+ \phi_{i-2,j}}{h} - \frac{\triangle_x^+ \phi_{i-1,j}}{h} \right)^2, \\
\beta_3^- &= \left(\frac{\triangle_x^+ \phi_{i-1,j}}{h} - \frac{\triangle_x^+ \phi_{i,j}}{h} \right)^2.
\end{aligned}$$

β_n^+'s are also in mirror symmetric with respect to β_n^-'s, and we omit them here to save space.

2.3 A Flowchart of Hybrid Finite Difference WENO-ZQ FSM

In this subsection, we will give a flowchart for the full fifth order finite difference WENO-ZQ FSM. Instead of using the WENO-ZQ reconstruction on the whole computational domain, here we propose a hybrid linear and WENO-ZQ reconstruction approach. Namely, we will use the fifth order linear scheme based on the big stencil S_0 or $\tilde{S}_0$ in the last subsection, when the numerical solution is monotone, that is $\{\triangle_x^+ \phi_{i,j}\}$ do not change sign in either S_0 or $\tilde{S}_0$, respectively. Otherwise, the WENO-ZQ reconstruction is used. For more details, we refer to [33]. For this hybrid approach, numerical tests show that it helps not only to save more computational cost, but also make the scheme more robust, since the dependence on the small parameter ϵ is further reduced.

Next we will describe the hybrid scheme in detail. We first divide the grid points $\{(x_i, y_j)\}$ into the following five categories:

Category I: For points on the boundary Γ, values are assigned from the exact boundary conditions and fixed during the fast sweeping iterations.

Category II: For points at the outflow boundary of the domain, where no physical boundary condition is given. Ghost points outside the computational domain near the outflow boundary are usually used due to the wide stencil of high order approximations. The numerical solution $\phi_{i,j}$ in this category is obtained by high order extrapolation.
Category III: For points near the inflow boundary (whose distances to Γ are less than or equal to $3h$). These points cannot be updated by the fifth order FSM because of its wide stencil. The numerical boundary treatment from [28] is used. If the inflow boundary Γ is a single point or a set of isolated points, these point values are obtained by the Richardson extrapolation, which is a combination of several first order solutions at different mesh sizes. Otherwise, if Γ is a smooth curve, the Lax–Wendroff type procedure (later named the inverse Lax–Wendroff method [22]) can be used, which repeatedly uses the PDE to obtain a high order approximation based on Taylor expansions.
Category IV: Those points whose distances to *Category III* are less than or equal to $3h$ (excluding *Category I*). We need to update these point values during the fast sweeping iterations.
Category V: All the remaining points. We also need to update these point values during the fast sweeping iterations.

Note that point values in *Category II* and *Category III* are obtained by the boundary treatments. We only need to update the point values in *Category IV* and *Category V* in the following sweepings. We now summarize our hybrid fifth order finite difference Godunov type WENO-ZQ FSM as follows:

Step 1. *Initialization*. We use the solution from the corresponding first order method base on (2.4) as the initial guess.

Step 2. *Gauss-Seidel iteration*. We solve the discretized nonlinear system (2.3) by Gauss-Seidel iterations with four alternating direction sweepings

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

In each sweeping, the updating procedure is as follows:
For *Category IV*: $(\phi_x)^{\pm}_{i,j}$ are computed directly by (2.18), similarly for $(\phi_y)^{\pm}_{i,j}$;
For *Category V*:

$$(\phi_x)^{-}_{i,j}=\begin{cases}(2.9), & \text{if } \{\triangle_x^{+}\phi_{i,j}\} \text{ have the same sign on } S_0,\\(2.18), & \text{otherwise,}\end{cases}\tag{2.20}$$

$$(\phi_x)^{+}_{i,j}=\begin{cases}(2.15), & \text{if } \{\triangle_x^{+}\phi_{i,j}\} \text{ have the same sign on } \widetilde{S}_0,\\(2.18), & \text{otherwise.}\end{cases}\tag{2.21}$$

$(\phi_y)^{\pm}_{i,j}$ can be obtained similarly along the y-direction. Then $\phi^{new}_{i,j}$ is updated by (2.5).

Step 3. *Convergence*. For two consecutive iteration steps, if

$$||\phi^{new}-\phi^{old}||_{L_1}\leq\delta,\tag{2.22}$$

then the convergence is declared and we stop the iteration. The threshold δ is a given small positive constant. We take $\delta=10^{-14}$ in our numerical tests.

Remark 1 The criteria (2.20) and (2.21) are based on the monotonicity of the numerical solution. Since oscillations usually happen around shocks, in which cases the sign of $\triangle_x^{+}\phi_{i,j}$ or $\triangle_y^{+}\phi_{i,j}$ on their corresponding big stencil would change, and (2.18) is needed. The idea is similar to that in [33], but here we do not explicitly get the extreme points of a quartic polynomial. We simply indicate a smooth cell when $\{\triangle_x^+\phi_{i,j}\}$ share the same sign on S_0 or $\widetilde{S}_0$, respectively. Numerical tests show it works well for the static Hamilton–Jacobi equations.

Remark 2 The first order Godunov scheme is upwind and monotone, fast convergence can be guaranteed [31]. It would be easier to pre-determine the sign for $\triangle_x^+\phi_{i,j}$ in (2.20) and (2.21) from the first order solution of initialisation at step 1, which will be fixed and directly used in (2.20) and (2.21) at step 2. This approach is used in our numerical tests and it can save a lot of computational costs.

Remark 3 If we take the classical fifth order finite difference WENO-JP reconstruction [8] to get $\phi_x^\pm$, we denote the scheme as hybrid WENO-JP FSM. We will compare these two schemes in the numerical tests.

2.4 Lax–Friedrichs Hamiltonian for General Static HJ Equations

For general static HJ equations, we can follow the same procedure as the high order Godunov type FSM for the Eikonal equation. Instead of the Godunov numerical Hamiltonian, here we use the Lax–Friedrichs (LF) numerical Hamiltonian [16], which is the monotone numerical Hamiltonian defined as follows:

$$\widehat{H}^{LF}(u^-,u^+,v^-,v^+) = H\left(\frac{u^-+u^+}{2},\frac{v^-+v^+}{2}\right) - \frac{\alpha^x}{2}(u^+-u^-) - \frac{\alpha^y}{2}(v^+-v^-), \tag{2.23}$$

where

$$\alpha^x = \max_{\substack{A\le u\le B\\ C\le v\le D}} |H_1(u,v)|, \qquad \alpha^y = \max_{\substack{A\le u\le B\\ C\le v\le D}} |H_2(u,v)|.$$

$H_p(u,v)$ $(p=1,2)$ is the partial derivative of H with respect to the p-th argument, or the Lipschitz constant of H with respect to the p-th argument. $[A,B]$ is the value range for $u^\pm$, and $[C,D]$ is the value range for $v^\pm$. The LF FSM for static HJ equations can be written as [30]

$$\begin{aligned}\phi_{i,j}^{new} = \left(\frac{\alpha^x+\alpha^y}{h}\right)\Bigg[f_{i,j} - H\left(\frac{(\phi_x)_{i,j}^+ + (\phi_x)_{i,j}^-}{2},\frac{(\phi_y)_{i,j}^+ + (\phi_y)_{i,j}^-}{2}\right)\\ +\alpha^x\frac{(\phi_x)_{i,j}^+ - (\phi_x)_{i,j}^-}{2} + \alpha^y\frac{(\phi_y)_{i,j}^+ - (\phi_y)_{i,j}^-}{2}\Bigg] + \phi_{i,j}^{old},\end{aligned} \tag{2.24}$$

where $\phi_{i,j}^{new}$ and $\phi_{i,j}^{old}$ have the same meanings as in the Godunov numerical Hamiltonian.

For the LF numerical Hamiltonian solving general static HJ equations, the flowchart of the full algorithm follows the same steps as in the Sect. 2.3. $(\phi_x)^\pm$ and $(\phi_y)^\pm$ are reconstructed the same as in the Sect. 2.2, only we use (2.24) instead of (2.5). However, in the initialization step 1, the LF numerical Hamiltonian in [30], a big enough value, e.g. 10^6, is used as the initial guess. A better initial value will help to reduce the number of iterations, so that to save CPU time cost, especially the LF numerical Hamiltonian usually requires a lot of iterations. As proposed in [9], similar to the Godunov type numerical Hamiltonian, the solution from a first order scheme is used as the initial guess for those points belonging to *Category IV* and *V*. However, the LF numerical Hamiltonian does not need very accurate initial guesses. Due to its slow convergence even for a first order scheme, we only take the convergence threshold

$\delta = 10^{-1}$ in (2.22) for the first order scheme, rather than $\delta = 10^{-10}$ as in [9]. Numerically we find that it works well as an initial guess for the corresponding high order schemes. This first order scheme is also used to pre-determine the signs of $\triangle_x^+ \phi_{i,j}$ in (2.20) and (2.21).

3 Numerical Examples

In this section, we will perform some numerical tests by using our proposed hybrid fifth order finite difference WENO-ZQ FSM for static HJ equations, especially the Eikonal equations. We will compare it with the fifth order finite difference WENO-JP FSM. For schemes without hybridization, the WENO type reconstruction is used for all cells in *Category IV* and *Category V*. In all numerical examples, we take the linear weights $\gamma_1 = 0.9$, $\gamma_2 = \gamma_3 = 0.05$, since the solutions do not contain strong discontinuities. $\epsilon = 10^{-6}$ is used unless otherwise specified. Errors and orders are compared at different scenarios. We use "iter" to denote the iterative number. Each iteration includes four alternating sweepings. For all examples, we take the mesh size $N_x = N_y = N$. All computations are carried out by using MATLAB 2018b on a ThinkPad computer with 1.70 GHz Intel Core i5 processor and 4GB RAM.

