Journal of Scientific Computing, Vol. 29, No. 1, October 2006 (© 2005)
DOI: 10.1007/s10915-005-9014-3

High Order Fast Sweeping Methods for Static Hamilton–Jacobi Equations

Yong-Tao Zhang,[1] Hong-Kai Zhao,[1] and Jianliang Qian[2]

Received June 3, 2004; accepted (in revised form) May 5, 2005; Published online December 28, 2005

We construct high order fast sweeping numerical methods for computing viscosity solutions of static Hamilton–Jacobi equations on rectangular grids. These methods combine high order weighted essentially non-oscillatory (WENO) approximations to derivatives, monotone numerical Hamiltonians and Gauss–Seidel iterations with alternating-direction sweepings. Based on well-developed first order sweeping methods, we design a novel approach to incorporate high order approximations to derivatives into numerical Hamiltonians such that the resulting numerical schemes are formally high order accurate and inherit the fast convergence from the alternating sweeping strategy. Extensive numerical examples verify efficiency, convergence and high order accuracy of the new methods.

KEY WORDS: fast sweeping methods; WENO approximation; high order accuracy; static Hamilton–Jacobi equations; Eikonal equations.

1. INTRODUCTION

We consider the static Hamilton–Jacobi equations

$$\begin{cases} H(\phi_{x_1}, \ldots, \phi_{x_d}, x) = 0, & x \in \Omega \setminus \Gamma, \\ \phi(x) = g(x), & x \in \Gamma \subset \Omega, \end{cases} \tag{1.1}$$

where Ω is a computational domain in R^d and Γ is a subset of Ω. The Hamiltonian H is a nonlinear Lipschitz continuous function. Such Hamilton–Jacobi (H–J) equations appear in many applications, such as optimal

[1]Department of Mathematics, University of California, Irvine, CA 92697-3875, USA. E-mail: {zyt, zhao}@math.uci.edu
[2]Department of Mathematics, University of California, Los Angeles, CA 90095-1555, USA. E-mail: qian@math.ucla.edu

control, differential games, image processing and computer vision, and geometric optics.

A very important member of the family of the static Hamilton–Jacobi equations is the Eikonal equation. The standard isotropic Eikonal equation is

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

where $f(x)$ is a positive function.

Since the boundary value problems (1.1), (1.2) are nonlinear first order partial differential equations, we may apply the classical method of characteristics to solve these equations in phase space; namely, consider the gradient components as independent variables and solve ODE systems to follow the propagation of characteristics. Although the characteristics may never intersect in phase space, their projection into physical space may intersect so that the solution in physical space is not uniquely defined at these intersections. By mimicking the entropy condition for hyperbolic conservation laws to pick out a physically relevant solution, Crandall and Lions [7] introduced the concept of viscosity solutions for Hamilton–Jacobi equations so that a global, physically relevant solution can be defined for such first order nonlinear equations. Moreover, monotone finite difference schemes are developed to compute such viscosity solutions stably.

There are mainly two classes of numerical methods for solving static Hamilton–Jacobi equations. The first class of numerical methods is based on reformulating the equations into suitable time-dependent problems. Osher [22] provides a natural link between static and time-dependent Hamilton–Jacobi equations by using the level-set idea and thus raising the problem one-dimensional higher. The zero-level set of the viscosity solution ψ of the time-dependent H–J equation

$$\psi_t + H(\psi_{x_1}, \ldots, \psi_{x_d}, x) = 0 \tag{1.3}$$

at time t is the set of x such that $\phi(x) = t$ of (1.1), where the Hamiltonian H is homogeneous of degree one. In the control framework, a semi-Lagrangian scheme is obtained for Hamilton–Jacobi equations by discretizing in time the dynamic programming principle [9, 10]. Another approach to obtaining a "time" dependent H–J equation from the static H–J equation is using the so called paraxial formulation in which a preferred spatial direction is assumed in the characteristic propagation [8, 11, 17, 27, 28]. High order numerical schemes are well developed for the time dependent H–J equation (1.3) on structured and unstructured meshes

[1,2,4–6,13,14,16,20,23–25,39]; see a recent review on high order numerical methods for time dependent H–J equations by Shu [35]. Due to the finite speed of propagation and the CFL condition for the discrete time step size, the number of time steps has to be of the same order as that for one of the spatial dimensions so that the solution converges in the entire domain.

The other class of numerical methods for static H–J equations is to treat the problem as a stationary boundary value problem: discretize the problem into a system of nonlinear equations and design an efficient numerical algorithm to solve the system. Among such methods are the fast marching method and the fast sweeping method. In the fast marching method [12,32–34,38], the solution is updated by following the causality in a sequential way; i.e., the solution is updated pointwise in the order that the solution is strictly increasing (decreasing); hence two essential ingredients are needed in the algorithm: an upwind difference scheme and a heap-sort algorithm. The resulting complexity of the fast marching method is of order $O(N \log N)$ for N grid points, where the $\log N$ factor comes from the heap-sort algorithm. In the fast sweeping method [3, 18,19,37,40,41], Gauss–Seidel iterations with alternating direction sweepings are incorporated into upwind finite differences. In contrast to the fast marching method, the fast sweeping method follows the causality along characteristics in a parallel way; i.e., all characteristics are divided into a finite number of groups according to their directions and each Gauss–Seidel iteration with a specific sweeping ordering covers a group of characteristics simultaneously; no heap-sort is needed. The fast sweeping method is optimal in the sense that a finite number of iterations is needed [40], so that the complexity of the algorithm is $O(N)$ for a total of N grid points; i.e., the number of iterations is independent of the grid size. The algorithm is extremely simple to implement.

Both the fast marching method and the fast sweeping method are extremely efficient algorithms for solving static Hamilton–Jacobi equations. In fact both of them solve the same system of discretized equations and try to order the nonlinear system of equations following the causality along characteristics. Here we point out a few interesting differences between these two methods. For the complexity, the fast marching method is $O(N \log N)$ and the fast sweeping method is $O(N)$. However the constant in the complexity for the fast marching method does not depend on the equation while the constant in the complexity for the fast sweeping method depends on the equation (but not on the grid size) [40]. On a given discretization which method is faster depends on the equation. In practice, they are more or less comparable. As to the ordering philosophy, these two methods are different. The fast marching method sorts out the ordering on the fly using the heap-sort algorithm; a strict causality

principle has to exist and to be enforced to order the equations; i.e., equations solved later has no influence on the previous equations. On the other hand, the fast sweeping method is in the general framework of iterative methods. If a strict causality exists (such as in Eikonal equations), it can be enforced in the iterative procedure, which guarantees convergence in a finite number of alternating sweepings. Therefore, the iterative framework is more robust and flexible for general equations and high order methods. With alternating ordering, the iterations can follow causality along characteristics, and handle global relaxation and smoothing effects as well. For examples, even if there is no strict causality, such as when the Hamiltonian is not convex, or there are viscosity terms (e.g., the solutions are coupled globally), or the numerical method (e.g., high order method) is not monotone, it is still possible that the iteration will converge. Another interesting remark is on the convergence mechanism for iterative methods. In general, the convergence of iterative methods is due to a certain contraction property as in the fixed point iteration. Infinite number of iterations are needed for full convergence. When a strict causality exists and is enforced in the fast sweeping method as in Eikonal equations, the convergence is achieved in a finite number of iterations due to the alternating sweeping order. In general, different sweeping orders will not change the contraction property of an iterative method. So both mechanisms can be in effect in more general situations.

In many applications, the numerical solutions from H–J equations are used to compute other quantities and thus their numerical derivatives are needed as well; for example, in geometrical optics the derivatives of travel-times are used to compute amplitudes [29]. In this paper, we develop a quite general framework for constructing high order fast sweeping methods. We adapt high order schemes for time dependent Hamilton–Jacobi equations in [14, 24, 39] to the static H–J equations in a novel way. First-order sweeping schemes are used as building blocks in our high order methods. The high order accuracy in our schemes results from the high order approximations for the partial derivatives because the monotone numerical Hamiltonians are Lipschitz continuous and consistent with the Hamiltonian H in the PDEs. In particular, we use the WENO approximations [14, 15], since the weighted essentially nonoscillatory (WENO) approximations have uniform high order accuracy, and are more robust and efficient than other schemes such as ENO schemes.

The algorithm is developed in Section 2. First we describe the algorithm for the Eikonal equation, then we extend it to the general static Hamilton–Jacobi equations. In Section 3, extensive numerical experiments are performed to demonstrate accuracy and fast convergence of the algorithms. High order accuracy in smooth regions and good resolution

of derivative singularities are observed. Concluding remarks are given in Section 4.

2. HIGH ORDER SWEEPING METHODS

We consider the two dimensional problems for simplicity. The extension to higher dimensions is straightforward.

2.1. Principles for Constructing Sweeping Methods

To compute the viscosity solution for Eq. (1.1), we first discretize the domain Ω. Suppose that a rectangular mesh Ω_h covers the computational domain Ω. Let (i, j) denote a grid point in Ω_h, i.e., $\Omega_h = \{(i, j), 1 \leqslant i \leqslant I, 1 \leqslant j \leqslant J\}$, and $\phi_{i,j}$ denote the numerical solution at the grid point (i, j). h_x and h_y denote uniform grid sizes in the x-direction and the y-direction, respectively; to simplify the notation, we take $h_x = h_y = h$.

Next we discretize the Hamiltonian H by a monotone numerical Hamiltonian $\hat{H}$ [24]:

$$\begin{cases} \hat{H}(\phi_x^-, \phi_x^+; \phi_y^-, \phi_y^+)_{ij} = 0, & (i, j) \in \Omega_h \setminus \Gamma_h, \\ \phi_{ij} = g_{ij}, & (i, j) \in \Gamma_h \subset \Omega_h. \end{cases} \tag{2.1}$$

Since the H–J equation is in general nonlinear, we end up with a coupled nonlinear system for unknown solutions at those mesh points. Because the boundary values are usually given on irregular geometries, it is very involved to solve the nonlinear system globally. Therefore, we appeal to construct an iterative method based to local solvers. An essential ingredient for constructing a sweeping method is an efficient local solver which expresses the value at the standing mesh point in terms of its neighboring values. If the Godunov's Hamiltonian is the numerical Hamiltonian in Eq. (2.1), then it is straightforward to carry out the local solution procedure for the Eikonal equation (1.2); in fact, the popularity of the Fast Marching method [32, 38] and Fast Sweeping method [40] is more or less due to such efficient local solvers. Furthermore, if the Godunov's Hamiltonian is the numerical Hamiltonian in Eq. (2.1), then it is also possible to carry out the local solution procedure for H–J equations (1.2) with convexity in the gradient component of Hamiltonians; see [37]. But if the Hamiltonian is nonconvex in the gradient component, then it is extremely involved to carry out the minmax optimization resulting from the Godunov monotone Hamiltonian. On the other hand, if the Lax–Friedrichs Hamiltonian is utilized, then it is extremely simple to carry out the local solution process, no matter whether the Hamiltonian is convex

or not, no matter how complicated the Hamiltonian could be; see [18]. In passing, we note that it is also possible to use the Roe-entropy fix type numerical Hamiltonian to discretize the H–J equation, and it is expected that the resulting scheme will interpolate the behavior of Godunov and Lax–Friedrichs schemes.

