Journal of Computational Physics 241 (2013) 104–117

Contents lists available at SciVerse ScienceDirect

Journal of Computational Physics

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

A uniformly second order fast sweeping method for eikonal equations

Songting Luo

Department of Mathematics, Iowa State University, Ames, IA 50011, USA

ARTICLE INFO

Article history:
Received 17 October 2012
Received in revised form 28 January 2013
Accepted 29 January 2013
Available online 13 February 2013

Keywords:
Discontinuous Galerkin method
Superconvergence
Fast sweeping method
Uniformly second order
Eikonal equations

ABSTRACT

A uniformly second order method with a local solver based on the piecewise linear discontinuous Galerkin formulation is introduced to solve the eikonal equation with Dirichlet boundary conditions. The method utilizes an interesting phenomenon, referred as the superconvergence phenomenon, that the numerical solution of monotone upwind schemes for the eikonal equation is first order accurate on both its value and gradient when the solution is smooth. This phenomenon greatly simplifies the local solver based on the discontinuous Galerkin formulation by reducing its local degrees of freedom from two (1-D) (or three (2-D), or four (3-D)) to one with the information of the gradient frozen. When considering the eikonal equation with point-source conditions, we further utilize a factorization approach to resolve the source singularities of the eikonal by decomposing it into two parts, either multiplicatively or additively. One part is known and captures the source singularities; the other part serves as a correction term that is differentiable at the sources and satisfies the factored eikonal equations. We extend the second order method to solve the factored eikonal equations to compute the correction term with second order accuracy, then recover the eikonal with second order accuracy. Numerical examples are presented to demonstrate the performance of the method.

1. Introduction

We consider the eikonal equation with Dirichlet boundary conditions,

$$\begin{aligned} H(\nabla\tau(\mathbf{x})) \equiv |\nabla\tau(\mathbf{x})| = s(\mathbf{x}), \quad & \mathbf{x}\in\Omega\setminus\Gamma; \\ \tau(\mathbf{x}) = g(\mathbf{x}), \quad & \mathbf{x}\in\Gamma\subset\Omega, \end{aligned} \tag{1}$$

where Ω is the domain, τ is the eikonal (or traveltime), $s(\mathbf{x}) \geqslant \eta > 0$ is the slowness field with η being a constant, and $g(\mathbf{x})$ is a Lipschitz continuous function. In particular, we consider point-source boundary conditions, e.g., $\Gamma = \{\mathbf{x}_i\}_{i=1}^n$ and $g(\{\mathbf{x}_i\}_{i=1}^n) = 0$. This boundary value problem (1) has wide applications from classical mechanics, geosciences, geometrical optics, computer vision to optimal control. The classical solution does not exist in general in that the characteristics may intersect. Crandall and Lions [11] introduced the concept of viscosity solutions for Hamilton–Jacobi equations, following which a globally unique weak solution can be defined. The solution is locally smooth with singularities in the gradient along certain sub-manifolds of codimension 1, 2 or 3 (in 3-D). Therefore it is possible to design high order methods that remain high order accurate away from the singularities. High order methods are highly desirable in many applications, e.g., in high frequency wave propagation where the eikonal equation is coupled to a transport equation through its gradient [14,25].

E-mail address: luos@iastate.edu

0021-9991/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.jcp.2013.01.042

The eikonal equation has been tackled numerically from many different perspectives, resulting in the vast literature on the topic. The methods can be divided mainly into two classes: (1) the problem is transferred into a time-dependent problem and solved with "time-marching" methods to reach the steady state; see [15,16,34,35] and references therein; and (2) the problem is treated as a stationary problem and solved directly with efficient iterative methods after discretization; see [1,3,12,13,18,19,22–24,26,38–49,51–54,59,60] and references therein. The essentially non-oscillatory (ENO) or the weighted essentially non-oscillatory (WENO) technique based high order finite difference methods have also been introduced to solve the eikonal equation (e.g., see [21,28,36,58]), where a wide stencil is used in the formulation, and the schemes are not fully upwind such that the computational complexity increases slightly more than linearly as the mesh is refined. Discontinuous Galerkin (DG) methods provide another high order accurate approach to solving the problem by using more compact stencils. The compactness of the stencil is a result of incorporating the information of the derivatives into the local formulation, which in return uses more degrees of freedom locally [4–10,20,55]. In [27,57], a second order DG method for the eikonal equation with linear polynomial approximations was introduced. This method utilizes the local DG solver introduced in [4] to derive updating formulas for the local unknowns. A complicated strategy was designed to enforce the causality condition in the local solver to obtain the second order accuracy without losing efficiency. The method was only uniformly second order in l_1 norm when first developed in [27]. And it was later improved to be uniformly second order in both l_1 norm and l_∞ norm in [57] by introducing an even more complicated strategy to choose the causality condition.

In this paper, we develop a new, simple and uniformly second order method based on a combination of the local DG formulation in [4,27,57] and an interesting phenomenon, referred as the superconvergence phenomenon, that the numerical solution for the eikonal equation computed with monotone upwind schemes is first order accurate on both its value and gradient when the solution is smooth. This superconvergence phenomenon allows us to freeze the information of the gradient in the local DG formulation through the information of the gradient obtained by monotone upwind schemes such that the local degrees of freedom are reduced from two (1-D) (or three (2-D), or four (3-D)) to only one, i.e., the cell average. A simple local updating formula for the cell average can be easily derived and is incorporated into an iterative method with Gauss–Seidel iterations and alternating orderings, which results in a simple and uniformly second order method for the eikonal equation. When there are point-source boundary conditions, we further utilize a factorization approach introduced in [17,30,31,33,37,56] to resolve the source singularities of the eikonal, where the eikonal is decomposed into two parts either multiplicatively or additively. One part contains the preknown information of the eikonal and captures the source singularities, which makes the other part differentiable at the sources such that it can be computed with high accuracy through the factored eikonal equations. The proposed method can be extended to solve the factored eikonal equations with second order accuracy, then the eikonal is recovered with second order accuracy. Here we also remark that the superconvergence phenomenon has already been observed and utilized in a compact upwind second order method developed in [2,29] by incorporating it into the Lagrangian structure of the eikonal equation to design a local solver which serves as a postprocess for the numerical solution obtained with monotone upwind schemes.

The rest of the paper is organized as follows. In Section 2, we first present the new method for the eikonal equation, then its extension to the factored eikonal equations is presented. In Section 3, numerical examples are presented to demonstrate the method. Conclusive remarks are given in Section 4.

2. Numerical method

We present the numerical formulations in 2-D. Extension to 3-D is straightforward. Assume the domain Ω is partitioned as $\Omega^h = \cup_{1\leqslant i\leqslant I, 1\leqslant j\leqslant J} I_{ij}$ where the cell $I_{ij} = I_i \times I_j$, $I_i = [x_{i-1/2}, x_{i+1/2}]$, and $I_j = [z_{j-1/2}, z_{j+1/2}]$. The centers of I_i and I_j are denoted as $x_i = (x_{i-1/2} + x_{i+1/2})/2$ and $z_j = (z_{j-1/2} + z_{j+1/2})/2$ respectively, and the cell sizes are $\Delta x_i = x_{i+1/2} - x_{i-1/2}$ and $\Delta z_j = z_{j+1/2} - z_{j-1/2}$ respectively. For simplicity we consider uniform meshes, i.e., $\Delta x_i = \Delta z_j = h$ for $1 \leqslant i \leqslant I$, $1 \leqslant j \leqslant J$. We further define the piecewise linear polynomial approximation space as

$$V_h^1 = \{v : v|_{I_{ij}} \in P^1(I_{ij}),\ \forall i, j\},$$

where $P^1(I_{ij})$ denotes linear polynomials on I_{ij}.

