Journal of Scientific Computing, Vol. 31, Nos. 1/2, May 2007 (© 2007)
DOI: 10.1007/s10915-006-9124-6

A Fast Sweeping Method for Static Convex Hamilton–Jacobi Equations*

Jianliang Qian,[1] Yong-Tao Zhang,[2] and Hong-Kai Zhao[3]

Received May 23, 2006; accepted (in revised form) November 15, 2006; Published online March 9, 2007

We develop a fast sweeping method for static Hamilton–Jacobi equations with convex Hamiltonians. Local solvers and fast sweeping strategies apply to structured and unstructured meshes. With causality correctly enforced during sweepings numerical evidence indicates that the fast sweeping method converges in a finite number of iterations independent of mesh size. Numerical examples validate both the accuracy and the efficiency of the new method.

KEY WORDS: Hamilton–Jacobi; fast sweeping; monotone scheme.

1. INTRODUCTION

We consider a class of static Hamilton–Jacobi equations of the following form,

$$\begin{aligned} H(\mathbf{x}, \nabla T(\mathbf{x})) &= 1, \quad \mathbf{x} \in \Omega \setminus \Gamma, \\ T(\mathbf{x}) &= g(\mathbf{x}), \quad \mathbf{x} \in \Gamma \subset \Omega, \end{aligned} \tag{1.1}$$

where $g(\mathbf{x})$ is a positive, Lipschitz continuous function, Ω is an open, bounded polygonal domain in R^d, and Γ is a subset of Ω; $H(\mathbf{x}, \mathbf{p})$ is Lipschitz continuous in both arguments, and it is convex in the second argument. If $H(\mathbf{x}, \mathbf{p}) = |\mathbf{p}| H(\mathbf{x}, \mathbf{p}/|\mathbf{p}|) = |\mathbf{p}| F(\mathbf{x})$, then the eikonal equation

* In memory of Xu-Dong Liu.

[1] Department of Mathematics and Statistics, Wichita State University, Wichita, KS 67260, USA. E-mail: qian@math.wichita.edu

[2] Department of Mathematics, University of Notre Dame, Notre Dame, IN 46556-4618, USA. E-mail: yzhang10@nd.edu

[3] Department of Mathematics, University of California, Irvine, CA 92697-3875, USA. E-mail: zhao@math.uci.edu

for isotropic wave propagation results; otherwise, the so-called anisotropic eikonal equation results.

Such equations arise in a multitude of applications, ranging from seismic waves, crystal growth, robotic navigation to optimal control. Consequently, it is necessary to develop accurate and efficient numerical methods for this nonlinear boundary value problem. In this paper, we extend the fast sweeping method [1,33,28,13,14,24] to solve the above static Hamilton–Jacobi equation on triangular meshes.

To tackle such anisotropic eikonal equations in seismics, Dellinger [7] extended upwind finite difference methods [30] to compute the first-arrival based viscosity solutions in an anisotropic medium; Qin and Schuster [25] proposed a wavefront expansion method, and it is based on Huygens' principle and computes the first-arrival traveltimes associated with seismic energy propagating at the group velocity. As pointed out in Dellinger and Symes [8], an anisotropic medium is different from an isotropic medium in that for an isotropic medium the ray velocity vector (i.e., the group velocity vector, or the characteristic direction) has the same direction as the (negative) traveltime gradient (i.e., the phase velocity vector), which enables us to use the traveltime gradient as a reliable indicator of energy flow (and thus the causality) in propagating the traveltime field [30,25,29, 26,33], while for an anisotropic medium this is no longer true. Therefore one may compute wrong solutions by extending fast marching methods designed for isotropic eikonal equations to anisotropic eikonal equations without taking into account the above essential differences as demonstrated in [27].

Based on the above observation, Qian and Symes [18,20–22] proposed a paraxial formulation for the static Hamilton–Jacobi equation by formulating a relation between the characteristic direction and the traveltime gradient direction, so that fast, efficient, and accurate methods with linear complexity can be obtained easily; furthermore, Qian *et al.* [23] made further improvement by removing the paraxial assumption.

On the other hand, Sethian and Vladimirsky [27] designed ordered upwind methods for the above static Hamilton–Jacobi equation. The spirit of their single-pass method is the following: at a considered node which is to be updated, first one estimates the possible numerical domain of dependency by using the so-called anisotropic coefficient [27], the accepted solution and the mesh size; secondly one uses the so-called control-theoretic update-from-a-single-simplex formula [9] to evaluate a tentative value at the standing node by taking the minimum among all possible values resulting from all the possible virtual simplexes constructed from its numerical domain of dependency; thirdly one accepts the smallest value of all the considered nodes to maintain causality; lastly one maintains the lists of accepted solutions and considered nodes. The resulting ordered upwind methods have the computational complexity of $O(\eta M \log M)$, where η is the anisotropic coefficient depending on the Hamilton–Jacobi equation, and M is the total number of mesh points.

As an iterative method for Hamilton–Jacobi equations, the fast sweeping method was originated in Boue and Dupuis [1], and its first PDE formulation was for implicit and nonparametric shape reconstruction from unorganized points using a variational level set method [35]; Zhao [33] proved the $O(N)$ convergence of the method for the eikonal equation based on the Godunov Hamiltonian on Cartesian meshes. Tsai *et al.* [28] applied the fast sweeping methods to a class of static Hamilton–Jacobi equations based on Godunov numerical Hamiltonians on uniform meshes, and they have derived some explicit update formulae so that the Gauss–Seidel based sweeping strategy can be easily carried out; numerical examples indicate that the sweeping method has linear complexity. Kao *et al.* [13] proposed a class of fast sweeping methods for the static Hamilton–Jacobi equations based on discretizing the Bellman formula resulting from the Hamiltonian directly on uniform meshes; numerical examples also indicate that the sweeping method has linear complexity. Kao *et al.* [14] have extended fast sweeping methods to deal with nonconvex Hamilton–Jacobi equations based on Lax–Friedrichs numerical Hamiltonians on uniform meshes. Zhang *et al.* [32] have developed higher-order fast sweeping methods based on weighted essentially nonoscillatory schemes [17, 12, 11] on uniform meshes. Zhang *et al.* [31] proposed fixed-point type sweeping methods on uniform meshes. All of the above cited fast sweeping methods are based upon uniform meshes. In [24] a class of novel fast sweeping methods was developed for isotropic eikonal equations on triangular meshes for the first time. In [3], Cecil *et al.* extended the fast sweeping method to deal with level set equations on adaptive tree-based unstructured meshes. Various parallel implementations of the fast sweeping method are developed in [34].

Mathematical foundation for the well-posedness of the Hamilton–Jacobi equation traces back to the theory of viscosity solution [5], and computability of viscosity solution by monotone finite difference methods is established in [6]. There are two crucial tasks in developing an efficient numerical method for such type of equations: one is designing a local solver or discretization scheme that can capture causality of the solution due to the hyperbolic nature of the equation, and the other is solving the resulting large system of nonlinear equations after discretization. Many finite difference schemes are available for discretizations on rectangular grids. In most of these schemes causality and consistency are coupled and enforced simultaneously in the local discretization. Due to nonlinearity of the equation, the corresponding local solver may be quite complicated.

In this paper, we develop a local solver that decouples consistency and causality. This approach allows one to deal with much more general Hamiltonians and applies to both structured and unstructured grids. Since we are solving a nonlinear boundary value problem, a large nonlinear system needs solving after discretization. We apply the fast sweeping strategy developed in [24] to solve the system which gives an efficient and unconditionally stable iterative method. In particular the fast sweeping method is an iterative method of Gauss–Seidel type, consisting of correct causality check and alternating sweepings. The key point is that all characteristics can be divided into a finite number of groups and each group can be captured simultaneously by one of the orderings. Recently this methodology has been studied extensively and has also been applied successfully to other hyperbolic problems [15].

Here is the outline of this paper. In Sect. 2, a local solver for general convex Hamiltonians is described. The fast sweeping algorithm is summarized in Sect. 3. Explicit formulae are derived for a class of anisotropic eikonal equations in Sect. 4. Finally numerical examples are shown in Sect. 5 to demonstrate both the efficiency and the accuracy of our method.

2. LOCAL SOLVERS

2.1. Some Basic Facts

Because Eq. (1.1) arises naturally from geometrical optics for wave propagation [2], without any hesitation we decide to adopt some common terminology from geometrical optics in the following presentation.

For the sake of simplicity in the following derivation we assume that H is strictly convex and homogeneous of degree one; we will comment on general cases later.

Wavefronts of the traveltime are level sets defined by

$$\{\mathbf{x} \in \Omega : T(\mathbf{x}) = t, t \in \mathcal{R}\}.$$

The wavefront normal at a point $\mathbf{x} \in \Omega$ is

$$\mathbf{n}(\mathbf{x}) = \frac{\nabla T(\mathbf{x})}{|\nabla T(\mathbf{x})|},$$

whenever the gradient of traveltime T is well-defined, where $\mathbf{p} = \nabla T(\mathbf{x})$ is the slowness vector because it has the dimension of the reciprocal of velocity. $V_{\mathrm{p}}(\mathbf{x},\mathbf{n}(\mathbf{x}))=\frac{1}{|\nabla T(\mathbf{x})|}$ is the so-called phase speed. Thus we have

$$\mathbf{p}(\mathbf{x}) = \frac{\mathbf{n}(\mathbf{x})}{V_p(\mathbf{x},\mathbf{n}(\mathbf{x}))}, \tag{2.1}$$

$$V_{\mathrm{p}}(\mathbf{x},\mathbf{n}(\mathbf{x})) = H(\mathbf{x},\mathbf{n}(\mathbf{x})), \tag{2.2}$$

$$H(\mathbf{x},\mathbf{p}) = 1. \tag{2.3}$$

Since Eq. (1.1) is a nonlinear first-order equation, applying the method of characteristics to the equation in phase space along ray trajectories $(\mathbf{x}(t),\mathbf{p}(t))$ yields

$$\frac{d\mathbf{x}}{dt} = \nabla_{\mathbf{p}}H, \tag{2.4}$$

$$\frac{d\mathbf{p}}{dt} = -\nabla_{\mathbf{x}}H, \tag{2.5}$$

$$\frac{dT}{dt} = \mathbf{p}\cdot\frac{d\mathbf{x}}{dt} = \mathbf{p}\cdot\nabla_{\mathbf{p}}H = 1. \tag{2.6}$$

The first equation defines the so-called group velocity vector, which points into the same direction as the characteristic (ray) direction. Its magnitude is

$$v_{\mathrm{g}}(\mathbf{x},\mathbf{p}) = \left|\frac{d\mathbf{x}}{dt}\right| = |\nabla_{\mathbf{p}}H|, \tag{2.7}$$

which is the so-called group speed depending on the position $\mathbf{x}$ and the slowness vector $\mathbf{p}$, so that the group speed varies as the traveltime gradient does, implying the so-called directional dependence.