Once a local solver is in place, symmetrical Gauss–Seidel nonlinear iterative methods can be applied immediately to obtain efficient fast sweeping methods.

2.2. Eikonal Equations

We take $d=2$ in (1.2):

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

A first-order Godunov upwind difference scheme is used to discretize the PDE (2.2) as in [40]:

$$\left[\left(\frac{\phi_{i,j}-\phi_{i,j}^{(x\min)}}{h}\right)^+\right]^2+\left[\left(\frac{\phi_{i,j}-\phi_{i,j}^{(y\min)}}{h}\right)^+\right]^2=f_{i,j}^2, \tag{2.3}$$

where $\phi_{i,j}^{(x\min)}=\min(\phi_{i-1,j},\phi_{i+1,j})$, $\phi_{i,j}^{(y\min)}=\min(\phi_{i,j-1},\phi_{i,j+1})$ and

$$(x)^+=\begin{cases} x, & x>0, \\ 0, & x\leqslant 0. \end{cases} \tag{2.4}$$

In order to construct a higher order scheme, we need to approximate the derivatives ϕ_x and ϕ_y with high order accuracy. We choose to use the popular WENO approximations developed in [14,15]. To illustrate the feasibility of the approach, we take the third order rather than the fifth order WENO approximations; one certainly can replace the third order WENO with the fifth order or even higher order WENO approximations. The interpolation stencil for the third order WENO scheme is shown in Fig. 1. $(\phi_x)_{i,j}$ and $(\phi_y)_{i,j}$ denote approximations for ϕ_x and ϕ_y at the grid point (i,j), respectively. The approximation for ϕ_x at the grid point (i,j) when the wind "blows" from the left to the right is

$$(\phi_x)_{i,j}^-=(1-w_-)\left(\frac{\phi_{i+1,j}-\phi_{i-1,j}}{2h}\right)+w_-\left(\frac{3\phi_{i,j}-4\phi_{i-1,j}+\phi_{i-2,j}}{2h}\right), \tag{2.5}$$

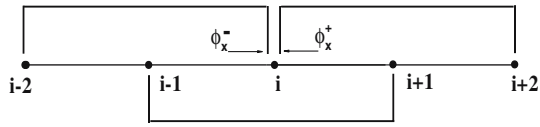

Fig. 1. Stencil of the third-order WENO Scheme.

where

$$w_- = \frac{1}{1+2r_-^2}, \quad r_- = \frac{\epsilon + (\phi_{i,j} - 2\phi_{i-1,j} + \phi_{i-2,j})^2}{\epsilon + (\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})^2}; \tag{2.6}$$

on the other hand, the approximation for ϕ_x at the grid point (i, j) when the wind "blows" from the right to the left is

$$(\phi_x)^+_{i,j} = (1 - w_+)\left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h}\right) + w_+\left(\frac{-\phi_{i+2,j} + 4\phi_{i+1,j} - 3\phi_{i,j}}{2h}\right), \tag{2.7}$$

where

$$w_+ = \frac{1}{1+2r_+^2}, \quad r_+ = \frac{\epsilon + (\phi_{i+2,j} - 2\phi_{i+1,j} + \phi_{i,j})^2}{\epsilon + (\phi_{i+1,j} - 2\phi_{i,j} + \phi_{i-1,j})^2}. \tag{2.8}$$

Similarly we define $(\phi_y)^-_{i,j}$ and $(\phi_y)^+_{i,j}$.

Next we have to incorporate these high order approximations (2.5)–(2.8) for derivatives into monotone numerical Hamiltonians. In the case of Eikonal equations, the numerical Hamiltonian under consideration is (2.3). In order to achieve this, we notice that the following identities hold:

$$(\phi_x)^-_{i,j} = \frac{\phi_{i,j} - [\phi_{i,j} - h \cdot (\phi_x)^-_{i,j}]}{h}, \quad (\phi_x)^+_{i,j} = \frac{[\phi_{i,j} + h \cdot (\phi_x)^+_{i,j}] - \phi_{i,j}}{h}, \tag{2.9}$$

$$(\phi_y)^-_{i,j} = \frac{\phi_{i,j} - [\phi_{i,j} - h \cdot (\phi_y)^-_{i,j}]}{h}, \quad (\phi_y)^+_{i,j} = \frac{[\phi_{i,j} + h \cdot (\phi_y)^+_{i,j}] - \phi_{i,j}}{h}. \tag{2.10}$$

According to the definitions of $(\phi_x)^-_{i,j}$ and $(\phi_x)^+_{i,j}$, $\phi_{i,j} - h \cdot (\phi_x)^-_{i,j}$ can be considered as an approximation to $\phi_{i-1,j}$ while $\phi_{i,j} + h \cdot (\phi_x)^+_{i,j}$ can be considered as an approximation to $\phi_{i+1,j}$. Similarly in (2.10), $\phi_{i,j} - h \cdot (\phi_y)^-_{i,j}$ and $\phi_{i,j} + h \cdot (\phi_y)^+_{i,j}$ can be considered as approximations to $\phi_{i,j-1}$ and $\phi_{i,j+1}$, respectively. Replacing $\phi_{i-1,j}, \phi_{i+1,j}, \phi_{i,j-1}, \phi_{i,j+1}$ with these approximations in Eq. (2.3), we have the following higher order schemes,

$$\left[\left(\frac{\phi^{\text{new}}_{i,j} - \phi^{(x\min)}_{i,j}}{h}\right)^+\right]^2 + \left[\left(\frac{\phi^{\text{new}}_{i,j} - \phi^{(y\min)}_{i,j}}{h}\right)^+\right]^2 = f^2_{i,j} \tag{2.11}$$

where

$$\begin{cases} \phi_{i,j}^{(x\min)} = \min(\phi_{i,j}^{\text{old}} - h\cdot(\phi_x)_{i,j}^-, \phi_{i,j}^{\text{old}} + h\cdot(\phi_x)_{i,j}^+), \\ \phi_{i,j}^{(y\min)} = \min(\phi_{i,j}^{\text{old}} - h\cdot(\phi_y)_{i,j}^-, \phi_{i,j}^{\text{old}} + h\cdot(\phi_y)_{i,j}^+). \end{cases} \tag{2.12}$$

Here $\phi_{i,j}^{\text{new}}$ denotes the to-be-updated numerical solution for ϕ at the grid point (i, j), and $\phi_{i,j}^{\text{old}}$ denotes the current old value for ϕ at the same grid point. When WENO approximations for derivatives $(\phi_x)_{i,j}^-$, $(\phi_x)_{i,j}^+$, $(\phi_y)_{i,j}^-$, $(\phi_y)_{i,j}^+$ in (2.12) are computed according to formulae (2.5)–(2.8), we always use the newest available values for ϕ in the interpolation stencils according to the philosophy of Gauss–Seidel type iterations. Of course, since we have not updated $\phi_{i,j}$ yet, $\phi_{i,j}^{\text{old}}$ is used in (2.5)–(2.8). The solution for Eq. (2.11) is:

$$\phi_{i,j}^{\text{new}} = \begin{cases} \min(\phi_{i,j}^{(x\min)}, \phi_{i,j}^{(y\min)}) + f_{i,j}h, & |\phi_{i,j}^{(x\min)} - \phi_{i,j}^{(y\min)}| \geqslant f_{i,j}h, \\ \dfrac{\phi_{i,j}^{(x\min)} + \phi_{i,j}^{(y\min)} + \sqrt{2f_{i,j}^2h^2 - (\phi_{i,j}^{(x\min)} - \phi_{i,j}^{(y\min)})^2}}{2}, & \text{otherwise.} \end{cases} \tag{2.13}$$

Remark 1. We introduce the high order accuracy by replacing $\phi_{i-1,j}, \phi_{i+1,j}, \phi_{i,j-1}, \phi_{i,j+1}$ with $\phi_{i,j} - h\cdot(\phi_x)_{i,j}^-, \phi_{i,j} + h\cdot(\phi_x)_{i,j}^+, \phi_{i,j} - h\cdot(\phi_y)_{i,j}^-, \phi_{i,j} + h\cdot(\phi_y)_{i,j}^+$, respectively, where $(\phi_x)_{i,j}^-$, $(\phi_x)_{i,j}^+$, $(\phi_y)_{i,j}^-$, $(\phi_y)_{i,j}^+$ are higher order WENO approximations for partial derivatives. When the iterations converge, we have solved the system

$$\sqrt{\max\{[(\phi_x)_{i,j}^-]^+, [-(\phi_x)_{i,j}^+]^+\}^2 + \max\{[(\phi_y)_{i,j}^-]^+, [-(\phi_y)_{i,j}^+]^+\}^2} = f_{i,j}, \quad 1 \leqslant i \leqslant I, 1 \leqslant j \leqslant J. \tag{2.14}$$

Since the Godunov numerical Hamiltonian is Lipschitz continuous with respect to all of its arguments, i.e., all of these high order approximations for partial derivatives, our schemes achieve the same formal higher order accuracy as the approximations for partial derivatives do.

Remark 2. The first order Godunov fast sweeping scheme (2.3) is purely upwind and monotone, hence the fast convergence is guaranteed (see [40]). There is no such monotonicity for the higher order scheme (2.11)–(2.12). A reliable initial guess is needed for the fast convergence of the Gauss–Seidel iterations. In other words, to achieve fast convergence we wish to have $\phi_{i,j} - h\cdot(\phi_x)_{i,j}^-, \phi_{i,j} + h\cdot(\phi_x)_{i,j}^+, \phi_{i,j} - h\cdot(\phi_y)_{i,j}^-, \phi_{i,j} + h\cdot(\phi_y)_{i,j}^+$ as good approximations to $\phi_{i-1,j}, \phi_{i+1,j}, \phi_{i,j-1}, \phi_{i,j+1}$, respectively, so that the causality of the true solution is approximately right. In the implementations of Godunov based high order fast sweeping schemes, we use the first order Godunov fast sweeping scheme, which is robust and efficient, to provide a good initial guess. Since the first order Godunov fast

sweeping scheme converges in just a few iterations, it is efficient to initialize the solution by such a scheme.