2.1. Eikonal equations

Following the formulations in [4,27,57], a numerical scheme based on the discontinuous Galerkin formulation for (1) is formulated as follows: find $\tau_h \in V_h^1$ such that $\forall w_h \in V_h^1$,

$$\begin{aligned}
&\int_{I_{ij}} |\nabla \tau_h| w_h(x,z)\,dx\,dz + \alpha_{e,ij} \int_{I_j} [\tau_h](x_{i+1/2}, z) w_h(x_{i+1/2}^-, z)\,dz + \alpha_{w,ij} \int_{I_j} [\tau_h](x_{i-1/2}, z) w_h(x_{i-1/2}^+, z)\,dz + \alpha_{n,ij} \int_{I_i} [\tau_h] \\
&\quad \times (x, z_{j+1/2}) w_h(x, z_{j+1/2}^-)\,dx + \alpha_{s,ij} \int_{I_i} [\tau_h](x, z_{j-1/2}) w_h(x, z_{j-1/2}^+)\,dx \\
&\quad = \int_{I_{ij}} s(x,z) w_h(x,z)\,dx\,dz. \qquad (2)
\end{aligned}$$

$[\tau_h]$ denotes the jump of τ_h across the cell boundary, and

$$w_h(x^-_{i+1/2},z)=\lim_{x\to x^-_{i+1/2}} w_h(x,z);\quad w_h(x^+_{i-1/2},z)=\lim_{x\to x^+_{i-1/2}} w_h(x,z);$$
$$w_h(x,z^-_{j+1/2})=\lim_{z\to z^-_{j+1/2}} w_h(x,z);\quad w_h(x,z^+_{j-1/2})=\lim_{z\to z^+_{j-1/2}} w_h(x,z).$$

$\alpha_{e,ij},\alpha_{w,ij},\alpha_{n,ij},$ and $\alpha_{s,ij}$ are local constants dependent of the numerical solutions on I_{ij} and its neighboring cells. $\alpha_{e,ij}$ and $\alpha_{w,ij}$ are local approximations of $\partial H/\partial\tau_x$, and $\alpha_{n,ij},$ and $\alpha_{s,ij}$ are local approximations of $\partial H/\partial\tau_z$. In order to enforce the causality property of the eikonal equation, i.e., the information propagates along characteristics from the boundary to the whole domain, we need to choose the local constants wisely.

The linear polynomial τ_h on I_{ij} is represented as $\tau_h|_{I_{ij}}=\bar\tau_{ij}+\xi_{ij}X_i+\eta_{ij}Z_j$ with $X_i=(x-x_i)/h$ and $Z_j=(z-z_j)/h$. The unknowns on I_{ij} are $\bar\tau_{ij},\ \xi_{ij}$ and η_{ij}. It is easy to see that $\bar\tau_{ij},\ \xi_{ij}/h$ and η_{ij}/h are approximations of $\tau,\ \tau_x$ and τ_z at (x_i,z_j), respectively.

Let us assume that a set of numerical solutions $\{(\tau^1,\tau^1_x,\tau^1_z)_{ij}\}_{1\leqslant i\leqslant I,1\leqslant j\leqslant J}$ of (1) have already been computed with certain monotone upwind schemes at cell centers. $(\tau^1,\tau^1_x,\tau^1_z)_{ij}$ denotes the numerical approximations of (τ,τ_x,τ_z) at (x_i,z_j). For our implementations, we use the fast sweeping method (FSM); see Appendix A.

With the concern of the causality enforcement, we choose the local constants $\alpha_{e,ij},\alpha_{w,ij},\alpha_{n,ij}$, and $\alpha_{s,ij}$ as follows:

$$\begin{aligned}\alpha_{e,ij}&=\min\left\{0,\frac{\xi_{ij}}{\sqrt{\xi^2_{ij}+\eta^2_{ij}}}\right\},\quad \alpha_{w,ij}=\max\left\{0,\frac{\xi_{ij}}{\sqrt{\xi^2_{ij}+\eta^2_{ij}}}\right\},\\ \alpha_{n,ij}&=\min\left\{0,\frac{\eta_{ij}}{\sqrt{\xi^2_{ij}+\eta^2_{ij}}}\right\},\quad \alpha_{s,ij}=\max\left\{0,\frac{\eta_{ij}}{\sqrt{\xi^2_{ij}+\eta^2_{ij}}}\right\}.\end{aligned}\tag{3}$$

Given cell I_{ij}, by choosing $w_h=1,X_i,Z_j$ and simply using the mid-point rule for numerical integrations, the local DG formulation (2) implies a system of nonlinear algebraic equations for determining $(\bar\tau_{ij},\xi_{ij},\eta_{ij})$:

$$\begin{aligned}\sqrt{\xi^2_{ij}+\eta^2_{ij}}+\gamma_{ij}\bar\tau_{ij}+\beta_{ij}\xi_{ij}+\lambda_{ij}\eta_{ij}&=R^1_{ij},\\ 12\beta_{ij}\bar\tau_{ij}+g_{ij}\xi_{ij}&=R^2_{ij},\\ 12\lambda_{ij}\bar\tau_{ij}+e_{ij}\eta_{ij}&=R^3_{ij},\end{aligned}\tag{4}$$

where

$$\begin{aligned}\gamma_{ij}&=\alpha_{w,ij}+\alpha_{s,ij}-\alpha_{e,ij}-\alpha_{n,ij},\\ \beta_{ij}&=-0.5(\alpha_{e,ij}+\alpha_{w,ij}),\quad \lambda_{ij}=-0.5(\alpha_{n,ij}+\alpha_{s,ij}),\\ g_{ij}&=-3\alpha_{e,ij}+3\alpha_{w,ij}-\alpha_{n,ij}+\alpha_{s,ij},\\ e_{ij}&=-3\alpha_{n,ij}+3\alpha_{s,ij}-\alpha_{e,ij}+\alpha_{w,ij}\end{aligned}\tag{5}$$

and

$$\begin{aligned}R^1_{ij}&=s_{ij}h-\alpha_{e,ij}(\bar\tau_{i+1,j}-0.5\xi_{i+1,j})+\alpha_{w,ij}(\bar\tau_{i-1,j}+0.5\xi_{i-1,j})\\ &\quad-\alpha_{n,ij}(\bar\tau_{i,j+1}-0.5\eta_{i,j+1})+\alpha_{s,ij}(\bar\tau_{i,j-1}+0.5\eta_{i,j-1}),\\ R^2_{ij}&=-6\alpha_{e,ij}(\bar\tau_{i+1,j}-0.5\xi_{i+1,j})-6\alpha_{w,ij}(\bar\tau_{i-1,j}-0.5\xi_{i-1,j})-\alpha_{n,ij}\xi_{i,j+1}+\alpha_{s,ij}\xi_{i,j-1},\\ R^3_{ij}&=-6\alpha_{n,ij}(\bar\tau_{i,j+1}-0.5\eta_{i,j+1})-6\alpha_{s,ij}(\bar\tau_{i,j-1}-0.5\eta_{i,j-1})-\alpha_{e,ij}\eta_{i+1,j}+\alpha_{w,ij}\eta_{i-1,j}.\end{aligned}\tag{6}$$

To solve this system, we utilize the superconvergence phenomenon. When the solution τ is smooth, the numerical solutions obtained by the FSM is first order accurate on both its value and gradient, i.e.,

$$|\tau^1-\tau|_\infty=O(h),\quad |\tau^1_x-\tau_x|_\infty=O(h),\quad \text{and}\quad |\tau^1_z-\tau_z|_\infty=O(h).\tag{7}$$

With (7), we solve (4) in the following way: since ξ_{ij}/h and η_{ij}/h are approximations of τ_x and τ_z at (x_i,z_j) respectively. We do not update ξ_{ij} and η_{ij} from (4); instead, we fix them as

$$\xi_{ij}=\tau^1_{x,ij}h,\quad \text{and}\quad \eta_{ij}=\tau^1_{z,ij}h.\tag{8}$$

In other words, we do not use the last two equations in (4). Only the first equation in (4) is used to update $\bar\tau_{ij}$. Therefore, we have a local solver given as

$$\bar\tau^{new}_{ij}=\frac{R^1_{ij}-\sqrt{\xi^2_{ij}+\eta^2_{ij}}-\beta_{ij}\xi_{ij}-\lambda_{ij}\eta_{ij}}{\gamma_{ij}},\tag{9}$$

where $\bar{\tau}_{ij}^{new}$ denotes the to-be-updated value on I_{ij}, and the right-hand side is evaluated with already-updated values of $\bar{\tau}$ on neighboring cells of I_{ij}, and ξ's and η's are given in (8). Furthermore, the local constants $\alpha_{e,ij}, \alpha_{w,ij}, \alpha_{n,ij}$ and $\alpha_{s,ij}$ are defined as in (3) with ξ's and η's being given in (8). With the local solver (9), our method for solving (1) numerically is summarized as the following:

Algorithm 1 (Second order fast sweeping method for eikonal equations).