Example 1 We solve the Eikonal equation (2.2) with

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

The computational domain is $[-1, 1]^2$, and the inflow boundary Γ is a single point source at $(0, 0)$. The exact solution is

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

The group velocity vectors [18] are pointing out along the same directions as the characteristics (rays), and ϕ is increasing along these characteristics. For mesh size $N = 80$, the group velocity vectors and contours are shown in Fig. 1a and the surface plot of the numerical solution is shown in Fig. 1b. Numerical errors and orders for different schemes are provided in Table 1. The fifth order Richardson extrapolation is used for those points belonging to

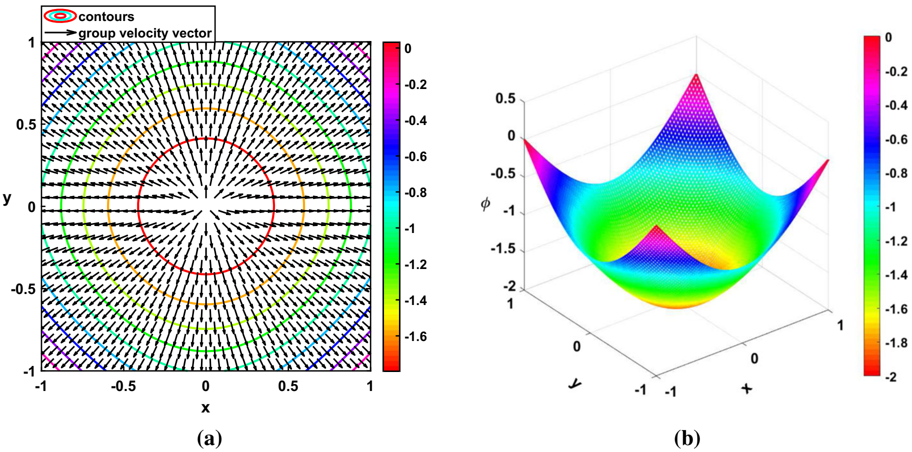

Fig. 1 Example 1, $N = 80$. **a** The group velocity vectors and contours of the numerical solution ϕ. **b** The surface plot of ϕ for the hybrid WENO-ZQ scheme

Table 1 Example 1. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ					WENO-JP				
40	5.81e−06	–	3.71e−05	–	38	1.54e−05	–	9.83e−05	–	47
80	1.00e−07	5.85	1.58e−06	4.55	47	1.13e−07	7.09	1.60e−06	5.94	51
160	6.92e−10	7.18	2.26e−08	6.12	61	8.45e−10	7.06	2.24e−08	6.15	63
320	2.16e−12	8.31	3.41e−11	9.37	82	3.01e−12	8.13	3.29e−11	9.40	80
640	5.49e−14	5.30	2.03e−13	7.39	117	5.68e−14	5.72	1.42e−13	7.85	116
	Hybrid WENO-ZQ					Hybrid WENO-JP				
40	5.85e−06	–	3.71e−05	–	39	7.56e−06	–	5.45e−05	–	44
80	1.00e−07	5.85	1.58e−06	4.55	47	1.03e−07	6.19	1.58e−06	5.10	46
160	6.92e−10	7.18	2.26e−08	6.12	60	7.20e−10	7.16	2.26e−08	6.12	59
320	2.16e−12	8.31	3.41e−11	9.37	78	2.26e−12	8.31	3.43e−11	9.36	78
640	5.49e−14	5.30	1.69e−13	7.65	115	5.51e−14	5.36	1.90e−13	7.49	115

The fifth order Richardson procedure is used for those points belonging to *Category III*. The errors are measured in the box $[-0.9, 0.9]^2$

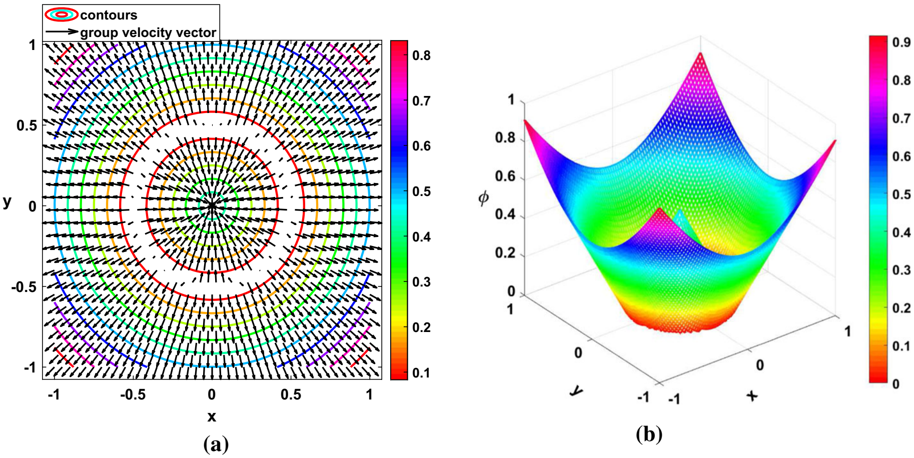

Fig. 2 Example 2, $N = 80$. **a** The group velocity vectors and contours of the numerical solution ϕ. **b** The surface plot of ϕ for the hybrid WENO-ZQ scheme

Category III. For the Richardson extrapolation, we refer to [5,28]. The third order extrapolation is used for those points belonging to *Category II*. We can see that the errors and orders obtained by these different methods are very similar. Moreover, the iterative numbers at the same mesh sizes are almost the same for all schemes. The CPU time is presented in Table 16, which indicates the hybrid WENO-ZQ scheme saves about 50% as compared to WENO-ZQ, while the hybrid WENO-JP scheme saves about 35% as compared to WENO-JP, similarly for the following examples.

Example 2 We solve the Eikonal equation (2.2) with $f(x, y) = 1$. The computational domain is $[-1, 1]^2$, and the inflow boundary Γ is a circle with center at (0, 0) and radius 0.5, that is

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

The boundary condition is $\phi(x, y) = 0$ on Γ. The exact solution is a distance function to the circle Γ, and it has a singular point at the center of the circle (where the characteristic lines intersect). The Lax–Wendroff procedure [5,28] is used for points belonging to *Category III*. The errors are measured in the box $[-0.9, 0.9]^2$ while outside the box $[-0.15, 0.15]^2$, which aim to remove the influence of the singularity and the outflow boundary treatment. When the mesh size is $N = 80$, the group velocity vectors and contours are shown in Fig. 2a and the surface plot of the numerical solution is shown in Fig. 2b. The numerical errors and orders are listed in Table 2. Similarly, the errors and orders are very close among the four methods, and the fifth order accuracies are all obtained. Besides the number of iterations are almost the same. The CPU time is provided in Table 16, which shows that the hybrid WENO-ZQ scheme saves about 40% as compared to WENO-ZQ, while the hybrid WENO-JP scheme saves about 50% as compared to WENO-JP for this example.

Example 3 We solve the Eikonal equation (2.2) with $f(x, y) = 1$. The computational domain is $[-3, 3]^2$, 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, that is

Table 2 Example 2. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ					WENO-JP				
80	3.74e−08	–	3.38e−06	–	37	3.79e−08	-	5.21e−06	–	35
160	4.96e−10	6.23	5.77e−08	5.87	44	6.80e−10	5.79	2.65e−07	4.29	45
320	1.64e−11	4.91	1.01e−09	5.82	59	1.65e−11	5.36	1.09e−09	7.91	59
640	5.65e−13	4.86	3.65e−11	4.79	80	5.65e−13	4.86	3.67e−11	4.90	82
	Hybrid WENO-ZQ					Hybrid WENO-JP				
80	3.49e−08	–	3.50e−06	–	57	2.91e−08	–	4.49e−06	–	69
160	4.91e−10	6.14	5.08e−08	6.10	52	5.62e−10	5.71	3.09e−07	3.87	44
320	1.64e−11	4.90	1.01e−09	5.64	59	1.65e−11	5.09	1.01e−09	8.24	59
640	5.67e−13	4.85	3.65e−11	4.79	81	5.64e−13	4.86	3.65e−11	4.80	82

The Lax–Wendroff procedure is used for those points belonging to *Category III*. The errors are measured in the box $[-0.9, 0.9]^2$ while outside the box $[-0.15, 0.15]^2$

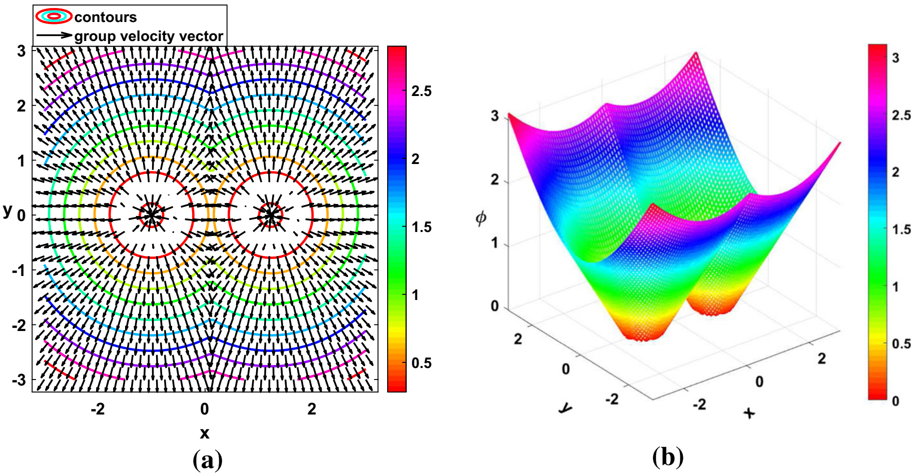

Fig. 3 Example 3, $N = 80$. **a** The group velocity vectors and contours of the numerical solution ϕ. **b** The surface plot of ϕ for the hybrid WENO-ZQ scheme

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

The exact solution is the distance function to the inflow boundary Γ. The Lax–Wendroff procedure is used for those points belonging to *Category III*. For the solution, the line with equal distances to the centers of the two circles is singular, where the characteristics would intersect. Here we measure the errors within the box $[-2.85, 2.85]^2$, but also exclude 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 the two centers of Γ and the singular line. When the mesh size is $N = 80$, the group velocity vectors and contours are shown in Fig. 3a and the surface plot of the numerical solution is shown in Fig. 3b. Numerical errors and orders are shown in Table 3. The CPU time is shown in Table 16, which shows that the hybrid WENO-ZQ scheme saves about 40% as compared to

Table 3 Example 3. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ					WENO-JP				
80	5.42e−06	-	1.25e−04	–	42	3.61e−06	-	1.22e−04	–	48
160	4.81e−07	3.49	3.47e−05	1.84	54	2.16e−07	4.05	6.73e−06	4.18	125
320	8.04e−09	5.90	1.70e−06	4.35	71	5.63e−09	5.26	4.41e−07	3.93	128
640	7.06e−11	6.83	8.75e−09	7.60	96	7.28e−11	6.27	8.87e−09	5.63	99
1280	2.21e−12	4.99	2.68e−10	5.02	150	2.21e−12	5.04	2.77e−10	4.99	152
	Hybrid WENO-ZQ					Hybrid WENO-JP				
80	4.82e−06	–	9.72e−05	–	42	3.69e−06	–	9.79e−05	–	45
160	4.79e−07	3.33	3.47e−05	0.76	54	2.28e−07	4.01	8.26e−06	3.56	121
320	8.01e−09	5.90	1.56e−06	4.47	71	5.04e−09	5.50	6.34e−07	3.70	115
640	7.06e−11	6.82	8.75e−09	7.48	97	7.37e−11	6.09	2.02e−08	4.96	97
1280	2.21e−12	4.99	2.68e−10	5.02	152	2.21e−12	5.05	2.68e−10	6.23	152