In a homogeneous anisotropic medium, $H(\mathbf{x},\mathbf{p})=H(\mathbf{p})$, the traveltime increment ΔT between any given two points of distance Δd is computed by the following relation:

$$\Delta T = \frac{\Delta d}{v_{\mathrm{g}}(\mathbf{p})}, \tag{2.8}$$

where $\mathbf{p}$ is determined implicitly by the condition that the ray direction is known which is the unit vector along the straight line connecting the two given points. In general, we have to use a numerical procedure to compute the above traveltime; for example, see [18].

Remark. In an isotropic medium the group velocity vector (the ray direction) and the phase velocity vector (the traveltime gradient) are in the same direction, and the group speed and the phase speed are equal. As a result the computation of traveltime between any two points is straightforward. In an anisotropic medium those properties are no longer true.

Remark. If $\lim_{\lambda\to 0} \nabla_{\mathbf{p}} H(\mathbf{x}, \lambda\mathbf{p}) = 0$, then the strict convexity of $H(\mathbf{x}, \mathbf{p})$ implies that

$$\big(\nabla_{\mathbf{p}} H(\mathbf{x}, \mathbf{p}) - \nabla_{\mathbf{p}} H(\mathbf{x}, \lambda\mathbf{q})\big) \cdot (\mathbf{p} - \lambda\mathbf{q}) > 0 \quad \text{for } \forall \mathbf{p}, \mathbf{q} \tag{2.9}$$

consequently, we have

$$\mathbf{p} \cdot \nabla_{\mathbf{p}} H(\mathbf{x}, \mathbf{p}) \geqslant 0 \quad \text{for } \forall \mathbf{p}. \tag{2.10}$$

Therefore without using the homogeneity of H in $\mathbf{p}$ we conclude that the solution value is nondecreasing along ray trajectories. Furthermore, if $\mathbf{p} \cdot \nabla_{\mathbf{p}} H(\mathbf{x}, \mathbf{p}) > 0$, then the solution value can be used as the running parameter along ray trajectories in the above formulation. This is the most essential condition for the fast sweeping method to work for Hamilton–Jacobi equations. Hence all the following algorithmic development applies as long as $\lim_{\lambda\to 0} \nabla_{\mathbf{p}} H(\mathbf{x}, \lambda\mathbf{p}) = 0$ holds.

2.2. A Local Solver Based on Fermat's Principle

We treat the two-dimensional case first. We consider a triangulation $\mathcal{T}_h$ of Ω into non-overlapping, nonempty, and closed triangles $\mathcal{T}$, with diameter $h_{\mathcal{T}}$ which is the longest side of the triangle $\mathcal{T}$, such that $\bar{\Omega} = \cup_{\mathcal{T} \in \mathcal{T}_h} \mathcal{T}$. We assume that $\mathcal{T}_h$ satisfies the following conditions:

- there are no obtuse triangles;
- no more than μ triangles have a common vertex, and the intersection of any two triangles is either empty, a single vertex or an entire edge;
- $h = \sup_{\mathcal{T} \in \mathcal{T}_h} h_{\mathcal{T}} < 1$;
- $\mathcal{T}_h$ is regular in the sense that there exists a constant ω_0, independent of h, such that if $\rho_{\mathcal{T}}$ is the diameter of the largest ball $B \subset \mathcal{T}$, then for all $\mathcal{T} \in \mathcal{T}_h$, $h_{\mathcal{T}} \leqslant \omega_0 \rho_{\mathcal{T}}$.

Therefore, Eq. (1.1) is solved in the domain Ω, which has a triangulation $\mathcal{T}_h$. We consider every vertex and all the triangles which are directly associated with this vertex; see Fig. 1 for a vertex C and its n triangles $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$. For a typical triangle $\triangle ABC$ we denote $A : (x_A, y_A)$, $B : (x_B, y_B)$, and $C : (x_C, y_C)$; $\angle A = \alpha$, $\angle B = \beta$, and $\angle C = \gamma$; $\overline{AB} = c$, $\overline{AC} = b$, and $\overline{BC} = a$ are the lengths of the edges AB, AC and BC, respectively.

During the solution process we need a local solver at the vertex C for each triangle; see Fig. 2. Given the values T_A and T_B at A and B of triangle $\triangle ABC$, we want to calculate the value T_C at C.

If T_A and T_B are used to update T_C, then there must be a ray emanating from the segment AB and hitting point C; namely there is an

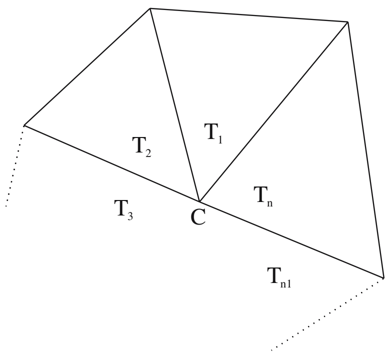


Fig. 1. The vertex C and the local mesh.

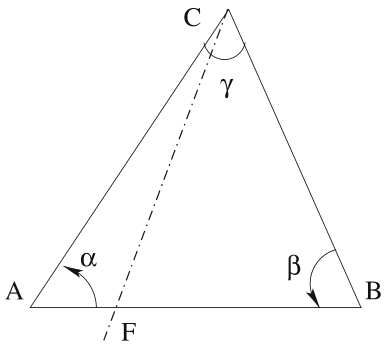


Fig. 2. Update the value at C in a triangle.

$F(s)$ located in between A and B, where s parametrizes the segment $\overline{AB}$: $F(0)=A$ and $F(1)=B$.

According to Fermat's principle the traveltime at C is given by minimizing the functional

$$T_C(s) = sT_B + (1-s)T_A + \frac{d(s)}{v_{\mathrm{g}}(C)} \tag{2.11}$$

with respect to s, where

$$d(s)=\overline{CF(s)} = \sqrt{b^2+c^2s^2-2bcs\cos\alpha}, \tag{2.12}$$

$$v_g(C;s) = v_{\mathrm{g}}(C;T_C(s),T_A,T_B) \tag{2.13}$$

namely,

$$T_C = \min_{s\in[0,1]} \left\{ sT_B + (1-s)T_A + \frac{d(s)}{v_g(C;s)} \right\}. \tag{2.14}$$

This is the so-called control-theoretic update-from-a-single-simplex formula as used in [9,27].

The main difficulty in implementing this formula is that we have to compute the group speed $v_g(C;s)$ by using the current ray direction defined by C and $F(s)$. By the above formula we can immediately conclude that

$$T_C \geqslant \min\{T_A, T_B\} \tag{2.15}$$

if there exists a characteristic emanating from the segment AB to hit point C. However, since the wave front normal ∇T does not coincide with the characteristic direction in general, ∇T may not fall into the triangle; therefore, it is not necessarily true that

$$T_C \geqslant \max\{T_A, T_B\}. \tag{2.16}$$

These two facts are observed in our numerical examples.

Remark. In the special case of the eikonal equation, the wave front normal coincides with the direction of characteristics and hence either one can be used to check causality condition; since ∇T points away from C and is in between the two sides CA and CB a causality-satisfying T_C must be larger than $\max\{T_A, T_B\}$.

Remark. The above formulation involves optimization which is avoidable by adopting a fully Eulerian viewpoint.

2.3. A Local Solver Based on an Eulerian discretization

Consider an acute triangle $\triangle ABC$. By definition we have

$$\frac{T_C - T_A}{b} = \nabla T(C) \cdot \left(\frac{x_C - x_A}{b}, \frac{y_C - y_A}{b} \right)^t + O(h^2), \tag{2.17}$$

$$\frac{T_C - T_B}{a} = \nabla T(C) \cdot \left(\frac{x_C - x_B}{a}, \frac{y_C - y_B}{a} \right)^t + O(h^2), \tag{2.18}$$

where t denotes transpose of vectors and matrices. Furthermore we have

$$\begin{pmatrix} \frac{T_C - T_A}{b} \\ \frac{T_C - T_B}{a} \end{pmatrix} = \mathbf{P}\nabla T(C) + O(h^2), \tag{2.19}$$

where

$$\mathbf{P} = \begin{pmatrix} \frac{x_C - x_A}{b}, & \frac{y_C - y_A}{b} \\ \frac{x_C - x_B}{a}, & \frac{y_C - y_B}{a} \end{pmatrix} \equiv \begin{pmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \end{pmatrix}. \tag{2.20}$$

Assuming a linear approximation of T locally near C to ignore higher-order terms and solve for ∇T_C, we have

$$\nabla T(C) \approx \mathbf{P}^{-1} \begin{pmatrix} \frac{T_C - T_A}{b} \\ \frac{T_C - T_B}{a} \end{pmatrix}, \tag{2.21}$$

where $\mathbf{P}^{-1} = \mathbf{P}^t\mathbf{Q}$,

$$\mathbf{P}\mathbf{P}^{\mathbf{t}} = \begin{pmatrix} 1, & \cos\gamma \\ \cos\gamma, & 1 \end{pmatrix}$$

and

$$\mathbf{Q} = (\mathbf{P}\mathbf{P}^{\mathbf{t}})^{-1} = \frac{1}{\sin^2\gamma} \begin{pmatrix} 1, & -\cos\gamma \\ -\cos\gamma, & 1 \end{pmatrix}.$$

Inserting $\nabla T(C)$ into the Hamilton–Jacobi equation at the mesh point C, we have a consistent discretization of the equation in the triangle $\triangle ABC$:

$$H(C, \nabla T(C)) \approx 1, \tag{2.22}$$

$$\hat{H}\left(C, \frac{T_C - T_A}{b}, \frac{T_C - T_B}{a}\right) = 1 \tag{2.23}$$

or

$$\hat{H}(C, T_C, T_A, T_B) = 1. \tag{2.24}$$

Since in general this is a nonlinear equation for T_C, we have to numerically solve the equation to obtain T_C if T_A and T_B are given.

It is possible that we have multiple solutions of T_C when solving the nonlinear Eq. (2.24); thus we have to choose the one that satisfies causality, the so-called characteristic condition.

The fundamental idea is the following. If there is no solution to Eq. (2.24) it means that this triangle does not support a consistent T_C. If there is one or more solutions we need to check the causality condition. Using the computed T_C and the above equation, we get a $\nabla T(C)$ which can be used to compute the ray direction $\nabla_{\mathbf{p}} H(C, \nabla T(C))$. The computed T_C satisfies the **causality condition** if the characteristic starting from C against the direction $\nabla_{\mathbf{p}} H(C, \nabla T(C))$ intersects the line segment AB, as illustrated in Fig. 2. If there is no T_C satisfying this causality condition it means that this triangle does not support a T_C that is both consistent and causality satisfying. If, after the causality check, there are multiple T_C's from this triangle, then we choose the smallest one using the first-arrival time principle.

Now we rigorously establish the above causality principle.