1. Initialization: assign numerical boundary conditions at cell centers on and near Γ according to given boundary conditions. These points are fixed during iterations.
2. Preprocessing: solve (1) with the FSM as in A to obtain $(\tau^1, \tau_x^1, \tau_z^1)|_{ij}$ at cell centers (x_i, z_j) for $1 \leqslant i \leqslant I,\ 1 \leqslant j \leqslant J$.
3. Gauss–Seidel iterations with alternating orderings: sweep the domain with four natural orderings,
$$\begin{aligned} &(1)\ i = 1 : I;\ j = 1 : J; \quad (2)\ i = I : 1;\ j = 1 : J; \\ &(3)\ i = I : 1;\ j = J : 1; \quad (4)\ i = 1 : I;\ j = J : 1. \end{aligned} \tag{10}$$
For each cell I_{ij}, update $\bar{\tau}_{ij}$ according to (9).
4. Stopping criterion: given $\delta > 0$, check if $|\bar{\tau}^{new} - \bar{\tau}^{old}|_\infty < \delta$.

Remark 1. γ_{ij} in (9) is positive as proved in [27,57]. The DG method introduced in [27,57] has a more complicated local solver which is derived from (4) with a much subtler procedure to update the local unknowns $\bar{\tau}_{ij},\ \xi_{ij}$ and β_{ij} and enforce the causality condition.

2.2. Factored eikonal equations

The derivation of Algorithm 1 relies on the superconvergence phenomenon (8), which may not be applicable in many cases. For example, for the eikonal equation with a point-source condition,

$$|\nabla\tau(\mathbf{x})| = s(\mathbf{x}), \quad \text{for } \mathbf{x} \in \Omega \setminus \{\mathbf{x}_0\}; \quad \tau(\mathbf{x}_0) = 0, \tag{11}$$

the eikonal τ has upwind source singularities that make it difficult to obtain high accurate numerical solutions even with high order finite difference methods unless the source singularities are resolved [50]. In order to resolve the source singularities, we adopt the factorization approach introduced in [17,30,31,37,56]. The eikonal is decomposed into two factors,

$$\tau = u\tilde{\tau},\ \text{multiplicatively}, \tag{12}$$

or

$$\tau = u + \tilde{\tau},\ \text{additively}, \tag{13}$$

where $\tilde{\tau}$ is chosen to be an appropriate function that captures the source singularities, and u is the unknown correction to be determined. Since $\tilde{\tau}$ captures the source singularities, u is differentiable at the source. Hence u can be computed with high accuracy through the following factored eikonal equations,

$$|\nabla\tau| = \sqrt{\tilde{\tau}^2|\nabla u|^2 + 2\tilde{\tau}u\nabla\tilde{\tau}\cdot\nabla u + u^2|\nabla\tilde{\tau}|^2} = s, \quad \text{for (12)} \tag{14}$$

or

$$|\nabla\tau| = \sqrt{|\nabla u|^2 + 2\nabla\tilde{\tau}\cdot\nabla u + |\nabla\tilde{\tau}|^2} = s, \quad \text{for (13)}, \tag{15}$$

after substituting (12) and (13) into (11), respectively. In particular, we choose $\tilde{\tau}$ as the distance function,

$$\tilde{\tau}(\mathbf{x}) = s(\mathbf{x}_0)|\mathbf{x} - \mathbf{x}_0|, \tag{16}$$

which approximates τ in the following way [32],

$$\tilde{\tau}(\mathbf{x}) = \tau(\mathbf{x}) + O(|\mathbf{x} - \mathbf{x}_0|^2), \quad \text{for } \mathbf{x} \text{ near source point } \mathbf{x}_0. \tag{17}$$

Therefore, it is natural to extend the second order method introduced above to solve the factored eikonal equations (14) or (15) to compute u, and recover τ with second order accuracy.

The DG formulation for (14) is: find $u_h \in V_h^1$ such that

$$\begin{aligned} &\int_{I_{ij}} \sqrt{\tilde{\tau}^2|\nabla u_h|^2 + 2\tilde{\tau}u_h\nabla\tilde{\tau}\cdot\nabla u_h + u_h^2|\nabla\tilde{\tau}|^2}\, w_h(x,z)dxdz + \alpha_{e,ij}\int_{I_j}[u_h](x_{i+1/2},z)w_h(x_{i+1/2}^-,z)dz + \alpha_{w,ij}\int_{I_j}[u_h] \\ &\quad\times(x_{i-1/2},z)w_h(x_{i-1/2}^+,z)dz + \alpha_{n,ij}\int_{I_i}[u_h](x,z_{j+1/2})w_h(x,z_{j+1/2}^-)dx + \alpha_{s,ij}\int_{I_i}[u_h](x,z_{j-1/2})w_h(x,z_{j-1/2}^+)dx \\ &\quad= \int_{I_{ij}} s(x,z)w_h(x,z)dxdz, \quad \forall w_h \in V_h^1. \end{aligned} \tag{18}$$

And the DG formulation for (15) is: find $u_h \in V_h^1$ such that

$$\begin{aligned}&\int_{I_{ij}} \sqrt{|\nabla u_h|^2 + 2\nabla\tilde{\tau}\cdot\nabla u_h + |\nabla\tilde{\tau}|^2}\, w_h(x,z)dxdz + \alpha_{e,ij}\int_{I_j}[u_h](x_{i+1/2},z)w_h(x_{i+1/2}^-,z)dz + \alpha_{w,ij}\int_{I_j}[u_h]\\ &\times(x_{i-1/2},z)w_h(x_{i-1/2}^+,z)dz + \alpha_{n,ij}\int_{I_i}[u_h](x,z_{j+1/2})w_h(x,z_{j+1/2}^-)dx + \alpha_{s,ij}\int_{I_i}[u_h](x,z_{j-1/2})w_h(x,z_{j-1/2}^+)dx\\ &= \int_{I_{ij}} s(x,z)w_h(x,z)dxdz, \quad \forall w_h \in V_h^1. \end{aligned} \tag{19}$$

Similarly, the piecewise linear polynomial approximation $u|_{I_{ij}}$ on I_{ij} is given as $u|_{I_{ij}} = \bar{u}_{ij} + \mu_{ij}X_i + \nu_{ij}Z_j$, where the new unknowns are $\bar{u}_{ij}$, μ_{ij} and ν_{ij}. $\bar{u}_{ij}$, μ_{ij}/h and ν_{ij}/h are approximations of u, u_x and u_z at (x_i, z_j) respectively. The local constants $\alpha_{e,ij}, \alpha_{w,ij}, \alpha_{n,ij}$, and $\alpha_{s,ij}$ are defined similarly as in (3) with ξ_{ij} and η_{ij} being replaced according to the following,

$$\begin{aligned}&\text{for (12)}: \frac{\xi_{ij}}{h} = \tilde{\tau}_{ij}\frac{\mu_{ij}}{h} + \tilde{\tau}_{x,ij}\bar{u}_{ij}, \quad \frac{\eta_{ij}}{h} = \tilde{\tau}_{ij}\frac{\nu_{ij}}{h} + \tilde{\tau}_{z,ij}\bar{u}_{ij};\\ &\text{for (13)}: \frac{\xi_{ij}}{h} = \tilde{\tau}_{x,ij} + \frac{\mu_{ij}}{h}, \quad \frac{\eta_{ij}}{h} = \tilde{\tau}_{z,ij} + \frac{\nu_{ij}}{h}.\end{aligned} \tag{20}$$

By choosing $w_h = 1, X_i, Z_j$ and using the mid-point rule for numerical integrations, a nonlinear algebraic system is formed locally to determine $(\bar{u}_{ij}, \mu_{ij}, \nu_{ij})$ on I_{ij}. For (14),

$$\begin{aligned}\sqrt{h^2\bar{u}_{ij}^2(\tilde{\tau}_{x,ij}^2 + \tilde{\tau}_{z,ij}^2) + 2\tilde{\tau}_{ij}\bar{u}_{ij}h(\tilde{\tau}_{x,ij}\mu_{ij} + \tilde{\tau}_{z,ij}\nu_{ij}) + \tilde{\tau}_{ij}^2(\mu_{ij}^2 + \nu_{ij}^2)} + \gamma_{ij}\bar{u}_{ij} + \beta_{ij}\mu_{ij} + \lambda_{ij}\nu_{ij} &= R_{ij}^1,\\ 12\beta_{ij}\bar{u}_{ij} + g_{ij}\mu_{ij} &= R_{ij}^2,\\ 12\lambda_{ij}\bar{u}_{ij} + e_{ij}\nu_{ij} &= R_{ij}^3.\end{aligned} \tag{21}$$

For (15),

$$\begin{aligned}\sqrt{(h\tilde{\tau}_{x,ij} + \mu_{ij})^2 + (h\tilde{\tau}_{z,ij} + \nu_{ij})^2} + \gamma_{ij}\bar{u}_{ij} + \beta_{ij}\mu_{ij} + \lambda_{ij}\nu_{ij} &= R_{ij}^1,\\ 12\beta_{ij}\bar{u}_{ij} + g_{ij}\mu_{ij} &= R_{ij}^2,\\ 12\lambda_{ij}\bar{u}_{ij} + e_{ij}\nu_{ij} &= R_{ij}^3.\end{aligned} \tag{22}$$

$\gamma_{ij}, \beta_{ij}, \lambda_{ij}, g_{ij}, e_{ij}, R_{ij}^1, R_{ij}^2$ and R_{ij}^3 are defined similarly as in (5) and (6) by replacing $\bar{\tau}$, ξ and η with $\bar{u}$, μ and ν respectively.