We summarize the higher order Godunov fast sweeping method for the eikonal equation (2.2) as follows:

1. *Initialization*: according to the boundary condition $\phi(x, y) = g(x, y)$, $(x, y) \in \Gamma$, assign exact values or interpolated values at grid points whose distances to Γ are less than or equal to $(n-1)h$, where n is the number of grid points in the small stencil in WENO approximations. For example, $n = 3$ for the third order WENO approximations. These values are fixed during iterations. The solution from the first-order Godunov fast sweeping method is used as the initial guess at all other grid points.
2. *Iterations*: solve the discretized nonlinear system (2.11) by Gauss–Seidel iterations with four alternating direction sweepings:

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

   Equations (2.5)–(2.8) and Eqs. (2.12), (2.13) are used to solve (2.11). High order extrapolations are used for the ghost points when calculating the high order WENO approximations of derivatives (2.5)–(2.8) for grid points on the boundary of the computational domain.
3. *Convergence*: if

$$||\phi^{\text{new}} - \phi^{\text{old}}||_{L^1} \leqslant \delta,$$

   where δ is a given convergence threshold value and $||\cdot||$ denotes the L^1 norm, the algorithm converges and stops.

Remark 3. In the first order fast sweeping method [40], at each grid point (i, j), when the solution, denoted by $\bar{\phi}$, of (2.3) is calculated from the current values of its neighboring grid points, $\phi_{i,j}^{\text{new}} = \min(\phi_{i,j}^{\text{old}}, \bar{\phi})$ is enforced so that the solution at each grid point is monotonically decreasing from an initially assigned large value. The reason for imposing this condition is that the first order scheme is monotone, thus the numerical solution at any iteration cannot get smaller than the numerical steady state solution for the discretized nonlinear system. However, for higher order schemes there is no such monotonicity any more. The solution may oscillate around the steady state solution. The oscillations are small.

However, we can NOT force $\phi_{i,j}^{\text{new}} = \min(\phi_{i,j}^{\text{old}}, \bar{\phi})$ as in the first order scheme. Otherwise, the order of accuracy of the schemes will deteriorate. Our numerical experiments have verified this.

Remark 4. In general the solution of this nonlinear PDE is not smooth. So the high order accuracy of our numerical schemes may not be achieved at singularities. However, the main issue is whether the breakdown at singularities will pollute the whole solution since errors can propagate along characteristics. There are two different scenarios for this issue. If the singularities are caused by the collision of characteristics analogous to the formation of shocks in hyperbolic conservation laws, the errors incurred at these singularities will not pollute solutions in other regions since no characteristics flow out of such singularities. If there are characteristics flowing out of the the singularities analogous to rarefaction waves in hyperbolic conservation laws, there will be pollution effects in the rarefaction wave region and more refined meshes are needed near these singularities to achieve uniform error bounds; see [29]. For the boundary value problem (2.2) all characteristics emanate from the boundary Γ. If there are singularities, such as corners or point sources, from which characteristics emanate, we assign exact solutions to a fixed local domain around those singularities when we refine the mesh, so that the high order convergence of the scheme can be observed. The above observation will be verified and discussed in more detail in the section for numerical examples.

Remark 5. Another important issue is to compare the computational cost of our higher order methods with that of the first order method. Suppose that the number of grids in each direction is N and the space dimension is d for the pth order method. To achieve the same accuracy for a first order method, we need N^p grid points in each direction. Since the first order method converges in a finite number of iterations independent of grid size, the computation cost is $O(N^{pd})$. So as long as the number of iterations for the pth order method does not grow faster than $N^{(p-1)d}$, we reduce the computational cost by using the high order method. For example, if we take the third order method in two dimensions, we can see from our numerical examples in Section 3 that the number of iterations is far fewer than N^4.

2.3. General Static Hamilton–Jacobi Equations

The idea of constructing high order sweeping methods for Eikonal equations can be straightforwardly extended to general static Hamilton–Jacobi equations

$$\begin{cases} H(\phi_x, \phi_y) = f(x, y), & x \in \Omega \setminus \Gamma, \\ \phi(x, y) = g(x, y), & x \in \Gamma \subset \Omega. \end{cases} \tag{2.15}$$

To discretize a general Hamiltonian $H(\phi_x, \phi_y)$, we may use the Lax-Friedrichs numerical Hamiltonian [24] which is the simplest among all monotone numerical Hamiltonians:

$$\hat{H}^{LF}(u^-, u^+; v^-, v^+) = H\left(\frac{u^- + u^+}{2}, \frac{v^- + v^+}{2}\right) - \frac{1}{2}\alpha^x(u^+ - u^-) - \frac{1}{2}\alpha^y(v^+ - v^-) \tag{2.16}$$

where

$$\alpha^x = \max_{\substack{A \leqslant u \leqslant B \\ C \leqslant v \leqslant D}} |H_1(u, v)|, \qquad \alpha^y = \max_{\substack{A \leqslant u \leqslant B \\ C \leqslant v \leqslant D}} |H_2(u, v)|. \tag{2.17}$$

Here $H_i(u, v)$ is the partial derivative of H with respect to the ith argument, or the Lipschitz constant of H with respect to the ith argument. $[A, B]$ is the value range for $u^{\pm}$, and $[C, D]$ is the value range for $v^{\pm}$.

Kao, Osher and Qian constructed the first order Lax-Friedrichs sweeping schemes for general static Hamilton–Jacobi equations in [18]

$$\phi_{i,j}^{\text{new}} = \left(\frac{1}{\frac{\alpha_x}{h_x} + \frac{\alpha_y}{h_y}}\right)\left[f - H\left(\frac{\phi_{i+1,j} - \phi_{i-1,j}}{2h_x}, \frac{\phi_{i,j+1} - \phi_{i,j-1}}{2h_y}\right) + \alpha_x \frac{\phi_{i+1,j} + \phi_{i-1,j}}{2h_x} + \alpha_y \frac{\phi_{i,j+1} + \phi_{i,j-1}}{2h_y}\right]. \tag{2.18}$$

Following the idea in Section (2.2), we replace $\phi_{i-1,j}$, $\phi_{i+1,j}$, $\phi_{i,j-1}$ and $\phi_{i,j+1}$ with $\phi_{i,j} - h_x \cdot (\phi_x)_{i,j}^-$, $\phi_{i,j} + h_x \cdot (\phi_x)_{i,j}^+$, $\phi_{i,j} - h_y \cdot (\phi_y)_{i,j}^-$ and $\phi_{i,j} + h_y \cdot (\phi_y)_{i,j}^+$, respectively, in (2.18), where $(\phi_x)_{i,j}^-$, $(\phi_x)_{i,j}^+$, $(\phi_y)_{i,j}^-$, and $(\phi_y)_{i,j}^+$ are higher order WENO approximations for partial derivatives of ϕ. Then the high order Lax-Friedrichs sweeping schemes for static H–J equations can be written as

$$\phi_{i,j}^{\text{new}} = \left(\frac{1}{\frac{\alpha_x}{h_x} + \frac{\alpha_y}{h_y}}\right)\left[f - H\left(\frac{(\phi_x)_{i,j}^- + (\phi_x)_{i,j}^+}{2}, \frac{(\phi_y)_{i,j}^- + (\phi_y)_{i,j}^+}{2}\right) + \alpha_x \frac{2\phi_{i,j}^{\text{old}} + h_x((\phi_x)_{i,j}^+ - (\phi_x)_{i,j}^-)}{2h_x} + \alpha_y \frac{2\phi_{i,j}^{\text{old}} + h_y((\phi_y)_{i,j}^+ - (\phi_y)_{i,j}^-)}{2h_y}\right], \tag{2.19}$$

or, written more compactly,

$$\phi_{i,j}^{\text{new}} = \left(\frac{1}{\frac{\alpha_x}{h_x}+\frac{\alpha_y}{h_y}}\right)\left[f - H\left(\frac{(\phi_x)_{i,j}^- + (\phi_x)_{i,j}^+}{2}, \frac{(\phi_y)_{i,j}^- + (\phi_y)_{i,j}^+}{2}\right) + \alpha_x \frac{(\phi_x)_{i,j}^+ - (\phi_x)_{i,j}^-}{2} + \alpha_y \frac{(\phi_y)_{i,j}^+ - (\phi_y)_{i,j}^-}{2}\right] + \phi_{i,j}^{\text{old}}, \tag{2.20}$$

where $\phi_{i,j}^{\text{new}}$ denotes the to-be-updated numerical solution for ϕ at the grid point (i, j), and $\phi_{i,j}^{\text{old}}$ denotes the current old value for ϕ at the same grid point. When WENO approximations for derivatives $(\phi_x)_{i,j}^-$, $(\phi_x)_{i,j}^+$, $(\phi_y)_{i,j}^-$, $(\phi_y)_{i,j}^+$ in (2.20) are calculated using formulae (2.5)–(2.8), we always use the newest value that we currently have for ϕ in the interpolation stencils according to the Gauss–Seidel type iteration. Of course, since we have not updated $\phi_{i,j}$ yet, $\phi_{i,j}^{\text{old}}$ is used in (2.5)–(2.8).

Since the Lax-Friedrichs numerical Hamiltonian is *not* a purely upwind flux, the causality along the characteristics is not strictly enforced. Thus the Lax-Friedrichs sweeping may take more iterations than the Godunov sweeping in general. However, it is easy to implement the Lax-Friedrichs Hamiltonians for more general equations. From our numerical experiments, the Lax-Friedrichs high order sweeping is more robust with respect to the initial guess than the Godunov high order sweeping. In our implementations we use big values, rather than the results from the first order Lax-Friedrichs fast sweeping method, as the initial guess in the high-order Lax-Friedrichs fast sweeping methods. The number of iterations is almost the same as the first order version in [18].