The Lax–Wendroff procedure is used for those points belonging to *Category III*. The errors are measured in the box $[-2.85, 2.85]^2$, while excluded 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 $[\sqrt{0.375} - 0.65, \sqrt{0.375} - 0.35] \times [-2.85, 2.85]$

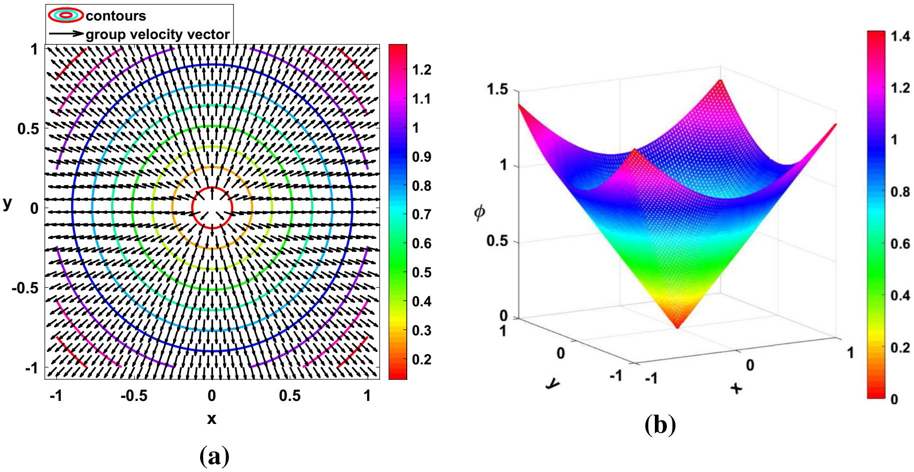

Fig. 4 Example 4, $N = 80$. **a** The group velocity vectors and contours of the numerical solution ϕ. **b** The surface plot of ϕ for the hybrid WENO-ZQ scheme

WENO-ZQ, while the hybrid WENO-JP scheme saves about 50% as compared to WENO-JP for this example.

Example 4 In this example, we consider to solve the Eikonal equation (2.2) with $f(x, y) = 1$ in the two-dimensional (2D) case, and also a corresponding three-dimensional (3D) problem for $\phi(x, y, z)$ with $f(x, y, z) = 1$. The computational domain is $[-1, 1]^2$ in 2D and $[-1, 1]^3$ in 3D. The inflow boundary Γ is a single point source at the origin. The exact solutions for these two problems are the distance functions to Γ correspondingly.

Both solutions are singular at the point source, a fifth order Richardson procedure for those points belonging to *Category III* does not give fifth order accuracy. Instead the exact solutions are pre-assigned in a small box with length 0.3 around the source point [28]. When the mesh size is $N = 80$, for the 2D case, the group velocity vectors and contours are shown in Fig. 4a and the surface plot of the numerical solution is shown in Fig. 4b. Numerical errors and orders for 2D and 3D are listed in Table 4 and Table 5, respectively. With this boundary treatment at the point source, the fifth order can be obtained for all schemes and the errors are almost the same. From these two tables, we can see that, the 3D case even has smaller iterative numbers as compared to 2D, and this is also observed for a third order WENO-JP FSM in [30] (Table V and Table VI). The CPU time is shown in Table 16, still hybrid schemes cost less CPU time than non-hybrid schemes.

Example 5 We solve the Eikonal equation (2.2) with $f(x, y) = 1$. The computational domain is $[-2, 2]^2$, 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 described as

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

The exact solution is the distance function to Γ. Singularities are at the two corners of Γ, which give rise to both shock and rarefaction wave in the solution. The Lax-Wendroff procedure

Table 4 Example 4 in 2D. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ					WENO-JP				
40	1.64e−06	–	7.18e−06	–	35	3.24e−06	–	7.74e−06	–	37
80	4.27e−08	5.26	2.72e−07	4.72	43	9.53e−08	5.08	2.26e−07	5.09	42
160	1.14e−09	5.22	4.36e−09	5.96	53	1.21e−09	6.29	3.76e−09	5.91	52
320	3.37e−11	5.08	8.83e−11	5.62	68	3.39e−11	5.16	9.07e−11	5.37	69
640	1.02e−12	5.03	2.68e−12	5.03	93	1.02e−12	5.04	2.69e−12	5.07	97
	Hybrid WENO-ZQ					Hybrid WENO-JP				
40	1.64e−06	–	7.18e−06	-	36	2.87e−06	–	6.61e−06	–	35
80	4.27e−08	5.26	2.72e−07	4.72	43	7.55e−08	5.24	2.17e−07	4.92	42
160	1.14e−09	5.22	4.36e−09	5.96	54	1.15e−09	6.03	3.98e−09	5.76	54
320	3.37e−11	5.08	8.83e−11	5.71	71	3.37e−11	5.09	8.91e−11	5.48	71
640	1.02e−12	5.03	2.68e−12	5.03	97	1.02e−12	5.04	2.69e−12	5.04	97

The exact values are assigned in a small box with length 0.3 around the center of the domain. The errors are measured in the box $[-0.9, 0.9]^2$

Table 5 Example 4 in 3D. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ					WENO-JP				
40	1.97e−06	–	8.31e−06	–	12	3.62e−06	–	1.01e−05	–	12
80	5.34e−08	5.20	3.31e−07	4.64	18	1.01e−07	5.16	3.20e−07	4.98	18
160	1.41e−09	5.24	5.38e−09	5.94	28	1.41e−09	6.15	5.56e−09	5.84	29
320	3.37e−11	5.07	1.22e−10	5.46	45	4.19e−11	5.08	1.27e−10	5.44	46
	Hybrid WENO-ZQ					Hybrid WENO-JP				
40	1.97e−06	–	8.31e−06	-	12	3.41e−06	–	9.71e−06	–	16
80	5.34e−08	5.20	3.29e−07	4.65	18	8.70e−08	5.29	3.22e−07	4.91	18
160	1.41e−09	5.24	5.37e−09	5.93	29	7.57e−09	6.02	4.21e−07	5.80	29
320	4.21e−11	5.06	1.22e−10	5.46	47	4.20e−11	5.08	1.23e−10	5.46	47

The exact solution values are assigned in a small box with length 0.3 around the center of the domain. The errors are measured in the box $[-0.9, 0.9]^3$

is used for those points belonging to *Category III*. 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 $[-0.5, 0.5]^2$. When the mesh size is $N = 80$, the group velocity vectors and contours are shown in Fig. 5a and the surface plot of the numerical solution is shown in Fig. 5b. Numerical errors and orders are listed in Table 6. They are also very similar for different methods. For this problem with the Lax–Wendroff boundary treatment, the fifth order accuracy is achieved. The CPU time is shown in Table 16, and hybrid schemes cost less time.

Example 6 We solve the Eikonal equation (2.2) with

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

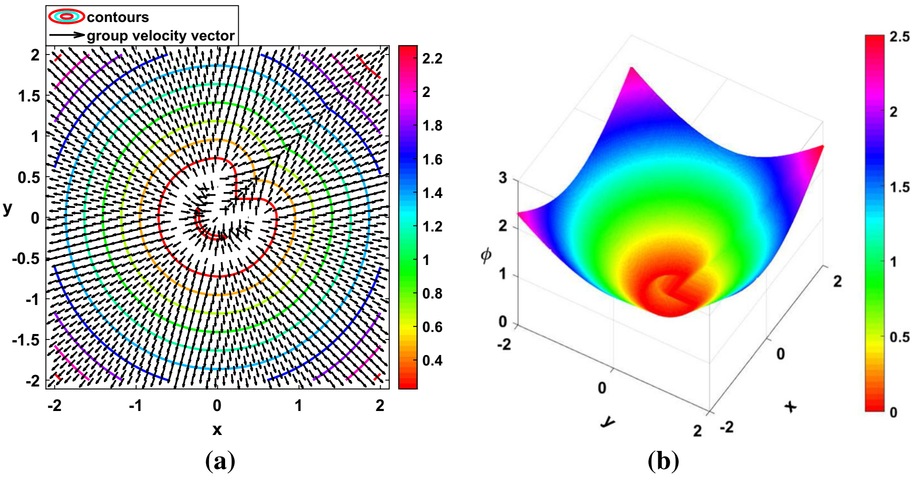

Fig. 5 Example 5, $N = 80$. **a** The group velocity vectors and contours of the numerical solution ϕ. **b** The surface plot of ϕ for the hybrid WENO-ZQ scheme

$\Gamma = \{(\frac{1}{4}, \frac{1}{4}), (\frac{3}{4}, \frac{3}{4}), (\frac{1}{4}, \frac{3}{4}), (\frac{3}{4}, \frac{1}{4}), (\frac{1}{2}, \frac{1}{2})\}$, consisting of five isolated points. The computational domain is $\Omega = [0, 1]^2$. $\phi(x, y) = 0$ is prescribed at the boundary of the unit square. The solution for this problem is the shape function [30]:

Case 1

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

the exact solution for this case is a smooth function

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

Case 2

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

the exact solution for this case is

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

which is continuous but not smooth. Exact solutions are set in a small box with a length $4h$ around these isolated points for both cases.

For *Case 1*, when the mesh size is $N = 80$, the group velocity vectors and contours are shown in Fig. 6a and the surface plot of the numerical solution is shown in Fig. 6c. Numerical errors and orders are shown in Table 7. With the exact solution pre-assigned around the point sources, we can see the fifth order accuracies can be obtained for all schemes. For this example, we would emphasize that, for the WENO-JP scheme, either hybrid or not, the iterative numbers depend on the parameter ϵ, in order to get the desired order. However, for the WENO-ZQ scheme, hybrid or not, we can take a fixed $\epsilon = 10^{-6}$, and fifth order accuracies are obtained. From this case, we can see the WENO-ZQ scheme is more robust than the WENO-JP scheme.