Let us also assume that the FSM in A has already been applied to solve (14) and (15) to obtain a set of numerical solutions $\{(u^1, u_x^1, u_z^1)|_{ij}\}$ which approximate $(u, u_x, u_z)|_{ij}$ at cell centers for $1 \leqslant i \leqslant I$, $1 \leqslant j \leqslant J$. The superconvergence phenomenon is also observed, i.e.,

$$|u^1 - u|_\infty = O(h), \quad |u_x^1 - u_x|_\infty = O(h), \quad \text{and} \quad |u_z^1 - u_z|_\infty = O(h). \tag{23}$$

With this observation, we proceed similarly as above. We fix μ_{ij} and ν_{ij} by

$$\mu_{ij} = u_{x,ij}^1 h, \quad \text{and} \quad \nu_{ij} = u_{z,ij}^1 h. \tag{24}$$

Therefore, we have a local solver for updating $\bar{u}_{ij}$. For (14),

$$\bar{u}_{ij}^{new} = \frac{R_{ij}^1 - \sqrt{h^2\bar{u}_{ij}^2(\tilde{\tau}_{x,ij}^2 + \tilde{\tau}_{z,ij}^2) + 2\tilde{\tau}_{ij}\bar{u}_{ij}h(\tilde{\tau}_{x,ij}\mu_{ij} + \tilde{\tau}_{z,ij}\nu_{ij}) + \tilde{\tau}_{ij}^2(\mu_{ij}^2 + \nu_{ij}^2)} - \beta_{ij}\mu_{ij} - \lambda_{ij}\nu_{ij}}{\gamma_{ij}}. \tag{25}$$

For (15),

$$\bar{u}_{ij}^{new} = \frac{R_{ij}^1 - \sqrt{(h\tilde{\tau}_{x,ij} + \mu_{ij})^2 + (h\tilde{\tau}_{z,ij} + \nu_{ij})^2} - \beta_{ij}\mu_{ij} - \lambda_{ij}\nu_{ij}}{\gamma_{ij}}. \tag{26}$$

$\bar{u}_{ij}^{new}$ is the to-be-updated value on I_{ij}. The right-hand sides of (25) and (26) are evaluated with already-updated values of $\bar{u}$ on neighboring cells of I_{ij}, and μ's and ν's are given as in (24). The local constants $\alpha_{e,ij}, \alpha_{w,ij}, \alpha_{n,ij}$, and $\alpha_{s,ij}$ are also obtained with μ_{ij} and ν_{ij} being fixed as in (24). With the local solver (25) and (26), we summarize the method for the factored eikonal equations.

Algorithm 2 (Second order fast sweeping method for factored eikonal equations).

1. Initialization: assign numerical boundary conditions at cell centers on and near Γ according to given boundary conditions. These points are fixed during iterations.
2. Preprocessing: solve (14) or (15) with the FSM as in A to obtain $(u^1, u_x^1, u_z^1)|_{ij}$ at cell centers (x_i, z_j) for $1 \leqslant i \leqslant I$, $1 \leqslant j \leqslant J$.

3. Gauss–Seidel iterations with alternating orderings: sweep the domain with four natural orderings as in (10). For each cell I_{ij}, update $\bar{u}_{ij}$ according to (25) or (26).
4. Stopping criterion: given $\delta > 0$, check if $|\bar{u}^{new} - \bar{u}^{old}|_\infty < \delta$.

Remark 2. Algorithm 1 and Algorithm 2 are second order in both l_1 norm and l_∞ norm when the solution is smooth. When there are shocks, both algorithms are second order in l_1 norm but reduces to first order in l_∞ norm since they are reduced to first order on the shocks. However, away from the shocks, they are still second order in l_∞ norm if the solution is smooth.

Remark 3. In Algorithm 1 or Algorithm 2, due to the upwind causality enforcement with (3) and (8) (or (24)), for cells on the computational boundary, only the information from the interior of the computational domain is needed to update the cell average according to (9) or (25), or (26).

We present several numerical examples with single- and multiple-source conditions in the following section to demonstrate the method.

3. Numerical examples

For our numerical implementations, we choose $\delta = 10^{-10}$. Numerical errors at the cell centers in both l_1 norm and l_∞ norm are recorded. And we denote one iteration as one sweep over all the cells.

3.1. Numerical examples for eikonal equations with Algorithm 1

We present several examples for Algorithm 1 for the eikonal equation (1).

Example 1. Constant velocity. The setup is the following,

- slowness field: $s \equiv 1$;
- domain: $[0,1]^2$;
- single source: $\mathbf{x}_0 = (0.5, 0.5)$.

The eikonal has source singularities. In our numerical tests, we first assign exact solutions on a disc centered at the source with radius 0.1, then carry out the calculation with Algorithm 1 outside the disc. The eikonal is smooth outside the disc. Tables 1 and 2 show the results. From Table 1, we observe the superconvergence phenomenon for the fast sweeping method. From Table 2, we observe second order convergence in both l_∞ norm and l_1 norm for Algorithm 1. And the number of iterations for Algorithm 1 does not increase as the mesh is refined.

Example 2. A velocity of constant gradient. The setup is the following,

- slowness field s satisfies $\frac{1}{s} = \frac{1}{s_0} + \mathbf{g} \cdot (\mathbf{x} - \mathbf{x}_0)$;
- domain: $[0, 0.5]^2$;
- the constant gradient $\mathbf{g} = (0, -1)$;
- $s_0 = 2$;

Table 1
Example 1: first order fast sweeping method for eikonal equation (11) with single-source condition; constant velocity.

Fast sweeping method for eikonal equation (11)				
Mesh	101×101	201×201	401×401	801×801
$\lvert\tau^1 - \tau\rvert_\infty$	6.85E−3	3.44E−3	1.72E−3	8.64E−4
Order of convergence	–	0.994	1.000	0.993
$\lvert\tau^1 - \tau\rvert_1$	3.13E−4	1.55E−4	7.67E−5	3.82E−5
Order of convergence	–	1.014	1.015	1.006
$\lvert\tau_x^1 - \tau_x\rvert_\infty$	5.25E−2	2.56E−2	1.27E−2	6.29E−3
Order of convergence	–	1.036	1.011	1.0137
$\lvert\tau_z^1 - \tau_z\rvert_\infty$	5.25E−2	2.56E−2	1.27E−2	6.29E−3
Order of convergence	–	1.036	1.011	1.0137
# Iter	5	5	5	5

Table 2
Example 1: Algorithm 1 for eikonal equation (11) with single-source condition; constant velocity.

Algorithm 1 for eikonal equation (11)				
Mesh	101×101	201×201	401×401	801×801
$\lvert\tau^1 - \tau\rvert_\infty$	1.84E−4	4.67E−5	1.13E−5	2.94E−6
Order of convergence	–	1.978	2.047	1.9424
$\lvert\tau^1 - \tau\rvert_1$	6.96E−5	1.67E−5	4.02E−6	1.01E−6
Order of convergence	–	2.059	2.055	1.993
# Iter	20	20	20	17

Table 3
Example 2: first order fast sweeping method for eikonal equation (11) with single-source condition; a velocity of constant gradient.

Fast sweeping method for eikonal equation (11): boundary condition 1				
Mesh	101×101	201×201	401×401	801×801
$\lvert\tau^1 - \tau\rvert_\infty$	8.66E−3	4.33E−3	2.16E−3	1.08E−3
Order of convergence	–	1.000	1.003	1.000
$\lvert\tau^1 - \tau\rvert_1$	5.10E−4	2.49E−4	1.23E−4	6.10E−5
Order of convergence	–	1.034	1.018	1.012
$\lvert\tau_x^1 - \tau_x\rvert_\infty$	5.75E−2	2.82E−2	1.40E−2	6.97E−3
Order of convergence	–	1.028	1.010	1.006
$\lvert\tau_z^1 - \tau_z\rvert_\infty$	5.09E−2	2.57E−2	1.33E−2	6.79E−3
Order of convergence	–	0.986	0.950	0.932
# Iter	8	8	8	8

Table 4
Example 2: Algorithm 1 for eikonal equation (11) with single-source condition; a velocity of constant gradient.