Since $H(\mathbf{x}, \mathbf{p})$ is strictly convex in the $\mathbf{p}$ argument, for a given $\mathbf{x}$ any straight line in the slowness space intersects the slowness surface defined by $H(\mathbf{x}, \mathbf{p}) = 1$ at two points at most; see Fig. 3(a). In general, there are three cases:

1. two different intersections;
2. two repeated identical intersections;
3. no intersection.

In the first case two outward normals $\nabla_{\mathbf{p}} H$ at those two intersections give two possible characteristic directions according to Eq. (2.4) at the given point $\mathbf{x}$. In the second case, we have two repeated identical intersections, and the outward normal at the intersection point also points into a characteristic direction at that point $\mathbf{x}$.

We have the following geometrical observation. For a given $\mathbf{x}$, the slowness surface $\{\mathbf{p}: H(\mathbf{x}, \mathbf{p}) = 1\}$ is strictly convex and closed; there are exactly two horizontal and two vertical lines in the slowness space which

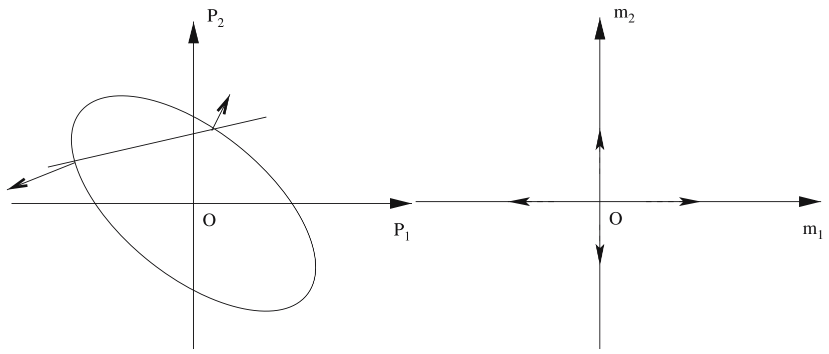


Fig. 3. (**a**). A straight line intersects the convex slowness surface at two points at most. (**b**) Embed the outward normals into a Cartesian coordinate system.

are tangent to the slowness surface. This fact can be proved rigorously by writing sectors of the slowness surface as graph functions; see [22].

According to the above observation, we have four tangent points denoted as $\mathbf{p}^{(i)} = (p_1^{(i)}, p_2^{(i)})$, $(i = 1,\ldots,4)$, yielding

$$\left(H_{p_1}(\mathbf{x}, \mathbf{p}^{(1)}) > 0,\, H_{p_2}(\mathbf{x}, \mathbf{p}^{(1)}) = 0\right),$$

$$\left(H_{p_1}(\mathbf{x}, \mathbf{p}^{(2)}) = 0,\, H_{p_2}(\mathbf{x}, \mathbf{p}^{(2)}) > 0\right),$$

$$\left(H_{p_1}(\mathbf{x}, \mathbf{p}^{(3)}) < 0,\, H_{p_2}(\mathbf{x}, \mathbf{p}^{(3)}) = 0\right)$$

and

$$\left(H_{p_1}(\mathbf{x}, \mathbf{p}^{(4)}) = 0,\, H_{p_2}(\mathbf{x}, \mathbf{p}^{(4)}) < 0\right).$$

Next we can embed the above four vectors into a two-dimensional m_1-m_2 Cartesian coordinate system by making $(m_1, m_2) = (1, 0)$ and $(m_1, m_2) = (0, 1)$ point into the same direction as $\nabla_{\mathbf{p}} H(\mathbf{x}, \mathbf{p}^{(1)})$ and $\nabla_{\mathbf{p}} H(\mathbf{x}, \mathbf{p}^{(2)})$, respectively. Accordingly, $(m_1, m_2) = (-1, 0)$ and $(m_1, m_2) = (0, -1)$ have the same direction as $\nabla_{\mathbf{p}} H(\mathbf{x}, \mathbf{p}^{(3)})$ and $\nabla_{\mathbf{p}} H(\mathbf{x}, \mathbf{p}^{(4)})$, respectively. See Fig. 3(b). By this embedding each nonzero vector in this coordinate system defines a possible ray direction at $\mathbf{x}$.

Now consider the straight line Eq. (2.21) parametrized by T_C in the slowness space, which is rewritten as

$$\mathbf{p} = \mathbf{r}\, T_C + \mathbf{r}_0, \tag{2.25}$$

where

$$\mathbf{r} = \mathbf{P}^{-1} \begin{pmatrix} b^{-1} \\ a^{-1} \end{pmatrix} = \frac{a - b\cos\gamma}{a\, b\, \sin^2\gamma}\, \mathbf{r}_1 + \frac{b - a\cos\gamma}{a\, b\, \sin^2\gamma}\, \mathbf{r}_2, \tag{2.26}$$

$$\mathbf{r}_0 = \mathbf{P}^{-1} \begin{pmatrix} -T_A\, b^{-1} \\ -T_B\, a^{-1} \end{pmatrix} = \frac{b\, T_B \cos\gamma - a\, T_A}{a\, b\, \sin^2\gamma}\, \mathbf{r}_1 + \frac{a\, T_A \cos\gamma - b\, T_B}{a\, b\, \sin^2\gamma}\, \mathbf{r}_2. \tag{2.27}$$

This straight line intersects the slowness surface in exactly the same way as described above in terms of the following three cases.

The first case: two different outward normals. In general there are three situations, referring to Fig. 3(b):

- **Case (i):** two outward normals are in the same quadrant in the m_1-m_2 coordinate system;
- **Case (ii):** two outward normals are in the neighboring quadrants in the m_1-m_2 coordinate system;
- **Case (iii):** two outward normals are in the opposite quadrants in the m_1-m_2 coordinate system.

Consider an acute triangle $\triangle ABC$ in Cartesian coordinates with the origin at the mesh point C.

In Case (i), we have two possible rays hitting C, and we choose the one yielding the smallest traveltime using the first-arrival time principle.

In Cases (ii) and (iii), these two outward normals must have *either* opposite signs in the second component, which correspond to downgoing rays and upgoing rays, respectively, *or* have opposite signs in the first component which correspond to left-going and right-going rays, respectively. We show that only one of those two ray directions may satisfy the causality condition, since the acute triangle must belong to one of the following cases:

1. if $\triangle ABC$ is completely located in the upper half plane, then a downgoing ray emanating from the segment AB should be selected to hit point C;
2. if $\triangle ABC$ is completely located in the lower half plane, then a upgoing ray emanating from the segment AB should be selected to hit point C;
3. if $\triangle ABC$ is located completely in the right half plane, then a left-going ray emanating from the segment AB should be selected to hit point C;
4. if $\triangle ABC$ is located completely in the left half plane, then a right-going ray emanating from the segment AB should be selected to hit point C.

Finally to ensure that the characteristic indeed emanates from the segment AB, one has to verify that the selected outward normal allows a ray to start from point C and intersect with the segment AB. If the above is true, we accept the corresponding T_C; if not we use the two rays along the edges AC and BC, respectively, to hit C; this will be dealt with in the **third case**.

The second case: two identical outward normals. We can use the similar arguments as in the **first case** to decide whether to accept the corresponding characteristic or not; if not we go to the **third case**.

The third case: no valid outward normals. In this case, we force rays to travel along either edge AC or edge BC. Since the ray direction is given and the Hamiltonian is convex, we can find the corresponding group speed by inversion of the relation (2.4) (see [18] for more details). Then the traveltime at C is given by

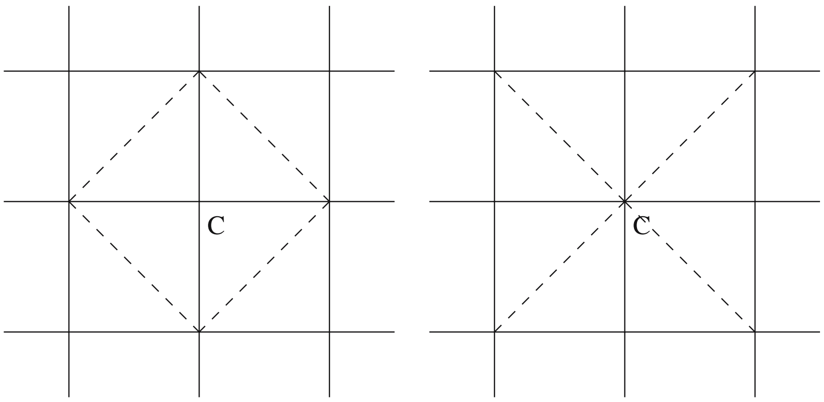
$$T_C = \min\left\{T_A + \frac{|AC|}{v_g^{AC}}, T_B + \frac{|BC|}{v_g^{BC}}\right\}, \tag{2.28}$$

where v_g^{AC} and v_g^{BC} denote the group speed along the edge AC and BC, respectively.

Remark. Rectangular grids can be considered as special cases. There are two possible virtual triangulations on a rectangular grid as illustrated in Fig. 4 for two spatial dimensions. In Case (a) four triangles are created, analogous to a five-point stencil used in finite-difference schemes. In Case (b) eight triangles are connected which result in a nine-point stencil. Case (b) will give more accurate solutions on the same Cartesian grid than Case (a) due to better directional resolution. As we will show in our numerical examples, the gain in accuracy of Case (b) justifies the extra cost compared to Case (a).

We can summarize the above into an algorithm for updating an acute triangle.

A two-dimensional local solver: (given T_A and T_B, determine $T_C = T_C(T_A, T_B)$.)



(a) Four three-point stencils. **(b)** Eight three-point stencils.

Fig. 4. Triangulation based on regular meshes.

1. Solve Eq. (2.24) for two possible roots, T_C^1 and T_C^2, either analytically or numerically.
2. If there are two roots, T_C^1 and T_C^2, then
   (a) if T_C^1 satisfies the characteristic condition, then
   $$T_C = \min\{T_C, T_C^1\};$$
   (b) if T_C^2 satisfies the characteristic condition, then
   $$T_C = \min\{T_C, T_C^2\};$$
   (c) if none of the two roots satisfies the characteristic condition, then
   $$T_C = \min\left\{T_C, T_A + \frac{|AC|}{v_g^{AC}}, T_B + \frac{|BC|}{v_g^{BC}}\right\};$$
3. else
$$T_C = \min\left\{T_C, T_A + \frac{|AC|}{v_g^{AC}}, T_B + \frac{|BC|}{v_g^{BC}}\right\}.$$

2.4. Consistency and Monotonicity

Considering an acute triangle $\triangle ABC$ in which T_A and T_B are given, we update the travel-time T_C at the vertex C. Denoting

$$q_1 = \frac{T_C - T_A}{b}, \qquad q_2 = \frac{T_C - T_B}{a}, \quad \mathbf{q} = (q_1, q_2)^t$$

we adopt the framework given in [4] to show consistency and monotonicity of the Godunov numerical Hamiltonian resulting from the local solver.

