Journal of Computational Physics 227 (2008) 10209–10225

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Legendre-transform-based fast sweeping methods for static Hamilton–Jacobi equations on triangulated meshes

Chiu-Yen Kao [a], Stanley Osher [b], Jianliang Qian [c,*]

[a] Department of Mathematics, The Ohio State University, Columbus, OH 43210, United States
[b] Department of Mathematics, University of California Los Angeles, Los Angeles, CA 90095, United States
[c] Department of Mathematics, Michigan State University, East Lansing, MI 48824, United States

ARTICLE INFO

Article history:
Received 12 April 2008
Received in revised form 21 August 2008
Accepted 25 August 2008
Available online 5 September 2008

Keywords:
Hamilton–Jacobi equations
Fast sweeping methods
Monotone schemes
Legendre transforms
Godunov numerical Hamiltonians

ABSTRACT

We propose a new sweeping algorithm which utilizes the Legendre transform of the Hamiltonian on triangulated meshes. The algorithm is a general extension of the previous proposed algorithm by Kao et al. [C.Y. Kao, S.J. Osher, Y.-H. Tsai, Fast sweeping method for static Hamilton–Jacobi equations, SIAM J. Numer. Anal. 42 (2005) 2612–2632]. The algorithm yields the numerical solution at a grid point using only its one-ring neighboring grid values and is easy to implement numerically. The minimization that is related to the Legendre transform in the sweeping algorithm can either be solved analytically or numerically. The scheme is shown to be monotone and consistent. We illustrate the efficiency and accuracy of the new method with several numerical examples in two and three dimensions.

1. Introduction

Hamilton–Jacobi equations arise from a multitude of applications, ranging from classical mechanics, semi-classical quantum mechanics to contemporary image processing and computer vision. Therefore it is crucial to develop efficient, accurate numerical methods for these equations. In this paper, we develop Legendre-transform-based fast sweeping methods for the following static Hamilton–Jacobi equation on unstructured meshes:

$$H(x, \nabla\phi(x)) = R(x), \quad x \in \Omega \setminus \Gamma, \tag{1}$$

$$\phi(x) = g(x), \quad x \in \Gamma \subset \Omega, \tag{2}$$

where $g(x)$ is a positive, Lipschitz continuous function, Ω is an open, bounded polygonal domain in R^d, and Γ is a subset of Ω; $H(x,p)$ is Lipschitz continuous in both arguments, and it is convex and homogeneous of degree one in the second argument. If $H(x,p) = |p|H(x, \frac{p}{|p|}) = |p|V(x)$, then the eikonal equation for isotropic wave propagation results.

Fast sweeping methods are a family of efficient methods for solving static Hamilton–Jacobi equations [26,23,9,10,5,18,19,27,12], and some essential ideas of these methods may trace back to [20,3]. In [26] the fast sweeping method was systematically analyzed for eikonal equations. Since then the fast sweeping methods have undergone intensive development for general static Hamilton–Jacobi equations in [26,23,9,10,5,18,19,27,12] and have found many different applications; see [11,8], for examples. On the other hand, the fast marching method and its relatives consist of another family of numerical methods for solving static Hamilton–Jacobi equations [24,21,7,22].

* Corresponding author. Tel.: +1 517 353 6334; fax: +1 517 432 1562.
E-mail addresses: kao@math.ohio-state.edu (C.-Y. Kao), sjo@math.ucla.edu (S. Osher), qian@math.msu.edu (J. Qian).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.08.016

A fast sweeping method consists of the following three essential ingredients: (1) an efficient local solver for a Hamilton–Jacobi equation on a given Cartesian mesh or triangulation, (2) systematic orderings of solution nodes according to some predetermined information-flowing directions, and (3) Gauss–Seidel type iterations based on a given order of solution nodes. As is well known by now, a Cartesian mesh provides the fast sweeping method with natural orderings of nodes so that efficient sweeping strategies can be easily designed as illustrated in [26,23,9,10,25]. On a triangulated mesh, however, such natural orderings no longer exist. In [18,5,19] some novel ordering strategies based on reference points and l^p-distances have been proposed, which are demonstrated to be effective.

As for local solvers for Hamilton–Jacobi equations, in general they are based on some monotone numerical Hamiltonians which discretize continuous Hamiltonians on a Cartesian mesh or a triangular mesh. Such monotone numerical Hamiltonians can be Godunov-type [1,13], Lax–Friedrichs-type [13], or optimal-control-type [24,2]. In [10] a Legendre-transform-based numerical Hamiltonian on Cartesian meshes is proposed to design a fast sweeping method for static Hamilton–Jacobi equations, and the resulting method is shown to be efficient and accurate; on Cartesian meshes this numerical Hamiltonian can be interpreted as Godunov-type Hamiltonians [10]. In [13] Osher and Shu derive the Riemann solver on Cartesian meshes for Hamilton–Jacobi equations from the Godunov scheme [1]; it might be possible to do the same on triangulated meshes for Hamilton–Jacobi equations. In this paper we propose a new Legendre-transform-based numerical Hamiltonian on triangulated meshes to discretize a continuous convex Hamiltonian; incorporating this new numerical Hamiltonian into a sweeping strategy on unstructured meshes [18] yields a new fast sweeping method for static Hamilton–Jacobi equations on triangulated meshes.

The rest of the paper is organized as follows. Section 2 presents Legendre-transform-based numerical Hamiltonians for 2-D and 3-D convex Hamilton–Jacobi equations. Section 3 details numerical Hamiltonians for a class of quadratic Hamiltonians, including classical eikonal equations. Section 4 gives some algorithmic and implementation details. Section 5 gives various examples to demonstrate the efficiency and accuracy of the new schemes.

2. Legendre-transform-based numerical Hamiltonians

2.1. Spherical Legendre transforms

Let $\gamma : S^{d-1} \rightarrow R^{+}$ be a continuous function defined on a curved space S^{d-1}, which is the unit sphere in R^d. Then the following results hold.

1. The first Legendre transformation of γ is

$$\gamma_*(v) = \min_{\theta\cdot v>0,|\theta|=1}\left[\frac{\gamma(\theta)}{(\theta\cdot v)}\right].$$

2. The second Legendre transformation of γ is

$$\gamma^*(v) = \max_{\theta\cdot v>0,|\theta|=1}[\gamma(\theta)(\theta\cdot v)].$$

The first and second Legendre transformations are dual to each other in the following sense: $(r_*)^* = r$ if r is convex, and $(r_*)^* = r$ if r is polar-convex; see [14]. We can extend γ to the whole space R^d by defining

$$\widetilde{\gamma}(x) = |x|\gamma\left(\frac{x}{|x|}\right),$$

where the extension $\widetilde{\gamma}$ is homogeneous of degree one and $x \in R^d$.

Using the Legendre transformations we can rewrite the convex Hamiltonian H in Eq. (1) in the following form:

$$H(\nabla\phi(x)) = \max_{\theta,\nabla\phi\cdot\theta>0}[(\nabla\phi\cdot\theta)w(\theta)], \quad \theta \in S^{d-1},$$

where

$$w(\theta) = \min_{v\cdot\theta>0,|v|=1}\left[\frac{H(v)}{(v\cdot\theta)}\right]. \tag{3}$$

In the above, for the sake of clarity in notation, we have suppressed the dependence of H on x. In addition, the θ notation will denote either the directional angle or the corresponding unit vector, and the specific meaning will be clear in the context.

Using homogeneity of H in Eq. (3), we can rewrite $w(\theta)$ as

$$w(\theta) = \min_{\frac{p}{|p|}\cdot\theta>0,|p|\neq 0}\left[\frac{H\left(\frac{p}{|p|}\right)}{\left(\frac{p}{|p|}\cdot\theta\right)}\right] = \min_{p\cdot\theta>0}\left[\frac{H(p)}{(p\cdot\theta)}\right]. \tag{4}$$

Then the minimization with respect to p implies that

$$\theta = \frac{(p \cdot \theta)}{H(p)} \nabla_p H(p). \tag{5}$$

According to the method of characteristics for the static Hamilton–Jacobi equation, $\nabla_p H(p)$ is the ray direction. Therefore, the above condition indicates that θ is exactly the local ray direction; see [19] for illustration of this in another setting.

Notice that if $H(p)$ is strictly convex in p, then the mapping

$$p \in \{p : H(p) = \text{const.}\} \to \frac{\nabla_p H(p)}{|\nabla_p H(p)|} \in S^{d-1}$$

is one-to-one and onto. Since $\nabla_p H(p)$ has the same direction as the ray according to the method of characteristics and θ is exactly the local ray direction, this implies that given θ there is a unique $p \in \{p : H(p) = \text{const.}\}$ so that Eq. (5) holds. This relation has been used in [16,17] to formulate paraxial anisotropic eikonal equations and in [19] to design a fast sweeping method for anisotropic eikonal equations.

2.2. 2-D numerical Hamiltonians

To illustrate the idea, we derive the 2-D numerical Hamiltonian 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 a 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;
- $h = \sup_{\mathcal{T} \in \mathcal{T}_h} h_{\mathcal{T}} < 1$;
- $\mathcal{T}_h$ 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) 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 O and its n triangles $\mathcal{T}_1, \mathcal{T}_2, \ldots, \mathcal{T}_n$. For a typical triangle $\triangle OP_1P_2$, we denote $O : (x_0, y_0), P_1 : (x_1, y_1)$ and $P_2 : (x_2, y_2)$; $|OP_1|$, $|OP_2|$ and $|P_1P_2|$ are the lengths of the edges OP_1, OP_2 and P_1P_2, respectively.

