HIGH-ORDER FACTORIZATION BASED HIGH-ORDER HYBRID FAST SWEEPING METHODS FOR POINT-SOURCE EIKONAL EQUATIONS

SONGTING LUO *, JIANLIANG QIAN †, AND ROBERT BURRIDGE ‡

Abstract. The solution for the eikonal equation with a point-source condition has an upwind singularity at the source point as the eikonal solution behaves like a distance function at and near the source. As such, the eikonal function is not differentiable at the source so that all formally high-order numerical schemes for the eikonal equation yield first-order convergence and relatively large errors. Therefore, it is a long standing challenge in computational geometrical optics how to compute a uniformly high-order accurate solution for the point-source eikonal equation in a global domain. In this paper, assuming that both the squared slowness and the squared eikonal are analytic near the source, we propose high-order factorization based high-order hybrid fast sweeping methods for point-source eikonal equations to compute just such solutions. Observing that the *squared* eikonal is differentiable at the source, we propose to factorize the eikonal into two multiplicative or additive factors one of which is specified to approximate the eikonal up to arbitrary order of accuracy near the source, and the other of which serves as a higher-order correction term. This decomposition is achieved by using the eikonal equation and applying power series expansions to both the squared eikonal and the squared slowness function. We develop recursive formulas to compute the approximate eikonal up to arbitrary order of accuracy near the source. Furthermore, these approximations enable us to decompose the eikonal into two factors either multiplicatively or additively so that we can design two new types of hybrid, high-order fast sweeping schemes for the point-source eikonal equation. We also show that the first-order hybrid fast sweeping methods are monotone and consistent so that they are convergent in computing viscosity solutions. Two- and three-dimensional numerical examples demonstrate that a hybrid p-th order fast sweeping method yields desired, uniform, clean p-th order convergence in a global domain by using a p-th order factorization.

Key words. eikonal equation, fast sweeping methods, higher-order factorizations, higher-order schemes

AMS subject classifications. 65N30, 65M60

1. Introduction.

We consider the eikonal equation with a point-source condition,

$$\begin{aligned} &H(\nabla\tau(\mathbf{x})) \equiv |\nabla\tau(\mathbf{x})| = s(\mathbf{x}), \quad \mathbf{x} \in \Omega \setminus \{\mathbf{x}_0\}; \\ &\tau(\mathbf{x}_0) = 0, \end{aligned} \tag{1.1}$$

where $\Omega \subset \mathcal{R}^m$, $\mathbf{x}_0$ is the source point, $\tau(\mathbf{x})$ is the so-called eikonal (sometimes called traveltime as well), and $s(\mathbf{x}) \geq \eta > 0$ is the slowness field with η a positive constant. The eikonal equation (1.1) has a wide variety of applications ranging from classical mechanics, geosciences, geometrical optics, computer vision to optimal control. One specific example of the point-source eikonal equation arises from computing asymptotic Green functions for Helmholtz equations in the high frequency regime, which are essential for seismic imaging and geophysical inverse problems. The particularity of the point-source eikonal equation is that the eikonal is not differentiable at the source because it behaves like distance from the source in the travel-time metric. Without special treatments at the source point, all formally high-order numerical schemes yield only first-order convergence as the high-order Taylor expansion based local truncation

*Department of Mathematics, Iowa State University, Ames, IA 50011. Email: luos@iastate.edu.

†Department of Mathematics, Michigan State University, East Lansing, MI 48824. Email: qian@math.msu.edu. Supported by grants from the National Science Foundation.

‡Department of Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131. Email: burridge137@gmail.com.

error analysis fails to hold at the source point. Therefore, to solve the point-source eikonal equation uniformly up to high-order accuracy near the source poses a challenging computational problem in Eulerian geometrical optics [4]. Assuming that both the squared slowness and the squared eikonal are analytic near the sources, in this paper we tackle this difficulty by proposing high-order factorization-based high order hybrid sweeping schemes for point-source eikonal equations, and these new schemes enable us to compute the point-source eikonal with uniform high-order accuracy globally.

To design efficient high-order schemes for point-source eikonals, we have to overcome several obstacles. The first obstacle is how to explicitly extract the high-order information of non-differentiable eikonals at the source point. Observing that the squared eikonal is differentiable at the source, we propose to carry out power-series expansions for both the squared eikonal and squared slowness functions and utilize the squared eikonal equation. The power-series expansion yields a recursive formula for computing arbitrary order expansions of the squared eikonal, which in turn provides us with arbitrary order truncations of the eikonal itself at the source point. This indirect expansion enables us to successfully extract high-order information of the point-source eikonal at the source.

The second obstacle is how to effectively utilize high-order truncations of the point-source eikonal to design efficient numerical schemes. We propose to factorize the eikonal into two multiplicative or additive factors[1], one of which is specified to approximate the eikonal up to arbitrary order of accuracy near the source by using high-order truncations of the point-source eikonal, and the other of which serves as a high-order correction term. These two factorizations allow us to design efficient schemes to compute the correction term with uniform order of accuracy near the source.