Lemma 2.1 (Consistency and Monotonicity). The numerical Hamiltonian $\hat{H}$ is consistent:

$$\hat{H}\left(C, \frac{T_C - T_A}{b}, \frac{T_C - T_B}{a}\right) = H(C, \mathbf{p}) \tag{2.29}$$

if $\nabla T_h = \mathbf{p} \in \mathcal{R}^2$, where T_h is the numerical solution defined by linear interpolation at the three vertexes on each triangle. The numerical Hamiltonian $\hat{H}$ constructed in the local solver is monotone if the causality condition holds.

Proof. By $\nabla T_h = \mathbf{p} \in \mathcal{R}^2$, we have

$$\mathbf{q} = \begin{pmatrix} \frac{T_C - T_A}{b} \\ \frac{T_C - T_B}{a} \end{pmatrix} = \mathbf{P}\mathbf{p}. \tag{2.30}$$

Inserting this into the numerical Hamiltonian, we have Eq. (2.29).

Differentiating $\hat{H}(C, q_1, q_2)$ with respect to q_1 and q_2, the monotonicity of the Hamiltonian requires

$$\frac{\partial \hat{H}(C, q_1, q_2)}{\partial q_1} \geqslant 0, \qquad \frac{\partial \hat{H}(C, q_1, q_2)}{\partial q_2} \geqslant 0 \tag{2.31}$$

since $\hat{H}(C, \mathbf{q}) = H(C, \mathbf{P}^{-1}\mathbf{q})$, the above inequalities can be satisfied if and only if the following holds component-wise:

$$(\mathbf{P}^t)^{-1} \nabla_{\mathbf{p}} H \geqslant \mathbf{0} \tag{2.32}$$

namely,

$$\begin{aligned} \nabla_{\mathbf{p}} H \cdot (\mathbf{r}_1 - \mathbf{r}_2 \cos\gamma) \geqslant 0, \\ \nabla_{\mathbf{p}} H \cdot (\mathbf{r}_2 - \mathbf{r}_1 \cos\gamma) \geqslant 0. \end{aligned}$$

The last inequalities mean that the characteristic direction $\nabla_{\mathbf{p}} H$ at C is a linear combination with positive coefficients of the two vectors CA and CB, which is exactly the condition that we have imposed in choosing the characteristic direction in the local solver, the upwind condition. □

We note that the above monotonicity in turn implies the monotonicity of T_C with respect to T_A and T_B by the implicit function theorem.

2.5. How to Compute the Group Speed?

In the above discussions, we have to compute the group speed from a given ray direction. Since in general anisotropic media we cannot find an explicit formula for the group speed in terms of a given ray direction, we have to use a numerical procedure to determine the group speed approximately.

Fortunately for the static Hamilton–Jacobi equation with the Hamiltonian H being convex in the gradient argument we can easily modify a shooting method [18] to compute the group speed from a given ray direction which is uniquely determined by two given points in a homogeneous medium. Since the algorithm is well explained in [18], we will not pursue it any further.

2.6. Acute Versus Obtuse Triangles

In developing our local solver, we have assumed that the triangulation does not consist of obtuse triangles. What happens if the triangulation does have obtuse triangles? We illustrate the consequences by using the paraxial eikonal theory and a geometrical argument.

2.6.1. Isotropic Cases

Consider the isotropic eikonal equation. Assuming that we have an obtuse triangle $\triangle ABC$ in which T_A and T_B are given, we update the traveltime T_C at the vertex C. Let the unit directional vector $\mathbf{r}_1$ along edge CA and $\mathbf{r}_2$ along edge CB be in the second and fourth quadrant, respectively; see Fig. 5.

Then according to Eq. (2.26) we have vector $\mathbf{r}$ located in between $\mathbf{r}_1$ and $\mathbf{r}_2$, as showed in Fig. 5, and the straight line defined by Eq. (2.25) has $\mathbf{r}$ as its directional vector. Depending on $\mathbf{r}_0$, i.e., on T_A, T_B and the triangle, the straight line may have no intersection, two identical intersections, and two different intersections with the slowness surface defined at C:

$$|\mathbf{p}|F(C) = 1 \tag{2.33}$$

see Fig. 5.

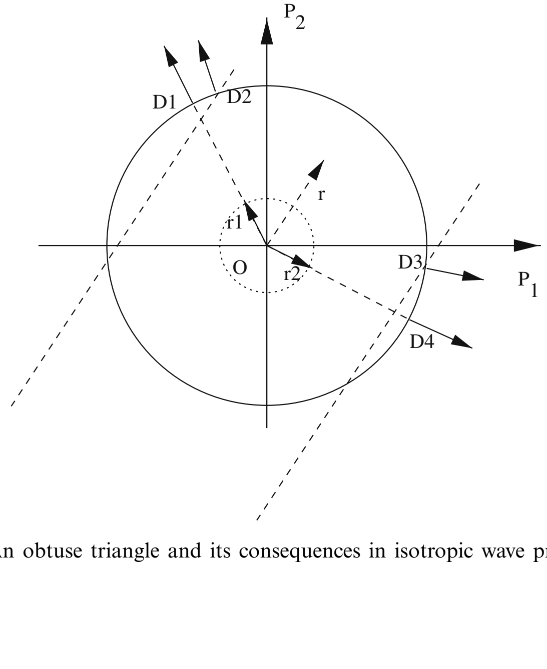


Fig. 5. An obtuse triangle and its consequences in isotropic wave propagation.

By the characteristic condition, if the triangle supports a consistent discretization then we will choose the intersection with the outward normal satisfying that the characteristic starting from C against the direction provided by the outward normal intersects the edge AB. As $\mathbf{r}$ varies in between $\mathbf{r}_1$ and $\mathbf{r}_2$, we have two extreme intersections defined by extending $\mathbf{r}_1$ and $\mathbf{r}_2$ to the isotropic slowness surface; they are D_1 and D_4 as illustrated in the figure. However, as observed from the figure the ray direction defined by the outward normal at D_1 has negative first-component and positive second-component while the ray direction defined by the outward normal at D_4 has positive first-component and negative second-component; similar observations can be made about D_2 and D_3.

In fact we can rigorously prove that as $\mathbf{r}$ varies from $\mathbf{r}_1$ to $\mathbf{r}_2$, the ray direction defined by the corresponding outward normal changes its signs from $(+,-)$ to $(-,+)$, noting that the ray direction hitting C is opposite to the outward normal direction in the figure. According to the proposed local solver, varying $\mathbf{r}$ between $\mathbf{r}_1$ and $\mathbf{r}_2$, the possible ray direction hitting C will change its sign from $(+,-)$ to $(-,+)$ going through

$$(+,-) \rightarrow (0,-) \rightarrow (-,-) \rightarrow (-,0) \rightarrow (-,+)$$

or

$$(+,-) \rightarrow (+,0) \rightarrow (+,+) \rightarrow (0,+) \rightarrow (-,+)$$

due to the convexity of the slowness surface.

Therefore according to the paraxial eikonal theory [18] one cannot define a locally stable, uni-directional propagation problem to update the traveltime at the vertex C by using traveltimes at A and B.

In summary, the criterion for splitting an obtuse angle is when the two ray directions corresponding to the two edges subtend an arc that contains two sonic points. For the isotropic case it simply means that the two ray directions have opposite signs in both corresponding components.

2.6.2. Anisotropic Cases

Next we consider the anisotropic eikonal equation. Assuming that we have an obtuse triangle $\triangle ABC$ in which T_A and T_B are given, we update the traveltime T_C at the vertex C.

Let the unit directional vectors $\mathbf{r}_1$ along edge CA and $\mathbf{r}_2$ along edge CB be in the second and fourth quadrant, respectively; the slowness surface defined at C is given as illustrated in Fig. 6. Then according to equation (2.26) we have vector $\mathbf{r}$ located in between $\mathbf{r}_1$ and $\mathbf{r}_2$, as showed in Fig. 6, and the straight line defined by Eq. (2.25) has $\mathbf{r}$ as its directional vector. Depending on $\mathbf{r}_0$, i.e., on T_A, T_B and the triangle, the straight line may have no intersection, two identical intersections, and two different intersections with the convex slowness surface defined at C:

$$H(C, \mathbf{p}) = 1 \tag{2.34}$$

see Fig. 6.

We note that this particular configuration as illustrated in Fig. 6 has the following property: the sector of the slowness surface subtended by $\mathbf{r}_1$, $\mathbf{r}$, and $\mathbf{r}_2$ has both horizontal and vertical tangent lines, and they are both unique due to the strict convexity in this sector.

By the characteristic condition, if the triangle supports a consistent discretization then we will choose the intersection with the outward normal satisfying that the characteristic starting from C against the direction provided by the outward normal intersects the edge AB. As $\mathbf{r}$ varies in between $\mathbf{r}_1$ and $\mathbf{r}_2$, we have two extreme intersections defined by $\mathbf{r}_1$ and $\mathbf{r}_2$, respectively; they are D_1 and D_4 as illustrated in the figure. However, as observed from the figure the ray direction defined by the outward normal at D_1 has negative first-component and positive second-component while the ray direction defined by the outward normal at D_4 has positive first-component and negative second-component; similar observations can be made about D_2 and D_3.

In fact we can rigorously prove that as $\mathbf{r}$ varies from $\mathbf{r}_1$ to $\mathbf{r}_2$, the ray direction defined by the corresponding outward normal changes its sign from $(+, -)$ to $(-, +)$. According to the proposed local solver, varying $\mathbf{r}$ between $\mathbf{r}_1$ and $\mathbf{r}_2$, the possible ray direction hitting C will change its sign

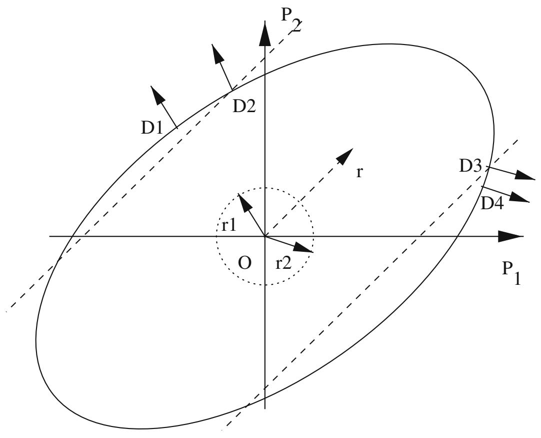


Fig. 6. An obtuse triangle and its consequences in anisotropic wave propagation.

from $(+,-)$ to $(-,+)$ going through

$$(+,-) \to (0,-) \to (-,-) \to (-,0) \to (-,+)$$

or

$$(+,-) \to (+,0) \to (+,+) \to (0,+) \to (-,+)$$

due to the convexity of the slowness surface.

According to the paraxial eikonal theory [18], if the traveltime at C is to be updated by the wave propagated from A and B to C, then the ray components at A and B must have a common sign in at least one of the components. However, by the above analysis, this is not true for the case under consideration. Therefore, one cannot define a locally stable, uni-directional propagation problem to update the traveltime at C by using traveltimes at A and B.