During the solution process we need a local solver at the vertex O for each triangle. To achieve such a purpose, we construct the numerical Hamiltonian as follows:

$$\widehat{H}(\nabla\phi(x)) = \max_{\theta}[(\widehat{\nabla\phi \cdot \theta})w(\theta)], \quad \theta \in S^{d-1},$$

where $\widehat{\nabla\phi \cdot \theta}$ is obtained by piecewise linear approximation.

Consider a vertex O and its associated triangles in a triangular mesh as shown in Fig. 1. Taking one typical triangle, such as ΔP_2OP_1, as an example, the vectors $\overrightarrow{OP_1}$ and $\overrightarrow{OP_2}$ have directional angles θ_1 and θ_2, respectively. Thus by linear approximation, we have

$$\frac{\phi(P_1) - \phi(O)}{|OP_1|} \approx \nabla\phi \cdot (\theta_1) = [\cos(\theta_1), \sin(\theta_1)] \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix}$$

and

$$\frac{\phi(P_2) - \phi(O)}{|OP_2|} \approx \nabla\phi \cdot (\theta_2) = [\cos(\theta_2), \sin(\theta_2)] \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix}.$$

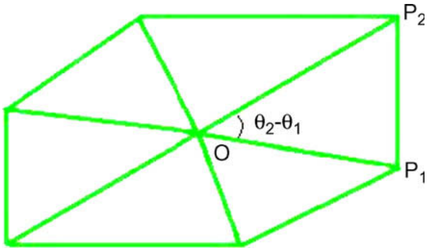


Fig. 1. A neighborhood of a vertex in a triangular mesh.

Thus

$$\begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) \\ \cos(\theta_2) & \sin(\theta_2) \end{bmatrix} \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} \approx \begin{bmatrix} \frac{\phi(P_1)-\phi(O)}{|OP_1|} \\ \frac{\phi(P_2)-\phi(O)}{|OP_2|} \end{bmatrix}$$

and this implies that

$$\nabla\phi = \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} \approx \frac{1}{\sin(\theta_2-\theta_1)} \begin{bmatrix} \sin(\theta_2) & -\sin(\theta_1) \\ -\cos(\theta_2) & \cos(\theta_1) \end{bmatrix} \begin{bmatrix} \frac{\phi(P_1)-\phi(O)}{|OP_1|} \\ \frac{\phi(P_2)-\phi(O)}{|OP_2|} \end{bmatrix} = \frac{\phi(P_1)-\phi(O)}{|OP_1|\sin(\theta_2-\theta_1)} \begin{bmatrix} \sin(\theta_2) \\ -\cos(\theta_2) \end{bmatrix}$$
$$+ \frac{\phi(P_2)-\phi(O)}{|OP_2|\sin(\theta_2-\theta_1)} \begin{bmatrix} -\sin(\theta_1) \\ \cos(\theta_1) \end{bmatrix}.$$

Now consider the ray starting from the segment P_1P_2 and reaching O with the direction θ, which has the same meaning as in the Legendre transform. Then $\theta-\pi$ points to the sector defined by P_1 and P_2. Thus

$$\nabla\widehat{\phi\cdot(\theta)} = -[\cos(\theta-\pi), \sin(\theta-\pi)] \begin{bmatrix} \phi_x \\ \phi_y \end{bmatrix} \approx \frac{\phi(O)-\phi(P_1)}{|OP_1|} \frac{\sin(\theta_2-(\theta-\pi))}{\sin(\theta_2-\theta_1)} + \frac{\phi(O)-\phi(P_2)}{|OP_2|} \frac{\sin((\theta-\pi)-\theta_1)}{\sin(\theta_2-\theta_1)},$$

where $\theta_1 \leqslant \theta-\pi \leqslant \theta_2$. Since the triangle is acute, $\sin(\theta_2-\theta_1) > 0$. Consequently, we also have that $0 \leqslant \frac{\sin(\theta_2-(\theta-\pi))}{\sin(\theta_2-\theta_1)} \leqslant 1$ and $0 \leqslant \frac{\sin((\theta-\pi)-\theta_1)}{\sin(\theta_2-\theta_1)} \leqslant 1$.

Therefore, locally in the neighborhood of O, the numerical Hamiltonian can be written as

$$\widehat{H}(\nabla\phi(O); p_1^j, p_2^j) = \max_\theta \left\{ \left(p_1^j \frac{\sin(\theta_2^j-(\theta-\pi))}{\sin(\theta_2^j-\theta_1^j)} + p_2^j \frac{\sin((\theta-\pi)-\theta_1^j)}{\sin(\theta_2^j-\theta_1^j)} \right) w(\theta) \right\},$$

where $\theta_1^j + \pi \leqslant \theta \leqslant \theta_2^j + \pi$

$$p_1^j = \frac{\phi(O)-\phi(P_1^j)}{|OP_1^j|}, \quad p_2^j = \frac{\phi(O)-\phi(P_2^j)}{|OP_2^j|};$$

the Hamilton–Jacobi equation (1) reduces to

$$\max_\theta \left\{ \left(p_1^j \frac{\sin(\theta_2^j-(\theta-\pi))}{\sin(\theta_2^j-\theta_1^j)} + p_2^j \frac{\sin((\theta-\pi)-\theta_1^j)}{\sin(\theta_2^j-\theta_1^j)} \right) w(\theta) \right\} - R(O) = 0,$$

where j is the index of all triangles sharing the same vertex O.

The consistency of the numerical Hamiltonian $\widehat{H}$ is a simple consequence of the definition; see [6,18].

Since

$$\frac{\partial\widehat{H}}{\partial p_1^j} \geqslant 0, \quad \frac{\partial\widehat{H}}{\partial p_2^j} \geqslant 0,$$

the above numerical Hamiltonian is monotone, and the resulting scheme is a monotone scheme [6,18].

2.3. 2-D local solvers

Since $w(\theta) > 0$, we may use Lemma 2.5 in [10] to simplify the above discretized Hamilton–Jacobi equation and solve for $\phi(O)$ in terms of its neighbors

$$\phi(O) = \min_\theta \left\{ \frac{\left(\frac{\phi(P_1^j)}{|OP_1^j|} \frac{\sin(\theta_2^j-(\theta-\pi))}{\sin(\theta_2^j-\theta_1^j)} + \frac{\phi(P_2^j)}{|OP_2^j|} \frac{\sin((\theta-\pi)-\theta_1^j)}{\sin(\theta_2^j-\theta_1^j)} \right) w(\theta) + R(O)}{\left(\frac{1}{|OP_1^j|} \frac{\sin(\theta_2^j-(\theta-\pi))}{\sin(\theta_2^j-\theta_1)} + \frac{1}{|OP_2^j|} \frac{\sin((\theta-\pi)-\theta_1^j)}{\sin(\theta_2^j-\theta_1)} \right) w(\theta)} \right\} = \min_\theta K(\theta), \tag{6}$$

where $\theta_1^j \leqslant \theta-\pi \leqslant \theta_2^j$ and $0 < \theta_2^j - \theta_1^j < \frac{\pi}{2}$.

Introducing the following notations:

$$a_1 = \frac{1}{|OP_1^j|\sin(\theta_2^j-\theta_1^j)}, \tag{7}$$

$$b_1 = \frac{1}{|OP_2^j|\sin(\theta_2^j-\theta_1^j)}, \tag{8}$$

$$f_1(\theta) = a_1\phi(P_1^j)\sin(\theta-\theta_2^j) + b_1\phi(P_2^j)\sin(\theta_1^j-\theta), \tag{9}$$

$$g_1(\theta) = a_1 \sin(\theta - \theta_2^j) + b_1 \sin(\theta_1^j - \theta), \tag{10}$$

$$f(\theta) = f_1(\theta)w(\theta) + R(O), \tag{11}$$

$$g(\theta) = g_1(\theta)w(\theta) \tag{12}$$

formula (6) becomes

$$\phi(O) = \min_\theta K(\theta) = \min_\theta \frac{f(\theta)}{g(\theta)}$$

and the minimum must be achieved in one of the following three cases: Case (i) $\theta_1^j + \pi$ for some j; Case (ii) $\theta_2^j + \pi$ for some j; Case (iii) $\arg_\theta\{\frac{dK(\theta)}{d\theta} = 0\}$.

Case (i): If $\theta = \theta_1^j + \pi$, then

$$K(\theta) = \phi(P_1^j) + \frac{R(O)}{w(\theta_1 + \pi)}|OP_1^j|.$$

Case (ii): If $\theta = \theta_2^j + \pi$, then

$$K(\theta) = \phi(P_2^j) + \frac{R(O)}{w(\theta_2 + \pi)}|OP_2^j|.$$

Case (iii): If $\theta_1^j < \theta - \pi < \theta_2^j$, we can verify that $\frac{dK(\theta)}{d\theta} = 0$ yields

$$0 = f'(\theta)g(\theta) - f(\theta)g'(\theta) = (f_1'g_1 - f_1g_1')w^2 - R(O)(g_1'w + g_1w') = a_1b_1(\phi(P_1^j) - \phi(P_2^j))\sin(\theta_1^j - \theta_2^j)w^2 - R(g_1'w + g_1w'),$$

where $'$ denotes the derivative with respect to θ. Therefore, we have

$$\frac{a_1b_1(\phi(P_2^j) - \phi(P_1^j))\sin(\theta_2^j - \theta_1^j)}{R(O)} = \frac{g_1'w + g_1w'}{w^2} \equiv F(\theta), \tag{13}$$

which is a nonlinear equation for θ. For some special cases, such as eikonal equations or quadratic type Hamilton–Jacobi equations, this equation can be solved exactly. Otherwise, numerical algorithms are needed to find the roots for

$$F(\theta) - \frac{a_1b_1(\phi(P_2^j) - \phi(P_1^j))\sin(\theta_2^j - \theta_1^j)}{R(O)} = 0.$$

2.4. 3-D numerical Hamiltonians

Next we consider Eq. (1) in a domain Ω in R^3, which has a triangulation $\mathcal{T}_h$ consisting of tetrahedrons. We consider every vertex and all tetrahedrons which are associated to this vertex. Again the question reduces to how to compute the numerical solution at the current central vertex for each tetrahedron.