The third obstacle is how to design high-order numerical schemes to compute the point-source eikonal with globally uniform high-order accuracy. Since the truncated high-order expansion of the point-source eikonal is only valid near the source point, upon which the high-order factorizations are based, we propose to partition the computational domain into two parts, one of which, called the source domain, is a small neighborhood of the source including the source point, and the other of which, called the non-source domain, is the complement of the source domain. In the source domain, we solve the factorized eikonal equation; in the non-source domain, we solve the original eikonal equation; the factorization formula serves as the bridge to link the two solutions obtained from the two versions of the eikonal equations. This strategy allows us to design hybrid, new, efficient high-order weighted essentially non-oscillatory (WENO) scheme based Lax-Friedrichs sweeping methods for solving point-source eikonal equations.

1.1. Related work. Because of the tremendous number of its applications, the eikonal equation has been tackled from many different perspectives, resulting in the vast literature on the topic; see [21, 35, 20, 34, 30, 6, 9, 23, 24, 25, 26, 33, 10, 12, 40, 39, 8, 27, 28, 11, 14, 5, 29, 2, 1, 38, 15] and references therein. When applied to the point-source eikonal equation, all of these algorithms yield polluted first-order accuracy without special treatments of the source point. To observe correct convergence order of a numerical algorithm for point-source eikonal equations, one has to initialize the eikonal analytically near the source by imposing a grid-independent region of constant

[1]For brevity we shall use the words 'factor', 'factorize', etc., even when the eikonal is split additively.

velocity near the source [36, 35, 30, 23, 25, 10, 40, 39, 8, 11, 14, 29, 1, 38] and measure the convergence order only for the solution outside that small region. Consequently, this special treatment of the point source has two essential drawbacks: (1) the slowness function may not be constant near the source, and (2) the convergence order is not globally uniform. In principle, highly accurate ray-tracing methods may be used to alleviate the first difficulty, but the second remains: non-uniformly convergent eikonal may hinder further application of the numerical eikonal to computing other quantities, such as amplitudes and take-off angles [24, 17, 19]. Other approaches to computing the point-source eikonal to high-order accuracy includes the adaptive grid refinement method [24], which compensates for the loss of accuracy near the source point, but the convergence order is still not globally uniform.

The factorization idea of dealing with the point-source eikonal equation has first appeared as the celerity transform in [22] and has been further developed in [37, 7, 17, 19, 18]; however, why the celerity transform yields highly accurate numerical solutions for the eikonal equation has not been fully understood until now. Our analysis-inspired high-order factorizations build a framework for dealing with the point-source eikonal up to arbitrary order of accuracy near the source, and include the celerity transform as a special case. Moreover, in comparison to the related works in [7, 17, 19, 18], our current work has advanced factorization-based fast sweeping methods in the following aspects: (a) it provides a theoretical basis for validity and performance of the factorization method in that the underlying mechanism of extraction of high-order information of the eikonal near the source is fully analyzed; (b) it extends the factorization techniques with both multiplicative and additive factors to higher order; and (c) it offers a systematic and efficient strategy to initialize the point-source condition for higher-order schemes when the exact solution is not available.

The hybrid, high-order WENO based Lax-Friedrichs sweeping methods are built upon the fast sweeping method [6, 10, 40], the first-order Lax-Friedrichs sweeping method [10], and the high-order WENO based Lax-Friedrichs sweeping method [39, 29]. Since the Lax-Friedrichs sweeping scheme [10] can handle convex and non-convex Hamiltonians with ease, we choose the Lax-Friedrichs numerical Hamiltonian as one of the building blocks in designing hybrid high-order sweeping methods.

We mention that high-order accurate eikonals are also important in solving linearized eikonal equations with respect to the velocity which arise in traveltime tomography [13, 32]. In these linearized eikonal equations, the traveltime gradient appears as the coefficient which usually is obtained by numerically differentiating computed traveltimes, thus high-order accurate traveltimes will be crucial for solving linearized eikonal equations with high accuracy. Therefore, the high-order schemes for eikonals proposed here will be useful in many applications, such as computational geometrical optics [24, 4, 19], traffic congestion equilibria [16], and traveltime tomography [13, 32].

A natural question is: what are the advantages of the proposed high-order schemes for eikonals? The advantages are at least two-fold. First, to achieve a certain specified accuracy, a high-order scheme needs a much coarser mesh than a first-order scheme does, thus high-order schemes are much more efficient than first-order schemes in terms of computational cost. Secondly, high-order accurate eikonals can be numerically differentiated to yield reliable eikonal gradients while first-order accurate eikonals cannot, as demonstrated in [24]; consequently, our proposed high-order schemes for point-source eikonal equations will be significant for solving linearized eikonal equations in traveltime tomography [13, 32] and other applications.

1.2. Layout. The rest of the paper is organized as follows. In Section 2, we first present arbitrary order expansions of the squared eikonals, which are followed by high-order truncations of the eikonals and related applications to multiplicative and additive factorizations. In Section 3, we present hybrid, new, high-order numerical schemes for computing the point-source eikonal and show that the hybrid, first-order scheme is monotone, consistent, and thus convergent. In Section 4, we use several 2-D and 3-D examples to demonstrate the performance and desired convergence order of the new schemes. Concluding remarks are given in Section 5.