For the anisotropic case, the criterion for splitting an obtuse angle is the same as that for the isotropic case. When the two ray directions corresponding to the two edges subtend an arc on the slowness surface that contains two sonic points, we need to split the obtuse angle.

2.7. Generalizations to Higher Dimensions

The above procedure can be easily extended to higher dimensions. The design principle for the local solver still holds; namely, we first use consistency to find possible candidates and then check the causality condition. The only thing to which we need to pay attention is how to compute the group speed in higher dimensions if we are given two points. Although, in this case, we have to solve an implicit nonlinear system, the problem still has a unique solution by the convexity of the slowness surface. Therefore, we can use a similar shooting method as the one in [18].

As an illustration, we consider the three-dimensional case. There are six special tangent planes to the slowness surface: two having the normal vectors $(\pm 1, 0, 0)$, two having the normal vectors $(0, \pm 1, 0)$ and two having the normal vectors $(0, 0, \pm 1)$. The corresponding tangent points on the slowness surface will be used to classify ray directions and design the shooting method, and so on.

3. THE FAST SWEEPING ALGORITHM

We now describe the complete algorithm combining the local solver explained in the previous section with the fast sweeping strategy that we developed in [24].

- Step 1, sorting. Sort all the nodes (vertexes) according to the l^p distance to a few reference points. In all our tests we use the l^1 distance.
- Step 2, initialization. Assign large positive values to all vertexes except those that belong to or near the boundary (the initial front). Those boundary nodes are assigned exact values or approximated values by a shooting method, and these values are fixed in later iterations.
- Step 3, sweeping. Start Gauss–Seidel iterations with alternating sweeping orderings according to the distances of nodes to the chosen reference points.

Given the consistency and monotonicity of the local solver by Lemma 2.1, one can establish the following result by using the ideas in [33,24]:

Theorem 3.1. There exists a unique solution for the discretized nonlinear system and the fast sweeping iteration converges.

Using similar arguments from [33,24] one can show that the total number of sweepings needed depends only on the specific properties of the PDE, such as the turns of characteristics, which can be computed from the characteristic Eq. (2.4); in turn, this indicates how many sweepings are needed to cover the tangent directions along a characteristic curve in the computational domain consecutively.

Letting M be the total number of nodes, the complexity of the above algorithm is the following:

- on rectangular grids: $O(M)$;
- on unstructured meshes: $O(M \log M)$.

Remark. On unstructured meshes the $\log M$ factor comes from the initial sorting of all nodes. Once the sorting is done the complexity of the fast sweeping iterations is $O(M)$. Usually the sorting can be incorporated into the mesh generation easily with little extra cost.

Remark. The constant in the complexity formula does not depend on the anisotropy of the Hamiltonian. As an example, for elliptical anisotropic eikonal equations, if the coefficients are constant, then the characteristics are straight lines; the number of iterations needed for the fast sweeping method to converge is independent of the anisotropy of the Hamiltonian.

Remark. The criterion for an optimal choice of reference points and their locations on a triangular mesh is: all directions of characteristics should be covered with minimal redundancy. In practice, it is better if these reference points are evenly spaced both spatially and angularly with respect to the data set or boundary where the solution is prescribed. In our numerical tests we use the corners as reference points if the computational domain is rectangular. Other points, such as the center point of the domain or middle points of each edge, can be used as well.

4. APPLICATION: AN ELLIPTICAL ANISOTROPIC EIKONAL EQUATION

We apply the above procedure to derive an explicit local solver for anisotropic eikonal equations of the following type

$$[\nabla T(\mathbf{x}) M(\mathbf{x}) \nabla T(\mathbf{x})]^{1/2} = 1, \quad \mathbf{x} \in R^d, \tag{4.35}$$

where $M(\mathbf{x})$ is a $d \times d$ symmetric positive definite matrix. In particular M can be considered as a specific metric for the medium in which the wave front is propagating or in which we want to compute geodesics.

For simplicity let us consider the two-dimensional case,

$$H = \sqrt{a(\mathbf{x})\, p_1^2 - 2c(\mathbf{x})\, p_1\, p_2 + b(\mathbf{x})\, p_2^2} = 1, \tag{4.36}$$

where $a > 0$, $b > 0$ and $c^2 - ab < 0$. Without abusing notations we use a, b, and c to denote the coefficients in the anisotropic eikonal equation in the sequel.

Denote matrix $\mathbf{P}$ in Eq. (2.20) by

$$\mathbf{P} = \begin{pmatrix} n_{11}, & n_{12} \\ n_{21}, & n_{22} \end{pmatrix},$$

where $n_{11} = (x_C - x_A)/l_b$, $n_{12} = (y_C - y_A)/l_b$, $n_{21} = (x_C - x_B)/l_a$, $n_{22} = (y_C - y_B)/l_a$; l_b, l_a, and l_c are the lengths of edge CA, CB, and AB, respectively. Then

$$\mathbf{P^{-1}} = \frac{1}{\sin^2\gamma} \begin{pmatrix} n_{11} - n_{21}\cos\gamma, & n_{21} - n_{11}\cos\gamma \\ n_{12} - n_{22}\cos\gamma, & n_{22} - n_{12}\cos\gamma \end{pmatrix} \equiv \begin{pmatrix} p_{11}, & p_{12} \\ p_{21}, & p_{22} \end{pmatrix}.$$

From (2.21), we have

$$\nabla T(C) \approx \begin{pmatrix} \left(\frac{p_{11}}{l_b} + \frac{p_{12}}{l_a}\right) T_C - \left(\frac{p_{11}}{l_b} T_A + \frac{p_{12}}{l_a} T_B\right) \\ \left(\frac{p_{21}}{l_b} + \frac{p_{22}}{l_a}\right) T_C - \left(\frac{p_{21}}{l_b} T_A + \frac{p_{22}}{l_a} T_B\right) \end{pmatrix} \equiv \begin{pmatrix} g_1 T_C + g_2 \\ g_3 T_C + g_4 \end{pmatrix}, \tag{4.37}$$

where

$$g_1 \equiv \frac{p_{11}}{l_b} + \frac{p_{12}}{l_a}, \tag{4.38}$$

$$g_2 \equiv -\left(\frac{p_{11}}{l_b}T_A + \frac{p_{12}}{l_a}T_B\right), \tag{4.39}$$

$$g_3 \equiv \left(\frac{p_{21}}{l_b} + \frac{p_{22}}{l_a}\right), \tag{4.40}$$

$$g_4 \equiv -\left(\frac{p_{21}}{l_b}T_A + \frac{p_{22}}{l_a}T_B\right). \tag{4.41}$$

Substituting $\nabla T(C)$ in (4.37) into the Hamilton–Jacobi Eq. (4.36), we obtain the quadratic equation

$$w_1 T_C^2 + w_2 T_C + w_3 - 1 = 0, \tag{4.42}$$

where

$$w_1 \equiv a g_1^2 + b g_3^2 - 2c g_1 g_3, \tag{4.43}$$

$$w_2 \equiv 2a g_1 g_2 + 2b g_3 g_4 - 2c(g_1 g_4 + g_2 g_3), \tag{4.44}$$

$$w_3 \equiv a g_2^2 + b g_4^2 - 2c g_2 g_4, \tag{4.45}$$

where $a = a(\mathbf{x_C})$, $b = b(\mathbf{x_C})$, and $c = c(\mathbf{x_C})$.

If the quadratic Eq. (4.42) has real roots

$$T_C = \frac{-w_2 \pm \sqrt{w_2^2 - 4w_1(w_3 - 1)}}{2w_1} \tag{4.46}$$

then we check the causality for the positive roots.

If T_C is a positive root, then we reconstruct $\nabla T(C) = (p, q)$ by (4.37) and calculate the characteristic direction

$$\mathbf{d} = \begin{pmatrix} ap - cq \\ bq - cp \end{pmatrix}$$

next we check whether the characteristic line with direction $\mathbf{d}$ passing vertex C falls inside the triangle $\triangle ABC$ or not. If the characteristic line is inside the $\triangle ABC$, causality is satisfied and we update T_C if this new value is smaller than the current numerical value of T_C. Otherwise, if the quadratic Eq. (4.42) has no positive root satisfying the causality condition, then we have to use the group speed along edges AC and BC, and

$$T_C = \min\left\{T_C, T_A + \frac{|AC|}{v_{\mathrm{g}}^{AC}}, T_B + \frac{|BC|}{v_{\mathrm{g}}^{BC}}\right\}.$$

5. NUMERICAL EXPERIMENTS

In all the examples, we choose four corners in a two-dimensional rectangular domain as the reference points and sort the nodes according to the l^1 metric by using the quicksort method, as in [24].

The convergence of iteration is measured in terms of the L^1-norm as advocated by Lin and Tadmor [16]; i.e., the iteration stops when the successive error satisfies $\|T^{n+1} - T^n\|_{L^1} < 10^{-10}$. To compute the L^1 error on a general triangular mesh, we follow the definition of the L^1 norm. First we calculate the errors e_1, e_2, e_3 at three vertices on each triangle, then we take $\frac{1}{3}(e_1 + e_2 + e_3)$ to be the average error on this triangle. Afterward, we multiply the average error on each triangle by the area of the triangle and sum up these products over all triangles. Finally we divide the sum by the area of the computational domain to get the normalized L^1 error.

In all of our test cases one sweeping means one Gauss–Seidel iteration through all nodes with a particular ordering.

A typical acute triangulation is shown in Fig. 7. When we check the accuracy of our methods, we refine the mesh uniformly, i.e., cutting each triangle into four smaller similar ones. Exact solutions needed in initializing the algorithm and checking accuracy are computed by the shooting method [18], whenever possible.

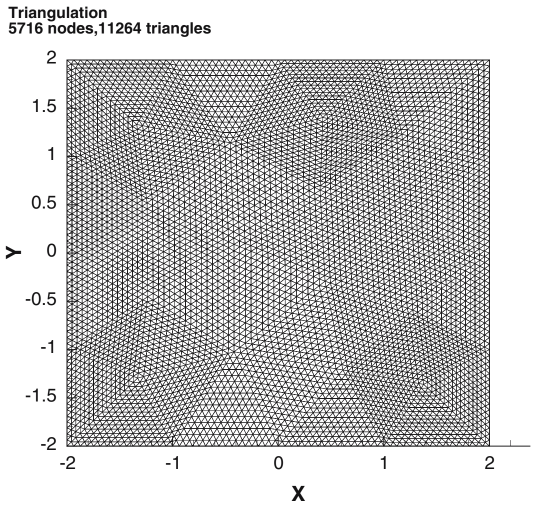


Fig. 7. A typical acute triangulation.