The high order Lax-Friedrichs fast sweeping methods for general static Hamilton–Jacobi equations are summarized as follows:

1. *Initialization*: according to the boundary condition $\phi(x, y) = g(x, y)$, $(x, y) \in \Gamma$, assign exact values or interpolated values at grid points whose distances to Γ are less than or equal to $(n-1)h$, where n is the number of grid points in the small stencil in WENO approximations. For example, $n = 3$ for the third order WENO approximations. These values are fixed during iterations. Big values are used as the initial guess at all other grid points.
2. *Iterations*: update $\phi_{i,j}^{\text{new}}$ in (2.20) by Gauss–Seidel iterations with four alternating direction sweepings:

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

WENO approximations (2.5)–(2.8) are used in (2.20). Linear Extrapolations are used for the ghost points when calculating the high order WENO approximations of derivatives (2.5)–(2.8) for grid points on the boundary of the computational domain.

3. *Convergence*: if

$$||\phi^{\text{new}} - \phi^{\text{old}}||_{L^1} \leqslant \delta,$$

where δ is a given convergence threshold value, the algorithm converges and stops.

Remark 6. High order approximations for derivatives can also be incorporated into other monotone numerical Hamiltonians in the same way as done here for Godunov and Lax-Friedrichs numerical Hamiltonians.

Remark 7. From our numerical experiments, the iterations needed for the Godunov based high order scheme to converge are fewer than those for the Lax-Friedrichs based high order scheme; however, which one to choose really depends on how easy to carry out the local solution procedure.

Remark 8. In our implementation for the third order Lax-Friedrichs sweeping scheme, we use linear extrapolation at the boundary points if there is no prescribed influx or other boundary conditions. It seems that high order extrapolations with bad initial guesses may cause instability because the causality along characteristics may not be followed correctly and no monotonicity exists for high order extrapolations. If the information is flowing out of the boundary, the low order accuracy at the boundary will not affect the accuracy in the interior, and the boundary points are excluded when we measure the error of our numerical solutions. But for the high order Godunov sweeping scheme, the initial guesses are the solutions from the first order scheme, so high order extrapolation boundary conditions are very robust. A fourth order extrapolation boundary condition is used in our implementation for the third order Godunov sweeping scheme.

3. NUMERICAL EXAMPLES

We apply the high order fast sweeping methods to some typical two dimensional and three dimensional problems. Third order WENO approximations are used. In all the examples, the threshold value at which the iteration stops is taken to be $\delta = 10^{-11}$. One iteration count includes four alternating sweepings for two-dimensional problems and eight alternating sweepings for three dimensional problems.

Example 1. Eikonal equation (2.2) with

$$f(x, y) = \frac{\pi}{2}\sqrt{\sin^2(\pi + \frac{\pi}{2}x) + \sin^2(\pi + \frac{\pi}{2}y)}, \tag{3.1}$$

and Γ is the point $(0, 0)$. The computational domain is $[-1, 1] \times [-1, 1]$. The exact solution for this problem is

$$\phi(x, y) = \cos\left(\pi + \frac{\pi}{2}x\right) + \cos\left(\pi + \frac{\pi}{2}y\right). \tag{3.2}$$

The Godunov Hamiltonian is used. Since the solution is smooth, errors and convergence rates in Table I indicate that the third order accuracy is obtained in the whole domain.

Example 2. Eikonal equation (2.2) with $f(x, y) = 1$. The computational domain is $\Omega = [-1, 1] \times [-1, 1]$, and Γ is a circle of center $(0, 0)$ and radius 0.5. The exact solution is the distance function to the circle Γ; thus the solution has a singularity at the center of the circle to which the characteristics converge. The Godunov Hamiltonian is used. Table II shows the errors and the third order accuracy in the smooth region (0.15 distance away from the center) of the solution. Table III shows the accuracy for the whole domain. Because of the singularity in the solution, the scheme achieves only the first order accuracy in the L^∞ norm; however, the L^1 accuracy for the whole domain still indicates the third order accuracy because the first order error is only made at the center (which is of measure $O(h^2)$) and there is no pollution effect. This indicates that the L^1 norm might be a more appropriate measure for the convergence rate of approximate solutions than the L^∞ norm. Theoretically, this was concluded by Lin–Tadmor [21].

Example 3. Case 1. 2-D Eikonal equation (2.2) with $f(x, y) = 1$.

The computational domain is $\Omega = [-3, 3] \times [-3, 3]$; Γ consists of two circles of equal radius 0.5 with centers located at $(-1, 0)$ and $(\sqrt{1.5}, 0)$, respectively. The exact solution is the distance function to Γ. The singular set for the solution is composed of the center of each circle and the line that

Table I. Smooth Solution. Godunov Numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
40 × 40	1.89E−4	–	6.32E−4	–	36
80 × 80	1.31E−5	3.85	3.08E−5	4.36	27
160 × 160	6.75E−7	4.28	1.19E−6	4.69	33
320 × 320	6.89E−8	3.29	1.14E−7	3.38	43
640 × 640	8.47E−9	3.02	1.40E−8	3.03	70

Table II. Γ Consists of a Circle. Smooth Region 0.15 Away from the Derivative Singularity. Godunov Numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
80 × 80	1.70E−6	–	1.27E−4	–	19
160 × 160	2.64E−7	2.69	3.70E−6	5.10	23
320 × 320	3.98E−8	2.73	1.16E−6	1.68	31
640 × 640	5.03E−9	2.99	1.52E−7	2.93	48

Table III. Γ Consists of a Circle. Errors for the Whole Domain. Godunov Numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
80 × 80	7.59E−6	–	5.87E−3	–	19
160 × 160	7.54E−7	3.33	2.80E−3	1.07	23
320 × 320	8.41E−8	3.16	1.27E−3	1.14	31
640 × 640	1.27E−8	2.72	5.34E−4	1.25	48

is equally distant to the two circles. All of these singularities correspond to the intersection of characteristics. The Godunov Hamiltonian is used. Table IV shows that the third order accuracy is obtained in the smooth region, where the errors are measured 0.15 distance away from the singular set. Table V shows that the scheme achieves only the first order accuracy in the L^∞ norm due to the singularity in the solution; moreover, the L^1 error in the whole domain indicates the second order accuracy since the singular set has a measure of $O(h)$.

Case 2. The 3-D eikonal equation.

Following the idea of high order fast sweeping algorithms for two dimensional problems in Section 2, we can construct the algorithms for three dimensional problems straightforwardly. For example, consider the 3-D eikonal equation

Table IV. Γ Consists of Two Circles. Smooth Region 0.15 Away from the Derivative Singularity. Godunov Numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
80 × 80	1.29E−4	–	2.35E−3	–	26
160 × 160	4.57E−6	4.82	6.72E−4	1.81	26
320 × 320	4.34E−7	3.40	2.99E−5	4.49	34
640 × 640	6.57E−8	2.72	2.70E−6	3.47	53

Table V. Γ is Two Circles. Whole Region. Godunov Numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
80 × 80	1.49E−4	–	9.57E−3	–	26
160 × 160	9.68E−6	3.94	4.84E−3	0.98	26
320 × 320	1.28E−6	2.92	2.35E−3	1.04	34
640 × 640	2.42E−7	2.40	9.97E−4	1.24	53

Table VI. Γ is Two Spheres. Whole Region. Godunov Numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
40 × 40 × 40	2.61E−3	–	3.67E−2	–	18
80 × 80 × 80	2.79E−4	3.22	1.49E−2	1.30	13
160 × 160 × 160	1.18E−5	4.57	7.64E−3	0.97	19
320 × 320 × 320	8.06E−7	3.87	3.75E−3	1.03	28

$$\begin{cases} \sqrt{\phi_x^2+\phi_y^2+\phi_z^2} = f(x,y,z), & (x,y,z)\in\Omega\subset R^3, \\ \phi(x,y,z) = g(x,y,z), & (x,y,z)\in\Gamma\subset\Omega. \end{cases} \tag{3.3}$$

The local solver by the Godunov numerical Hamiltonian is

$$\left[\left(\frac{\phi_{i,j}^{\text{new}}-\phi_{i,j}^{(x\min)}}{h}\right)^+\right]^2 + \left[\left(\frac{\phi_{i,j}^{\text{new}}-\phi_{i,j}^{(y\min)}}{h}\right)^+\right]^2 + \left[\left(\frac{\phi_{i,j}^{\text{new}}-\phi_{i,j}^{(z\min)}}{h}\right)^+\right]^2 = f_{i,j}^2 \tag{3.4}$$

where

$$\begin{cases} \phi_{i,j}^{(x\min)} = \min(\phi_{i,j}^{\text{old}} - h\cdot(\phi_x)_{i,j}^-, \phi_{i,j}^{\text{old}} + h\cdot(\phi_x)_{i,j}^+), \\ \phi_{i,j}^{(y\min)} = \min(\phi_{i,j}^{\text{old}} - h\cdot(\phi_y)_{i,j}^-, \phi_{i,j}^{\text{old}} + h\cdot(\phi_y)_{i,j}^+), \\ \phi_{i,j}^{(z\min)} = \min(\phi_{i,j}^{\text{old}} - h\cdot(\phi_z)_{i,j}^-, \phi_{i,j}^{\text{old}} + h\cdot(\phi_z)_{i,j}^+). \end{cases} \tag{3.5}$$

$(\phi_x)_{i,j}^-, (\phi_x)_{i,j}^+, (\phi_y)_{i,j}^-, (\phi_y)_{i,j}^+, (\phi_z)_{i,j}^-, (\phi_z)_{i,j}^+$ are the WENO approximations for left and right derivatives in x, y, z directions. The discretized nonlinear system (3.4) is solved by the systematic method in [40] and Gauss–Seidel