2. High-Order Factored Eikonal Equations.

2.1. Arbitrary Order Expansion of Squared Eikonals. Without loss of generality, we assume that the source point is at the origin: $\mathbf{x}_0 = \mathbf{0}$. We first derive the eikonal equation for τ^2 and then proceed to solve it by power series about the origin. We assume that $T \equiv \tau^2$ is analytic and zero at the source, and $S \equiv s^2$ is analytic at the source. As shown in [31], the eikonal is locally smooth near the source except the source point itself; therefore, the analyticity assumption of the squared eikonal is reasonable. Let us expand T and S as power series,

$$\begin{array}{rcl} T(\mathbf{x}) & = & \sum_{\nu=0}^{\infty} T_\nu(\mathbf{x}), \\ S(\mathbf{x}) & = & \sum_{\nu=0}^{\infty} S_\nu(\mathbf{x}), \end{array} \tag{2.1}$$

where $T_\nu(\mathbf{x})$ and $S_\nu(\mathbf{x})$ are homogeneous polynomials of degree ν in $\mathbf{x}$. Since

$$\nabla T = \nabla \tau^2 = 2\tau \nabla \tau, \tag{2.2}$$

we find that the eikonal equation in terms of T is

$$|\nabla T|^2 = 4\tau^2 |\nabla \tau|^2 = 4ST. \tag{2.3}$$

We write this in the form

$$ST = \frac{1}{4}|\nabla T|^2. \tag{2.4}$$

Hence

$$\sum_{\nu=0}^{\infty} S_\nu(\mathbf{x}) \sum_{\nu=0}^{\infty} T_\nu(\mathbf{x}) = \frac{1}{4}\left(\sum_{\nu=0}^{\infty} \nabla T_\nu(\mathbf{x})\right)^2. \tag{2.5}$$

Since $\tau(\mathbf{0}) = 0$, we find that $T_0 = 0$. So we may write

$$\left(\sum_{\nu=0}^{\infty} S_\nu(\mathbf{x})\right)\left(\sum_{\mu=1}^{\infty} T_\mu(\mathbf{x})\right) = \frac{1}{4}\left(\sum_{\nu=1}^{\infty} \nabla T_\nu(\mathbf{x})\right)^2. \tag{2.6}$$

Comparing the constant terms we find that

$$0 = \frac{1}{4}|\nabla T_1|^2 \tag{2.7}$$

so that, since $T_1(\mathbf{x})$ is linear in $\mathbf{x}$,

$$T_1(\mathbf{x}) = 0. \tag{2.8}$$

Hence we rewrite (2.6) as

$$\left(\sum_{\nu=0}^{\infty} S_\nu(\mathbf{x})\right)\left(\sum_{\mu=2}^{\infty} T_\mu(\mathbf{x})\right) = \frac{1}{4}\left(\sum_{\nu=2}^{\infty} \nabla T_\nu(\mathbf{x})\right)^2. \tag{2.9}$$

Let us consider the quadratic term

$$S_0 T_2(\mathbf{x}) = \frac{1}{4}|\nabla T_2(\mathbf{x})|^2. \tag{2.10}$$

But since $T_2(\mathbf{x})$ is a quadratic form we may write it as

$$T_2(\mathbf{x}) = \mathbf{x}^T \mathbf{A}\mathbf{x}\,, \quad \text{with} \quad \nabla T_2(\mathbf{x}) = 2\mathbf{A}\,\mathbf{x}\,, \tag{2.11}$$

where $\mathbf{A}$ is a symmetric matrix. So

$$S_0 \mathbf{A} = \mathbf{A}^2. \tag{2.12}$$

We reject the zero solution and assume that $\mathbf{A}$ is invertible. Then

$$\mathbf{A} = S_0 \mathbf{I}, \tag{2.13}$$

where $\mathbf{I}$ is the identity matrix, and hence

$$T_2(\mathbf{x}) = S_0\,\mathbf{x}^2. \tag{2.14}$$

Here and in the following, without confusions we use the nonstandard notation $\mathbf{y}^2$ to denote $\mathbf{y}^2 = \mathbf{y}^T\mathbf{y}$, where $\mathbf{y}$ is any column vector in $\mathcal{R}^m$.

Let us now equate the terms of P-th degree on the left and right of equation (2.9),

$$\begin{aligned}
\textstyle\sum_{\nu=0}^{P-2} S_\nu(\mathbf{x})\, T_{P-\nu}(\mathbf{x}) &= \frac{1}{4}\sum_{\nu=1}^{P-1} \nabla T_{\nu+1}(\mathbf{x}) \cdot \nabla T_{P-\nu+1}(\mathbf{x}) \\
&= S_0\,\mathbf{x}\cdot\nabla T_P(\mathbf{x}) + \frac{1}{4}\sum_{\nu=2}^{P-2} \nabla T_{\nu+1}(\mathbf{x}) \cdot \nabla T_{P-\nu+1}(\mathbf{x}) \qquad (2.15)\\
&= P\, S_0\, T_P(\mathbf{x}) + \frac{1}{4}\sum_{\nu=2}^{P-2} \nabla T_{\nu+1}(\mathbf{x}) \cdot \nabla T_{P-\nu+1}(\mathbf{x}),
\end{aligned}$$

using Euler's theorem on homogeneous functions in the last step. Separating the term $\nu = 0$ on the left and rearranging we find that

$$(P-1)\, S_0\, T_P(\mathbf{x}) = \sum_{\nu=1}^{P-2} S_\nu(\mathbf{x})\, T_{P-\nu}(\mathbf{x}) - \frac{1}{4}\sum_{\nu=2}^{P-2} \nabla T_{\nu+1}(\mathbf{x}) \cdot \nabla T_{P-\nu+1}(\mathbf{x}). \tag{2.16}$$