5.1. Example 1

We consider the following equation

$$\sqrt{a\,T_x^2 + b\,T_y^2 - 2c\,T_x\,T_y} = 1, \quad (x, y) \in (-2, 2) \times (-2, 2), \tag{5.1}$$

$$T(0, 0) = 0, \tag{5.2}$$

where $a > 0$, $b > 0$, and $c^2 - ab < 0$.

In a homogeneous anisotropic metric, the eigenvalues of the symmetric positive definite matrix

$$\mathbf{M} = \begin{pmatrix} a, & -c \\ -c, & b \end{pmatrix}$$

characterizes the anisotropy of the metric. According to [27] the anisotropy coefficient of the metric is defined by

$$\eta = \sqrt{\frac{\lambda_{\max}(\mathbf{M})}{\lambda_{\min}(\mathbf{M})}},$$

where $\lambda_{\max}(\mathbf{M})$ and $\lambda_{\min}(\mathbf{M})$ are the larger and smaller eigenvalues of $\mathbf{M}$, respectively.

Case 1: a homogeneous, mild anisotropic case. We take $a = 1$, $b = 1$, and $c = 0.9$; $\eta = \sqrt{19}$. Since the point source problem has an upwind singularity at the source [19] we have to measure the order of accuracy of the fast sweeping method away from the singularity. Otherwise the accuracy will degenerate to $h \log h$ [33]. To achieve this we fix a small region, $[-0.2, 0.2] \times [-0.2, 0.2]$, around the source, assign the exact solution to the grid points inside this small region, and compute the numerical error only for the grid points outside the small region; this is the so-called wrapping technique. As shown in Table I, with this wrapping technique we are able to observe the expected first-order accuracy while without the wrapping

Table I. The Order of Convergence; $a = 1, b = 1, c = 0.9$

		Wrapping $[-0.2, 0.2] \times [-0.2, 0.2]$			No wrapping		
Nodes	Elements	L^1 error	Order	Iter	L^1 error	Order	Iter
1473	2816	6.45E − 2	–	4	7.47E − 2	–	4
5716	11264	3.27E − 2	0.98	4	4.68E − 2	0.68	4
22785	45056	1.75E − 2	0.91	4	2.87E − 2	0.71	5
90625	180224	8.88E − 3	0.97	4	1.71E − 2	0.75	5

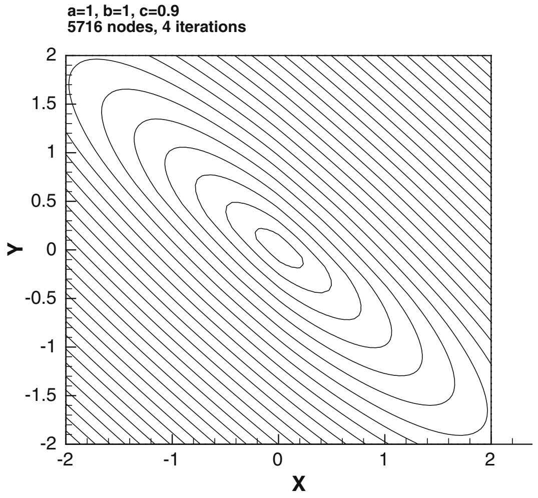


Fig. 8. $a=1, b=1, c=0.9$, $\eta=\sqrt{19}$; convergence after four sweepings.

Table II. Comparison of the Accuracy between the Four Three-Point Stencils and the Eight Three-Point Stencils (Fig. 4); Wrapping $[-0.2, 0.2]\times[-0.2, 0.2]$; $a=1, b=1, c=0.9$

	Four three-point stencils			Eight three-point stencils		
Mesh	L^1 error	Order	Iter	L^1 error	Order	Iter
40×40	1.17E−1	–	4	1.57E−2	–	4
80×80	6.35E−2	0.88	4	8.18E−3	0.94	4
160×160	3.39E−2	0.90	4	4.18E−3	0.97	4
320×320	1.78E−2	0.93	4	2.12E−3	0.98	4

technique we are only able to observe the degraded first-order accuracy. In Table I, we also observe that the number of iterations needed for convergence is almost a constant which is independent of mesh sizes. Figure 8 shows the contour plot for this test case.

Next we test our algorithm on rectangular grids, which can be considered as special cases. See the discussion in Sect. 2 and Fig. 4. The sweeping directions are based on the l^1 metric. The accuracy and the number of iterations are listed in Table II for both the four three-point stencils and the eight three-point stencils (Fig. 4). Apparently the eight three-point stencils can result in a higher accuracy than the four three-point stencils.

In this case of rectangular grids we also test the i, j orderings as that used in the fast sweeping method on rectangular meshes; the same

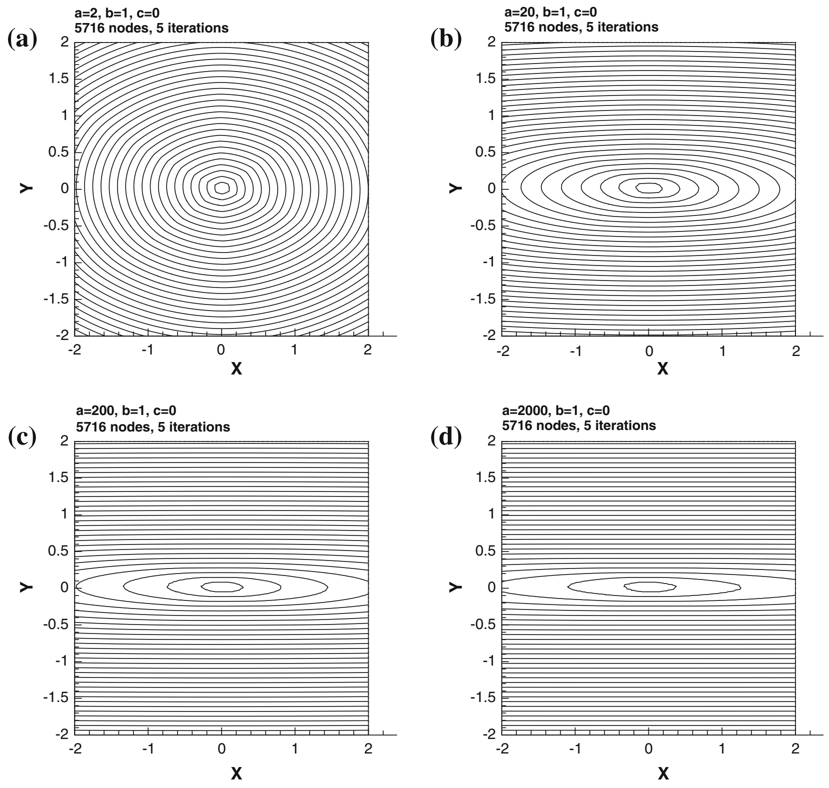
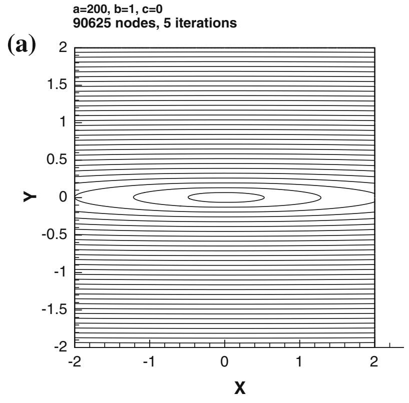


Fig. 9. (**a**). $a=2, b=1, c=0,\ \eta=\sqrt{2}$; convergence after five sweepings; (**b**). $a=20, b=1, c=0,\ \eta=\sqrt{20}$; convergence after five sweepings; (**c**). $a=200, b=1, c=0,\ \eta=\sqrt{200}$; convergence after five sweepings; (**d**). $a=2000, b=1, c=0,\ \eta=\sqrt{2000}$; convergence after five sweepings.

accuracy and the same number of sweepings are obtained; we omit the results here.

Case 2: Homogeneous, strong anisotropic cases. We carry out a sequence of tests to study the power and the robustness of our sweeping methods.

First we increase the coefficient a successively from 2 to 2000 with $b=1$ and $c=0$ fixed. Figure 9 shows the contour plots of the results for $a=2, 20, 200, 2000$ computed by using the same mesh with 5716 nodes and 11264 triangles. In addition for $a=200, 2000$ we use a finer mesh with 90625 nodes and 180224 triangles; the results are shown in Fig. 10, which have higher resolution than the ones shown in Fig. 9.



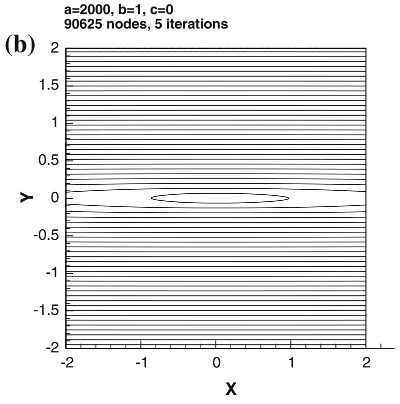
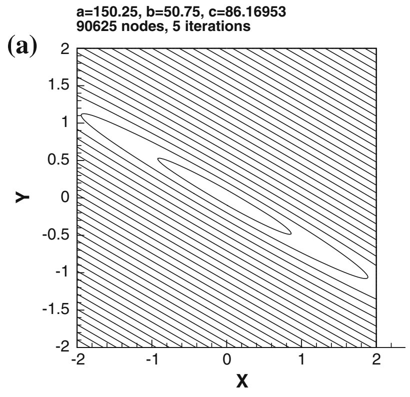


Fig. 10. Refined mesh. (**a**). $a=200, b=1, c=0$, $\eta=\sqrt{200}$; convergence after five sweepings; (**b**). $a=2000, b=1, c=0$, $\eta=\sqrt{2000}$; convergence after five sweepings.



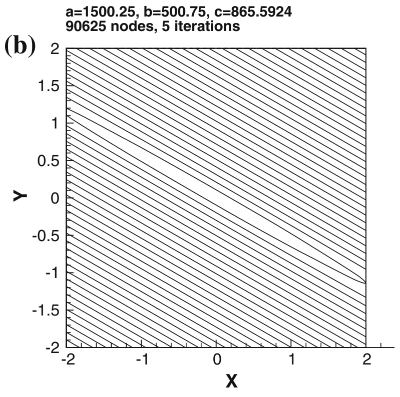


Fig. 11. (**a**). $a=150.25, b=50.75, c=86.16953$, $\eta=\sqrt{200}$; convergence after five sweepings; (**b**). $a=1500.25, b=500.75, c=865.5924$, $\eta=\sqrt{2000}$; convergence after five sweepings.

The above cases have the symmetrical axis of the slowness surface aligned with the Cartesian axis, which may not be able to test out the full power of the sweeping method. To test out the full power of the method we take out the two cases, $(a=200, b=1, c=0)$ and $(a=2000, b=1, c=0)$, and apply to the resulting matrices M a similarity transform defined by a rotation with angle $\frac{\pi}{6}$. Then we have $(a=150.25, b=50.75, c=86.16953)$ and $(a=1500.25, b=500.75, c=865.5924)$; the resulting anisotropic coefficients are $\eta=\sqrt{200}$ and $\eta=\sqrt{2000}$, respectively.