Taking a typical tetrahedron with four vertexes O, P_1, P_2, and P_3 as an example, the vectors $\overrightarrow{OP_1}, \overrightarrow{OP_2}, \overrightarrow{OP_3}$ have directional angles $(\theta_1^1, \theta_2^1), (\theta_1^2, \theta_2^2)$, and (θ_1^3, θ_2^3), respectively where the subscripts 1 and 2 indicate the angles intersecting with the z-axis and x-axis, respectively. In the computation, we assume that the angles between $\overrightarrow{OP_1}$ and $\overrightarrow{OP_2}$, $\overrightarrow{OP_2}$ and $\overrightarrow{OP_3}$, $\overrightarrow{OP_1}$ and $\overrightarrow{OP_3}$ are non-obtuse; such a tetrahedron is termed acute [4]. If a tetrahedron is not acute, then one may use the splitting strategy proposed in [4] to pre-process this tetrahedron first.

Thus by linear approximation, we have

$$\frac{\phi(P_1) - \phi(O)}{|OP_1|} \approx \nabla\phi \cdot (\theta_1^1, \theta_2^1) = [\sin(\theta_1^1)\cos(\theta_2^1), \sin(\theta_1^1)\sin(\theta_2^1), \cos(\theta_1^1)]\begin{bmatrix}\phi_x \\ \phi_y \\ \phi_z\end{bmatrix},$$

$$\frac{\phi(P_2) - \phi(O)}{|OP_2|} \approx \nabla\phi \cdot (\theta_1^2, \theta_2^2) = [\sin(\theta_1^2)\cos(\theta_2^2), \sin(\theta_1^2)\sin(\theta_2^2), \cos(\theta_1^2)]\begin{bmatrix}\phi_x \\ \phi_y \\ \phi_z\end{bmatrix}$$

and

$$\frac{\phi(P_3) - \phi(O)}{|OP_3|} \approx \nabla\phi \cdot (\theta_1^3, \theta_2^3) = [\sin(\theta_1^3)\cos(\theta_2^3), \sin(\theta_1^3)\sin(\theta_2^3), \cos(\theta_1^3)]\begin{bmatrix}\phi_x \\ \phi_y \\ \phi_z\end{bmatrix}.$$

Let

$$\Theta = \begin{bmatrix} \sin(\theta_1^1)\cos(\theta_2^1) & \sin(\theta_1^1)\sin(\theta_2^1) & \cos(\theta_1^1) \\ \sin(\theta_1^2)\cos(\theta_2^2) & \sin(\theta_1^2)\sin(\theta_2^2) & \cos(\theta_1^2) \\ \sin(\theta_1^3)\cos(\theta_2^3) & \sin(\theta_1^3)\sin(\theta_2^3) & \cos(\theta_1^3) \end{bmatrix}.$$

Then we have

$$\Theta \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} \approx \begin{bmatrix} \frac{\phi(P_1)-\phi(O)}{|OP_1|} \\ \frac{\phi(P_2)-\phi(O)}{|OP_2|} \\ \frac{\phi(P_3)-\phi(O)}{|OP_3|} \end{bmatrix}$$

and this implies that

$$\nabla\phi = \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} \approx \frac{\begin{bmatrix} \Theta_{11} & \Theta_{21} & \Theta_{13} \\ \Theta_{21} & \Theta_{22} & \Theta_{23} \\ \Theta_{31} & \Theta_{23} & \Theta_{33} \end{bmatrix} \begin{bmatrix} \frac{\phi(P_1)-\phi(O)}{|OP_1|} \\ \frac{\phi(P_2)-\phi(O)}{|OP_2|} \\ \frac{\phi(P_3)-\phi(O)}{|OP_3|} \end{bmatrix}}{\cos(\theta_1^1)\Theta_1 + \cos(\theta_1^2)\Theta_2 + \cos(\theta_1^3)\Theta_3},$$

where

$$\begin{aligned}
\Theta_{11} &= \sin(\theta_1^2)\sin(\theta_2^2)\cos(\theta_1^3) - \sin(\theta_1^3)\sin(\theta_2^3)\cos(\theta_1^2), \\
\Theta_{21} &= \sin(\theta_1^3)\sin(\theta_2^3)\cos(\theta_1^1) - \sin(\theta_1^1)\sin(\theta_2^1)\cos(\theta_1^3), \\
\Theta_{31} &= \sin(\theta_1^1)\sin(\theta_2^1)\cos(\theta_1^2) - \sin(\theta_1^3)\sin(\theta_2^3)\cos(\theta_1^1), \\
\Theta_{12} &= \sin(\theta_1^3)\cos(\theta_2^3)\cos(\theta_1^2) - \sin(\theta_1^2)\cos(\theta_2^2)\cos(\theta_1^3), \\
\Theta_{22} &= \sin(\theta_1^1)\cos(\theta_2^1)\cos(\theta_1^3) - \sin(\theta_1^3)\cos(\theta_2^3)\cos(\theta_1^1), \\
\Theta_{32} &= \sin(\theta_1^2)\cos(\theta_2^2)\cos(\theta_1^1) - \sin(\theta_1^1)\cos(\theta_2^1)\cos(\theta_1^2), \\
\Theta_{13} &= \sin(\theta_1^3)\sin(\theta_1^2)\sin(\theta_2^3 - \theta_1^2), \\
\Theta_{23} &= \sin(\theta_1^1)\sin(\theta_1^3)\sin(\theta_2^1 - \theta_1^3), \\
\Theta_{33} &= \sin(\theta_1^2)\sin(\theta_1^1)\sin(\theta_2^2 - \theta_1^1)
\end{aligned}$$

and

$$\begin{aligned}
\Theta_1 &= \sin(\theta_1^2)\sin(\theta_1^3)\sin(\theta_2^3 - \theta_2^2), \\
\Theta_2 &= \sin(\theta_1^3)\sin(\theta_1^1)\sin(\theta_2^1 - \theta_2^3), \\
\Theta_3 &= \sin(\theta_1^1)\sin(\theta_1^2)\sin(\theta_2^2 - \theta_2^1).
\end{aligned}$$

Thus

$$\nabla\phi = \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} = \Theta^{-1} \begin{bmatrix} \frac{\phi(P_1)}{|OP_1|} \\ \frac{\phi(P_2)}{|OP_2|} \\ \frac{\phi(P_3)}{|OP_3|} \end{bmatrix} - \phi(O)\Theta^{-1} \begin{bmatrix} \frac{1}{|OP_1|} \\ \frac{1}{|OP_2|} \\ \frac{1}{|OP_3|} \end{bmatrix}.$$

Now consider the ray starting from the plane $P_1P_2P_3$ and reaching O has the direction (θ_1, θ_2), which has the same meaning as in the Legendre transform. Then $(\pi - \theta_1, \theta_2 - \pi)$ points to the plane defined by P_1, P_2 and P_3. Thus

$$\widehat{\nabla\phi \cdot (\theta)} = -n(\theta_1, \theta_2) \cdot \begin{bmatrix} \phi_x \\ \phi_y \\ \phi_z \end{bmatrix} = n(\theta_1, \theta_2) \cdot \Theta^{-1} p,$$

where

$$\begin{aligned}
n(\theta_1, \theta_2) &= \begin{bmatrix} \sin(\pi - \theta_1)\cos(\theta_2 - \pi) \\ \sin(\pi - \theta_1)\sin(\theta_2 - \pi) \\ \cos(\pi - \theta_1) \end{bmatrix}, \\
p^{\mathrm{t}} &= (p_1, p_2, p_3), \\
p_i &= \frac{\phi(O) - \phi(P_i)}{|OP_i|} \quad (i = 1, 2, 3).
\end{aligned}$$

Substituting this into the numerical Hamiltonian, we have

$$\widehat{H}(\nabla\phi(O);p_1,p_2,p_3)=\max_{\theta}\left\{(n(\theta)\cdot\Theta^{-1}p)w(\theta)\right\}.$$

Hence, the Hamilton–Jacobi equation reduces to

$$\max_{(\theta_1,\theta_2)}\left\{-n(\theta_1,\theta_2)^{\mathrm{T}}\left\{\Theta^{-1}\begin{bmatrix}\frac{\phi(P_1)}{|OP_1|}\\ \frac{\phi(P_2)}{|OP_2|}\\ \frac{\phi(P_3)}{|OP_3|}\end{bmatrix}-\phi(O)\Theta^{-1}\begin{bmatrix}\frac{1}{|OP_1|}\\ \frac{1}{|OP_2|}\\ \frac{1}{|OP_3|}\end{bmatrix}\right\}w(\theta_1,\theta_2)\right\}-R(O)=0.$$

Thus

$$\phi(O)=\min_{\theta_1,\theta_2}\left\{\frac{n(\theta_1,\theta_2)^{\mathrm{T}}\Theta^{-1}\begin{bmatrix}\frac{\phi(P_1)}{|OP_1|}\\ \frac{\phi(P_2)}{|OP_2|}\\ \frac{\phi(P_3)}{|OP_3|}\end{bmatrix}w(\theta_1,\theta_2)+R(O)}{n(\theta_1,\theta_2)^{\mathrm{T}}\Theta^{-1}\begin{bmatrix}\frac{1}{|OP_1|}\\ \frac{1}{|OP_2|}\\ \frac{1}{|OP_3|}\end{bmatrix}w(\theta_1,\theta_2)}\right\},$$

which yields a formula for $\phi(O)$ in terms of its neighbors.