Algorithm 1 for eikonal equation (11): boundary condition 1				
Mesh	101×101	201×201	401×401	801×801
$\lvert\tau^1 - \tau\rvert_\infty$	1.17E−4	2.87E−5	7.12E−6	1.77E−6
Order of convergence	–	2.027	2.011	2.008
$\lvert\tau^1 - \tau\rvert_1$	8.19E−6	2.00E−6	4.98E−7	1.24E−7
Order of convergence	–	2.034	2.006	2.006
# Iter	12	15	15	15

- boundary condition 1: single source point: $(0.25, 0.25)$;
- boundary condition 2: two source points: $(0.12, 0.12)$ and $(0.37, 0.37)$;
- boundary condition 3: four source points: $(0.12, 0.12)$, $(0.37, 0.12)$, $(0.37, 0.37)$, and $(0.12, 0.37)$;
- the exact solution is known analytically [17].

The eikonal has source singularities. In our numerical tests, we first assign exact solutions on a disc centered at the sources with radius 0.1, then carry out the calculation with Algorithm 1 outside the discs. For single-source conditions, the solution is smooth outside the disc. For boundary condition 1 with a single source, the results are recorded in Tables 3 and 4. From Table 3, we observe the superconvergence phenomenon. From Table 4, we observe second order convergence in both l_∞ norm and l_1 norm for Algorithm 1. For boundary condition 2 with two sources, the results are recorded in Table 5. For boundary condition 3 with four sources, the results are recorded in Table 6. From Tables 5 and 6, we observe second order convergence in l_1 norm, but only first order convergence in l_∞ norm. This is because the eikonal has shocks and the second order Algorithm 1 reduces to be first order accurate on the shocks. However, the error on the shocks is localized, therefore it is still second order accurate away from the shocks. Fig. 1 shows the contour plots of l_∞ error. It is clear that the error on the shocks is localized. Moreover, the number of iterations for Algorithm 1 does not increase as the mesh is refined.

3.2. Numerical examples for factored eikonal equations with Algorithm 2

We present several examples for Algorithm 2 for the factored eikonal equations.

Example 3. A velocity of constant gradient. The setup is the same as in Example 2. With the factorization approach, there is no need to assign the exact solution on a disc centered at sources with a fixed size.

Table 5
Example 2: Algorithm 1 for eikonal equation (11) with two-source condition; a velocity of constant gradient.

Algorithm 1 for eikonal equation (11): boundary condition 2				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	2.49E−3	1.56E−3	9.63E−4	4.64E−4
Order of convergence	–	0.675	0.696	1.053
$\|\tau^1 - \tau\|_1$	1.06E−5	2.70E−6	6.87E−7	1.77E−7
Order of convergence	–	1.973	1.975	1.957
# Iter	19	19	15	15

Table 6
Example 2: Algorithm 1 for eikonal equation (11) with four-source condition; a velocity of constant gradient.

Algorithm 1 for eikonal equation (11): boundary condition 3				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	1.64E−3	9.66E−4	5.31E−4	3.03E−4
Order of convergence	–	0.764	0.863	0.809
$\|\tau^1 - \tau\|_1$	4.87E−6	1.15E−6	2.68E−7	6.78E−8
Order of convergence	–	2.082	2.101	1.983
# Iter	10	11	11	9

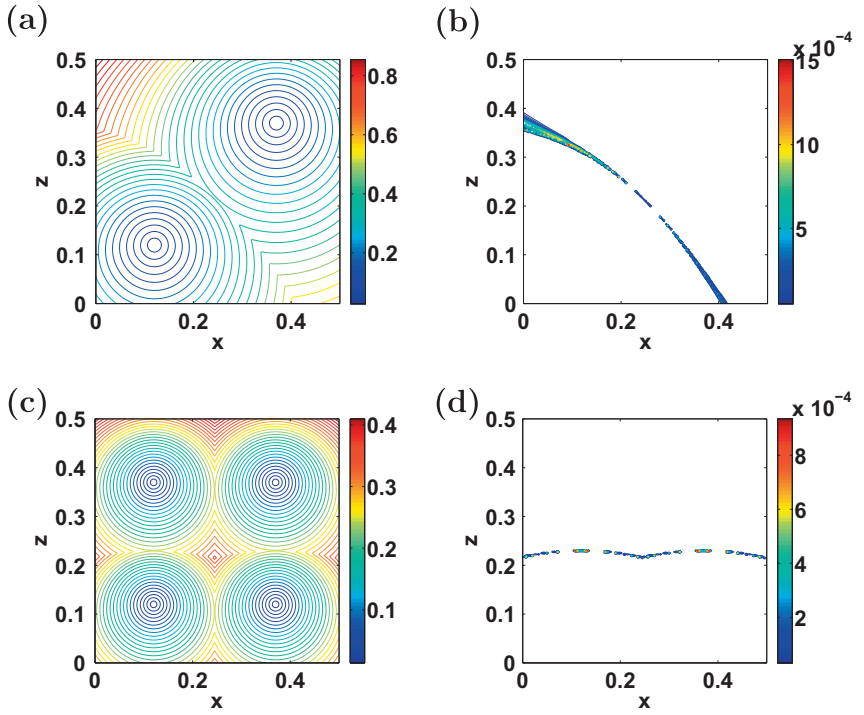

Fig. 1. Example 2. Numerical solutions by Algorithm 1. (a) contour plots of the numerical solution with two-source conditions; (b) contour plots of the l_∞ error with two-source conditions; (c) contour plots of the numerical solution with four-source conditions; and (d) contour plots of the l_∞ error with four-source conditions; mesh: 201×201.

For boundary condition 1 with a single source, the results are recorded in Tables 7–10. From Tables 7 and 8, we observe the superconvergence phenomenon. From Tables 9 and 10, we observe second order convergence in both l_∞ norm and l_1 norm for Algorithm 2. For boundary condition 2 with two sources, the results are recorded in Tables 11 and 12. For boundary condition 3 with four sources, the results are recorded in Tables 13 and 14. From Tables 11–14, we observe second order

Table 7
Example 3: first order fast sweeping method for factored eikonal equation (14) with single-source condition; a velocity of constant gradient.

Fast sweeping method for factored eikonal equation (14): boundary condition 1				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	1.57E−3	7.84E−4	3.92E−4	1.96E−4
Order of convergence	–	1.002	1.000	1.000
$\|\tau^1 - \tau\|_1$	5.83E−5	2.85E−5	1.41E−5	7.00E−6
Order of convergence	–	1.033	1.015	1.010
$\|u_x^1 - u_x\|_\infty$	2.83E−3	1.40E−3	6.98E−4	3.48E−4
Order of convergence	–	1.015	1.004	1.004
$\|u_z^1 - u_z\|_\infty$	9.38E−3	4.80E−3	2.44E−3	1.23E−3
Order of convergence	–	0.967	0.976	0.988
$\|\tau_x^1 - \tau_x\|_\infty$	4.34E−3	2.16E−3	1.08E−3	5.41E−4
Order of convergence	–	1.007	1.000	0.997
$\|\tau_z^1 - \tau_z\|_\infty$	3.80E−3	1.93E−3	9.71E−4	4.86E−4
Order of convergence	–	0.977	0.991	1.004
# Iter	18	18	18	18

Table 8
Example 3: first order fast sweeping method for factored eikonal equation (15) with single-source condition; a velocity of constant gradient.

Fast sweeping method for factored eikonal equation (15): boundary condition 1				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	5.27E−3	2.65E−3	1.33E−3	6.68E−4
Order of convergence	–	0.992	0.995	0.994
$\|\tau^1 - \tau\|_1$	3.62E−4	1.81E−4	9.06E−5	4.53E−5
Order of convergence	–	1.000	0.998	1.000
$\|u_x^1 - u_x\|_\infty$	1.24E−2	6.24E−3	3.13E−3	1.56E−3
Order of convergence	–	0.991	0.995	1.005
$\|u_z^1 - u_z\|_\infty$	7.16E−3	3.56E−3	1.77E−3	8.86E−4
Order of convergence	–	1.008	1.008	1.048
$\|\tau_x^1 - \tau_x\|_\infty$	1.24E−2	6.24E−3	3.13E−3	1.56E−3
Order of convergence	–	0.991	0.995	1.005
$\|\tau_z^1 - \tau_z\|_\infty$	7.16E−3	3.56E−3	1.77E−3	8.86E−4
Order of convergence	–	1.008	1.008	1.048
# Iter	16	16	16	16

Table 9
Example 3: Algorithm 2 for factored eikonal equation (14) with single-source condition; a velocity of constant gradient.