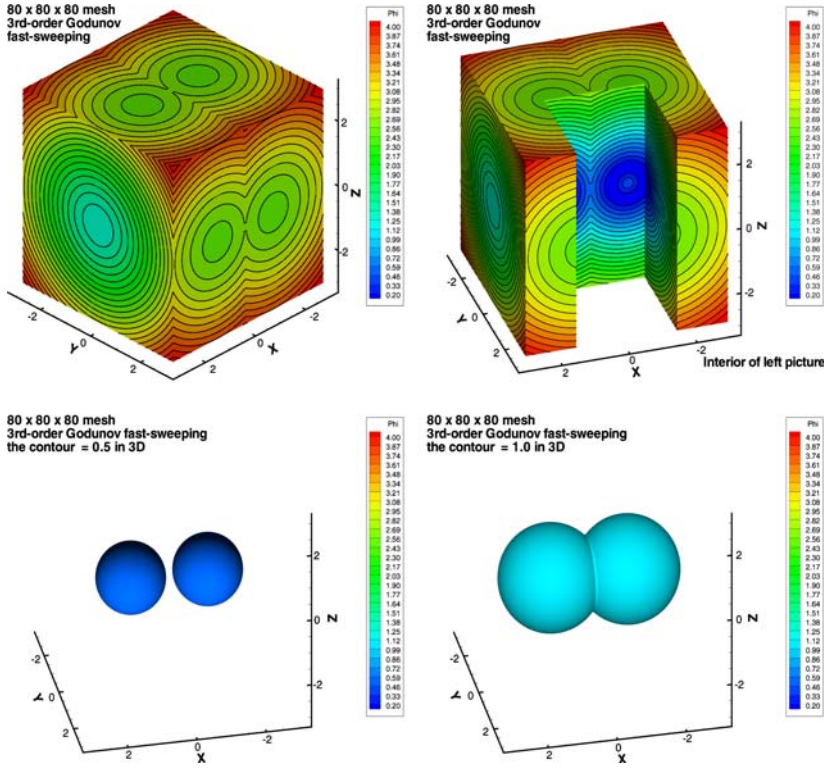

Fig. 2. Two Spheres problem. Godunov numerical Hamiltonian. Top left: contour plot for the whole surface; top right: a part of the interior region; bottom left: the contour plot for $\phi = 0.5$; bottom right: the contour plot for $\phi = 1.0$. For the contour lines in the top two pictures, there are 30 equally spaced of them from $\phi = 0.2$ to $\phi = 4$. Third order numerical solution with $80 \times 80 \times 80$ mesh.

iterations with eight alternating direction sweepings. We use the two-sphere problem to test the third order algorithm for three dimensional problems. The computational domain is $\Omega = [-3, 3] \times [-3, 3] \times [-3, 3]$; Γ consists of two spheres of equal radius 0.5 with centers located at $(-1, 0, 0)$ and $(\sqrt{1.5}, 0, 0)$, respectively. The accuracy for the whole domain and iteration numbers are listed in Table VI, and the contour plots of the solution by a $80 \times 80 \times 80$ mesh are presented in Fig. 2 for the surface of the domain, a part of the interior of the domain and in the cases of two contour values 0.5 and 1.0 in the 3-D case, respectively. Similar conclusions as in the 2-D example can be drawn here.

Example 4. Eikonal equation (2.2) with $f(x, y) = 1$. The Godunov Hamiltonian is used. The computational domain is $\Omega = [-1, 1] \times [-1, 1]$, and Γ is a source point with coordinates $(0, 0)$. So the exact solution is

Table VII. Γ is a Point. Initial Values are Given in the Box with Length 0.3 which Includes the Source Point. Godunov Numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
40×40	1.56E−4	–	2.88E−4	–	30
80×80	2.80E−6	5.80	5.60E−6	5.69	28
160×160	6.64E−7	2.08	1.28E−6	2.12	32
320×320	9.70E−8	2.78	1.83E−7	2.81	44
640×640	1.23E−8	2.98	2.32E−8	2.98	68

the distance function to the source point Γ. The solution is singular at the source point. Since all characteristics emanate from this source point analogous to rarefaction waves in hyperbolic conservation laws, errors incurred at the source point will propagate out and pollute the solution in the whole computational domain. We illustrate this subtlety with different initializations near the source point. This was also treated in the geometrical optics setting in [29].

First we assign the exact values to a small region which encloses the source point, say, a small box with length 0.3. This box size is fixed during the mesh refinement study. Accuracy and errors are reported in Table VII, and the third order accuracy is obtained.

Next we initialize the solution near the source point by fixing the number of grid points during the mesh refinement study. Thus the box which encloses the source point is taken to have length $2h$, where h is the mesh size. Accuracy and errors are reported in Table VIII, and the third order accuracy is polluted to some extent. We remark that in Examples 2 and 3 where singularities are of shock type, both of the two initializations led to neat third order accuracy. The reason is that the rarefaction wave is different from the shock wave. Characteristics intersect to form shocks. Errors incurred at the shocks will not propagate from shocks to other locations to degrade the high order accuracy of the solution in smooth regions. But if Γ has source points or corners from which the characteristics emanate so that the rarefaction wave forms, the errors incurred at the singular point will propagate to pollute the solution in the smooth region, hence leading to the loss of accuracy. Of course, if a small fixed domain is used to wrap up the singular corner point, i.e., accurate values are assigned to the small region, the high order accuracy can be achieved. The most efficient way to recover such a loss of accuracy is to use adaptive meshes with good a posteriori estimates near singularities; see [29] for such an adaptive method in the paraxial formulation.

Table VIII. Γ is a Point. Initial Values are Given in the Box with Length $2h$ which Includes the Source Point. Godunov numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
40 × 40	4.51E−4	–	8.41E−4	–	52
80 × 80	1.71E−4	1.40	3.16E−4	1.41	37
160 × 160	5.42E−5	1.66	1.07E−4	1.56	37
320 × 320	9.90E−6	2.45	2.36E−5	2.18	52
640 × 640	1.46E−6	2.76	2.21E−6	3.41	84

Table IX. Γ is a Point. Initial Values are Given in the Box with Length $2h$ which Includes the Source Point. First Order Scheme. Godunov Numerical Hamiltonian

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
40 × 40	6.29E−3	–	1.27E−2	–	1
80 × 80	3.52E−3	0.84	6.66E−3	0.93	1
160 × 160	1.90E−3	0.89	3.47E−3	0.94	1
320 × 320	1.01E−3	0.92	1.79E−3	0.96	1
640 × 640	5.26E−4	0.94	9.16E−4	0.96	1

In Table IX, we also list the errors and convergence rates for the first order fast sweeping method (2.3). In comparison with the results in Table VII and VIII, the high order schemes do achieve much smaller errors than a low order method does on the same grids.

Example 5. Eikonal equation (2.2) with $f(x, y) = 1$. The Godunov Hamiltonian is used. The computational domain $\Omega = [-2, 2] \times [-2, 2]$, and Γ is a sector of three quarters of a circle shown in Fig. 3. So the exact solution is the distance function to the sector Γ. Singularities at two corners in Γ give rise to different scenarios in different regions, which include both shocks and rarefaction waves. In Fig. 3, we illustrate the shocks by a boldfaced solid line and three rarefaction wave regions with letter(s) "R" or "Rarefaction"; the solution is smooth in other regions. During the mesh refinement, we fix the number of grid points when we initialize the domain. Errors and convergence rates are presented in Table X. We only list results for L^1 errors. The third order accuracy is obtained in the regions where the solution is smooth. Moreover, we still obtain the second order accuracy in the whole computational domain. The accuracy in the rarefaction wave region is consistent with the results in Example 4. We also plot the 3-D pictures of the exact solution and numerical solution with an 80×80 mesh in Fig. 4; the sharp shock transition is apparent.

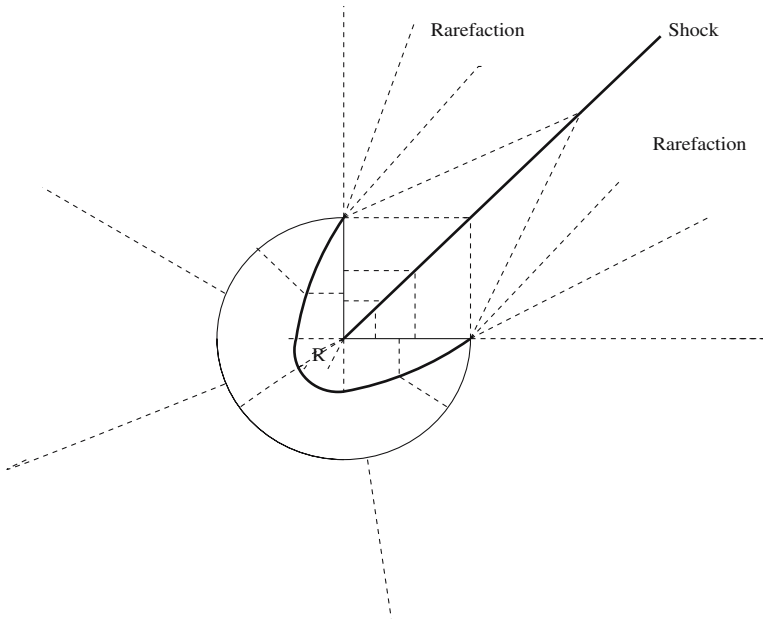

Fig. 3. Γ is a sector.

Table X. Γ is a Sector in Fig. 3. Accuracy in Different Regions. Godunov Numerical Hamiltonian

	Smooth region		Whole region		Rarefaction		
Mesh	L^1 error	Order	L^1 error	Order	L^1 error	Order	Iter
80 × 80	1.03E−5	–	1.70E−4	–	5.71E−4	–	38
160×160	1.32E−6	2.96	6.09E−5	1.48	2.26E−4	1.33	26
320 × 320	2.89E−7	2.19	1.72E−5	1.82	5.83E−5	1.96	34
640 × 640	4.03E−8	2.84	4.13E−6	2.06	1.03E−5	2.50	53

Example 6 (shape-from-shading). Eikonal equation (2.2) with

$$f(x, y) = 2\pi\sqrt{[\cos(2\pi x)\sin(2\pi y)]^2 + [\sin(2\pi x)\cos(2\pi y)]^2}. \tag{3.6}$$

$\Gamma = \{(\frac{1}{4}, \frac{1}{4}), (\frac{3}{4}, \frac{3}{4}), (\frac{1}{4}, \frac{3}{4}), (\frac{3}{4}, \frac{1}{4}), (\frac{1}{2}, \frac{1}{2})\}$, consisting of five isolated points. The computational domain $\Omega = [0, 1] \times [0, 1]$. $\phi(x, y) = 0$ is prescribed at the boundary of the unit square. The solution for this problem is the shape function, which has the brightness $I(x, y) = 1/\sqrt{1 + f(x, y)^2}$ under vertical lighting. See [31] for details. In [14], high order time marching

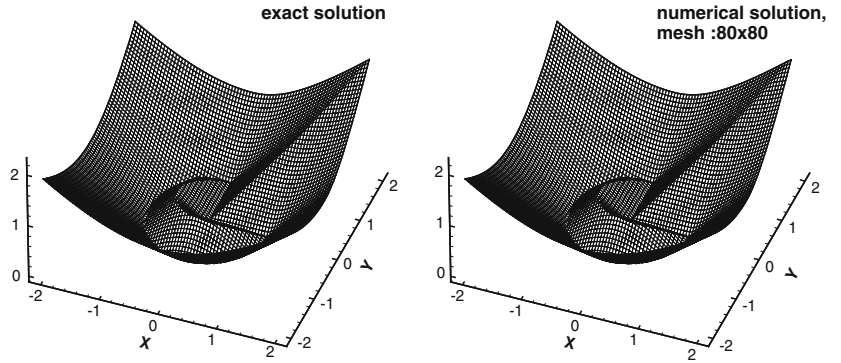

Fig. 4. Distance to Γ. where Γ is a sector in Fig. 3. Godunov numerical Hamiltonian. Left: exact solution; right: third order numerical solution with 80×80 mesh.