Table 6 Example 5. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ					WENO-JP				
80	1.12e−07	–	1.50e−06	–	42	2.41e−07	–	1.10e−06	–	44
160	5.15e−09	4.45	6.52e−08	4.52	55	8.43e−09	4.83	5.38e−08	4.51	58
320	1.94e−10	4.72	1.14e−09	5.83	72	1.67e−10	5.65	9.95e−10	5.60	73
640	7.09e−12	4.77	1.40e−11	6.35	99	7.00e−12	4.57	1.36e−11	6.19	100
1280	2.40e−13	4.88	4.60e−13	5.22	167	2.40e−13	4.86	4.58e−13	4.89	167
	Hybrid WENO-ZQ					Hybrid WENO-JP				
80	1.12e−07	–	1.50e−06	–	42	1.85e−07	3.99	1.57e−06	–	40
160	5.15e−09	4.45	6.52e−08	4.52	56	5.42e−09	5.09	5.60e−08	4.48	55
320	1.94e−10	4.72	1.14e−09	5.82	70	1.81e−10	4.90	1.08e−09	5.69	71
640	7.09e−12	4.77	1.49e−11	6.52	98	7.07e−12	4.68	1.48e−11	6.19	98
1280	2.40e−13	4.88	4.59e−13	5.02	167	2.40e−13	4.87	4.58e−13	5.01	167

The Lax–Wendroff procedure is used for those points belonging to *Category III*. The errors are measured inside the box $[-1.9, 1.9]^2$ with $x \leq 0$ or $y \leq 0$, and outside the box $[-0.5, 0.5]^2$

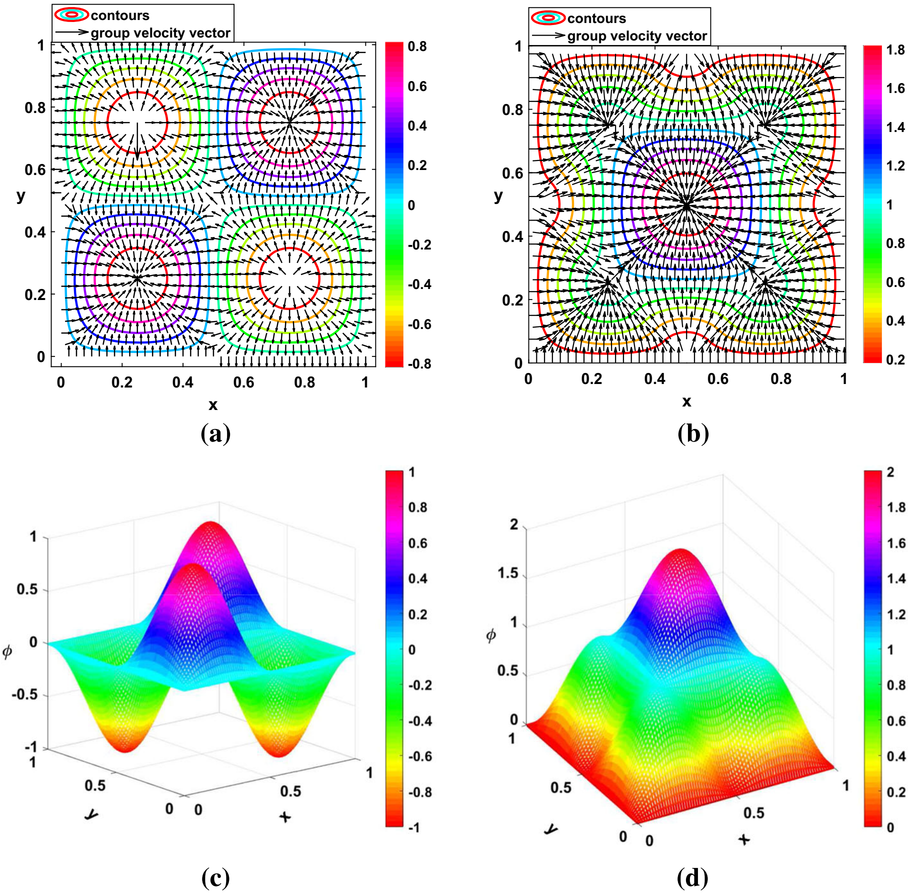

Fig. 6 Example 6, $N = 80$. **a** and **b** are the group velocity vectors and contours of the numerical solution ϕ for *Case 1* and *Case 2*, respectively. **c** and **d** are the surface plots of ϕ from the hybrid WENO-ZQ scheme for *Case 1* and *Case 2*, respectively

For *Case 2*, when the mesh size is $N = 80$, the group velocity vectors and contours are shown in Fig. 6b and the surface plot of the numerical solution is shown in Fig. 6d. The numerical errors and orders are listed in Table 8. We can observe that, due to the non-smoothness of the exact solution, only almost second order accuracy can be obtained. For this example, we can observe that the WENO-JP scheme, hybrid or not, the iterative numbers depend on the parameter ϵ. Furthermore, the WENO-ZQ scheme without hybridization also depends on the parameter ϵ, only the hybrid WENO-ZQ scheme does not. This extreme case shows that the hybrid scheme is the most robust one.

For these two cases, we can find that with variant ϵ's, all four schemes have similar errors and orders. However, for the WENO-JP scheme, we would emphasize that if we take a fixed ϵ, e.g., $\epsilon = 10^{-3}$ or $\epsilon = 10^{-6}$, the WENO-JP scheme may either lose order or even blow up as mesh refinement (we omit the tables here to save space). This shows the great importance to develop a scheme which does not depend on the choice of this artificial parameter. More cases can be seen in the next example. The CPU time for these two cases are also presented in Table 16. Similar CPU time savings are obtained for both cases.

Example 7 We solve the Eikonal equation (2.2) with

Table 7 Example 6 *Case 1*. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes under different ϵ's

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ $\epsilon = 10^{-6}$					WENO-JP $\epsilon = 10^{-3}, 10^{-4}, 10^{-4}, 10^{-6}$				
80	2.63e−07	–	4.86e−06	–	42	4.83e−08	–	1.93e−07	–	36
160	1.05e−09	7.96	5.50e−09	9.78	51	1.97e−09	4.61	7.18e−09	4.75	51
320	3.88e−11	4.77	1.41e−10	5.27	73	4.12e−11	5.58	1.50e−10	5.58	70
640	1.30e−12	4.89	4.83e−12	4.87	114	1.83e−12	4.48	6.39e−12	4.55	108
	Hybrid WENO-ZQ $\epsilon = 10^{-6}$					Hybrid WENO-JP $\epsilon = 10^{-3}, 10^{-4}, 10^{-4}, 10^{-6}$				
80	4.31e−07	–	1.09e−05	–	121	4.03e−08	–	1.62e−07	–	36
160	1.14e−09	8.56	5.50e−09	10.95	50	1.40e−09	4.84	5.03e−09	5.00	50
320	3.84e−11	4.89	1.41e−10	5.28	69	3.91e−11	5.16	1.42e−10	5.14	70
640	1.29e−12	4.89	4.61e−12	4.94	101	1.36e−12	4.84	4.77e−12	4.90	102

Exact values are set on grid points in a small box with length of $4h$ around the isolated points. The errors are measured on the whole computational domain. For the (hybrid) WENO-JP scheme, $\epsilon = 10^{-3}$ if $N = 80$, $\epsilon = 10^{-4}$ if $N = 160$, and so on

Table 8 Example 6 *Case 2*. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ $10^{-3}-10^{-6}$					WENO-JP $10^{-3}-10^{-6}$				
80	1.30e−04	–	1.15e−03	–	41	1.14e−04	–	1.12e−03	–	37
160	4.25e−05	1.61	3.27e−04	1.81	49	3.78e−05	1.60	3.18e−04	1.81	48
320	1.09e−05	1.96	9.02e−05	1.85	69	1.05e−05	1.83	8.79e−05	1.85	67
640	2.77e−06	1.97	2.71e−05	1.73	108	2.79e−06	1.91	2.75e−05	1.67	101
	Hybrid WENO-ZQ 10^{-6}					Hybrid WENO-JP $10^{-3}-10^{-6}$				
80	1.02e−04	–	9.55e−04	–	66	1.02e−04	–	1.03e−03	–	36
160	3.45e−05	1.59	2.61e−04	1.87	49	3.44e−05	1.57	3.14e−04	1.71	47
320	9.88e−06	1.80	8.40e−05	1.63	66	9.82e−06	1.80	8.37e−05	1.90	65
640	2.66e−06	1.89	2.67e−05	1.65	98	2.64e−06	1.89	2.67e−05	1.64	97

The exact values are set on grid points in a small box with length of $4h$ to Γ. The errors are measured on the whole computational domain. $\epsilon = 10^{-3} - 10^{-6}$ means that $\epsilon = 10^{-3}$ if $N = 80$, $\epsilon = 10^{-4}$ if $N = 160$, and so on

$$Case\ (a): \quad f(x, y) = \sqrt{(1-|x|)^2 + (1-|y|^2)};$$
$$Case\ (b): \quad f(x, y) = 2\sqrt{y^2(1-x^2)^2 + x^2(1-y^2)^2}.$$

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

$$Case\ (\text{a}): \ \phi(x, y) = (1-|x|)(1-|y|),$$
$$Case\ (\text{b}): \ \phi(x, y) = (1-x^2)(1-y^2).$$

For *Case* (a), we take $\epsilon = 10^{-14}$ in order to get the exact solution (see [28]). We use the Lax–Wendroff procedure for those points belonging to *Category III*. When the mesh size is $N = 80$, the group velocity vectors and contours are shown in Fig. 7a, and the surface plot of the numerical solution is shown in Fig. 7c. The errors are measured on the whole domain

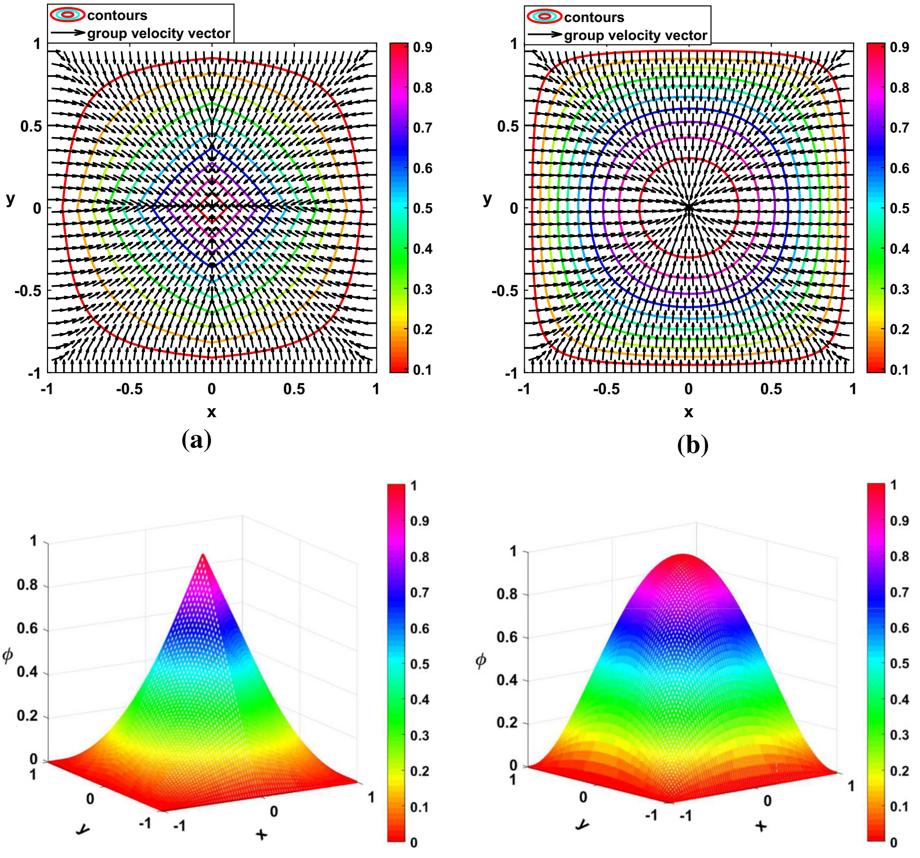

Fig. 7 Example 7, $N = 80$. **a** and **b** is group velocity vectors and contours of the numerical solution ϕ for *Case (a)* and *Case (b)*, respectively. **c** and **d** are the surface plots of ϕ from the hybrid WENO-ZQ scheme for *Case (a)* and *Case (b)*, respectively

and listed in Table 9. Due to the exact solution is a bi-linear polynomial, for high order approximations, we achieve machine error precision within one iteration of four sweepings.