Since the right side of this contains only T_ν for $\nu < P$, we may solve this system recursively starting with $P = 3$ (for which the second term on the right does not arise).

For example, we can set $P = 3$ in (2.16) to get

$$2\, S_0\, T_3 = S_1\, T_2, \tag{2.17}$$

so that

$$T_3 = \frac{1}{2S_0} S_1\, T_2 = \frac{1}{2}\, S_1\, \mathbf{x}^2. \tag{2.18}$$

For $P = 4$, (2.16) becomes

$$3\,S_0\,T_4 = S_1\,T_3 + S_2\,T_2 - \frac{1}{4}(\nabla T_3)^2. \tag{2.19}$$

By computing ∇T_3 from (2.18), we obtain the following,

$$T_4 \quad = \quad \frac{\mathbf{x}^2}{48S_0}\left[16\,S_0\,S_2 - \mathbf{x}^2\,(\nabla S_1)^2\right]. \tag{2.20}$$

We may proceed to obtain further terms in a similar way. For example,

$$T_5 = \frac{(\mathbf{x}^2)^2}{96S_0^2}\Big[-2\,S_0\,\nabla S_1{\cdot}\nabla S_2 + S_1\,(\nabla S_1)^2\Big] + \frac{1}{4}\mathbf{x}^2\,S_3. \tag{2.21}$$

2.2. High-order Truncation of Eikonals. With the above recursive formulas at our disposal, we can now truncate the analytical expansion of the squared eikonal,

$$\tilde{\tau}_N^2(\mathbf{x}) \equiv \tilde{T}_N(\mathbf{x}) = \sum_{v=2}^{N} T_v(\mathbf{x}), \tag{2.22}$$

where $N \geq 2$ is a user-specified positive integer. It is this truncation that allows us to extract high-order information of the eikonal τ in a neighborhood of the source.

For future reference we note that as $r \equiv |\mathbf{x}| \to 0$ we see from (2.14) that

$$T = O(r^2) \quad \text{and} \quad \tau = \sqrt{T} = O(r), \tag{2.23}$$

and we also have

$$\tilde{T}_N = O(r^2) \quad \text{and} \quad \tilde{\tau}_N = \sqrt{\tilde{T}_N} = O(r). \tag{2.24}$$

For example, with $N = 2$, we have

$$\tilde{\tau}_2^2(\mathbf{x}) = \tilde{T}_2(\mathbf{x}) = T_2(\mathbf{x}) = S_0\mathbf{x}^2, \tag{2.25}$$

where $\tilde{\tau}_2$ is the distance function satisfying

$$|\nabla\tilde{\tau}_2| = \tilde{s}, \quad \tilde{\tau}_2(\mathbf{0}) = 0, \tag{2.26}$$

with $\tilde{s} \equiv \sqrt{S_0} = \sqrt{S(\mathbf{0})}$.

And with $N = 3$, we have

$$\tilde{\tau}_3^2(\mathbf{x}) = \tilde{T}_3(\mathbf{x}) = T_2(\mathbf{x}) + T_3(\mathbf{x}) = (S_0 + \frac{1}{2}S_1)\mathbf{x}^2. \tag{2.27}$$

Then,

$$2\tilde{\tau}_3\nabla\tilde{\tau}_3 = \nabla\tilde{T}_3 = \mathbf{x}^2\nabla(S_0 + \frac{1}{2}S_1) + 2(S_0 + \frac{1}{2}S_1)\mathbf{x} \tag{2.28}$$

and

$$4\tilde{\tau}_3^2|\nabla\tilde{\tau}_3|^2 = \frac{1}{4}(\mathbf{x}\cdot\mathbf{x})^2\nabla\,S_1\cdot\nabla\,S_1 + 2\mathbf{x}\cdot\mathbf{x}S_1\,(S_0 + \frac{1}{2}S_1) + 4(S_0 + \frac{1}{2}S_1)^2\mathbf{x}\cdot\mathbf{x}. \tag{2.29}$$

Thus, we have

$$|\nabla\tilde{\tau}_3|^2 = \frac{1}{16(S_0+\frac{1}{2}S_1)}\left((\mathbf{x}\cdot\mathbf{x})\nabla S_1\cdot\nabla S_1 + 8S_1(S_0+\frac{1}{2}S_1) + 16(S_0+\frac{1}{2}S_1)^2\right). \tag{2.30}$$

In general, for $N \geq 2$ we have the following to hold,

$$|\nabla\tilde{\tau}_N|^2 = f(\mathbf{x}; S), \tag{2.31}$$

where the function $f(\mathbf{x}; S)$ depends on both $\mathbf{x}$ and the function S. However, we do not need to actually solve the above eikonal equation, as we know the solution $\tilde{\tau}_N$ already by above power series expansions.