WENO schemes are used to calculate the solution for this problem. We apply both high order Godunov based and Lax-Friedrichs based fast sweeping schemes to the following two cases.

Case 1.

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

The exact solution for this case is

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

a smooth function.

Case 2.

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

The exact solution for this case is

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

this solution is *not* smooth.

Errors, convergence rates and iteration numbers are reported in Table XI and XII for both Godunov and Lax-Friedrichs sweeping methods, respectively. Both methods yield fully third order accuracy in the case of smooth solutions (Case 1). Since the exact solution in Case 2 is *not* smooth, globally numerical errors indicate that the convergence order is

Table XI. Example 6, Shape-from-Shading Problem, Godunov Numerical Hamiltonian

	Case 1 (smooth)			Case 2 (non-smooth)		
Mesh	L^1 error	Order	Iteration number	L^1 error	Order	Iteration number
80 × 80	1.24E−4	–	16	1.14E−3	–	18
160 × 160	5.78E−6	4.43	23	1.98E−4	2.53	23
320 × 320	3.27E−7	4.14	29	2.95E−5	2.74	31
640 × 640	3.56E−8	3.20	52	7.28E−6	2.02	57

Table XII. Example 6, Shape-from-Shading Problem, Lax-Friedrichs Numerical Hamiltonian, $\alpha_x = \alpha_y = 1$

	Case 1 (smooth)			Case 2 (non-smooth)		
Mesh	L^1 error	Order	Iteration number	L^1 error	Order	Iteration number
80 × 80	1.44E−4	–	22	2.94E−3	–	28
160 × 160	7.68E−6	4.23	33	8.35E−4	1.81	38
320 × 320	4.93E−7	3.96	58	2.20E−4	1.92	75
640 × 640	5.58E−8	3.14	111	4.67E−5	2.24	136

higher than the second order, which is consistent with the results in the examples shown above. In terms of the iteration numbers shown in Tables XI and XII, the high order Godunov fast sweeping method needs fewer iterations than the high order Lax-Friedrichs fast sweeping method does. Figures 5 and 6 illustrate three-dimensional pictures and contour plots for numerical solutions of these two cases on the 80×80 mesh; these numerical solutions are from both the third order Godunov scheme and the first order Godunov fast sweeping method. Obviously, the high order method achieves much higher accuracy and resolution than the first order scheme does on the same mesh. Figures by the Lax-Friedrichs third order fast sweeping schemes are very similar to those by the Godunov Hamiltonian, hence they are omitted to save space.

Remark 9. Using this example, we compared our third order fast sweeping method with the time marching approach in [14], based on both Godunov and Lax-Friedrichs numerical Hamiltonians, and the results are reported in Table XIII. For the time marching approach, we use the third order WENO in spatial direction [14] and a third order TVD Runge-Kutta scheme [36] in time direction, and the CFL number is taken to be 0.6 as in [14]. The initial guesses of the time marching method are totally the same as our third order fast sweeping method; i.e., for the Godunov numeri-

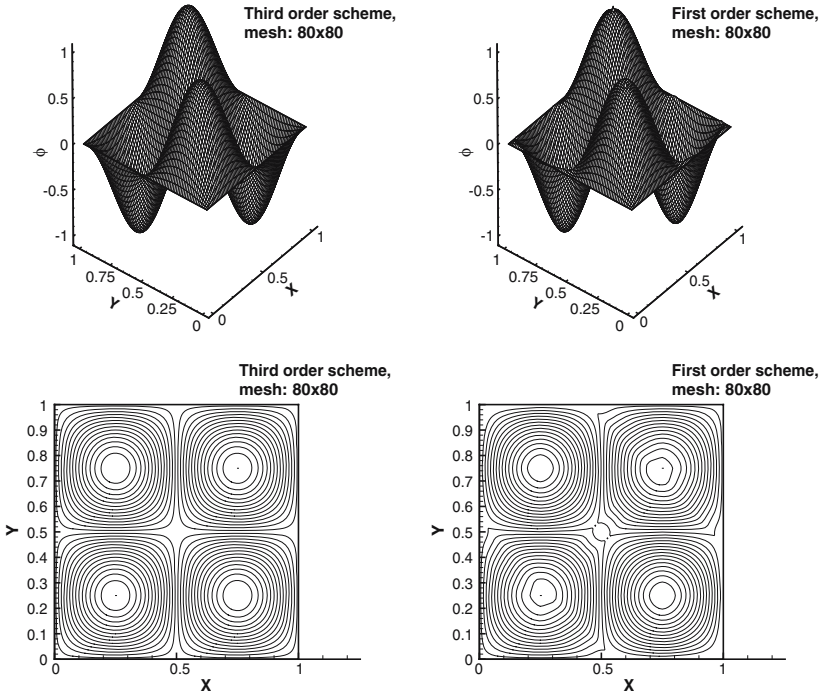

Fig. 5. Example 6, shape-from-shading, case 1. Godunov numerical Hamiltonian. Left: third order scheme; right: first order scheme; top: three-dimensional view; bottom: contour lines, 30 equally spaced contour lines from $\phi = -1$ to $\phi = 1$.

cal Hamiltonian, they are generated by the first order Godunov sweeping method; for the Lax-Friedrichs numerical Hamiltonian, the initial guesses are just big values. The convergence criteria of both methods are the same too. In Table XIII, one iteration count of the time-marching approach includes three stages of the third order TVD Runge-Kutta scheme, and one iteration count of the fast sweeping method includes four alternating sweepings. For this example, the third order fast sweeping method with the Godunov numerical Hamiltonian is about three times faster than the third order time-marching approach on a 80×80 mesh. When the mesh is refined, the advantage of our fast sweeping algorithm with the Godunov numerical Hamiltonian becomes more significant due to the upwind property of the Godunov numerical Hamiltonian, and it is about eight times faster than the third order time-marching approach on a 640×640 mesh. For the Lax-Friedrichs numerical Hamiltonian, the third order fast sweeping method is about six times faster than the third order time-marching approach for all mesh sizes. So our method is much more efficient

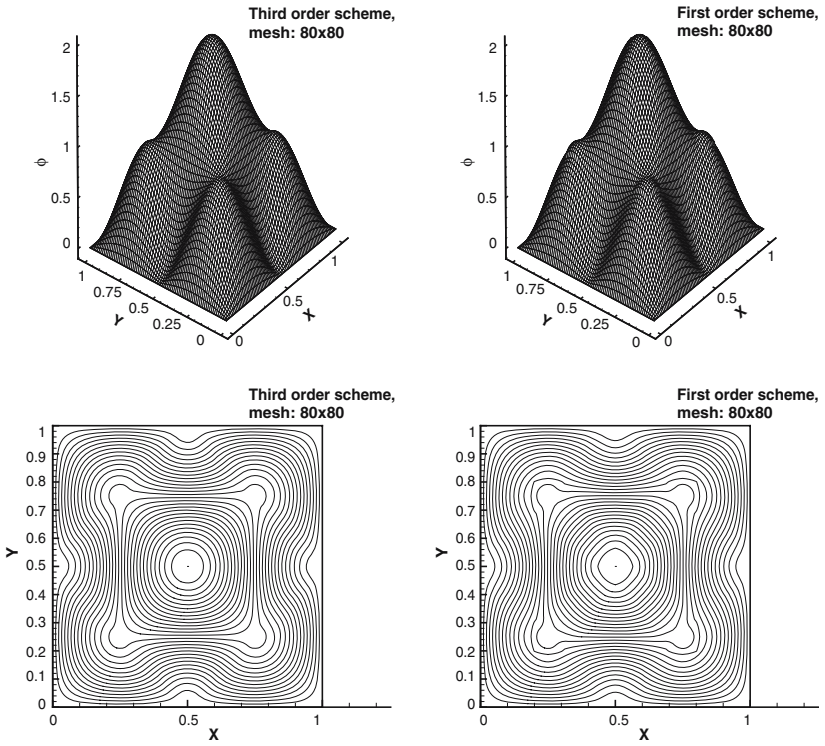

Fig. 6. Example 6, shape-from-shading, case 2. Godunov numerical Hamiltonian. Left: third order scheme; right: first order scheme; top: three-dimensional view; bottom: contour lines, 30 equally spaced contour lines from $\phi = 0$ to $\phi = 2$.

than the time-marching approach, for both Godunov and Lax-Friedrichs numerical Hamiltonians.

Example 7 (shape-from-shading). Eikonal equation (2.2) with

$$\textbf{Case (a):} \qquad f(x, y) = \sqrt{(1 - |x|)^2 + (1 - |y|)^2}; \tag{3.7}$$

$$\textbf{Case (b):} \qquad f(x, y) = 2\sqrt{y^2(1 - x^2)^2 + x^2(1 - y^2)^2}. \tag{3.8}$$

The computational domain $\Omega = [-1, 1] \times [-1, 1]$. $\phi(x, y) = 0$ is prescribed at the boundary of the square for both cases. Additional boundary condition $\phi(0, 0) = 1$ is prescribed for case (b). The exact solutions for these two cases are

Table XIII. The Comparison between the Third Order Fast Sweeping Method and the Time-marching Method (Third-Order WENO Schemes with Third-Order TVD Runge-Kutta in Time Direction, see [14]) for the Example 6, Shape-from-shading Problem, Case 2 (Non-smooth Solution)

	3rd-order Time-marching			3rd-order Fast Sweeping		
Mesh	L^1 error	Order	Iteration number	L^1 error	Order	Iteration number
Godunov Numerical Hamiltonian						
80 × 80	1.14E−3	–	70	1.14E−3	–	18
160 × 160	1.98E−4	2.53	133	1.98E−4	2.53	23
320 × 320	2.96E−5	2.74	281	2.95E−5	2.74	31
640 × 640	7.31E−6	2.02	578	7.28E−6	2.02	57
Lax–Friedrichs Numerical Hamiltonian						
80 × 80	2.94E−3	–	217	2.94E−3	–	28
160 × 160	8.35E−4	1.81	327	8.35E−4	1.81	38
320 × 320	2.19E−4	1.93	589	2.20E−4	1.92	75
640 × 640	4.79E−5	2.19	1092	4.67E−5	2.24	136

Godunov numerical Hamiltonian & Lax–Friedrichs numerical Hamiltonian.

$$\textbf{Case (a):} \quad \phi(x, y) = (1-|x|)(1-|y|); \tag{3.9}$$

$$\textbf{Case (b):} \quad \phi(x, y) = (1-x^2)(1-y^2). \tag{3.10}$$

In [13], a time-marching discontinuous Galerkin method is used to compute the solution for these two cases. We apply our high order Godunov fast sweeping method to these problems. Because the exact solution of case (a) is a piecewise bi-linear polynomial and the exact solution of case (b) is a bi-quadratic polynomial, the numerical solutions by the third order scheme are accurate up to round-off errors. Table XIV indicates that the errors for both cases are round-off errors, and the iteration numbers in Table XIV demonstrate the fast convergence of the scheme, which is again much faster than the time marching results. The three-dimensional pictures and contour plots for the numerical solution of both cases on an 80×80 mesh are presented in Fig. 7.