For *Case* (b), we use the Lax–Wendroff procedure for those points belonging to *Category III*. Similarly we set exact values in a small box with length $3h$ around the point $(0, 0)$. The errors are measured on the whole domain. When the mesh size is $N = 80$, the group velocity vectors and contours are shown in Fig. 7b, and the surface plot of the numerical solution is shown in Fig. 7d. The numerical errors and iterative numbers are listed in Table 10. For this case, the (hybrid) WENO-JP scheme depends on the parameter ϵ, while the (hybrid) WENO-ZQ scheme does not, which also show that the WENO-ZQ scheme is more robust. We list the CPU time for *Case* (b) in Table 16, the hybrid WENO-ZQ scheme saves about 55% as compared to WENO-ZQ, while the hybrid WENO-JP scheme saves about 70% as compared to WENO-JP for this case.

Table 9 Example 7 *Case (a)*. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP with $\epsilon = 10^{-14}$

N	L_1 Error	L_∞ Error	Iter	L_1 Error	L_∞ Error	Iter
	WENO-ZQ			WENO-JP		
80	6.28e−17	2.10e−15	1	3.73e−17	3.33e−16	1
160	5.44e−17	4.53e−15	1	3.75e−17	4.44e−16	1
320	7.21e−17	9.96e−15	1	6.13e−17	1.11e−15	1
640	9.80e−17	2.08e−15	1	9.75e−17	9.99e−16	1
	Hybrid WENO-ZQ			Hybrid WENO-JP		
80	6.29e−17	2.10e−15	1	3.73e−17	3.33e−16	1
160	5.46e−17	4.50e−15	1	3.73e−17	4.44e−16	1
320	7.18e−17	9.97e−15	1	6.14e−17	1.11e−15	1
640	9.80e−17	2.08e−15	1	9.75e−17	8.88e−16	1

The Lax–Wendroff procedure is used for those points belonging to *Category III*. The errors are measured on the whole computational domain

Table 10 Example 7 *Case (b)*. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	L_∞ Error	Iter	L_1 Error	L_∞ Error	Iter	ϵ
	WENO-ZQ $\epsilon = 10^{-6}$			WENO-JP			
80	5.15e−15	3.79e−13	36	4.56e−15	2.83e−13	35	10^{-3}
160	3.25e−15	5.25e−13	46	2.66e−15	2.50e−13	44	10^{-4}
320	4.66e−15	4.29e−13	68	2.74e−15	7.81e−13	58	10^{-5}
640	3.85e−15	2.74e−13	98	3.74e−15	2.73e−13	98	10^{-6}
	Hybrid WENO-ZQ $\epsilon = 10^{-6}$			Hybrid WENO-JP			
80	7.21e−15	4.65e−13	37	4.86e−15	2.67e−13	35	10^{-3}
160	5.55e−15	9.58e−14	53	2.65e−15	2.49e−13	44	10^{-4}
320	1.34e−16	1.33e−15	66	4.38e−16	7.98e−14	62	10^{-5}
640	1.15e−16	1.44e−15	94	3.76e−16	2.74e−13	98	10^{-6}

The Lax–Wendroff procedure is used for the outer boundary of the domain. Exact values are set in a small box with length $3h$ around the point $(0, 0)$. The errors are measured on the whole computational domain

Table 11 Example 7 *Case (c)*. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ 10^{-6}					WENO-JP $10^{-2}-10^{-5}$				
80	7.74e−07	–	4.10e−06	–	64	7.39e−07	–	3.99e−06	–	61
160	2.59e−08	4.89	9.49e−08	5.43	78	3.49e−08	4.40	1.35e−07	4.88	72
320	7.65e−10	5.08	2.42e−09	5.29	77	1.18e−09	4.88	3.91e−09	5.11	73
640	2.33e−11	5.03	7.01e−11	5.11	86	3.52e−11	5.07	1.08e−10	5.17	78
	Hybrid WENO-ZQ 10^{-6}					Hybrid WENO-JP $10^{-2}-10^{-5}$				
80	6.69e−07	-	4.10e−06	–	64	7.26e−07	–	3.99e−06	–	61
160	2.12e−08	4.97	8.31e−08	5.62	73	2.41e−08	4.90	9.23e−08	5.43	72
320	6.81e−10	4.96	2.20e−09	5.23	75	7.40e−10	5.03	2.43e−09	5.24	74
640	2.23e−11	4.93	6.78e−11	5.01	79	2.31e−11	4.99	7.11e−11	5.09	77

The errors are measured on the whole computational domain. $\epsilon = 10^{-2} - 10^{-5}$ means $\epsilon = 10^{-2}$ if $N = 80$, $\epsilon = 10^{-3}$ if $N = 160$, and so on

From Example 6 and Example 7-*Case(b)*, we find that the iterative numbers of the WENO-JP scheme are very sensitive to the parameter ϵ, when the solution has point sources. In this case, we further consider an extreme one-dimensional (1D) problem with multiple point sources. We solve the 1D Eikonal equation $|\phi_x| = f(x)$, where $f(x) = 2\pi\sqrt{cos^2(2\pi x)}$. $\phi(x) = 0$ is prescribed at those point sources inside the computational domain:

Case (c): the computational domain is $[0, 2]$, and $\Gamma = \{\frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1, \frac{5}{4}, \frac{3}{2}, \frac{7}{4}\}$, and

$$g\left(\frac{1}{4}\right) = g\left(\frac{5}{4}\right) = 1, \quad g\left(\frac{3}{4}\right) = g\left(\frac{7}{4}\right) = -1, \quad g\left(\frac{1}{2}\right) = g(1) = g\left(\frac{3}{2}\right) = 0.$$

The fifth order Richardson procedure is used for those points belonging to *Category III*. The numerical results are presented in Table 11. For this 1D problem, if we use the WENO-JP scheme with $\epsilon = 10^{-6}$, the iteration does not converge with mesh sizes less than $N = 640$. If we use the hybrid WENO-JP FSM with $\epsilon = 10^{-6}$, although the iteration converges, only third order accuracies are obtained. The WENO-ZQ scheme and its hybrid one show to be more robust.

Example 8 (Travel-time problem in elastic wave propagation) The quasi-P and the quasi-SV slowness surfaces are defined as follows [17]

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

where

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

and $a_{i,j}$'s 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

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

Table 12 Example 8, quasi-P wave. Comparison of iterative numbers and CPU cost (in seconds) for three different initial choices, *Case (i-iii)*. “ratio2” represents the CPU cost of *Case (ii)* over *Case (i)*, and “ratio3” for *Case (iii)* over *Case (i)*

N	*Case (i)*		*Case (ii)*			*Case (iii)*		
	Iter	Time	Iter	Time	Ratio2	Iter	Time	Ratio3 (%)
WENO-JP								
80	1197	20.69	52	1.38	6.71%	52	1.79	8.69
160	2315	141.23	69	6.46	4.57%	69	8.21	5.81
320	4612	1092.50	105	38.46	3.52%	105	48.67	4.45
640	8198	7828.10	175	260.28	3.32%	175	320.41	4.09
WENO-ZQ								
80	323	7.48	54	1.72	23.07%	54	2.17	29.07
160	622	52.58	70	8.15	15.51%	70	10.11	19.23
320	1191	393.80	105	47.77	12.13%	105	57.79	14.67
640	2429	3216.80	173	320.72	9.97%	173	379.03	11.78

Table 13 Example 8, quasi-SV wave. Comparison of iterative numbers and CPU cost (in seconds) for three different initial choices, *Case (i-iii)*. “ratio2” represents the CPU cost of *Case (ii)* over *Case (i)*, and “ratio3” for *Case (iii)* over *Case (i)*

N	*Case (i)*		*Case (ii)*			*Case (iii)*		
	Iter	Time	Iter	Time	Ratio2	Iter	Time	Ratio3
WENO-JP								
80	10000	–	50	1.4571	–	50	1.7329	–
160	10000	–	71	6.8315	–	71	8.3846	–
320	2830	672.49	108	39.647	5.89%	108	50.893	7.56%
640	5804	5559	181	265.09	4.76%	181	333.94	6.00%
WENO-ZQ								
80	350	8.1	105	2.98	36.79%	105	4.03	49.79%
160	646	54.25	162	15.63	28.82%	162	18.13	33.42%
320	1052	343.49	109	47.89	13.94%	109	59.94	17.45%
640	2085	2737.1	179	319.27	11.66%	179	390.35	14.26%

The quasi-SV 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 nonconvex HJ equation, and the elastic parameters are taken to be

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

The computational domain is $\Omega = [-1, 1]^2$, and the inflow boundary is $\Gamma = (0, 0)$. Exact values are assigned in a box with length 0.3, which includes the source point. For this problem, the Lax–Friedrichs (LF) numerical Hamiltonian is used.