The consistency of the numerical Hamiltonian $\widehat{H}$ is a consequence of the definition. To show that the numerical Hamiltonian $\widehat{H}$ is monotone, we rewrite the matrix Θ in terms of unit vectors

$$n_i=(\sin(\theta_1^i)\cos(\theta_2^i),\sin(\theta_1^i)\sin(\theta_2^i),\cos(\theta_1^i))^{\mathrm{t}}\quad(i=1,2,3)$$

and

$$\Theta^{\mathrm{t}}=(n_1,n_2,n_3).$$

Letting

$$\beta^{\mathrm{t}}=(\beta_1,\beta_2,\beta_3)=n^{\mathrm{t}}\Theta^{-1},$$

we have

$$n=\beta_1n_1+\beta_2n_2+\beta_3n_3$$

and

$$\widehat{H}(\nabla\phi(O);p_1,p_2,p_3)=\max_{\theta}\{(\beta_1p_1+\beta_2p_2+\beta_3p_3)w(\theta)\}.$$

Accordingly, the following holds:

$$\beta_1=\frac{n\cdot(n_2\times n_3)}{n_1\cdot(n_2\times n_3)},\quad\beta_2=\frac{n\cdot(n_1\times n_3)}{n_2\cdot(n_1\times n_3)},\quad\beta_3=\frac{n\cdot(n_1\times n_2)}{n_3\cdot(n_1\times n_2)}.$$

Notice that $n_1\cdot(n_2\times n_3)$ is the volume of the parallelepiped whose edges are n_1,n_2,n_3. Therefore, if n lies inside the cone defined by n_1,n_2 and n_3, then we conclude that $0\leqslant\beta_i\leqslant 1$ $(i=1,2,3)$, and the monotonicity of the numerical Hamiltonian follows.

3. A 2-D quadratic Hamiltonian

If

$$H(\cos v,\sin v)=\sqrt{a\cos^2 v+b\sin^2 v-2c\sin v\cos v},\tag{14}$$

where a, b, and c are given functions, then we can compute $w(\theta)$ directly

$$w(\theta)=\sqrt{\frac{ab-c^2}{a\sin^2\theta+b\cos^2\theta+2c\cos(\theta)\sin(\theta)}}.$$

Therefore, on a typical triangle $\triangle OP_1^jP_2^j$, by Eq. (13) we have

$$F(\theta)=\frac{g_1'w+g_1w'}{w^2}=\frac{\sin(\theta)(aa_2+cb_2)+\cos(\theta)(bb_2+ca_2)}{\sqrt{(ab-c^2)\left(a\sin^2\theta+b\cos^2(\theta)+2c\sin(\theta)\cos(\theta)\right)}},$$

where a_1 and b_1 are defined in Eqs. (7) and (8), respectively,

$$a_2 = a_1 \sin(\theta_2^j) - b_1 \sin(\theta_1^j),$$
$$b_2 = a_1 \cos(\theta_2^j) - b_1 \cos(\theta_1^j).$$

Letting

$$m = \frac{a_1 b_1(\phi(P_2^j) - \phi(P_1^j)) \sin(\theta_2^j - \theta_1^j)}{R(O)},$$

the nonlinear equation (13) is quadratic in $\tan\theta$ if $\cos(\theta) \neq 0$

$$q_2 \tan^2(\theta) + q_1 \tan(\theta) + q_0 = 0,$$

where

$$q_2 = (aa_2 + b_2c)^2 - am^2(ab - c^2),$$
$$q_1 = 2(aa_2 + b_2c)(bb_2 + ca_2) - 2cm^2(ab - c^2),$$
$$q_0 = (bb_2 + ca_2)^2 - bm^2(ab - c^2).$$

If the leading coefficient q_2 is not zero, we have

$$\theta^j = \tan^{-1}\left(\frac{-q_1 \pm \sqrt{q_1^2 - 4q_0q_2}}{2q_2}\right) + k\pi, \tag{15}$$

where k is chosen such that $\theta_1^j \leqslant \theta^j - \pi \leqslant \theta_2^j$ and $F(\theta^j) = m$; if $q_2 = 0$, then we choose $\theta^j = \tan^{-1}\left(-\frac{q_0}{q_1}\right) + k\pi$.

3.1. 2-D eikonal equations

Notice that for the eikonal equation, we have $a = b = 1$ and $c = 0$; as a result, we have the following simplified equation:

$$\frac{a_1 b_1(\phi(P_2^j) - \phi(P_1^j)) \sin(\theta_2^j - \theta_1^j)}{R(O)} = a_2 \sin(\theta) + b_2 \cos(\theta)$$

leading to

$$|OP_2^j| \cos(\theta_2^j - \theta) - |OP_1^j| \cos(\theta - \theta_1^j) = \frac{\phi(P_2^j) - \phi(P_1^j)}{R(O)}.$$

By using the subtraction formula of cosine, we have

$$\sin\theta(|OP_2^j| \sin\theta_2 - |OP_1^j| \sin\theta_1) + \cos\theta(|OP_2^j| \cos\theta_2 - |OP_1^j| \cos\theta_1) = \frac{\phi(P_2^j) - \phi(P_1^j)}{R(O)}.$$

Let $a_3 = |OP_2^j| \sin\theta_2 - |OP_1^j| \sin\theta_1$, $b_3 = |OP_2^j| \cos\theta_2 - |OP_1^j| \cos\theta_1$, and

$$c_3 \equiv \sqrt{a_3^2 + b_3^2} = \sqrt{|OP_1^j|^2 + |OP_2^j|^2 - 2|OP_1^j||OP_2^j| \cos(\theta_2 - \theta_1)} = |P_2^j P_1^j|.$$

Furthermore, introduce angle μ by letting $\frac{a_3}{c_3} = \cos(\mu)$ and $\frac{b_3}{c_3} = \sin(\mu)$. Therefore, we have

$$\sin(\theta + \mu) = \frac{\phi(P_2^j) - \phi(P_1^j)}{R(O)|P_2^j P_1^j|}.$$

Only when

$$\left|\frac{\phi(P_2^j) - \phi(P_1^j)}{R(O)|P_2^j P_1^j|}\right| \leqslant 1, \tag{16}$$

we have a solution for $\theta + \mu$. This condition is essentially a causality condition as stated in [18] for eikonal equations.

Letting

$$v = \arcsin\left(\frac{\phi(P_2^j) - \phi(P_1^j)}{R(O)|P_2^j P_1^j|}\right),$$

we have

$$\theta = v - \mu \quad \text{or} \quad \theta = \pi - v - \mu.$$

Since $\theta_1 < \theta - \pi < \theta_2$, we need to have

$$\theta_1 + \pi + \mu < \nu < \theta_2 + \pi + \mu,$$

or

$$-\theta_2 - \mu < \nu < -\theta_1 - \mu.$$

The algorithm for a typical simplex can be summarized as the following.

Step 1. Find θ by solving (13) for a general Hamiltonian or (15) for a quadratic Hamiltonian. Choose θ which satisfies $\theta_1 < \theta - \pi < \theta_2$. Compute $\phi^{\text{tmp}} = \phi(O)$ in (6).
Step 2. Compute $\phi_1^{\text{tmp}} = \phi(P_1^j) + \frac{R(O)}{w(\theta_1+\pi)}|OP_1^j|$ and $\phi_2^{\text{tmp}} = \phi(P_2^j) + \frac{R(O)}{w(\theta_2+\pi)}|OP_{21}^j|$.
Step 3. Update $\phi(O) = \min(\phi(O), \phi^{\text{tmp}}, \phi_1^{\text{tmp}}, \phi_2^{\text{tmp}})$.

If we go through each simplex j one by one, due to the sharing of vertexes of the adjacent simplexes, the update from a specific vertex in Step 2 is computed twice which is unnecessary. Also, a simplex need not considering if both $\phi(P_1^j)$ and $\phi(P_2^j)$ are larger than the current new candidate value. Thus a more efficient way to implement the above algorithm is

Step 1. Compute $\phi^v = \min_j \phi(P_1^j) + \frac{R(O)}{w(\theta_1+\pi)}|OP_1^j|$.
Step 2. Only consider the simplex j such that either $\phi(P_1^j)$ or $\phi(P_2^j)$ is smaller than ϕ^v. Find θ by solving (13) for a general Hamiltonian or (15) for a quadratic Hamiltonian. Choose θ which satisfies $\theta_1 < \theta - \pi < \theta_2$. Compute $\phi^e = \phi(O)$ in (6).
Step 3. Update $\phi(O) = \min(\phi(O), \phi^v, \phi^e)$.

4. Algorithms and implementation

We now describe the complete algorithm combining the local solver explained in the previous sections with the fast sweeping strategy developed in [18].