Example 8 (Travel-time problem in elastic wave propagation). The quasi-P and the quasi-SV slowness surfaces are defined by the quadratic equation [26]:

$$c_1\phi_x^4 + c_2\phi_x^2\phi_y^2 + c_3\phi_y^4 + c_4\phi_x^2 + c_5\phi_y^2 + 1 = 0, \tag{3.11}$$

where

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

Table XIV. Example 7, Shape-from-Shading Problem, Third Order Scheme, Godunov Numerical Hamiltonian

	Case a			Case b		
Mesh	L^1 error	L^∞ error	Iter number	L^1 error	L^∞ error	Iter number
80 × 80	1.68E−16	2.23E−14	2	1.62E−16	3.38E−14	27
160 × 160	8.00E−16	3.30E−13	4	3.49E−16	4.83E−14	32
320 × 320	3.88E−15	4.39E−12	7	4.05E−16	1.41E−13	46
640 × 640	1.81E−14	5.30E−11	10	4.05E−16	4.30E−13	76

Here a_{ij}s are given elastic parameters. The corresponding quasi-P wave eikonal equation is

$$\sqrt{-\frac{1}{2}(c_4\phi_x^2+c_5\phi_y^2)+\sqrt{\frac{1}{4}(c_4\phi_x^2+c_5\phi_y^2)^2-(c_1\phi_x^4+c_2\phi_x^2\phi_y^2+c_3\phi_y^4)}}=1, \tag{3.12}$$

which is a convex Hamilton–Jacobi equation. The elastic parameters are taken to be

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

in the numerical example to be shown; Figure 8 shows the corresponding convex slowness surface in the gradient component.

The corresponding quasi-SV wave eikonal equation is

$$\sqrt{-\frac{1}{2}(c_4\phi_x^2+c_5\phi_y^2)-\sqrt{\frac{1}{4}(c_4\phi_x^2+c_5\phi_y^2)^2-(c_1\phi_x^4+c_2\phi_x^2\phi_y^2+c_3\phi_y^4)}}=1, \tag{3.13}$$

which is a nonconvex H–J equation. The elastic parameters are taken to be

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

in the numerical example to be shown; Figure 8 shows the corresponding non-convex slowness surface in the gradient component.

The computational domain is $[-1,1]\times[-1,1]$, and $\Gamma=\{(0,0)\}$. Initial values are assigned in a box with length 0.3 which includes the source point. Because both Hamiltonians are pretty complicated, it is troublesome to apply the Godunov Hamiltonian to these examples. Thus we

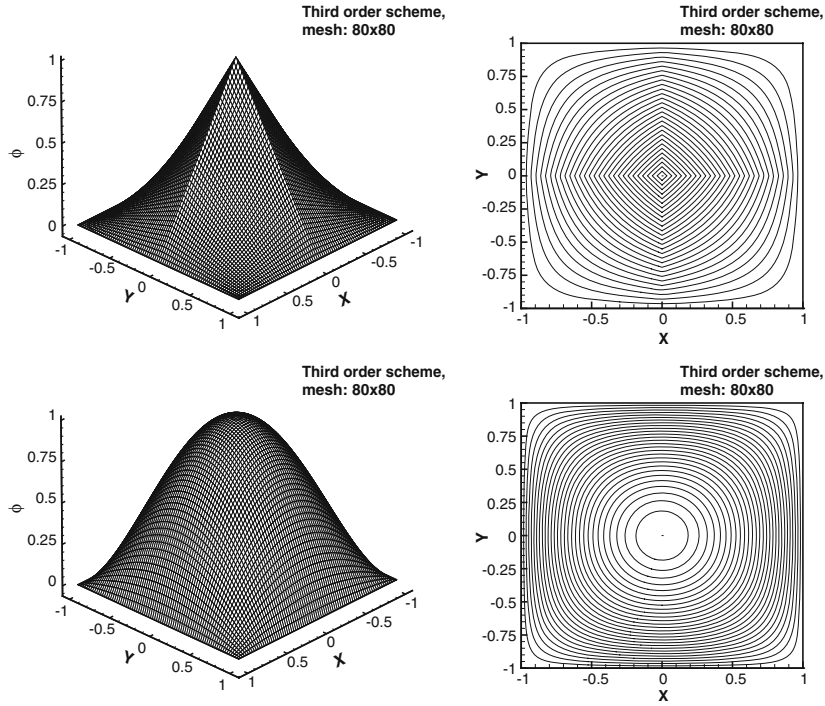

Fig. 7. Example 7, shape-from-shading. Third order scheme, Godunov numerical Hamiltonian. Top left: three-dimensional view for case (a); bottom left: three-dimensional view for case (b); top right: contour lines for case (a), 30 equally spaced contour lines from $\phi = 0$ to $\phi = 1$; bottom right: contour lines for case (b), 30 equally spaced contour lines from $\phi = 0$ to $\phi = 1$.

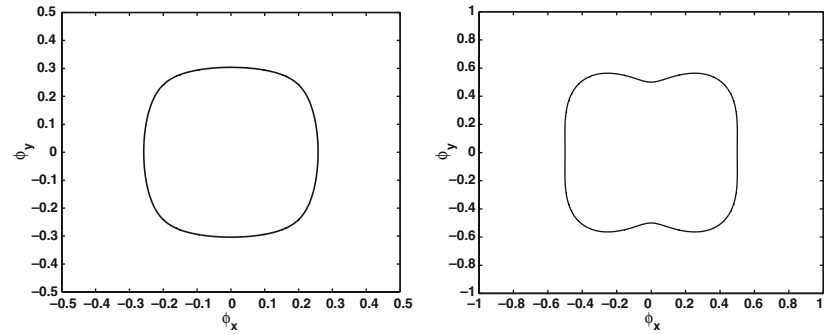

Fig. 8. Example 8. Slowness surfaces. Left: quasi-P wave slowness surface; right: quasi-SV wave slowness surface.

Table XV. Quasi-P Wave, Lax-Friedrichs Numerical Hamiltonian, Initial Values are Given in the Box with Length 0.3 which Includes the Source Point, $\alpha_x = \alpha_y = 4$

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
40 × 40	7.23E−4	–	3.07E−3	–	27
80 × 80	9.33E−5	2.95	5.81E−4	2.40	36
160 × 160	9.58E−6	3.28	6.72E−5	3.11	58
320 × 320	1.27E−6	2.92	8.94E−6	2.91	98
640 × 640	1.62E−7	2.97	1.16E−6	2.95	178

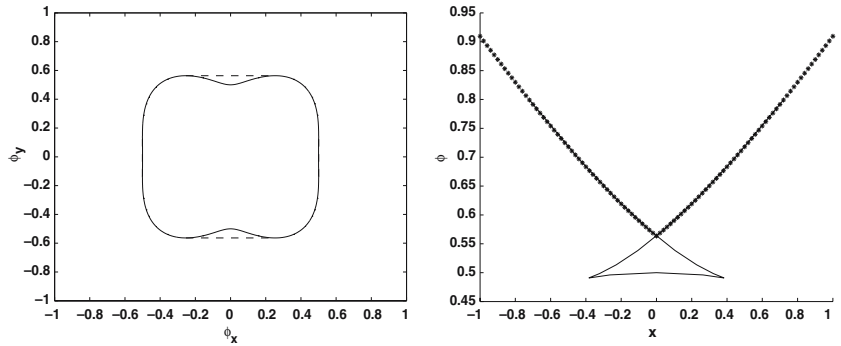

Fig. 9. Example 8. Left: Convexified quasi-SV slowness surface; right: ray tracing solutions at y=1.0: solid lines; convexified solutions at y=1.0: stars.

apply the third order Lax-Friedrichs fast sweeping method (2.20) to these problems.

Errors and convergence rates are listed in Table XV for the quasi-P wave travel-time, which is smooth in the whole domain except at the source point. In fact, since the slowness surface is convex, the resulting point-source solution is a pure rarefaction wave. To initialize a fixed box around the source point, we use a shooting method by solving a two-point boundary value problem; see [27] for details. The third order accuracy is obtained.