For this example with the LF numerical Hamiltonian, it is important on how to choose the initial values to start the iteration, in order to result low computational cost. We first study the

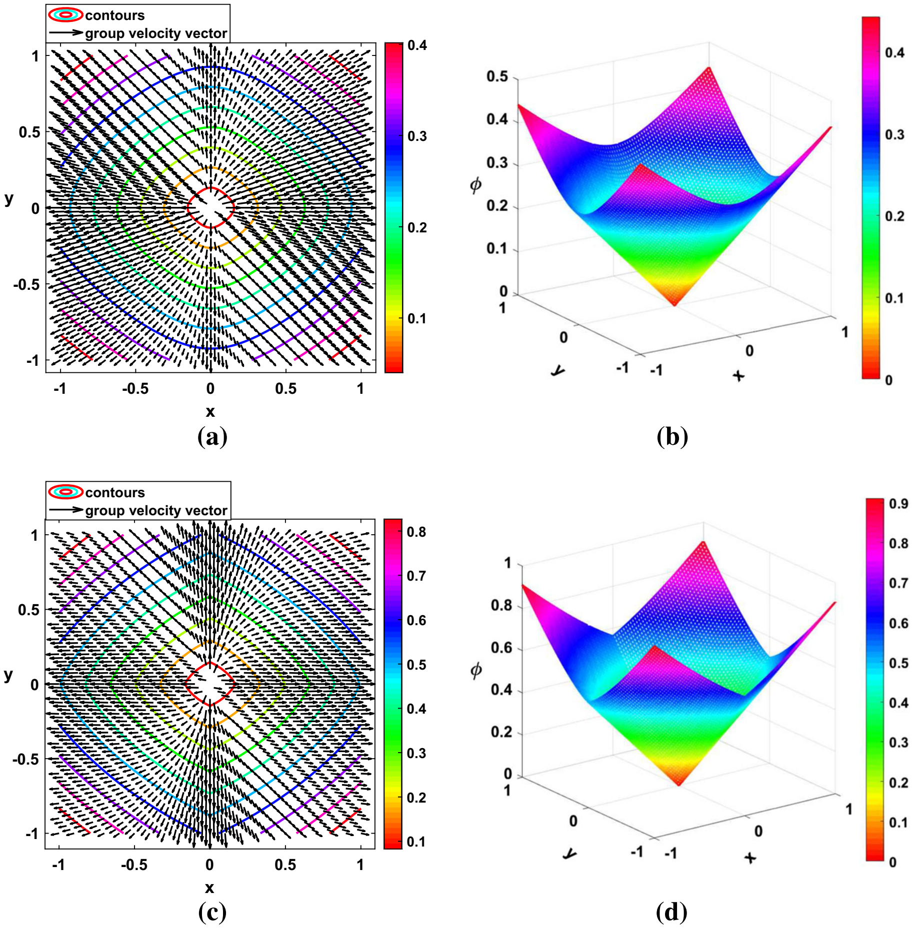

Fig. 8 Example 8, $N = 80$. **a** and **b** are the group velocity vectors, contours and the surface plot of the numerical solution for the hybrid WENO-ZQ scheme with quasi-P wave. **c** and **d** are the group velocity vectors, contours and the surface plot of the numerical solution for the hybrid WENO-ZQ scheme with quasi-SV wave

computational cost and iterative numbers for three different choices of initial values: *Case (i)* big enough values such as 100; *Case (ii)* the corresponding first order method with the convergence threshold $\delta = 10^{-1}$; *Case (iii)* the corresponding first order method with the convergence threshold $\delta = 10^{-10}$. *Case (ii)* means an incomplete first order initial guesses. In Table 12 and Table 13, we show the CPU cost as well as the iterative numbers from mesh refinements. The comparison shows that the first order initial guess is better than big values, and *Case (ii)* with an incomplete convergence could save more. Besides, with big initial values, WENO-ZQ has much less iterative numbers than WENO-JP, which shows that WENO-ZQ is more robust. In the following, we will take *Case (ii)* as the initial guess, and we will show it is also effective to obtain fifth order accuracy for the fifth order method.

For quasi-P wave, in Fig. 8a, b, we display the group velocity vectors, contours and the surface plot of the numerical solution on the mesh $N = 80$. Numerical errors and orders for four schemes are presented in Table 14, we can see the errors and orders are very close. For this

Table 14 Example 8, quasi-P wave. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ					WENO-JP				
80	1.18e−06	–	9.86e−06	–	54	1.65e−06	–	2.05e−05	–	52
160	4.46e−08	4.73	3.96e−07	4.63	70	5.49e−08	4.91	6.85e−07	4.90	69
320	1.46e−09	4.92	1.30e−08	4.92	105	1.51e−09	5.18	1.44e−08	5.57	105
640	4.65e−11	4.98	4.14e−10	4.98	172	4.66e−11	5.01	4.17e−10	5.10	175
	hybrid WENO-ZQ					hybrid WENO-JP				
80	1.18e−06	–	9.86e−06	–	54	1.46e−06	–	1.60e−05	–	51
160	4.46e−08	4.73	3.96e−07	4.63	69	4.96e−08	4.93	5.28e−07	4.89	70
320	1.46e−09	4.92	1.30e−08	4.92	104	1.48e−09	5.19	1.34e−08	5.57	104
640	4.65e−11	4.98	4.14e−10	4.98	172	4.66e−11	5.00	4.15e−10	5.11	174

The exact values are assigned in a box with length 0.3 which includes the source point. The errors are measured in the box $[-0.95, 0.95]^2$

Table 15 Example 8, quasi-SV wave. Comparison between (hybrid) WENO-ZQ and (hybrid) WENO-JP schemes

N	L_1 Error	Order	L_∞ Error	Order	Iter	L_1 Error	Order	L_∞ Error	Order	Iter
	WENO-ZQ					WENO-JP				
80	3.24e−06	–	1.11e−04	–	105	1.34e−06	–	2.11e−05	–	50
160	6.08e−08	5.73	2.66e−06	5.38	162	2.09e−08	6.00	8.27e−07	4.67	71
320	2.41e−10	7.97	4.42e−08	5.91	109	1.82e−10	6.83	1.24e−08	6.05	108
640	4.10e−12	5.86	1.19e−10	8.53	179	4.11e−12	5.47	1.18e−10	6.71	180
1280	1.26e−13	5.01	3.68e−12	5.02	325	1.26e−13	5.02	3.68e−12	5.00	328
	hybrid WENO-ZQ					hybrid WENO-JP				
80	3.23e−06	–	1.10e−04	–	94	1.10e−06	-	2.54e−05	–	51
160	6.08e−08	5.73	2.67e−06	5.34	167	1.42e−08	6.27	5.12e−07	5.63	71
320	2.41e−10	7.97	4.42e−08	5.98	107	1.58e−10	6.48	1.47e−08	5.11	108
640	4.10e−12	5.88	1.19e−10	8.53	177	4.10e−12	5.27	1.18e−10	6.96	180
1280	1.20e−13	5.07	3.67e−12	5.00	323	1.26e−13	5.01	3.68e−12	5.00	328

The exact values are assigned in a box with length 0.3 which includes the source point. The errors are measured in the region away from two singular lines of $x = 0$ and $y = 0$

Table 16 The total CPU cost (in seconds) for examples 1–8 with four schemes.

Example	WENO-ZQ	h-WENO-ZQ	ratio-ZQ	WENO-JP	h-WENO-JP	ratio-JP
1	66.21	33.00	49.84%	63.72	38.60	60.57%
2	43.68	26.16	59.89%	42.68	27.79	65.11%
3	333.00	191.40	57.47%	351.69	193.39	54.99%
4-2D	49.71	29.42	59.18%	49.82	28.76	57.72%
4-3D	21599	11800	54.63%	22553	14032	62.21%
5	368.46	198.08	53.75%	341.23	204.99	60.07%
6-1	63.40	30.64	48.32%	97.17	35.78	36.82%
6-2	59.77	29.72	49.72%	95.26	33.07	34.71%
7-b	47.73	21.46	44.96%	82.65	27.35	33.09%
8-p	288.96	246.00	85.13%	284.27	248.77	87.51%
8-sv	1986.80	1771.70	89.17%	1969.9	1731.8	87.91%

The "ratio-ZQ" or "ratio-JP" denotes the total CPU cost of the hybrid scheme over the original one, for WENO-ZQ and WENO-JP respectively

example with the LF numerical Hamiltonian, we can see that all schemes give the expected fifth order accuracies, except the iterative numbers are larger than the Godunov numerical Hamiltonian for the Eikonal equation. The CPU time is shown in Table 16. For this type of numerical Hamiltonian, the hybrid approach seems to save not that much computational cost.

For quasi-SV wave, in Fig. 8c, d, we display the group velocity vectors, contours and the surface plot of the numerical solution on the mesh $N = 80$. Numerical errors and orders for four schemes are presented in Table 15, we can see the errors and orders are also similar, and the fifth order accuracies are all obtained. The CPU time is shown in Table 16, for this non-convex case, the CPU time is greatly increased and hybrid approach does not save too much computational cost too.

Example 9 (*The Marmousi Model*) This model is designed to compare different velocity estimation methods behind seismic data acquisition and processing [25]. It is based on a complex synthetic 2D acoustic data set, namely, the Marmousi data set, which involves strong horizontal and vertical velocity changes. In this example, we will apply our method to the Marmousi model using both a point source and a plane-wave source as in [3].

For our fifth order method, the fifth order Richardson extrapolation is used for those points belonging to *Category III*, on which the source f is obtained by high order interpolation on a refined mesh. We only present the numerical results of the hybrid WENO-ZQ scheme for this example. In Figs. 9 and 10, we show the numerical results on the mesh 231×76, for the point source and the plane-wave source, respectively. We compute two reference solutions on the mesh 921×301. We compare the solutions of the fifth order scheme and the first order scheme, to the reference solution. As we can see that the fifth order results on a coarser mesh are very close to the results on a much finer mesh. In order to clearly see the differences, we present the absolute errors between the reference solution and the numerical solutions in Fig. 11 for the point source and Fig. 12 for the plane-wave source. We can see that the fifth-order numerical results are obviously better than the first-order numerical results. As concerning to the CPU cost, for the fifth order scheme, it takes about 112 iterations and CPU time $4.74s$ for the point source, while 72 iterations and CPU time $1.96s$ for the plane-wave source.