- 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.

Notice that the local solver is carried out at each node by the theoretical formula for a quadratic Hamiltonian mentioned in Section 3 or numerical minimization. Basically one-ring neighboring vertexes at each node are stored at the beginning of the algorithm. Taking the three dimensional case as an example, a tetrahedral mesh is constructed by subdividing each cube

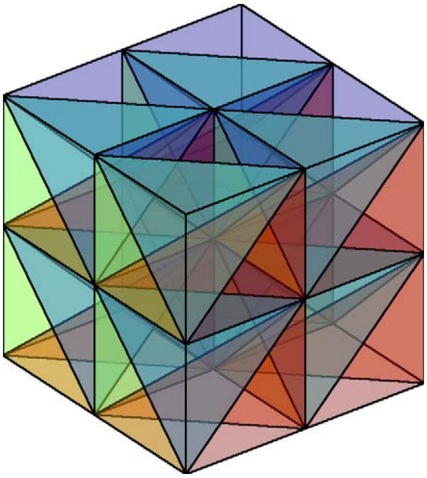


Fig. 2. A tetrahedron mesh is constructed by subdividing each cube with eight vertexes in a rectangular mesh into six tetrahedrons.

with eight vertexes in a rectangular mesh into six tetrahedrons as shown in Fig. 2. Denoting the neighboring grids $(i-1:i+1, j-1:j+1, k-1:k+1)$ of a node (i,j,k) as v_1 to v_{27}, the one-ring neighbors are v_2, $v_4, v_5, v_6, v_8, v_9, v_{10}, v_{11}$, $v_{13}, v_{15}, v_{17}, v_{18}, v_{19}, v_{20}, v_{22}, v_{23}, v_{24}$, and v_{26}. Totally there are 18 neighboring vertexes which form 48 different triangular faces and 24 tetrahedral elements. In a general irregular mesh, the number of the neighboring vertexes may be different. Following a similar updating process mentioned in Section 3.1, we first find the minimum ϕ^v of the candidate values by using one neighboring point (traveling along a line). Second, we find the minimum ϕ^e of the candidate values by using two neighboring points (traveling on a face) when at least one of the neighboring points is smaller than ϕ^v. Third, we find the minimum ϕ^t of the candidate values by using three neighboring points (traveling through a tetrahedron) when at least one of the neighboring points is smaller than ϕ^e. In this way, unnecessary computation can be avoided at each node.

5. Numerical results

5.1. 2-D examples

Example 1. We start with the simplest example which is the eikonal equation with the unit speed function. The value of ϕ is specified as 0 at $(x,y) = (0,0)$. The exact solution is the distance function to the origin, i.e.,

$$\phi(x,y) = \sqrt{x^2 + y^2}.$$

We first obtain a triangularization of the computational domain $\Omega = [-1,1] \times [-1,1]$. This can be done by using Matlab Toolbox "pdetool" or other mesh generation software. Here we simply use the "pdetool" toolbox to specify a square computational domain and then generate an initial mesh. The mesh can be further refined by dividing each triangle into four smaller ones. The data which includes points, edges, and triangles of refined meshes can then be exported for usage. In Fig. 3(a), we show a triangulated mesh with 448 triangles and 249 vertexes for the square domain. Furthermore, the refined mesh with 1792 triangles and 945 vertexes is shown in Fig. 3(b). In Fig. 4, we show the numerical result on a further refined mesh with 114688 triangles and 57729 vertexes. In Table 1, the l^1 error is computed for various meshes. We can clearly see the first-order convergence after the mesh is fine enough.

Example 2. The second example is an eikonal equation with two boundary points at (0.5,0.5) and (−0.5,−0.5). The solution is the minimum distance to these two seed points, and it is not smooth (non-differentiable) along the equal distance line $x + y = 0$ shown in Fig. 5. In Table 2, the first order convergence is observed.

Example 3. In Fig. 6, we consider the shape-from-shading examples. The computational domain is the unit square $\Omega = [0,1] \times [0,1]$. The governing equation is

$$|\nabla\phi| = f(x,y),$$

where

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

The boundary condition is given at $\Gamma = \{(\frac{1}{4},\frac{1}{4}), (\frac{3}{4},\frac{3}{4}), (\frac{1}{4},\frac{3}{4}), (\frac{3}{4},\frac{1}{4}), (\frac{1}{2},\frac{1}{2})\}$, consisting of five isolated points and the boundary of Ω. We consider two different cases.

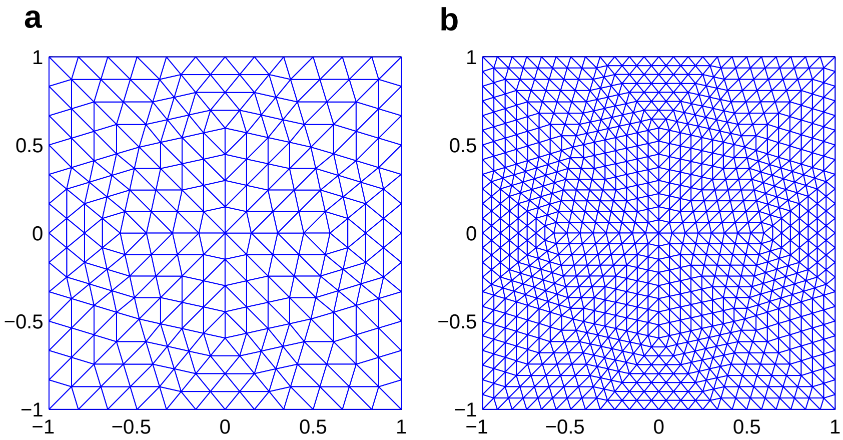


Fig. 3. (a) A triangulated mesh with 448 triangles and 249 vertexes. (b) The refined mesh with 1792 triangles and 945 vertexes.

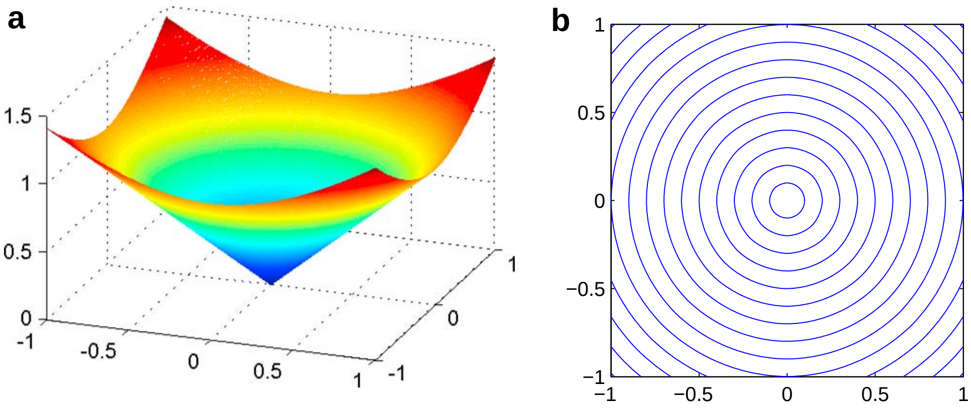


Fig. 4. $|\nabla\phi| = 1$ with $\phi(0,0) = 0$ on the domain $\Omega = [-1,1] \times [-1,1]$. (a) Left: $\phi(x,y)$. (b) Right: The contour lines of ϕ from $\phi = 0.1$ to 1.4.

Table 1
Accuracy tests for $|\nabla\phi| = 1$ with $\phi(0,0) = 0$ on the domain $\Omega = [-1,1] \times [-1,1]$

Nodes	Elements	L^1 error	Order
249	448	0.0600	
945	1792	0.0362	0.7264
3681	7168	0.0219	0.7303
14529	28672	0.0113	0.9513
57729	114688	0.0055	1.0439

Case I:

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

and

$$\phi(x,y) = 0 \quad \text{for } (x,y) \in \partial\Omega.$$

The exact solution is

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

Case II:

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

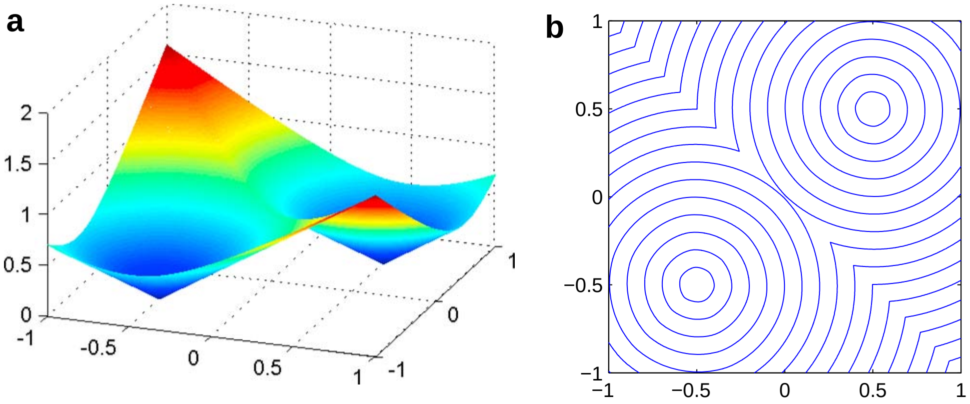
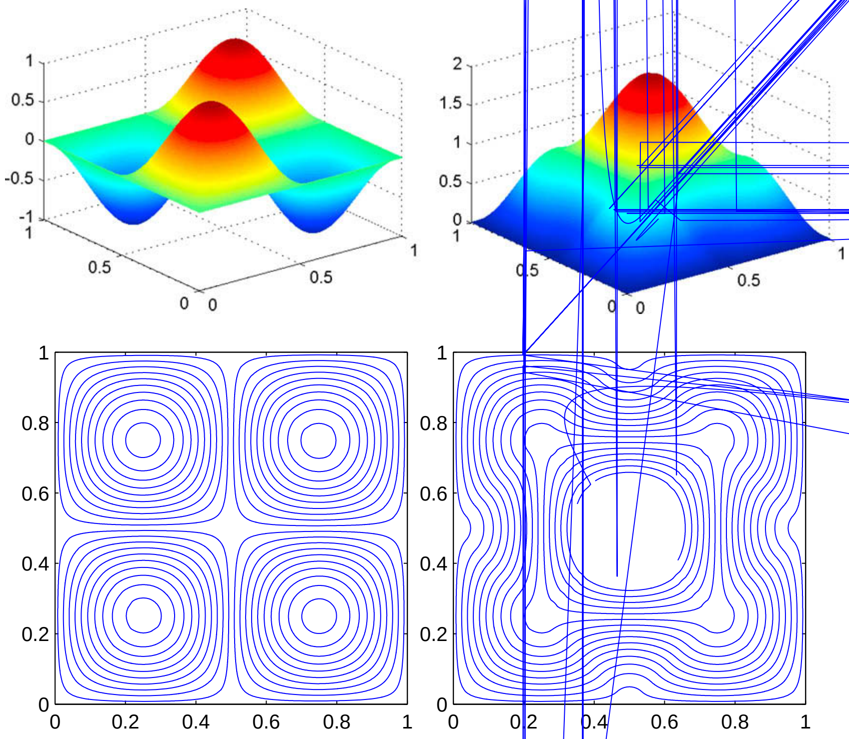


Fig. 5. $|\nabla\phi| = 1$ with $\phi(0.5,0.5) = \phi(-0.5,-0.5) = 0$ on the domain $\Omega = [-1,1] \times [-1,1]$. (a) Left: $\phi(x,y)$. (b) Right: The contour lines of ϕ from $\phi = 0.1$ to $\phi = 1.5$.

Table 2
Accuracy tests for $|\nabla\phi| = 1$ with $\phi(0.5, 0.5) = \phi(-0.5, -0.5) = 0$ on the domain $\Omega = [-1, 1] \times [-1, 1]$

Nodes	Elements	L^1 error	Order
249	448	0.0850	
945	1792	0.0516	0.7198
3681	7168	0.0285	0.8543
14 529	28 672	0.0143	0.9935
57 729	114 688	0.0070	1.0261



and

$$\phi(x, y) = 0 \quad \text{for } (x, y) \in \partial\Omega.$$

The exact solution is

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

In Fig. 6, we show the numerical results for both cases on the mesh with 458 752 triangles. The accuracy results in Table 3 also demonstrate the first order convergence.

Example 4. Now we consider anisotropic eikonal equations in the high frequency asymptotics for linear elastic wave equations. Since our algorithm works for a general convex Hamiltonian, we compute the travel time function for the quasi-SH wave in a typical anisotropic elastic model, the transversely isotropic solid with horizontal symmetry [15]. Then the quasi-SH eikonal equation is defined by the following equation:

$$\frac{1}{2}(a_{11} - a_{12})\phi_x^2 + a_{44}\phi_y^2 = 1,$$

where a_{ij} are given elastic parameters. Fig. 7(a) shows the simulation for $a_{11} = 15.6038, a_{12} = 6.5616$, and $a_{44} = 3.1258$. Fig. 7(b) shows the computational result for a model with two layers, so that the corresponding Hamilton–Jacobi equations have discontinuous coefficients; therefore, this model is used to test the stability and robustness of the sweeping scheme. As we can see from the figure, the Snell law for anisotropic media is well enforced.

Table 3
Accuracy tests for shape-from-shading examples. Left: case I; Right: case II

Nodes	Elements	L^1 error	Order	Nodes	Elements	L^1 error	Order
945	1792	0.0664		945	1792	0.0374	
3681	7168	0.0349	0.9276	3681	7168	0.0211	0.8266
14529	28672	0.0179	0.9645	14529	28672	0.0114	0.8886
57729	114688	0.0091	0.9817	57729	114688	0.0060	0.9253
230145	458752	0.0046	0.9901	230145	458752	0.0031	0.9528

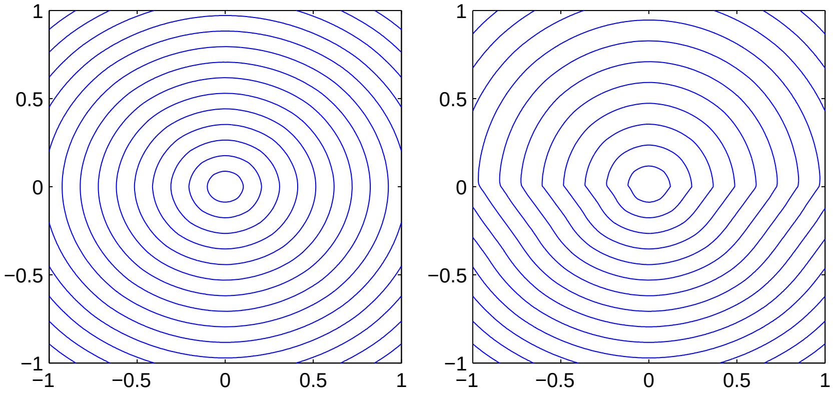


Fig. 7. Left: $a_{11} = 15.0638, a_{12} = 6.5616$, and $a_{44} = 3.1258$, contour difference = 0.05, mesh with 114688 triangles and 57729 vertexes. Right: $a_{11} = 15.3871, a_{12} = 3.4993$, and $a_{44} = 5.6074$ for upper half domain; $a_{11} = 15.0638$, $a_{12} = 6.5616$, and $a_{44} = 3.1258$ for the lower half domain; contour difference = 0.05, mesh with 114688 triangles and 57729 vertexes.

Example 5. Another important application for the quadratic type Hamiltonian (14) is related to the geodesic distance on a Riemannian manifold. Suppose that $P(x,y)$ is a point on a manifold M defined as the graph of a smooth function $f(x,y)$ and that γ are the curves connecting P and $\Gamma \subset M$ on the manifold. The minimizing curve of γ is called the geodesic. Let ϕ be the distance function such that

$$\phi(x,y) = \min_{\gamma \subset M} \int_{\gamma} ds.$$

Then ϕ is the solution of

$$\sqrt{\left(\frac{1+f_y^2}{1+f_x^2+f_y^2}\right)\phi_x^2 + \left(\frac{1+f_x^2}{1+f_x^2+f_y^2}\right)\phi_y^2 - 2\frac{f_x f_y}{1+f_x^2+f_y^2}\phi_x\phi_y} = 1, \quad \phi|_{\Gamma} = 0.$$

We show the result for $f(x,y) = \cos(2\pi x)\cos(2\pi y)$ in Fig. 8.

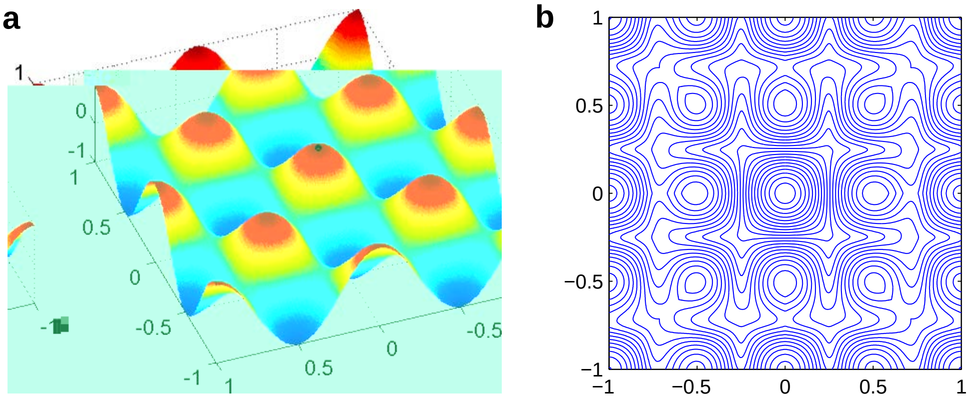


Fig. 8. (a) The surface $f(x,y) = \cos(2\pi x)\cos(2\pi y)$; (b) the distance contour from $(x,y,f(x,y)) = (0,0,1)$ on the $f(x,y) = \cos(2\pi x)\cos(2\pi y)$; contour difference = 0.1; the mesh consists of 114688 triangles and 57729 vertexes.

Table 4
The number of iterations

Nodes	Elements	Fig. 4	Fig. 5	Fig. 6 Case I	Fig. 6 Case II
945	1792	2	2	3	3
3681	7168	2	2	4	3
14529	28672	2	2	4	4
57729	114688	2	2	4	4

Nodes	Elements	Fig. 7(a)	Fig. 7(b)	Fig. 8
945	1792	2	2	12
3681	7168	2	2	12
14529	28672	2	2	12
57729	114688	2	2	12

Each iteration consists of four sweepings corresponding to four reference points, and each sweeping enumerates all the vertexes.

Table 4 summarizes the number of iterations for Examples 1–5. Here each iteration consists of four sweepings corresponding to four reference points, and each sweeping enumerates all the vertexes. As we can see, the number of iterations is almost constant, which indicates that fast sweeping methods may have linear complexity in these examples.