Figure 11 shows the contour plots for the two cases with the computational mesh of 90625 nodes and 180224 triangles. Numerical errors and order of convergence are shown in Table III and Table IV.

Table III. The Order of Convergence; $a = 150.25, b = 50.75, c = 86.16953$

		Wrapping $[-0.2, 0.2] \times [-0.2, 0.2]$			No wrapping		
Nodes	Elements	L^1 error	Order	Iter	L^1 error	order	iter
1473	2816	8.78E − 3	–	4	8.87E − 3	–	4
5716	11264	4.04E − 3	1.12	4	5.38E − 3	0.72	4
22785	45056	2.10E − 3	0.94	4	3.22E − 3	0.74	5
90625	180224	1.04E − 3	1.02	4	1.88E − 3	0.77	5

Table IV. The Order of Convergence; $a = 1500.25, b = 500.75, c = 865.5924$

		Wrapping $[-0.2, 0.2] \times [-0.2, 0.2]$			No wrapping		
Nodes	Elements	L^1 error	Order	Iter	L^1 error	Order	Iter
1473	2816	6.23E − 3	–	4	5.72E − 3	–	4
5716	11264	2.89E − 3	1.11	4	3.33E − 3	0.78	4
22785	45056	1.53E − 3	0.92	4	1.93E − 3	0.79	5
90625	180224	7.66E − 4	1.00	4	1.10E − 3	0.80	5

These results demonstrate that our sweeping method is robust enough to handle the equation with a very high-anisotropy coefficient. Unlike other methods with numerical domain of dependency depending on the anisotropy coefficient η which may be very large when η is large, our iterative methods do not have such a shortcoming and thus are efficient and robust. As discussed in Sect. 3, the number of sweepings is independent of mesh size and anisotropy, but it depends on the behavior of characteristics. In the case of homogeneous media, the characteristics are straight-lines, and, that is, why we only need 4–5 sweepings.

5.2. Example 2

We consider the following equation with variable coefficients

$$\sqrt{a(x,y)\,T_x^2 + b(x,y)\,T_y^2 - 2c(x,y)\,T_x\,T_y} = 1, \quad (x,y) \in (-1,1) \times (-1,1),$$
$$T(x,y) = 0,\ (x,y) \in \Gamma,$$

where Γ is a unit square in the middle of the domain: $\Gamma = \{x = \pm 0.5, |y| \leqslant 0.5\} \cup \{y = \pm 0.5, |x| \leqslant 0.5\}$.

We choose $a(x,y) = 150.25(1 + \lambda \sin^2(\pi xy))$, $b(x,y) = 50.75(1 + \delta \cos^2(\pi xy))$, $c(x,y) = 86.16953(1 - \epsilon \sin^2(\pi xy))$, where λ, δ, and ϵ are constants

Table V. Comparison of Iteration Numbers of the Four Three-Point Stencil (Fig. 4 (a)), the Eight Three-Point Stencil (Fig. 4 (b)) and the Regular Triangular Stencil (Fig. 1), $a(x, y) = 150.25(1 + \lambda \sin^2(\pi xy)), b(x, y) = 50.75(1 + \delta \cos^2(\pi xy)), c(x, y) = 86.16953(1 - \epsilon \sin^2(\pi xy))$ where $\lambda = 1, \delta = 1, \epsilon = 0.125$

Four three-point stencil		Eight three-point stencil		Triangular stencil	
Mesh	Iter	Mesh	Iter	Mesh nodes	Iter
40×40	11	40×40	9	1473	9
80×80	13	80×80	10	5761	7
160×160	13	160×160	11	22785	8
320×320	13	320×320	13	90625	9

to be selected; this is a perturbation of the homogeneous case $(a, b, c) = (150.25, 50.75, 86.16953)$.

We solve this equation on three different sets of stencils: a regular triangular mesh and two virtual meshes constructed from uniform rectangular grids; the latter two consists of four three-point stencils at each node and eight three-point stencils at each node, respectively.

We take $\lambda = 1$, $\delta = 1$, and $\epsilon = 0.125$. The results in terms of mesh refinement are shown in Table V. Figure 12 shows contours of the solutions on different meshes.

5.3. Example 3: The Five-Ring Problem

We consider the following equation

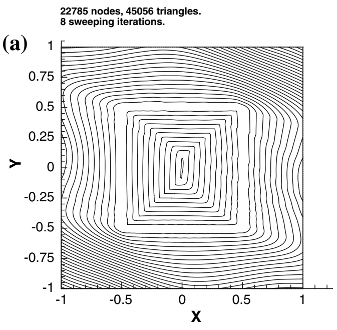
$$\sqrt{a T_x^2 + b T_y^2 - 2c T_x T_y} = 1, \quad (x, y) \in (0, 1) \times (0, 1), \tag{5.3}$$

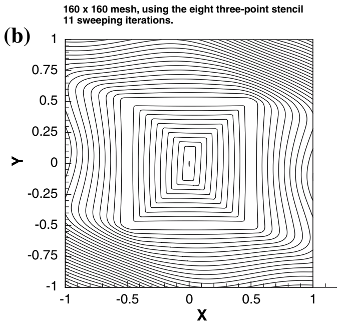
$$T(0, 0) = 0, \tag{5.4}$$

where $a > 0$, $b > 0$, and $c^2 - ab < 0$. A five-ring obstacle is placed in the computational domain. This example is borrowed from [10]. Due to the geometrical complexity of the obstacle, it is difficult to use a rectangular mesh; on the other hand, our method based on unstructured meshes can be used with an advantage. We treat the five rings as a part of the computational boundary, and we triangulate the resulting computational domain. A typical triangulation is shown in Fig. 13.

We apply our fast sweeping method to cases with increasing anisotropy; the method converges in almost the same number of sweepings in all the cases.

Figure 14(a) shows the result of the isotropic case; the wavefronts near the source are circular. Figure 14(a), (b), (c), and (d) show the





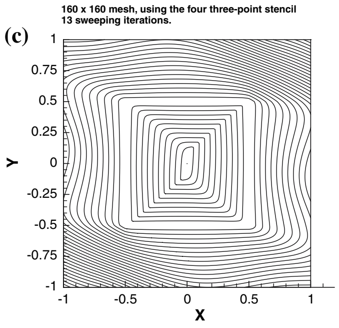
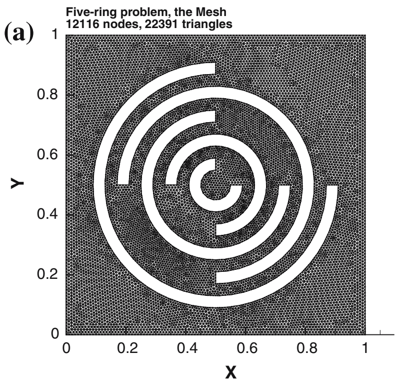


Fig. 12. $a(x, y) = 150.25(1 + \lambda \sin^2(\pi xy)), b(x, y) = 50.75(1 + \delta \cos^2(\pi xy)), c(x, y) = 86.16953(1 - \epsilon \sin^2(\pi xy))$ where $\lambda = 1, \delta = 1, \epsilon = 0.125$. (**a**). On a general triangular mesh with 22785 nodes, convergence after eight sweepings; (**b**). on the 160×160 rectangular mesh, using the eight three-point stencils, convergence after 11 sweepings; (**c**) on the 160×160 rectangular mesh, using the four three-point stencils, convergence after 13 sweepings.



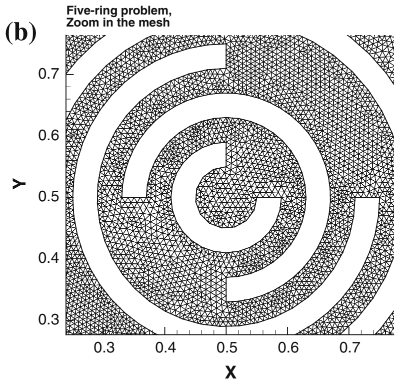


Fig. 13. A typical triangular mesh for the five-ring problem. 12116 nodes, 22391 triangles. (**a**). the whole mesh; (**b**). the zoom in.

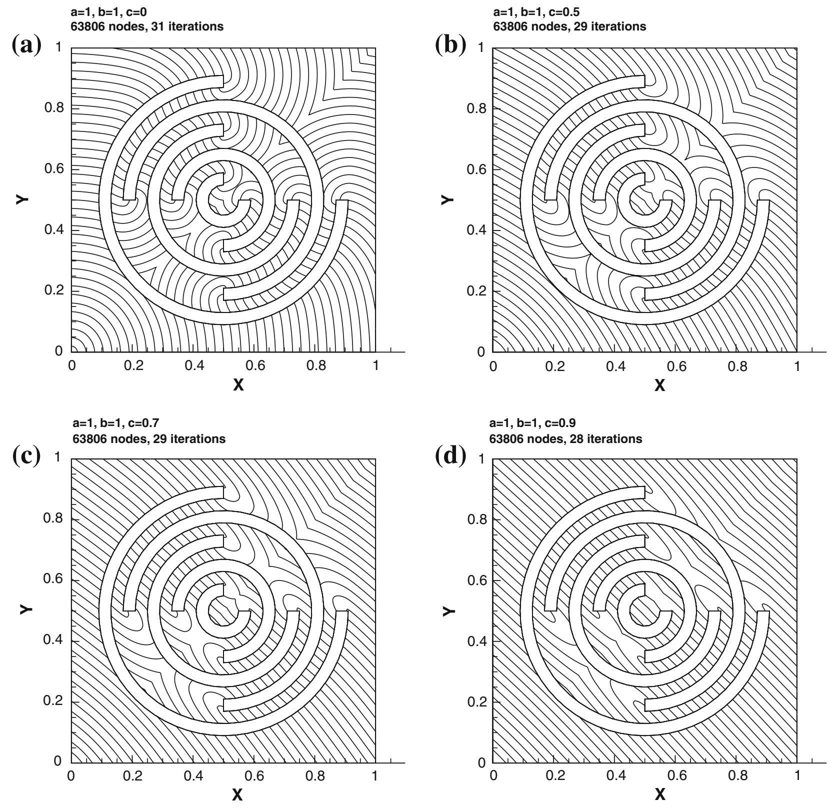
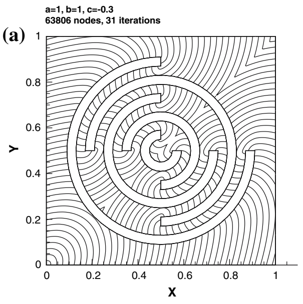
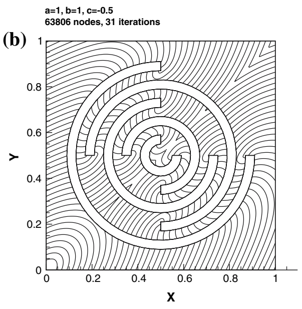
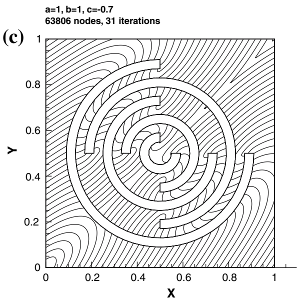