Algorithm 2 for (14): boundary condition 1				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	1.57E−5	3.95E−6	9.91E−7	2.48E−7
Order of convergence	–	1.991	1.995	1.999
$\|\tau^1 - \tau\|_1$	5.10E−7	1.24E−7	3.11E−8	7.85E−9
Order of convergence	–	2.040	1.995	1.986
# Iter	41	49	57	61

convergence in l_1 norm, but only first order convergence in l_∞ norm. This is because the eikonal has shocks and the second order Algorithm 2 reduces to be first order accurate on the shocks. However, the error on the shocks is localized, therefore it is still second order accurate away from the shocks. Figs. 2 and 3 show the contour plots of l_∞ error. It is clear that the error on the shocks is localized. And for the decomposition with multiplicative factors (12), the number of iterations increases slightly as the mesh is refined, while it does not increase for the decomposition with additive factors (13).

For the cases with multiple sources, the choice of $\bar{\tau}$ is explained in B.

Example 4. Sinusoidal model. The setup is the following,

Table 10
Example 3: Algorithm 2 for factored eikonal equation (15) with single-source condition; a velocity of constant gradient.

Algorithm 2 for (15): boundary condition 1				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	1.22E−4	3.10E−5	7.74E−6	1.96E−6
Order of convergence	–	1.977	2.002	1.981
$\|\tau^1 - \tau\|_1$	8.40E−6	1.99E−6	4.79E−7	1.22E−7
Order of convergence	–	2.078	2.055	1.973
# Iter	17	15	13	13

Table 11
Example 3: Algorithm 2 for factored eikonal equation (14) with two-source condition; a velocity of constant gradient.

Algorithm 2 for (14): boundary condition 2				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	2.53E−3	1.57E−3	9.73E−4	4.67E−4
Order of convergence	–	0.688	0.690	1.059
$\|\tau^1 - \tau\|_1$	1.10E−5	2.83E−6	7.28E−7	1.88E−7
Order of convergence	–	1.959	1.959	1.953
# Iter	41	49	59	64

Table 12
Example 3: Algorithm 2 for factored eikonal equation (15) with two-source condition; a velocity of constant gradient.

Algorithm 2 for (15): boundary condition 2				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	2.42E−3	1.56E−3	9.55E−4	4.60E−4
Order of convergence	–	0.639	0.708	1.054
$\|\tau^1 - \tau\|_1$	1.22E−5	2.80E−6	6.70E−7	1.72E−7
Order of convergence	–	2.123	2.063	1.962
# Iter	19	19	19	19

Table 13
Example 3: Algorithm 2 for factored eikonal equation (14) with four-source condition; a velocity of constant gradient.

Algorithm 2 for (14): boundary condition 3				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	2.48E−3	1.24E−3	5.75E−4	3.20E−4
Order of convergence	–	1.000	1.109	0.845
$\|\tau^1 - \tau\|_1$	8.16E−6	1.92E−6	4.44E−7	1.10E−7
Order of convergence	–	2.087	2.112	2.013
# Iter	37	52	64	76

Table 14
Example 3: Algorithm 2 for factored eikonal equation (15) with four-source condition; a velocity of constant gradient.

Algorithm 2 for (15): boundary condition 3				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	1.23E−3	8.31E−4	5.80E−4	2.87E−4
Order of convergence	–	0.566	0.519	1.015
$\|\tau^1 - \tau\|_1$	1.57E−5	3.51E−6	9.58E−7	2.38E−7
Order of convergence	–	2.161	1.873	2.009
# Iter	23	23	23	21

- slowness function s satisfies $\frac{1}{s} = 1 + 0.2\sin(0.5\pi z)\sin(3\pi(x + 0.05))$;
- domain: $[0,1]^2$;
- single source: $\mathbf{x}_0 = (0.5, 0.5)$;
- the numerical solution on a mesh 3201×3201 is obtained as the "exact" solution.

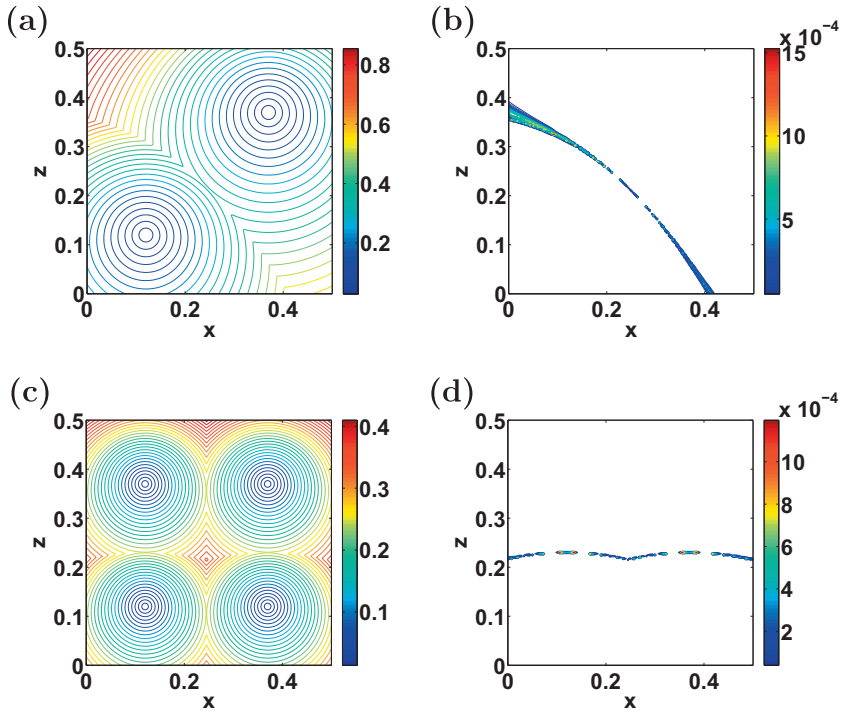

Fig. 2. Example 3. Numerical solutions by Algorithm 2 for the factored eikonal equation (14). (a) contour plots of the numerical solution with two-source conditions; (b) contour plots of the l_∞ error with two-source conditions; (c) contour plots of the numerical solution with four-source conditions; and (d) contour plots of the l_∞ error with four-source conditions; mesh: 201×201.

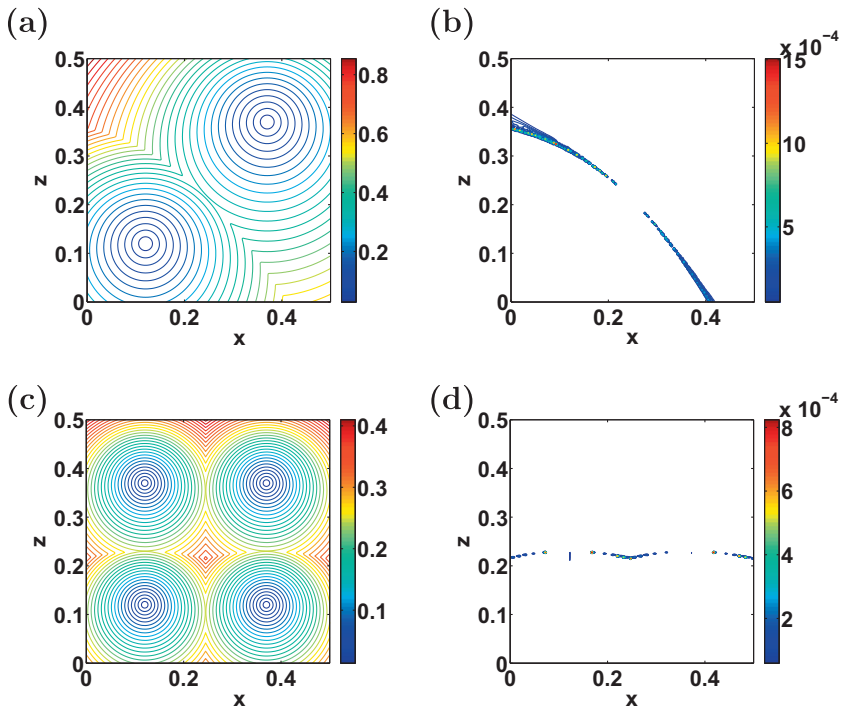

Fig. 3. Example 3. Numerical solutions by Algorithm 2 for the factored eikonal equation (15). (a) contour plots of the numerical solution with two-source conditions; (b) contour plots of the l_∞ error with two-source conditions; (c) contour plots of the numerical solution with four-source conditions; and (d) contour plots of the l_∞ error with four-source conditions; mesh: 201×201.