Moreover, we have the following Lemma.

LEMMA 2.1. *The truncation $\tilde{\tau}_N$ approximates τ in the following way:*

$$\tau = \tilde{\tau}_N + O(|\mathbf{x}|^N) \tag{2.32}$$

near the source, where $N \geq 2$ is a given arbitrary integer.

On dividing (2.32) by $\tilde{\tau}_N$ and invoking (2.24) we have

LEMMA 2.2.

$$\frac{\tau}{\tilde{\tau}_N} = 1 + O(|\mathbf{x}|^{N-1}). \tag{2.33}$$

To prove Lemma 2.1, we write

$$\begin{aligned}\tau - \tilde{\tau}_N &= \frac{T-\tilde{T}_N}{\tau+\tilde{\tau}_N} = \frac{T-\tilde{T}_N}{\tau(1+\frac{\tilde{\tau}_N}{\tau})} \\ &= \frac{\sum_{k=N+1}^{\infty} T_k(\omega) r^{k-1}}{(1+\frac{\tilde{\tau}_N}{\tau})\sqrt{\sum_{k=2}^{\infty} T_k(\omega) r^{k-2}}}.\end{aligned} \tag{2.34}$$

Since $T_2(\omega) \neq 0$, we have $\tau - \tilde{\tau}_N = O(r^N)$. □

To facilitate further discussion, we introduce the following definition.

DEFINITION 2.3. *Let τ_h be a numerical approximation to the eikonal τ which is obtained by a certain numerical method on a computational mesh of size h. We say that the numerical method is of order k if the following estimate holds,*

$$\|\tau - \tau_h\| \leq C\, h^k, \tag{2.35}$$

where $\|\cdot\|$ is either ℓ_1 or ℓ_∞ norm, k is a positive integer, and C is a positive constant independent of h and k.

According to Definition 2.3, Lemma 2.1 implies that

$$\|\tau - \tilde{\tau}_N\| \leq C\, h^N \tag{2.36}$$

holds in a small neighborhood $\Omega_\alpha = \{|\mathbf{x}| \leq \alpha\, h\}$ of the source, where C is a positive constant independent of h and N, and α is a small positive integer. Therefore, $\tilde{\tau}_N$ approximates τ with N-th order accuracy near the source.

2.3. Applications of Truncated Eikonals. With the above high-order expansions of the eikonal near the source, we present the factored eikonal equations.

2.3.1. Multiplicative Factorization of Eikonal Equations. Since $\tilde{\tau}_N$ is an N-th order approximation to the eikonal τ in a neighborhood of the source, by Lemma 2.2 $u = \frac{\tau}{\tilde{\tau}_N} = 1 + O(r^{N-1})$ can be approximated by the constant function 1 with $(N-1)$-th order accuracy in that neighborhood. This leads us to consider the following multiplicative factorization in a neighborhood of the source,

$$\tau = u\,\tilde{\tau}_N, \tag{2.37}$$

where $\tilde{\tau}_N$ is specified according to equation (2.22), and the unknown correction u should satisfy the following factored eikonal equation

$$|\nabla\tau| = \sqrt{\tilde{\tau}_N^2|\nabla u|^2 + 2\tilde{\tau}_N u\nabla\tilde{\tau}_N\cdot\nabla u + u^2|\nabla\tilde{\tau}_N|^2} = s. \tag{2.38}$$

The point-source boundary condition for u is

$$\lim_{\mathbf{x}\to\mathbf{0}} u(\mathbf{x}) = \lim_{\mathbf{x}\to\mathbf{0}} \frac{\tau(\mathbf{x})}{\tilde{\tau}_N(\mathbf{x})} = 1. \tag{2.39}$$

Now that the function u is smooth with up to N-th order derivatives at the source due to the singularity cancellation and in fact can be approximated by the constant function 1 with $(N-1)$-th order accuracy in a small neighborhood of the point source, an N-th order scheme is effective for solving the factored eikonal equation with N-th order convergence in a neighborhood of the source so that τ can be recovered from $\tau = u\,\tilde{\tau}_N$ with N-th order accuracy.

In particular, if $N = 2$, we have $\tilde{\tau}_2 = s(\mathbf{0})|\mathbf{x}|$ which is the distance function corresponding to a homogeneous medium. This particular case has been discovered as the so-called celerity transform in [22] for obtaining highly accurate finite-difference solutions for the eikonal equation, which has been further developed in [37, 7, 17, 19, 18]. According to our derivation and analysis above, we now understand why the celerity transform is effective for the point-source eikonal equations: it yields a second-order approximation to the eikonal in a neighborhood of the source point.