Fig. 14. The five-ring problem. The source is at $(0, 0)$. (**a**). the isotropic case, $a = 1, b = 1, c = 0$; convergence after 31 sweepings; (**b**). $a = 1, b = 1, c = 0.5$; convergence after 29 sweepings; (**c**). $a = 1, b = 1, c = 0.7$; convergence after 29 sweepings; (**d**): $a = 1, b = 1, c = 0.9$; convergence after 28 sweepings.

wavefronts with increasing anisotropy; near the source, we can see that the wavefronts change from circular to noncircular to nearly planar shapes; inside the rings, we can see that local secondary sources generate noncircular wavefronts in Fig. 14(b), (c), and (d). We note that two families of wavefronts are colliding along the diagonal near the upper-right corner because the geometry of the five-ring obstacle is symmetric with respect to the diagonal from the lower-left to the upper-right corner, and the source is located on this diagonal; for convenience we name this diagonal the lower-right diagonal. As c is decreased from 0.9 to 0.0 while a and b are fixed, we can observe that near the upper-right corner the two families of

(a)


(b)


(c)


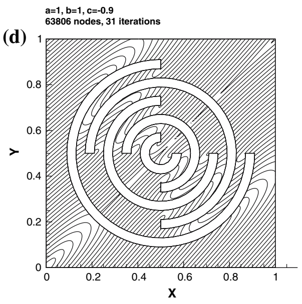
(d)


Fig. 15. The five-ring problem. The source is at $(0,0)$. (**a**). $a=1, b=1, c=-0.3$; convergence after 31 sweepings; (**b**). $a=1, b=1, c=-0.5$; convergence after 31 sweepings; (**c**). $a=1, b=1, c=-0.7$; convergence after 31 sweepings; (**d**): $a=1, b=1, c=-0.9$; convergence after 31 sweepings.

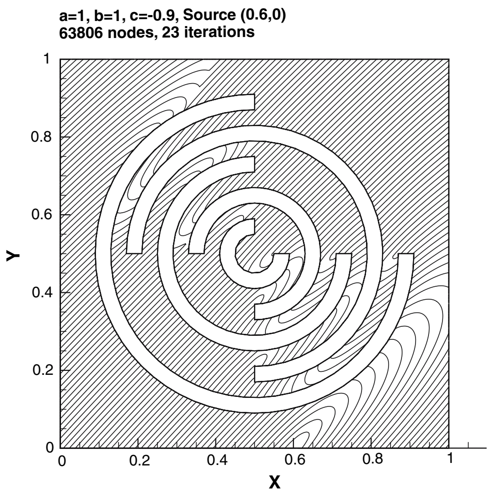


Fig. 16. The five-ring problem. The source is at $(0.6,0)$. $a=1, b=1, c=-0.9$; convergence after 23 sweepings.

wavefronts are colliding at angles which are becoming smaller, the resulting kinks becoming sharper as shown in Fig. 14(d), (c), (b), and (a).

If c is decreased further from 0.0 to -0.9 as shown in Fig. 15, we can observe that near the upper-right corner the two families of wavefronts are colliding at even smaller angles, the resulting kinks becoming even sharper. In particular, when $c = -0.9$, the two families of wavefronts are nearly planar and nearly parallel, and the colliding angles are very small; even for such a case the fast sweeping method is robust enough to give good results based on the given mesh. To verify the above reasoning, we may move the source location from $(0, 0)$ to $(0.6, 0.0)$ so that it is no longer located on the lower-right diagonal; as shown in Fig. 16, two families of wavefronts are colliding at locations near $(0.4, 0.9)$, the resulting kinks no longer sharp.

6. CONCLUSIONS

We develop a fast sweeping method for static Hamilton–Jacobi equations with convex Hamiltonians. Local solvers and fast sweeping strategies apply to structured and unstructured meshes. With causality correctly enforced during sweepings numerical evidence indicates that the fast sweeping method converges in a finite number of iterations independent of mesh size. Numerical examples validate both the accuracy and the efficiency of the new methods.

ACKNOWLEDGMENTS

The research of H.-K. Zhao is partially supported by ONR Grant #N00014-02-1-0090, DARPA Grant #N00014-02-1-0603 and Sloan Foundation Fellowship. The research of J. Qian is supported by NSF Grant #0542174.

REFERENCES

1. Boue, M., and Dupuis, P. (1999). Markov chain approximations for deterministic control problems with affine dynamics and quadratic costs in the control. *SIAM J. Numer. Anal.* **36**, 667–695.
2. Burridge, R., de Hoop, M. V., Miller, D., and Spencer, C. (1998). Multiparameter inversion in anisotropic media. *Geophys. J. Internat.* **134**, 757–777.
3. Cecil, T., Osher, S. J., and Qian, J. (2006). Simplex free adaptive tree fast sweeping and evolution methods for solving level set equations in arbitrary dimension. *J. Comp: Phys.* **213**, 458–473.
4. Cockburn, B., and Qian, J. (2002). Continuous dependence results for Hamilton–Jacobi equations. In Estep, D., and Tavener, S. (eds.) *Collected Lectures on the Preservation of Stability Under Discretization*. SIAM, Philadelphia, PA, pp. 67–90.

5. Crandall, M. G., and Lions, P. L. (1983). Viscosity solutions of Hamilton–Jacobi equations. *Trans. Am. Math. Soc.* **277**, 1–42.
6. Crandall, M. G., and Lions, P. L. (1984). Two approximations of solutions of Hamilton–Jacobi equations. *Math. Comput.* **43**, 1–19.
7. Dellinger, J. (1991). *Anisotropic Seismic Wave Propagation*. Ph.D. Thesis, Stanford University, Stanford, CA94305.
8. Dellinger, J., and Symes, W. W. (1997). Anisotropic finite-difference traveltimes using a Hamilton–Jacobi solver. In *Proc. 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts*, Soc. Expl. Geophys., Tulsa, OK, pp. 1786–1789.
9. Gonzales, R., and Rofman, E. (1985). On deterministic control problems: an approximation procedure for the optimal cost. I. the stationary problem. *SIAM J. Control Optim.* **23**, 242–266.
10. Gremaud, P. A., and Kuster, C. M. (2006). Computational study of fast methods for the eikonal equations. *SIAM J. Sci. Comput.* **27**, 1803–1816.
11. Jiang, G. S., and Peng, D. (2000). Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM J. Sci. Comput.* **21**, 2126–2143.
12. Jiang, G. S., and Shu, C. W. (1996). Efficient implementation of weighted ENO schemes. *J. Comput. Phys.* **126**, 202–228.
13. Kao, C. Y., Osher, S. J., and Tsai, Y.-H. (2005). Fast sweeping method for static Hamilton–Jacobi equations. *SIAM J. Numer. Anal.* **42**, 2612–2632.
14. Kao, C. Y., Osher, S. J., and Qian, J. (2004). Lax–Friedrichs sweeping schemes for static Hamilton–Jacobi equations. *J. Comput. Phys.* **196**, 367–391.
15. Leung, S., and Qian, J. (2006). An adjoint state method for three-dimensional transmission traveltime tomography using first-arrivals. *Commun. Math. Sci.* **4**, 249–266.
16. Lin, C. T., and Tadmor, E. (2001). L^1-stability and error estimates for approximate Hamilton–Jacobi equations. *Numer. Math.* **88**, 2163–2186.
17. Liu, X. D., Osher, S. J., and Chan, T. (1994). Weighted essentially nonoscillatory schemes. *J. Comput. Phys.* **115**, 200–212.
18. Qian, J., and Symes, W. W. (2001). Paraxial eikonal solvers for anisotropic quasi-P traveltimes. *J. Comput. Phys.* **173**, 1–23.
19. Qian, J., and Symes, W. W. (2002). Adaptive finite difference method for traveltime and amplitude. *Geophysics* **67**, 167–176.
20. Qian, J., and Symes, W. W. (2002). Finite-difference quasi-P traveltimes for anisotropic media. *Geophysics* **67**, 147–155.
21. Qian, J., and Symes, W. W. (2002). Paraxial geometrical optics for quasi-P waves: theories and numerical methods. *Wave Motion* **35**, 205–221.
22. Qian, J., and Symes, W. W. (2003). A paraxial formualtion for the viscosity solution of quasi-p eikonal equations. *Comput. Math. Appl.* **46**, 1691–1701.
23. Qian, J., Symes, W. W., and Dellinger, J. A. (2001). A full-aperture anisotropic eikonal solver for quasi-P traveltimes. In *Proc. 71st Ann. Internat. Mtg., Expanded Abstracts*, Soc. Expl. Geophys., Tulsa, OK, pp. 129–132.
24. Qian, J., Zhang, Y. T., and Zhao, H. K. (2007). Fast sweeping methods for eikonal equations on triangular meshes. *SIAM J. Numer. Anal.* **45**, 83–107.
25. Qin, F., and Schuster, G. T. (1993). First-arrival traveltime calculation for anisotropic media. *Geophysics* **58**, 1349–1358.
26. Sethian, J. A. (1996). *Level Set Methods*. Cambridge University Press, Cambridge.
27. Sethian, J. A., and Vladimirsky, A. (2001). Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms. In *Proc. PAM-792*. University of California at Berkeley, Berkeley, CA94720.

28. Tsai, R., Cheng, L.-T., Osher, S. J., and Zhao, H. K. (2003). Fast sweeping method for a class of Hamilton–Jacobi equations. *SIAM J. Numer. Anal.* **41**, 673–694.
29. Tsitsiklis, J. N. (1995). Efficient algorithms for globally optimal trajectories. *IEEE Tran. Automatic Control* **40**, 1528–1538.
30. van Trier, J., and Symes, W. W. (1991). Upwind finite-difference calculation of traveltimes. *Geophysics* **56**, 812–821.
31. Zhang, Y. T., Zhao, H. K., and Chen, S. (2005). Fixed-point iterative sweeping methods for static Hamilton–Jacobi equations. *Methods Appl. Anal*. (accepted).
32. Zhang, Y. T., Zhao, H. K., and Qian, J. (2006). High order fast sweeping methods for static Hamilton–Jacobi equations. *J. Sci. Comp.* **29**, 25–56.
33. Zhao, H. K. (2005). Fast sweeping method for eikonal equations. *Math. Comp.* **74**, 603–627.
34. Zhao, H. K. (2006). Parallel implementations of the fast sweeping method. UCLA CAM06-13.
35. Zhao, H. K., Osher, S., Merriman, B., and Kang, M. (2000). Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Comput. Vis. Image Underst.* **80**, 295–319.