Table 15
Example 4: Algorithm 2 for factored eikonal equation (14) with single-source condition; Sinusoidal model.

Algorithm 2 for factored eikonal equation (14)				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	7.14E−5	1.70E−5	4.09E−6	9.67E−7
Order of convergence	–	2.070	2.055	2.081
$\|\tau^1 - \tau\|_1$	2.19E−5	5.39E−6	1.33E−6	3.15E−7
Order of convergence	–	2.023	2.019	2.078
# Iter	46	54	60	64

Table 16
Example 4: Algorithm 2 for factored eikonal equation (15) with single-source condition; Sinusoidal model.

Algorithm 2 for factored eikonal equation (15)				
Mesh	101×101	201×201	401×401	801×801
$\|\tau^1 - \tau\|_\infty$	1.10E−4	2.84E−5	7.12E−6	1.74E−6
Order of convergence	–	1.954	1.996	2.033
$\|\tau^1 - \tau\|_1$	3.71E−5	8.60E−6	2.01E−6	4.71E−7
Order of convergence	–	2.109	2.097	2.093
# Iter	18	17	17	19

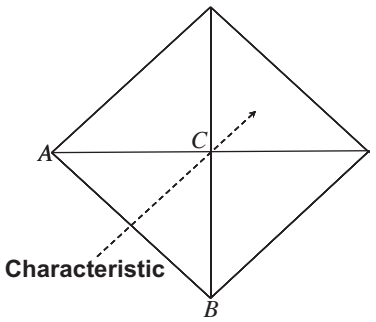

Fig. 4. Local mesh of point C.

Tables 15 and 16 show the results for Algorithm 2. We observe second order convergence in both l_1 norm and l_∞ norm. Moreover, the number of iterations for the decomposition (12) with multiplicative factors increases slightly as the mesh is refined, while it does not increase for the decomposition (13) with additive factors.

4. Conclusion

We introduce a simple and efficient method for solving the eikonal equation and the factored eikonal equations with uniform second order accuracy. The method has a simple local solver which is the combination of the piecewise linear discontinuous Galerkin formulation with a superconvergence phenomenon that the numerical solution of monotone upwind schemes for the eikonal equation is first order accurate on both its value and gradient. The information of the gradient is obtained first with the first order fast sweeping method, then fixed in the local discontinuous Galerkin formulation, which results in a simple local solver for updating the cell average. Numerical examples verify that the method is uniformly second order accurate. Future projects include the development of high order methods for static Hamilton–Jacobi equations with local discontinuous Galerkin formulations. Lax–Friedrichs type numerical Hamiltonians will be chosen instead of Godunov type numerical Hamiltonians [55].

Acknowledgments

The author thanks the anonymous reviewers for helpful comments on this work. The author thanks the Department of Mathematics at Iowa State University for the support of this work.

Appendix A. Fast sweeping method

We summarize the fast sweeping scheme for (1), (14) and (15) in 2-D on a rectangular mesh with grid size h [17,31,42,43,59]. Without loss of generality, let us consider Hamilton–Jacobi equations in the following generic form in 2-D,

$$F(x,z,u,u_x,u_z) = f(x,z), \tag{A.1}$$

where F is convex in the gradient variable.

Taking a local mesh of point $C = (x_C, z_C)$ as shown in Fig. 4, we consider discretizations on the triangle with neighbors $A = (x_A, z_A)$ and $B = (x_B, z_B)$,

$$\nabla u(C) \approx \left(\frac{u(C)-u(A)}{x_C - x_A}, \frac{u(C)-u(B)}{z_C - z_B}\right), \tag{A.2}$$

which defines the numerical Hamiltonian $\hat{F}$ as

$$\hat{F}(C, u(C), u(A), u(B) \equiv F\left(C, u(C), \frac{u(C)-u(A)}{x_C - x_A}, \frac{u(C)-u(B)}{z_C - z_B}\right) - f(C) = 0. \tag{A.3}$$

Given $u(A)$ and $u(B)$, we wish to solve (A.3) for $u(C)$. There are only three scenarios due to the convexity of F:

1. Scenario 1: there is no solution for $u(C)$ from (A.3);
2. Scenario 2: there is one solution for $u(C)$ from (A.3);
3. Scenario 3: there are two solutions for $u(C)$ from (A.3).

In Scenario 1, we enforce the characteristic equation for Hamilton–Jacobi equations along the edges $\mathbf{r}_{AC}$ and $\mathbf{r}_{BC}$ to get possible values of $u(C)$, where $\mathbf{r}_{AC}$ is the vector from A to C, and $\mathbf{r}_{BC}$ is the vector from B to C; see [17,31,42,43,59]. In Scenario 2 or 3, we need to further check whether a candidate value for $u(C)$ that is consistent with Eq. (A.1) satisfies the following *causality condition*: the characteristic passing through C is in between the two vectors $\mathbf{r}_{AC}$ and $\mathbf{r}_{BC}$. This is a crucial condition for the monotonicity of the scheme. We call a value $u(C)$ a possible candidate if it satisfies both the consistency and causality conditions. We can use the same procedure to find possible candidates for $u(C)$ from other triangles with C as one of their vertices. If there are multiple candidates, we choose the minimum among them. We summarize the method.

Algorithm 3 (First-order fast sweeping method (FSM) [17,31,42,43,59]).

1. Initialization: for vertices on or near the boundary, their values are set according to the given boundary condition. All other vertices are assigned a large value, for instance, infinity, initially.
2. Gauss–Seidel iterations with four alternating orderings (sweepings):
   - Update during each iteration: at a vertex C, the updated value $u^{new}(C)$ at C is

$$u^{new}(C) = \min\{u^{old}(C), u^{comp}(C)\}, \tag{A.4}$$

where $u^{old}(C)$ is the current value at C and $u^{comp}(C)$ is the value at C computed from the current given neighboring values according to (A.3) and the procedure detailed as above.
   - Orderings: four alternating orderings as in (10) are needed.
   - Stopping criterion: given $\delta > 0$, check if $|u^{new} - u^{old}|_\infty < \delta$.

Appendix B. $\tilde{\tau}$ for multiple sources

We explain how to choose the preknown part $\tilde{\tau}$ in the decomposition (12) or (13). For a single-source condition $\Gamma = \{\mathbf{x}_0\}$,

$$\tilde{\tau}(\mathbf{x}) = s(\mathbf{x}_0)|\mathbf{x} - \mathbf{x}_0|, \text{ for both decompositions.}$$

For a multiple-source condition $\Gamma = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$,

$$\tilde{\tau}(\mathbf{x}) = \Pi_{i=1}^n s(\mathbf{x}_i)|\mathbf{x} - \mathbf{x}_i|, \quad \text{for (12)}; \quad \tilde{\tau}(\mathbf{x}) = \sum_{i=1}^n s(\mathbf{x}_i)|\mathbf{x} - \mathbf{x}_i|, \quad \text{for (13)}.$$

References

[1] S. Bak, J.R. McLaughlin, D. Renzi, Some improvements for the fast sweeping method, SIAM J. Sci. Comput. 32 (2010) 2853–2874.
[2] J.D. Benamou, S. Luo, H.K. Zhao, A compact upwind second order scheme for the eikonal equation, J. Comput. Math. 28 (2010) 489–516.
[3] M. Boué, P. Dupuis, Markov chain approximations for deterministic control problems with affine dynamics and quadratic cost in the control, SIAM J. Numer. Anal. 36 (3) (1999) 667–695.
[4] Y. Cheng, C.W. Shu, A discontinuous Galerkin finite element method for directly solving the Hamilton–Jacobi equations, J. Comput. Phys. 223 (2007) 398–415.
[5] B. Cockburn, S. Hou, C.W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case, Math. Comput. 54 (1990) 545–581.
[6] B. Cockburn, S.Y. Lin, C.W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems, J. Comput. Phys. 84 (1989) 90–113.
[7] B. Cockburn, C.W. Shu, TVB Runge–Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework, Math. Comput. 52 (1989) 411–435.