2.3.2. Eikonal Equations for Additive Splitting. We can also decompose τ into two additive factors,

$$\tau = \tilde{\tau}_N + u$$

where $\tilde{\tau}_N$ is specified according to equation (2.22), and the unknown correction u is to be determined. Substituting the factorization into the eikonal equation (1.1), we have the following factored eikonal equation,

$$|\nabla\tau| = \sqrt{|\nabla u|^2 + 2\nabla\tilde{\tau}_N\cdot\nabla u + |\nabla\tilde{\tau}_N|^2} = s. \tag{2.40}$$

Since $\tilde{\tau}_N$ captures the source singularities, u is differentiable up to N-th order near the source. In addition, we know

$$\lim_{\mathbf{x}\to\mathbf{0}} u(\mathbf{x}) = \lim_{\mathbf{x}\to\mathbf{0}} (\tau(\mathbf{x}) - \tilde{\tau}_N(\mathbf{x})) = 0. \tag{2.41}$$

Thus we can solve (2.40) for u with an N-th order accurate scheme so as to recover τ with N-th order accuracy.

In particular, when $N = 2$, the corresponding factorization has been first proposed in [18].

Next we design hybrid numerical schemes to make use of the above two factorizations.

3. Hybrid High-Order Fast Sweeping Methods. For the factored eikonal equations, when $N = 2$, the choice of $\tilde{\tau}_2$ coincides with the one used in [22, 37, 7, 17, 19, 18]; in this case, $\tilde{T}_2$ and $\tilde{\tau}_2$ are guaranteed to be non-negative in the whole domain.

For $N > 2$, $\tilde{T}_N$ is not guaranteed to be non-negative in the whole domain Ω except near the source, which implies that $\tilde{\tau}_N$ may not be defined away from the source; therefore the factored eikonal equation (2.38) or (2.40) cannot be applied in the whole domain. In order to resolve this issue, we propose a hybrid strategy: solving the factored eikonal equations locally in a neighborhood of the source where the non-negativity of $\tilde{T}_N$ is guaranteed while switching to the original eikonal equation away from this neighborhood. The setup is illustrated in Figure 3.1(a): $\Omega = \Omega_u \cup \Omega_\tau$, where Ω_u is closed and Ω_τ is open. The intersection of the two regions is $\Gamma = \Omega_u \cap \bar{\Omega}_\tau$; in region Ω_u, we solve the factored eikonal equations; in region Ω_τ, we solve the original eikonal equation.

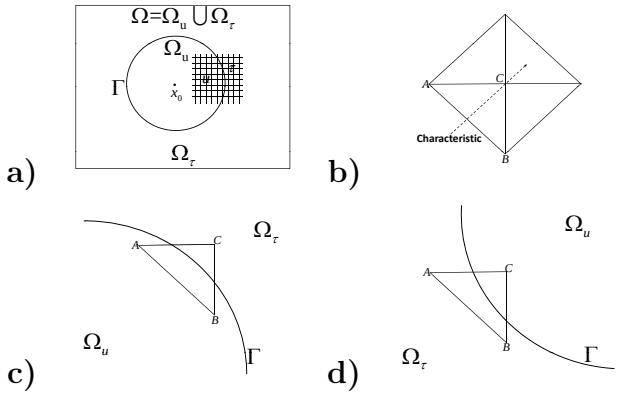

FIG. 3.1. *Domain and Mesh (2-D). (a) Domain Ω and different regions Ω_u and Ω_τ; (b) Local mesh of point C; (c) and (d): A triangle/simplex across two regions Ω_u and Ω_τ.*

We present both first-order and high-order hybrid sweeping methods.

3.1. First-Order Fast Sweeping Schemes. To set up the stage for further development, we summarize the fast sweeping scheme for the original eikonal equation (1.1) and factored eikonal equations (2.38) and (2.40) with $N = 2$, on a rectangular mesh Ω^h with grid size h covering the domain Ω (also see [40, 27, 28, 7, 18] and references therein). Without loss of generality, let us consider Hamilton-Jacobi equations in the following generic form in 2-D,

$$F(x, z, u, u_x, u_z) = f(x, z), \tag{3.1}$$

where F is convex in the gradient variable.

Taking a local mesh of point $C = (x_C, z_C)$ as shown in Figure 3.1(b), we consider discretizations on the triangle with neighbors $A = (x_A, z_A)$ and $B = (x_B, z_B)$,

$$\nabla u(C) \approx \left(\frac{u(C) - u(A)}{x_C - x_A}, \frac{u(C) - u(B)}{z_C - z_B} \right), \tag{3.2}$$

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

$$\hat{F}(C, u(C), u(A), u(B)) \equiv F\left(C, u(C), \frac{u(C) - u(A)}{x_C - x_A}, \frac{u(C) - u(B)}{z_C - z_B} \right) - f(C) = 0. \tag{3.3}$$

Given $u(A)$ and $u(B)$, we wish to solve (3.3) 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 (3.3);
2. Scenario 2: There is one solution for $u(C)$ from (3.3);
3. Scenario 3: There are two solutions for $u(C)$ from (3.3).

In Scenario 1, we enforce the characteristic equation for the Hamilton-Jacobi equation 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 [27, 28, 7, 18]. In Scenario 2 or 3, we need to further check whether a candidate value for $u(C)$ that is consistent with equation (3.1) 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 is more than one candidate, we choose the minimum among all possible candidates.