For the quasi-SV wave, the solution is smooth except along the lines $x = 0$ and $y = 0$ because of its nonconvexity in the slowness space along the two axes. In this case, since the outward normals of the slowness surface correspond to the ray directions in the physical space according to the method of characteristics, we have so-called instantaneous singularities along the lines $x = 0$ and $y = 0$ and the resulting travel-time field is multivalued; see the solid lines in Fig. 9. To pick out a unique, physically

Table XVI. Quasi-SV Wave, Lax-Friedrichs Numerical Hamiltonian, Initial Values are Given in the Box with Length 0.3 Which Includes the Source Point, $\alpha_x = \alpha_y = 2$

Mesh	L^1 error	Order	L^∞ error	Order	Iteration number
	Smooth region 0.15 away from $x=0$ and $y=0$				
40 × 40	9.20E−4	–	3.65E−3	–	42
80 × 80	6.21E−5	3.89	6.32E−4	2.53	34
160 × 160	2.39E−6	4.70	2.28E−5	4.79	57
320 × 320	4.61E−7	2.37	1.47E−6	3.96	99
640 × 640	5.97E−8	2.95	2.97E−7	2.30	181
	Whole region				
40 × 40	2.19E−3	–	1.53E−2	–	42
80 × 80	6.09E−4	1.84	8.02E−3	0.93	34
160 × 160	1.62E−4	1.91	4.19E−3	0.94	57
320 × 320	3.70E−5	2.13	2.00E−3	1.06	99
640 × 640	1.02E−5	1.86	8.21E−4	1.29	181

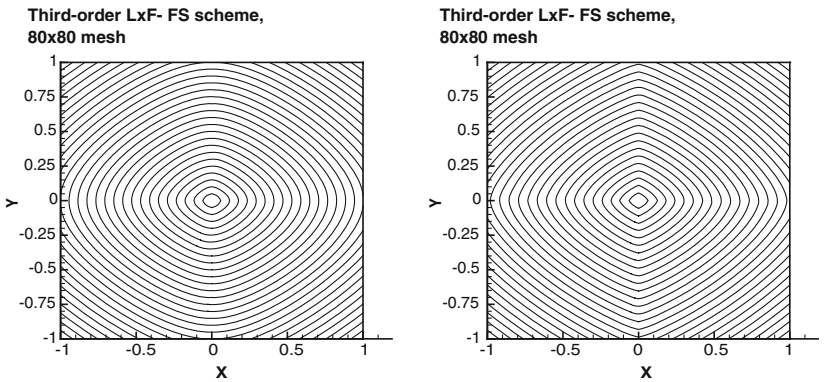

Fig. 10. Example 8, Travel-time problem in elastic wave propagation. Third order Lax-Friedrichs fast sweeping scheme. 80 × 80 mesh. Left: quasi-P wave, $\alpha_x = \alpha_y = 4$, 30 equally spaced contour lines from $\phi = 0$ to $\phi = 0.44173$; right: quasi-SV wave, $\alpha_x = \alpha_y = 2$, 30 equally spaced contour lines from $\phi = 0$ to $\phi = 0.909652$.

relevant solution, we convexify the nonconvex slowness surface first and then adapt a shooting method, similar to the one used for quasi-P waves, to pick out a continuous solution; see the stars in Fig. 9. This construction agrees with a method based on the Huygens's principle demonstrated in [30]. The shooting method is also used to initialize the traveltime field in a specified box around the source point. Therefore, the point source

problem for quasi-SV wave traveltime produces both rarefaction and shock singularities.

Table XVI illustrates that the third order accuracy is obtained in the smooth region of the solution in both L^1 and L^∞ norms; however, in terms of the whole domain including the shock wave region, the second order accuracy is obtained in the L^1 norm and only the first order accuracy is obtained in the L^∞ norm. Overall, the third-order scheme achieves much smaller errors than the first order scheme does, even in the shock wave region. The contour plots of the solutions are shown in Fig. 10. The quasi-SV solution agrees with the results based on the Huygens's principle shown in [30].

4. CONCLUDING REMARKS

We have developed high order fast sweeping methods for static Hamilton–Jacobi equations on rectangular meshes. A general procedure is given to incorporate the high order approximations into monotone numerical Hamiltonians so that the first order sweeping schemes can be extended to high order schemes. Extensive numerical examples demonstrate that the high order methods yield higher order accuracy in the smooth region of the solution, higher resolution for the singularities of derivatives, and fast convergence to viscosity solutions of the Hamilton–Jacobi equations.

ACKNOWLEDGEMENTS

The research of Hong-Kai Zhao is partially supported by ONR Grant #N00014-02-1-0090, DARPA Grant #N00014-02-1-0603 and Sloan Fellowship Foundation. The research of Jianliang Qian is supported by ONR Grant #N00014-02-1-0720.

REFERENCES

1. Abgrall, R. (1996). Numerical discretization of the first-order Hamilton–Jacobi equation on triangular meshes. *Commun. Pure Appl. Math.* **49**, 1339–1373.
2. Augoula, S., and Abgrall, R. (2000). High order numerical discretization for Hamilton–Jacobi equations on triangular meshes. *J. Sci. Comput.* **15**, 197–229.
3. Boué, M., and Dupuis, P. (1999). Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control. *SIAM J. Numer. Anal.* **36**, 667–695.
4. Bryson, S., and Levy, D. (2003). High-order central WENO schemes for multidimensional Hamilton–Jacobi equations. *SIAM J. Numer. Anal.* **41**, 1339–1369.
5. Barth, T., and Sethian, J. (1998). Numerical schemes for the Hamilton–Jacobi and level set equations on triangulated domains. *J. Comput. Phys.* **145**, 1–40.

6. Cecil, T., Qian, J., and Osher, S. (2004). Numerical methods for high dimensional Hamilton–Jacobi equations using radial basis functions. *J. Comput. Phys.* **196**, 327–347.
7. Crandall, M. G., and Lions, P. L. (1983). Viscosity solutions of Hamilton–Jacobi equations. *Trans. Amer. Math. Soc.* **277**, 1–42.
8. Dellinger, J., and Symes, W. W. (1997). Anisotropic finite-difference traveltimes using a Hamilton–Jacobi solver, 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1786–1789.
9. Falcone, M., and Ferretti, R. (1994). Discrete time high-order schemes for viscosity solutions of Hamilton–Jacobi–Bellman equations. *Numer. Math.* **67**, 315–344.
10. Falcone, M., and Ferretti, R. (2002). Semi-Lagrangian schemes for Hamilton–Jacobi equations, discrete representation formulae and Godunov methods. *J. Comput. Phys.* **175**, 559–575.
11. Gray, S., and May, W. (1994). Kirchhoff migration using eikonal equation travel-times. *Geophysics* **59**, 810-817.
12. Helmsen, J., Puckett, E., Colella, P., and Dorr, M. (1996) Two new methods for simulating photolithography development in 3d, Proc. SPIE, 2726: 253–261.
13. Hu, C., and Shu, C.-W. (1999). A discontinuous Galerkin finite element method for Hamilton–Jacobi equations. *SIAM J. Sci. Comput.* **20**, 666–690.
14. Jiang, G.-S., and Peng, D. (2000). Weighted ENO Schemes for Hamilton–Jacobi equations, *SIAM* J. Sci. Comput. **21**, 2126–2143.
15. Jiang, G.-S., and Shu, C.-W. (1996). Efficient implementation of weighted ENO schemes. *J. Comput. Phy.* **126**, 202–228.
16. Jin, S., and Xin, Z. (1998). Numerical passage from systems of conservation laws to Hamilton–Jacobi equations and relaxation schemes. *SIAM J. Numer. Anal.* **35**, 2385–2404.
17. Kim, S., and Cook, R. (1999). 3D traveltime computation using second-order ENO scheme. *Geophys.* **64**, 1867–1876.
18. Kao, C. Y., Osher, S., and Qian, J. (2004). Lax-Friedrichs sweeping scheme for static Hamilton–Jacobi equations. *J. Comput. Phys.* **196**, 367–391.
19. Kao, C. Y., Osher, S., and Tsai, Y. H. (2005). Fast sweeping methods for Hamilton–Jacobi equations. *SIAM J. Numer. Anal.* **42**, 2612–2632.
20. Lin, C.-T., and Tadmor, E. (2000). High-resolution non-oscillatory central schemes for approximate Hamilton–Jacobi equations. *SIAM J. Sci. Comput.* **21**, 2163–2186.
21. Lin, C.-T., and Tadmor, E. (2001). L^1-stability and error estimates for approximate Hamilton–Jacobi solutions. *Numer. Math.* **87**, 701–735.
22. Osher, S. (1993). A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations. *SIAM J. Math. Anal.* **24**, 1145–1152.
23. Osher, S., and Sethian, J. (1988). Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.* **79**, 12–49.
24. Osher, S., and Shu, C.-W. (1991). High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations. *SIAM J. Numer. Anal.* **28**, 907–922.
25. Peng, D., Osher, S., Merriman, B., Zhao, H.-K., and Rang, M. (1999). A PDE-based fast local level set method. *J. Comput. Phys.* **155**, 410–438.
26. Qian, J., Cheng, L. T., and Osher, S. (2003). A level set based Eulerian approach for anisotropic wave propagations. *Wave Motion* **37**, 365–379.
27. Qian, J., and Symes, W. W. (2001). Paraxial eikonal solvers for anisotropic quasi-P travel times. *J. Comput. Phys.* **174**, 256–278.
28. Qian, J., and Symes, W. W. (2002). Finite-difference quasi-P traveltimes for anisotropic media. *Geophysics* **67**, 147–155.

29. Qian, J., and Symes, W. W. (2002). An adaptive finite-difference method for traveltime and amplitude. *Geophysics* **67**, 166–176.
30. Qin, F., and Schuster, G. T. (1993). First-arrival traveltime calculation for anisotropic media. *Geophysics* **58**, 1349–1358.
31. Rouy, E., and Tourin, A. (1992). A viscosity solutions approach to shape-from-shading. *SIAM J. Numer. Anal.* **29**, 867–884.
32. Sethian, J. A. (1996). A fast marching level set method for monotonically advancing fronts. *Proc. Nat. Acad. Sci.* **93**, 1591–1595.
33. Sethian, J. A., and Vladimirsky, A. (2001). Ordered upwind methods for static Hamilton–Jacobi equations. *Proc. Natl. Acad. Sci.* **98**, 11069–11074.
34. Sethian, J. A., and Vladimirsky, A. (2003). Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms. *SIAM J. Numer. Anal.* **41**, 325–363.
35. Chi-Wang Shu, (2004). *High Order Numerical Methods for Time Dependent Hamilton–Jacobi Equations*, WSPC/Lecture Notes Series.
36. Shu, C.-W., and Osher, S. (1988). Efficient Implementation of essentially non-oscillatory shock-capturing schemes. *J. Comput. Phys.* **77**, 439–471.
37. Tsai, Y.-H. R., Cheng, L.-T., Osher, S., and Zhao, H.-K. (2003). Fast sweeping algorithms for a class of Hamilton–Jacobi equations. *SIAM J. Numer. Anal.* **41**, 673–694.
38. Tsitsiklis, J. N. (1995). Efficient algorithms for globally optimal trajectories. *IEEE Trans. Auto. Control* **40**, 1528–1538.
39. Zhang, Y.-T., and Shu, C.-W. (2003). High order WENO schemes for Hamilton–Jacobi equations on triangular meshes. *SIAM J. Sci. Comput.* **24**, 1005–1030.
40. Zhao, H. (2004). A fast sweeping method for Eikonal equations. *Math. Comp.* **74**, 603–627.
41. Zhao, H., Osher, S., Merriman, B., and Kang, M. (2000). Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Comput. Vision Image Understand.* **80**, 295–319.