5.2. 3-D examples

Example 6. Fig. 9(a) and (b) shows numerical results for the eikonal equation with different boundary conditions. In Fig. 9(a), one seed point in the center $(0,0,0)$ is given. The exact solution is the distance function to $(0,0,0)$. In Fig. 9(b), two seed points, $(-0.5,-0.5,-0.5)$ and $(0.5,0.5,0.5)$, are given. The exact solution would be the minimum of the distance functions to these two seed points. These two simple examples provide an easy check for the numerical accuracy.

Example 7. In Fig. 10, we generalize one of the shape-from-shading examples to the 3-D case. The computational domain is the unit cube $\Omega = [0,1]^3$. The governing equation is

$$|\nabla\phi| = f(x,y,z),$$

where

$$f(x,y,z) = \pi\sqrt{f_1^2 + f_2^2 + f_3^2}$$

and

$$\begin{aligned} f_1(x,y,z) &= \cos(\pi x)\sin(\pi y)\sin(\pi z),\\ f_2(x,y,z) &= \sin(\pi x)\cos(\pi y)\sin(\pi z),\\ f_3(x,y,z) &= \sin(\pi x)\sin(\pi y)\cos(\pi z). \end{aligned}$$

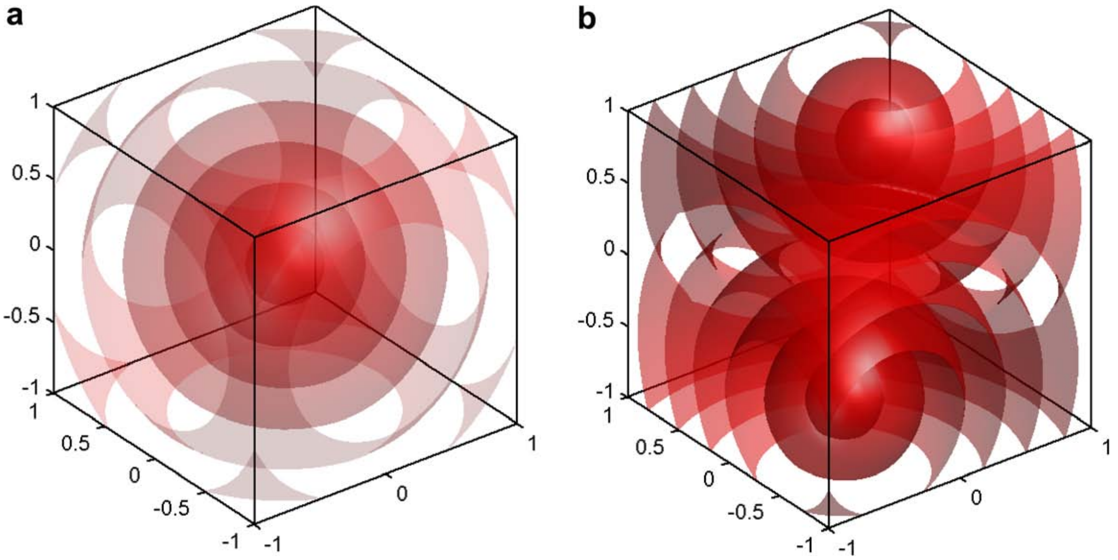


Fig. 9. The numerical results for eikonal equations with (a) one boundary condition $\phi(0,0,0) = 0$ (b) two boundary conditions $\phi(-0.5,-0.5,-0.5) = 0$ and $\phi(0.5,0.5,0.5) = 0$. The isocontours are plotted for contour difference 0.25.

The boundary condition is $\phi|_{\partial\Omega} = 0$. The exact solution is

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

The isosurfaces for $\phi = 0.1$, 0.4 and 0.7 are shown in Fig. 10(a)–(c) and the isocontour for $z = 0.5$ is shown in Fig. 10(d).

Example 8. We demonstrate an example with anisotropy. The governing equation is

$$\sqrt{a\phi_x^2 + b\phi_y^2 + c\phi_z^2 - 2d\phi_x\phi_y - 2e\phi_y\phi_z - 2f\phi_z\phi_x} = 1,$$

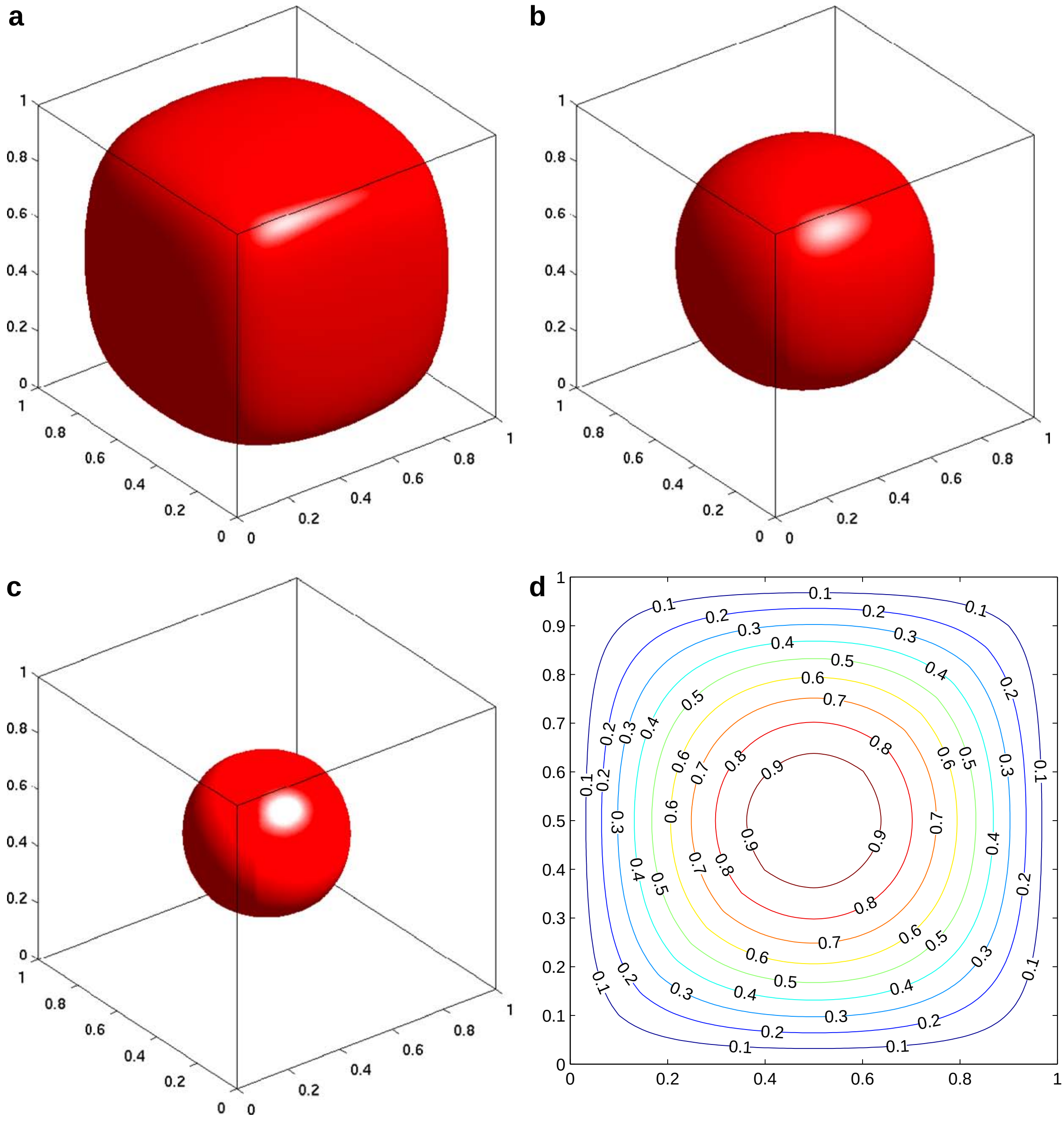


Fig. 10. The 3-D shape-from-shading example. (a) The isosurface for $\phi = 0.1$. (b) The isosurface for $\phi = 0.4$. (c) The isosurface for $\phi = 0.7$. (d) The isocontour for $z = 0.5$.

where

$$a=\frac{1+f_y^2+f_z^2}{1+f_x^2+f_y^2+f_z^2},\quad b=\frac{1+f_x^2+f_z^2}{1+f_x^2+f_y^2+f_z^2},\quad c=\frac{1+f_x^2+f_y^2}{1+f_x^2+f_y^2+f_z^2},$$
$$d=\frac{f_xf_y}{1+f_x^2+f_y^2+f_z^2},\quad e=\frac{f_yf_z}{1+f_x^2+f_y^2+f_z^2},\quad f=\frac{f_zf_x}{1+f_x^2+f_y^2+f_z^2}$$

and $f(x,y,z)=\cos(\pi x)\cos(\pi y)\cos(\pi z)$. The function w is

$$w(\theta_1,\theta_2)=\sqrt{\frac{1}{1+(f_x\sin\theta_1\cos\theta_2+f_y\sin\theta_1\sin\theta_2+f_z\cos\theta_1)^2}}.$$