[8] B. Cockburn, C.W. Shu, The Runge–Kutta local projection P1-discontinuous Galerkin finite element method for scalar conservation laws, Math. Modell. Numer. Anal. 25 (1991) 337–361.
[9] B. Cockburn, C.W. Shu, The local discontinuous Galerkin method for time-dependent convection–diffusion system, SIAM J. Numer. Anal. 35 (1998) 2440–2463.
[10] B. Cockburn, C.W. Shu, The Runge–Kutta discontinuous Galerkin finite element method for conservation laws V: multidimensional systems, J. Comput. Phys. 141 (1998) 199–224.
[11] M.G. Crandall, P.L. Lions, Viscosity solutions of Hamilton–Jacobi equations, Trans. Am. Math. Soc. 277 (1983) 1–42.
[12] P. Danielsson, Euclidean distance mapping, Comput. Graphics Image Process. 14 (1980) 227–248.
[13] E.W. Dijkstra, A note on two problems in connection with graphs, Numer. Math. 1 (1959) 269–271.
[14] B. Engquist, O. Runborg, Computational high frequency wave propagation, Acta Numer. 12 (2003) 181–266.
[15] M. Falcone, R. Ferretti, Discrete time high-order schemes for viscosity solutions of Hamilton–Jacobi–Bellman equations, Numer. Math. 67 (1994) 315–344.
[16] M. Falcone, R. Ferretti, Semi-Lagrangian schemes for Hamilton–Jacobi equations, discrete representation formulae and Godunov methods, J. Comput. Phys. 175 (2002) 559–575.
[17] S. Fomel, S. Luo, H.K. Zhao, Fast sweeping method for the factored eikonal equation, J. Comput. Phys. 228 (2009) 6440–6455.
[18] P.A. Gremaud, C.M. Kuster, Computational study of fast methods for the eikonal equations, SIAM J. Sci. Comput. 27 (2006) 1803–1816.
[19] J. Helmsen, E. Puckett, P. Colella, M. Dorr, Two new methods for simulating photolithography development in 3d, Proc. SPIE 2726 (1996) 253–261.
[20] C. Hu, C. Shu, A discontinuous Galerkin finite element method for Hamilton–Jacobi equations, SIAM J. Sci. Comput. 21 (2000) 666–690.
[21] G.S. Jiang, D. Peng, Weighted ENO schemes for Hamilton–Jacobi equations, SIAM J. Sci. Comput. 21 (2000) 2126–2143.
[22] C. Kao, S. Osher, J. Qian, Legendre transform based fast sweeping methods for static Hamilton–Jacobi equations on triangulated meshes, J. Comput. Phys. 227 (2008) 10209–10225.
[23] C.Y. Kao, S. Osher, J. Qian, Lax–Friedrichs sweeping schemes for static Hamilton–Jacobi equations, J. Comput. Phys. 196 (2004) 367–391.
[24] C.Y. Kao, S. Osher, Y.H. Tsai, Fast sweeping method for static Hamilton–Jacobi equations, SIAM J. Numer. Anal. 42 (2005) 2612–2632.
[25] J.B. Keller, R.M. Lewis, Asymptotic methods for partial differential equations: the reduced wave equation and Maxwell's equations, Surv. Appl. Math. 1 (1995) 1–82.
[26] S. Kim, R. Cook, 3-D traveltime computation using second-order ENO scheme, Geophysics 64 (1999) 1867–1876.
[27] F. Li, C.W. Shu, Y.T. Zhang, H.K. Zhao, Second order discontinuous fast sweeping method for eikonal equations, J. Comput. Phys. 227 (2008) 8191–8208.
[28] X.D. Liu, S.J. Osher, T. Chan, Weighted essentially non oscillatory schemes, J. Comput. Phys. 115 (1994) 200–212.
[29] S. Luo, Numerical methods for static Hamilton–Jacobi equations, Ph.D Thesis, University of California, Irvine, 2009.
[30] S. Luo, J. Qian, Factored singularities and high-order Lax–Friedrichs sweeping schemes for point-source traveltimes and amplitudes, J. Comput. Phys. 230 (2011) 4742–4755.
[31] S. Luo, J. Qian, Fast sweeping methods for factored anisotropic eikonal equations: multiplicative and additive factors, J. Sci. Comput. 52 (2012) 360–382.
[32] S. Luo, J. Qian, R. Burridge, High-Order Factorization Based High-Order Hybrid Fast Sweeping Methods for Point-Source Eikonal Equations, submitted for publication.
[33] S. Luo, J. Qian, H.K. Zhao, Higher-order schemes for 3-D first-arrival traveltimes and amplitudes, Geophysics 77 (2012) 1–10.
[34] S. Osher, A level set formulation for the solution of the Dirichlet problem for Hamilton–Jacobi equations, SIAM J. Math. Anal. 24 (1993) 1145–1152.
[35] S. Osher, J.A. Sethian, Fronts propagating with curvature dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[36] S. Osher, C.W. Shu, High-order essentially nonoscillatory schemes for Hamilton–Jacobi equations, SIAM J. Numer. Anal. 28 (1991) 907–922.
[37] A. Pica, Fast and accurate finite-difference solutions of the 3D eikonal equation parametrized in celerity, in: 67th Annual International Meeting Society of Exploration Geophysicists, 1997, pp. 1774–1777.
[38] J. Qian, W.W. Symes, Paraxial eikonal solvers for anisotropic quasi-P traveltimes, J. Comput. Phys. 173 (2001) 1–23.
[39] J. Qian, W.W. Symes, Adaptive finite difference method for traveltime and amplitude, Geophysics 67 (2002) 167–176.
[40] J. Qian, W.W. Symes, Finite-difference quasi-P traveltimes for anisotropic media, Geophysics 67 (2002) 147–155.
[41] J. Qian, W.W. Symes, Paraxial geometrical optics for quasi-P waves: theories and numerical methods, Wave Motion 35 (2002) 205–221.
[42] J. Qian, Y.T. Zhang, H.K. Zhao, A fast sweeping methods for static convex Hamilton–Jacobi equations, J. Sci. Comput. 31 (1/2) (2007) 237–271.
[43] 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.
[44] F. Qin, Y. Luo, K.B. Olsen, W. Cai, G.T. Schuster, Finite difference solution of the eikonal equation along expanding wavefronts, Geophysics 57 (1992) 478–487.
[45] E. Rouy, A. Tourin, A viscosity solutions approach to shape-from-shading, SIAM J. Numer. Anal. 29 (1992) 867–884.
[46] W.A.J. Schneider, K. Ranzinger, A. Balch, C. Kruse, A dynamic programming approach to first arrival traveltime computation in media with arbitrarily distributed velocities, Geophysics 57 (1992) 39–50.
[47] J.A. Sethian, A fast marching level set method for monotonically advancing fronts, in: Proceedings of the National Academy of Sciences, vol. 93, 1996.
[48] J.A. Sethian, A. Vladimirsky, Ordered upwind methods for static Hamilton–Jacobi equations, Proc. Natl. Acad. Sci. 98 (2001) 11069–11074.
[49] J.A. Sethian, A. Vladimirsky, Ordered upwind methods for static Hamilton–Jacobi equations: theory and algorithms, SIAM J. Numer. Anal. 41 (2003) 325–363.
[50] W.W. Symes, J. Qian, A slowness matching Eulerian method for multivalued solutions of eikonal equations, J. Sci. Comput. 19 (2003) 501–526.
[51] J. van Trier, W.W. Symes, Upwind finite-difference calculation of traveltimes, Geophysics 56 (1991) 812–821.
[52] Y.H. Tsai, L.T. Cheng, S. Osher, H.K. Zhao, Fast sweeping algorithms for a class of Hamilton–Jacobi equations, SIAM J. Numer. Anal. 41 (2003) 673–694.
[53] Y.R. Tsai, Rapid and accurate computation of the distance function using grids, J. Comput. Phys. 178 (2002) 175–195.
[54] J.N. Tsitsiklis, Efficient algorithms for globally optimal trajectories, IEEE Trans. Autom. Control 40 (1995) 1528–1538.
[55] J. Yan, S. Osher, A local discontinuous Galerkin method for directly solving Hamilton–Jacobi equations, J. Comput. Phys. 230 (2011) 232–244.
[56] L. Zhang, J.W. Rector, G.M. Hoversten, Eikonal solver in the celerity domain, Geophys. J. Int. 162 (2005) 1–8.
[57] Y.T. Zhang, S. Chen, F. Li, H.K. Zhao, C.W. Shu, Uniformly accurate discontinuous Galerkin fast sweeping methods for eikonal equations, SIAM J. Sci. Comput. 33 (2011) 1873–1896.
[58] 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.
[59] H.K. Zhao, A fast sweeping method for eikonal equations, Math. Comput. 74 (2005) 603–627.
[60] H.K. Zhao, Parallel implementations of the fast sweeping method, J. Comput. Math. 25 (2007) 421–429.