We summarize the method into the following algorithm.

ALGORITHM 1 (First-order fast sweeping method (FSM) [40, 27, 28, 7, 18]).

1. *Initial guess (enforce the boundary condition):*
   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 alternating orderings (sweepings):*
   - *Update during each iteration: at a vertex C, the updated value $u^{new}(C)$ at C is*

$$u^{new}(C) = \min\{u^{old}(C), u^{comp}(C)\}, \tag{3.4}$$

   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 (3.3) *and the procedure detailed as above.*
   - *Orderings: four alternating orderings are needed,*

$$\begin{array}{ll} (1)\ i = 1 : I; j = 1 : J; & (2)\ i = I : 1; j = 1 : J; \\ (3)\ i = I : 1; j = J : 1; & (4)\ i = 1 : I; j = J : 1. \end{array} \tag{3.5}$$

3.2. First-Order Hybrid Fast Sweeping Schemes. With the first-order fast sweeping scheme at our disposal, we present first-order hybrid fast sweeping methods for the 2-D case only, as an extension to the 3-D case is not difficult. A first-order hybrid scheme is in general structured as follows, referring to Figure 3.1:

- in region Ω_u, we first solve the factored-eikonal equations for u, and we then recover τ corresponding to different factorizations;
- in region Ω_τ, we solve the original equation for τ.

Similarly, for the local mesh of point C as shown in Figure 3.1(b), without loss of generality, let us also focus on the triangle with neighbors A and B. We consider discretizations corresponding to different regions and equations:

- For the original eikonal equation (1.1) in Ω_τ, $\nabla\tau$ is approximated as

$$\nabla\tau(C) \approx \left(\frac{\tau(C) - \tau(A)}{x_C - x_A}, \frac{\tau(C) - \tau(B)}{z_C - z_B} \right). \tag{3.6}$$

When C is near or on Γ, A and/or B may be in region Ω_u (see Figure 3.1(c)). In this case, for example, if A is in region Ω_u, then $\tau(A)$ should be given by $\tilde{\tau}_2(A)$ and $u(A)$ corresponding to the specific factorization used since $\tilde{\tau}_2$ and u are known at A, .

- For the factored eikonal equation (2.38) in Ω_u, $\nabla\tau$ is given as

$$\nabla\tau(C) = \nabla\tilde{\tau}_2(C)u(C) + \tilde{\tau}_2(C)\nabla u(C), \tag{3.7}$$

with ∇u approximated as

$$\nabla u(C) \approx \left(\frac{u(C)-u(A)}{x_C - x_A}, \frac{u(C)-u(B)}{z_C - z_B}\right); \tag{3.8}$$

thus,

$$\nabla\tau(C) \approx \left(\frac{\tau(C) - \frac{\tau(A)}{\tilde{\tau}_2(A)}\tilde{\tau}_2(C)}{x_C - x_A}, \frac{\tau(C) - \frac{\tau(B)}{\tilde{\tau}_2(B)}\tilde{\tau}_2(C)}{z_C - z_B}\right) + \frac{\tau(C)}{\tilde{\tau}_2(C)}\nabla\tilde{\tau}_2(C). \tag{3.9}$$

When C is near or on Γ, A and/or B may be in region Ω_τ (see Figure 3.1(d)). In this case, for example, if A is in region Ω_τ, then $u(A)$ should be given as $\dfrac{\tau(A)}{\tilde{\tau}_2(A)}$ since $\tilde{\tau}_2$ and τ are known at A.

- For the factored eikonal equation (2.40) in Ω_u, $\nabla\tau$ is given as

$$\nabla\tau(C) = \nabla\tilde{\tau}_2(C) + \nabla u(C), \tag{3.10}$$

with $\nabla u(C)$ as in (3.8); therefore,

$$\begin{aligned}\nabla\tau(C) \approx &\left(\frac{\tau(C)-\tau(A)}{x_C - x_A}, \frac{\tau(C)-\tau(B)}{z_C - z_B}\right)\\ &- \left(\frac{\tilde{\tau}_2(C)-\tilde{\tau}_2(A)}{x_C - x_A}, \frac{\tilde{\tau}_2(C)-\tilde{\tau}_2(B)}{z_C - z_B}\right) + \nabla\tilde{\tau}_2(C).\end{aligned} \tag{3.11}$$

When C is near or on Γ, A and/or B may be in region Ω_τ (see Figure 3.1(d)). In this case, for example, if A is in region Ω_τ, then $u(A)$ should be given as $\tau(A) - \tilde{\tau}_2(A)$ since $\tilde{\tau}_2$ and τ are known at A.

With the above discretizations, let us define the numerical Hamiltonian $\hat{H}$ as

$$\hat{H}(C, \tau(C), \tau(A), \tau(B)) \equiv H(\nabla\tau(C)) - s(C) = 0, \tag{3.12}$$

with $\nabla\tau(C)$ approximated according to the above different cases.