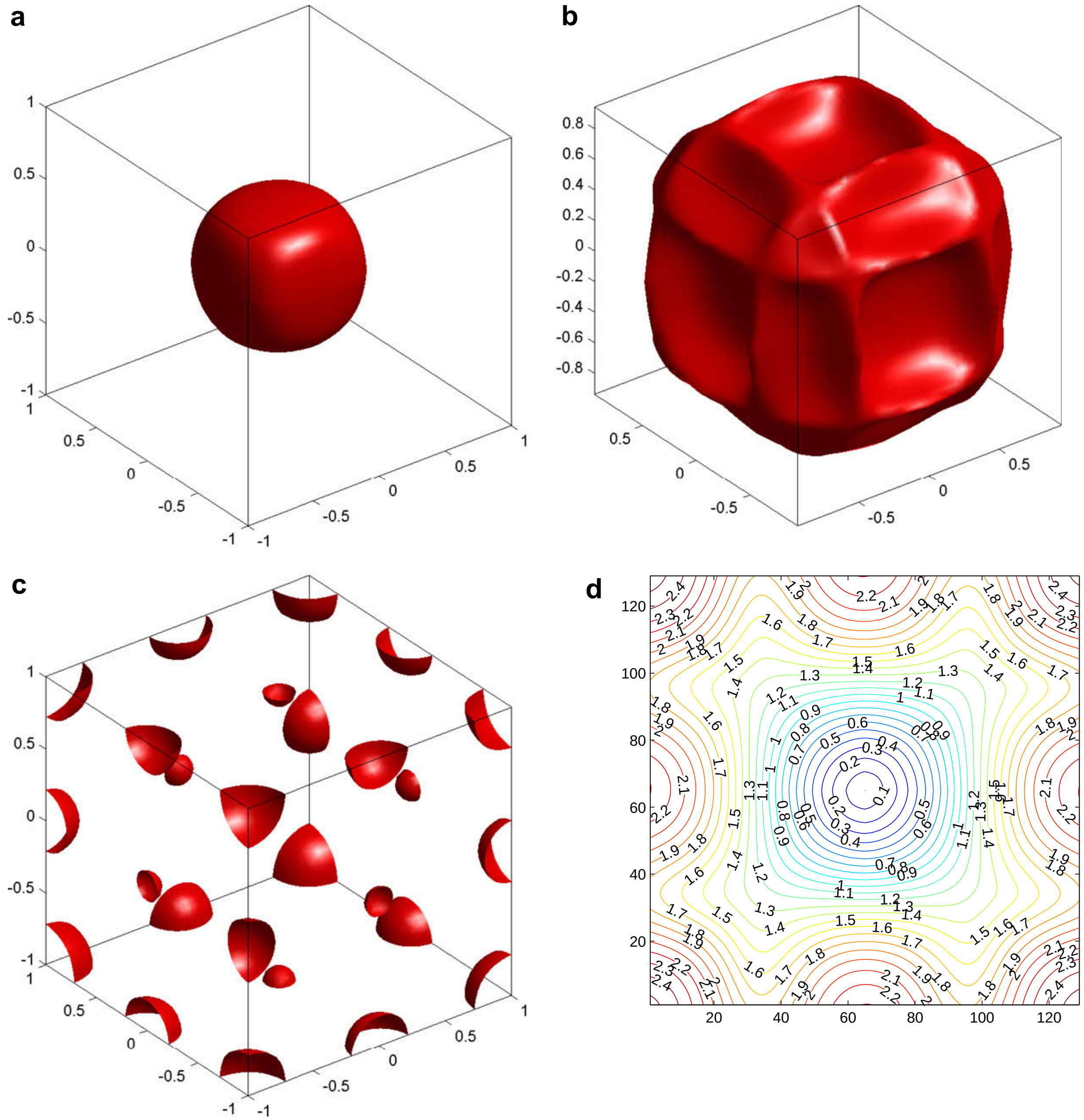


Fig. 11. The 3-D anisotropic example. (a) The isosurface for $\phi=0.8$. (b) The isosurface for $\phi=1.6$. (c) The isosurface for $\phi=2.3$. (d) The isocontour for $z=0$.

The boundary condition is $\phi(0,0,0) = 0$ and the computational domain is $[-1,1]^3$. The isosurfaces for $\phi = 0.8$, 1.6 and 2.3 are shown in Fig. 11(a)–(c) and the isocontour for $z = 0.0$ is shown in Fig. 11(d).

6. Conclusion

The new sweeping algorithm utilizes the Legendre transform of Hamiltonians on triangulated meshes. The algorithm yields the numerical solution at a grid point using only its one-ring neighboring grid values and is easy to implement numerically. We illustrate the efficiency and accuracy of the new method with several numerical examples in two and three dimensions.

Future work includes developing high-order schemes based on this first order scheme and developing a fast algorithm for non-convex Hamilton–Jacobi equations.

Acknowledgments

Osher is supported by an ONR MURI subcontract from Stanford University and Qian is supported in part by NSF DMS #0753797 and in part by NSF DMS #0810104.

References

[1] M. Bardi, S.J. Osher, The nonconvex multi-dimensional Riemann problem for Hamilton–Jacobi equations, SIAM J. Math. Anal. 22 (1991) 344–351.
[2] F. Bornemann, C. Rasch, Finite element discretization of static Hamilton–Jacobi equations based on a local variational principle, Comput. Visual. Sci. 9 (2006) 57–69.
[3] M. Boue, P. Dupuis, Markov chain approximations for deterministic control problems with affine dynamics and quadratic costs in the control, SIAM J. Numer. Anal. 36 (1999) 667–695.
[4] A.M. Bronstein, M.M. Bronstein, R. Kimmel, Weighted distance maps computation on parametric three-dimensional manifolds, J. Comput. Phys. 225 (2007) 771–784.
[5] T. Cecil, S.J. Osher, J. Qian, Simplex free adaptive tree fast sweeping and evolution methods for solving level set equations in arbitrary dimension, J. Comput. Phys. 213 (2006) 458–473.
[6] B. Cockburn, J. Qian, Continuous dependence results for Hamilton–Jacobi equations, in: D. Estep, S. Tavener (Eds.), Collected Lectures on the Preservation of Stability Under Discretization, SIAM, Philadelphia, PA, 2002, pp. 67–90.
[7] J. Helmsen, E. Puckett, P. Colella, M. Dorr. Two new methods for simulating photolithography development in 3-D, in: Proc. SPIE, vol. 2726, 1996, pp. 253–261.
[8] M. Jackowski, C. Kao, L. Staib. Estimation of anatomical connectivity by anisotropic front propagation and diffusion tensor imaging, in: MICCAI, 2004.
[9] C.Y. Kao, S.J. Osher, J. Qian, Lax–Friedrichs sweeping schemes for static Hamilton–Jacobi equations, J. Comput. Phys. 196 (2004) 367–391.
[10] C.Y. Kao, S.J. Osher, Y.-H. Tsai, Fast sweeping method for static Hamilton–Jacobi equations, SIAM J. Numer. Anal. 42 (2005) 2612–2632.
[11] S. Leung, J. Qian, An adjoint state method for three-dimensional transmission traveltime tomography using first-arrivals, Commun. Math. Sci. 4 (2006) 249–266.
[12] F. Li, C.W. Shu, Y.T. Zhang, H.-K. Zhao, A second order discontinuous galerkin fast sweeping method for eikonal equations, Preprint, 2007.
[13] S.J. Osher, C.W. Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, SIAM J. Numer. Anal. 28 (1991) 907–922.
[14] D. Peng, S.J. Osher, B. Merriman, H.-K. Zhao, The geometry of Wulff crystal shapes and its relations with Riemann problems, in: Nonlinear Partial Differential Equations, Amer. Math. Soc., Providence, RI, 1999, pp. 251–303.
[15] J. Qian, L.-T. Cheng, S.J. Osher, A level set based Eulerian approach for anisotropic wave propagations, Wave Motion 37 (2003) 365–379.
[16] J. Qian, W.W. Symes, Paraxial eikonal solvers for anisotropic quasi-P traveltimes, J. Comput. Phys. 173 (2001) 1–23.
[17] J. Qian, W.W. Symes, Finite-difference quasi-P traveltimes for anisotropic media, Geophysics 67 (2002) 147–155.
[18] J. Qian, Y.T. Zhang, H.K. Zhao, Fast sweeping methods for eikonal equations on triangulated meshes, SIAM J. Numer. Anal. 45 (2007) 83–107.
[19] J. Qian, Y.T. Zhang, H.K. Zhao, Fast sweeping methods for static Hamilton–Jacobi equations on triangulated meshes, J. Sci. Comput. 31 (2007) 237–271.
[20] E. Rouy, A. Tourin, A viscosity solutions approach to shape-from-shading, SIAM J. Numer. Anal. 298 (1992) 67–884.
[21] J.A. Sethian, Level Set Methods, Cambridge Univ. Press, 1996.
[22] J.A. Sethian, A. Vladimirsky. Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms, in: PAM-792, University of California at Berkeley, Berkeley, CA94720, 2001.
[23] R. Tsai, L.-T. Cheng, S.J. Osher, H.K. Zhao, Fast sweeping method for a class of Hamilton–Jacobi equations, SIAM J. Numer. Anal. 41 (2003) 673–694.
[24] J.N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, IEEE Trans. Automatic Control 40 (1995) 1528–1538.
[25] Y.T. Zhang, H.K. Zhao, J. Qian, High order fast sweeping methods for static Hamilton–Jacobi equations, J. Sci. Comput. 29 (2006) 25–56.
[26] H.K. Zhao, Fast sweeping method for eikonal equations, Math. Comput. 74 (2005) 603–627.
[27] H.K. Zhao, Parallel implementations of the fast sweeping method, J. Comput. Math. 25 (2007) 421–429.