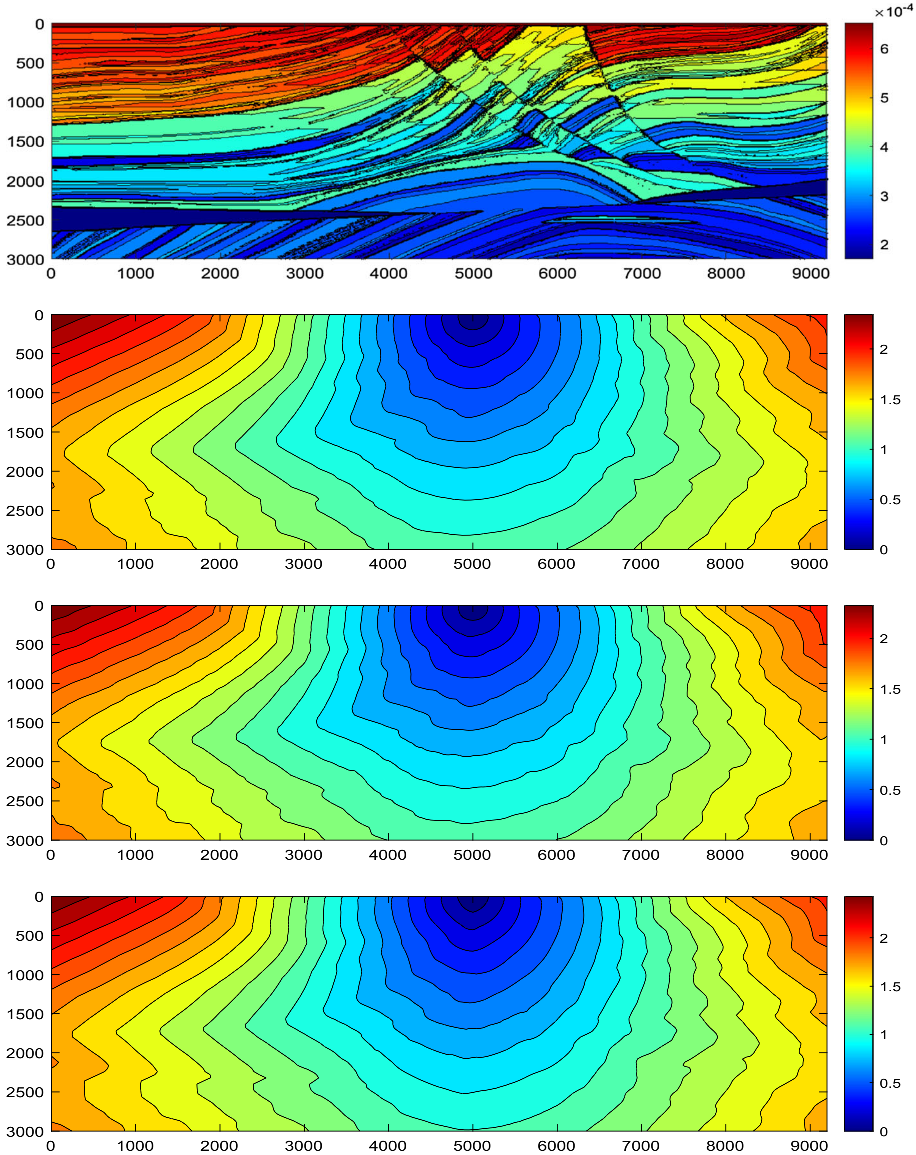

Fig. 9 The Marmousi model with a point source. From top to bottom: the slowness field of Marmousi model on 921×301; the 5th order reference solution on the mesh 921×301; the fifth order result on the mesh 231×76; the first order result on the mesh 231×76

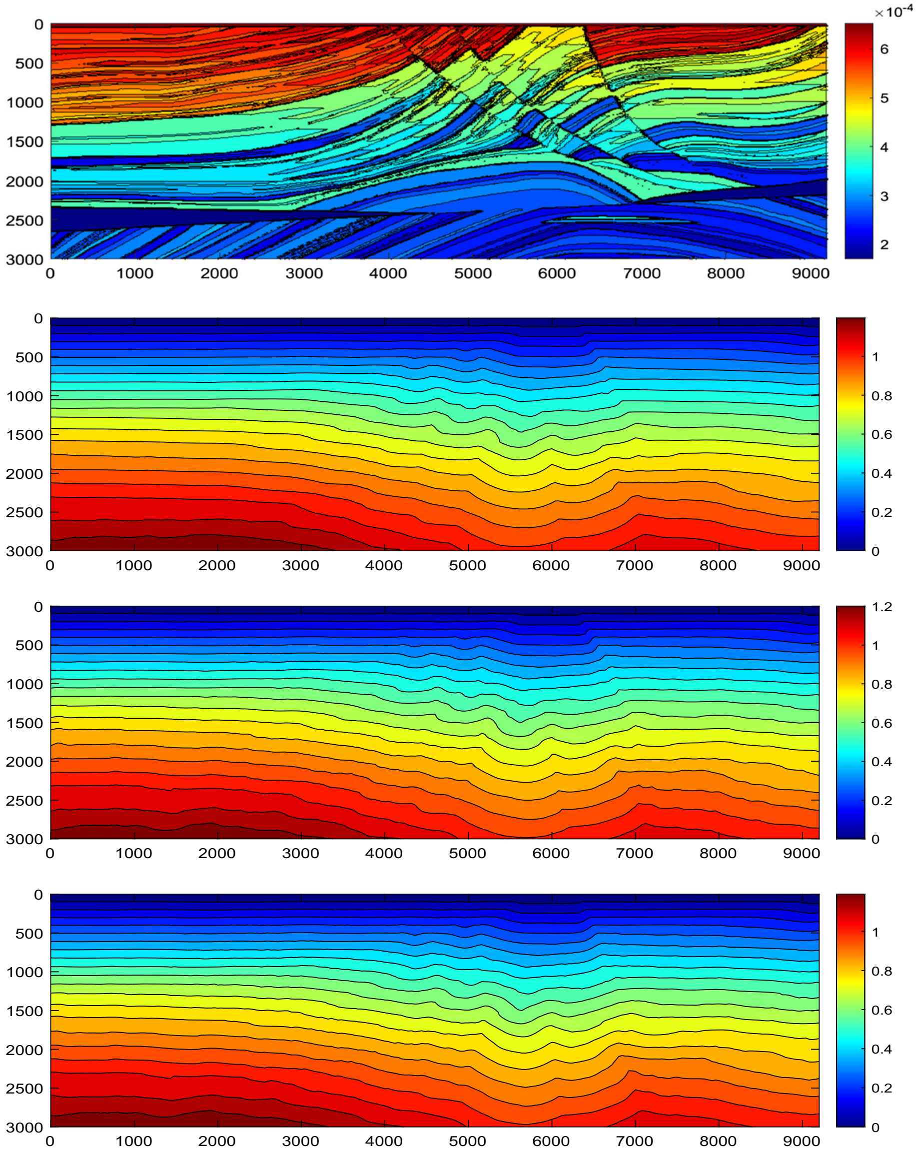

Fig. 10 The Marmousi model with a plane wave source. From top to bottom: the slowness field of Marmousi model on 921×301; the 5th order reference solution on the mesh 921×301; the fifth order result on the mesh 231×76; the first order result on the mesh 231×76

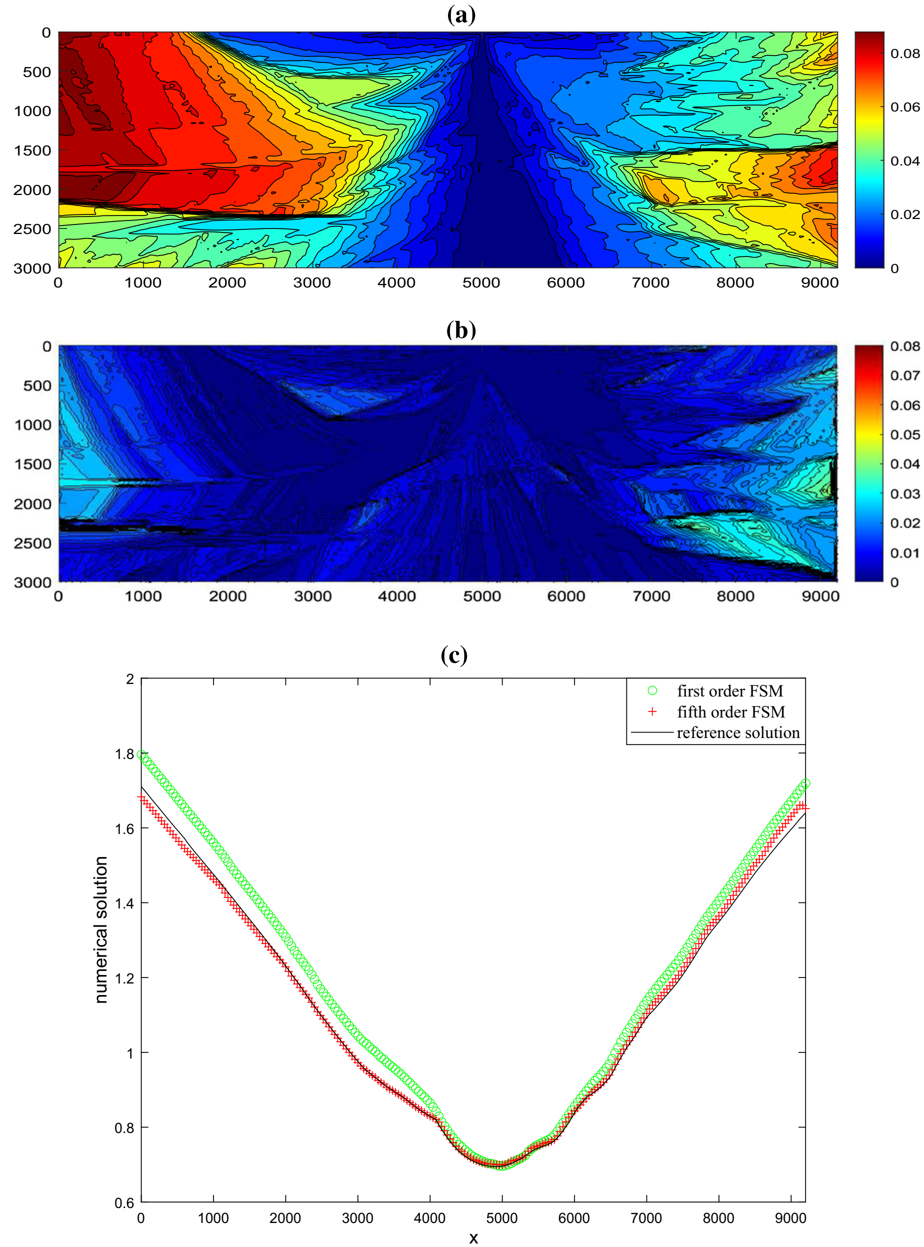

Fig. 11 The absolute errors between the numerical solutions and the reference solution, for the point source. **a** the first order; **b** the fifth order; **c** cutting plot along $y = 1600$

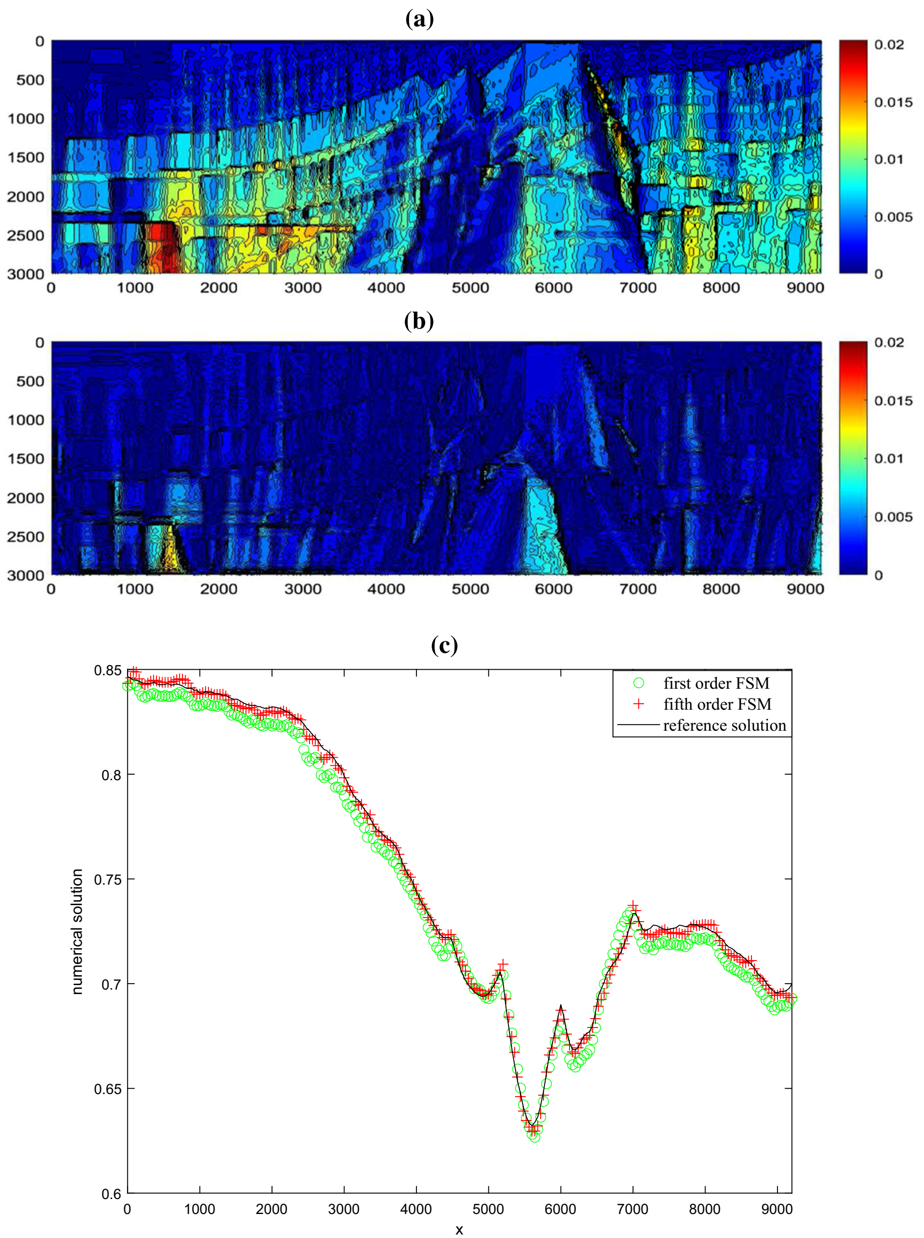

Fig. 12 The absolute errors between the numerical solutions and the reference solution, for the plane wave. **a** the first order; **b** the fifth order; **c** cutting plot along $y = 1600$

4 Concluding Remark

In this work, we have combined a fifth order finite difference WENO-ZQ scheme with a high order fast sweeping method, to develop a new fifth order WENO-ZQ fast sweeping scheme for directly solving static Hamilton–Jacobi equations. Due to the unequal stencils in the hybrid WENO-ZQ scheme, it can alleviate the dependence of iterative numbers on the parameter ϵ which the fifth order WENO-JP FSM does. Furthermore, a hybrid scheme is proposed, which on one aspect saves much more computational cost, on the other it is more robust. Numerical results have demonstrated the effectiveness of our proposed approach. For the Godunov type numerical Hamiltonian solving the Eikonal equation, the hybrid scheme can save about half of the computational cost. For the Lax–Friedrichs type numerical Hamiltonian solving general static Hamilton–Jacobi equations, the savings are not significant.

References

1. Boué, M., Dupuis, P.: Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control. SIAM J. Numer. Anal. **36**(3), 667–695 (1999)
2. Crandall, M.G., Lions, P.-L.: Viscosity solutions of Hamilton–Jacobi equations. Trans. Am. Math. Soc. **277**(1), 1–42 (1983)
3. Fomel, S., Luo, S., Zhao, H.K.: Fast sweeping method for the factored eikonal equation. J. Comput. Phys. **228**, 6440–6455 (2009)
4. Helmsen, J., Puckett, E., Colella, P., Dorr, M.: Two new methods for simulating photolithography development in 3D. Proc. SPIE **2726**, 253–262 (1996)
5. Huang, L., Shu, C.-W., Zhang, M.P.: Numerical boundary conditions for the fast sweeping high order WENO methods for solving the Eikonal equation. J. Comput. Math. **26**(3), 336–346 (2008)
6. Huang, L., Wong, S.C., Zhang, M., Shu, C.-W., Lam, W.H.K.: Revisiting Hughes' dynamic continuum model for pedestrian flow and the development of an efficient solution algorithm. Transp. Res. B-Meth. **43**(1), 127–141 (2009)
7. Jiang, G.S., Peng, D.P.: Weighted ENO schemes for Hamilton–Jacobi equations. SIAM J. Sci. Comput **21**(6), 2126–2143 (2000)
8. Jiang, G.S., Shu, C.-W.: Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**(1), 202–228 (1996)
9. Kao, C.-Y., Osher, S., Qian, J.: Lax–Friedrichs sweeping scheme for static Hamilton–Jacobi equations. J. Comput. Phys. **196**(1), 367–391 (2004)
10. Kao, C.-Y., Osher, S., Tsai, Y.H.: Fast sweeping methods for static Hamilton–Jacobi equations. SIAM J. Numer. Anal. **42**(6), 2612–2632 (2005)
11. Levy, D., Puppo, G., Russo, G.: Central WENO schemes for hyperbolic systems of conservation laws, M2AN. Math. Model. Numer. Anal. **33**(3), 547–571 (1999)
12. Levy, D., Puppo, G., Russo, G.: Compact central WENO schemes for multidimensional conservation laws. SIAM J. Sci. Comput. **22**(2), 656–672 (2000)
13. Li, F., Shu, C.-W., Zhang, Y.-T., Zhao, H.: Second order discontinuous Galerkin fast sweeping method for Eikonal equations. J. Comput. Phys. **227**(17), 8191–8208 (2008)
14. Luo, S.: A uniformly second order fast sweeping method for Eikonal equations. J. Comput. Phys. **241**(10), 104–117 (2013)
15. Lin, J., Abgrall, R., Qiu, J.: High order residual distribution for steady state problems for hyperbolic conservation laws. J. Sci. Comput **79**(2), 891–913 (2019)
16. Osher, S., Shu, C.-W.: High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations. SIAM J. Numer. Anal. **28**(4), 907–922 (1991)
17. Qian, J., Cheng, L.T., Osher, S.: A level set based Eulerian approach for anisotropic wave propagations. Wave. Motion. **37**(4), 365–379 (2003)
18. Qian, J., Zhang, Y.-T., Zhao, H.-K.: A fast sweeping method for static convex Hamilton–Jacobi equations. J. Sci. Comput. **31**(1), 237–271 (2007)
19. Serna, S., Qian, J.: A stopping criterion for higher-order sweeping schemes for static Hamilton–Jacobi equations. J. Comput. Math. **28**(4), 552–568 (2010)

20. Sethian, J.A.: A fast marching level set method for monotonically advancing fronts. Proc. Nat. Acad. Sci. **93**(4), 1591–1595 (1996)
21. Shu, C.-W.: High order numerical methods for time dependent Hamilton–Jacobi equations. Math. Comput. Imaging Sci. Inf. Process. (2007)
22. Tan, S., Shu, C.-W.: Inverse Lax–Wendroff procedure for numerical boundary conditions of conservation laws. J. Comput. Phys. **229**(21), 8144–8166 (2010)
23. Tsai, R., Cheng, L.T., Osher, S., Zhao, H.-K.: Fast sweeping algorithms for a class of Hamilton–Jacobi equations. SIAM J. Numer. Anal. **41**(2), 673–694 (2003)
24. Tsitsiklis, J.N.: Efficient algorithms for globally optimal trajectories. IEEE Trans. Autom. Contr. **40**(9), 1528–1538 (1995)
25. Versteeg, R.: The Marmousi experience: velocity model determination on a synthetic complex data set. Lead. Edge **13**(09), 927–936 (1994)
26. Wu, L., Zhang, Y.-T.: A third order fast sweeping method with linear computational complexity for Eikonal equations. J. Sci. Comput. **62**(1), 198–229 (2015)
27. Xia, Y., Wong, S.C., Zhang, M., Shu, C.-W., Lam, W.H.K.: An efficient discontinuous Galerkin method on triangular meshes for a pedestrian flow model. Int. J. Numer. Meth. Eng. **76**(3), 337–350 (2008)
28. Xiong, T., Zhang, M.P., Zhang, Y.-T., Shu, C.-W.: Fast sweeping fifth order WENO scheme for static Hamilton–Jacobi equations with accurate boundary treatment. J. Sci. Comput. **45**(1–3), 514–536 (2010)
29. Zhang, Y.-T., Chen, S., Li, F., Zhao, H.-K., Shu, C.-W.: Uniformly accurate discontinuous Galerkin fast sweeping methods for Eikonal equations. SIAM J. Sci. Comput. **33**(4), 1873–1896 (2011)
30. Zhang, Y.-T., Zhao, H.-K., Qian, J.: High Order fast sweeping methods for static Hamilton–Jacobi equations. J. Sci. Comput. **29**(1), 25–56 (2006)
31. Zhao, H.-K.: A fast sweeping method for Eikonal equations. Math. Comput. **74**(250), 603–627 (2005)
32. Zhao, H.-K., Osher, S., Merriman, B., Kang, M.: Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. Comput. Vis. Image. Und. **80**(3), 295–314 (2000)
33. Zhao, Z., Zhu, J., Chen, Y., Qiu, J.: A new hybrid WENO scheme for hyperbolic conservation laws. Comput. Fluids. **179**, 422–436 (2019)
34. Zhu, J., Qiu, J.: A new fifth order finite difference WENO scheme for Hamilton–Jacobi equations. Numer. Meth. Part. D. E. **33**(4), 1095–1113 (2017)
35. Zhu, J., Qiu, J.: A new fifth order finite difference WENO scheme for solving hyperbolic conservation laws. J. Comput. Phys. **318**, 110–121 (2016)
36. Zhu, J., Qiu, J.: A new type of finite volume WENO schemes for hyperbolic conservation laws **73**(2–3), 1338–1359 (2017)