As indicated in Section 3.1, the causality condition enforced in the first-order fast sweeping scheme for the eikonal and factored eikonal equations is

$$\nabla\tau(C)\cdot\mathbf{r}_{AC} \geq 0, \text{ and } \nabla\tau(C)\cdot\mathbf{r}_{BC} \geq 0, \tag{3.13}$$

since in this particular case the characteristic direction is in the same direction as $\nabla\tau(C)$.

Using the above causality condition, we show that the scheme (3.12) is monotone and consistent.

LEMMA 3.1. *The scheme* (3.12) *under the causality condition* (3.13) *is consistent and monotone; that is,*

$$\frac{\partial \hat{H}(C,\tau(C),\tau(A),\tau(B))}{\partial \tau(C)} \geq 0;\ \frac{\partial \hat{H}(C,\tau(C),\tau(A),\tau(B))}{\partial \{\tau(A),\tau(B)\}} \leq 0. \tag{3.14}$$

Proof. The consistency is obvious. We prove monotonicity. Denote $\mathbf{p} = (p_1, p_2) \equiv \nabla\tau(C)$. The causality condition (3.13) given on a rectangular mesh is

$$p_1(x_C - x_A) \geq 0, \text{ and } p_2(z_C - z_B) \geq 0. \tag{3.15}$$

For the discretization of the original eikonal equation with $\nabla\tau$ as given in (3.6), we have

$$\begin{aligned}
\frac{\partial \hat{H}}{\partial \tau(C)} &= \frac{1}{s(C)}\left(\frac{p_1}{x_C - x_A} + \frac{p_2}{z_C - z_B}\right) \geq 0;\\
\frac{\partial \hat{H}}{\partial \tau(A)} &= -\frac{1}{s(C)}\frac{p_1}{(x_C - x_A)} \leq 0;\\
\frac{\partial \hat{H}}{\partial \tau(B)} &= -\frac{1}{s(C)}\frac{p_2}{(z_C - z_B)} \leq 0.
\end{aligned} \tag{3.16}$$

For the discretization of the factored eikonal equation with multiplicative factors and $\nabla\tau(C)$ as given in (3.9), the causality condition (3.15) gives

$$\begin{aligned}
\tau(C) - \frac{\tau(A)}{\tilde{\tau}_2(A)}\tilde{\tau}_2(C) + \frac{\tau(C)}{\tilde{\tau}_2(C)}\tilde{\tau}_{2,x}(C)(x_C - x_A) \geq 0,\\
\tau(C) - \frac{\tau(B)}{\tilde{\tau}_2(B)}\tilde{\tau}_2(C) + \frac{\tau(C)}{\tilde{\tau}_2(C)}\tilde{\tau}_{2,z}(C)(z_C - z_B) \geq 0,
\end{aligned} \tag{3.17}$$

which imply that

$$\begin{aligned}
1 + \frac{\tilde{\tau}_{2,x}(C)}{\tilde{\tau}_2(C)}(x_C - x_A) \geq \frac{\tilde{\tau}_2(C)}{\tau(C)}\frac{\tau(A)}{\tilde{\tau}_2(A)} \geq 0,\\
1 + \frac{\tilde{\tau}_{2,z}(C)}{\tilde{\tau}_2(C)}(z_C - z_B) \geq \frac{\tilde{\tau}_2(C)}{\tau(C)}\frac{\tau(B)}{\tilde{\tau}_2(B)} \geq 0.
\end{aligned} \tag{3.18}$$

Then the following results hold,

$$\begin{aligned}
\frac{\partial \hat{H}}{\partial \tau(C)} &= \frac{1}{s(C)}\left\{\frac{p_1}{x_C - x_A}\left(1 + \frac{\tilde{\tau}_{2,x}(C)}{\tilde{\tau}_2(C)}(x_C - x_A)\right)\right\}\\
&\quad + \frac{1}{s(C)}\left\{\frac{p_2}{z_C - z_B}\left(1 + \frac{\tilde{\tau}_{2,z}(C)}{\tilde{\tau}_2(C)}(z_C - z_B)\right)\right\} \geq 0,\\
\frac{\partial \hat{H}}{\partial \tau(A)} &= -\frac{1}{s(C)}p_1\frac{\tilde{\tau}_2(C)}{\tilde{\tau}_2(A)(x_C - x_A)} = -\frac{p_1}{s(C)(x_C - x_A)}\frac{\tilde{\tau}_2(C)}{\tilde{\tau}_2(A)} \leq 0,\\
\frac{\partial \hat{H}}{\partial \tau(B)} &= -\frac{1}{s(C)}p_2\frac{\tilde{\tau}_2(C)}{\tilde{\tau}_2(B)(z_C - z_B)} = -\frac{p_2}{s(C)(z_C - z_B)}\frac{\tilde{\tau}_2(C)}{\tilde{\tau}_2(B)} \leq 0.
\end{aligned} \tag{3.19}$$

For the discretization of the factored eikonal equation with additive factors and $\nabla\tau(C)$ as given in (3.11), we have

$$\begin{aligned}
\frac{\partial \hat{H}}{\partial \tau(C)} &= \frac{1}{s(C)}\left(\frac{p_1}{x_C - x_A} + \frac{p_2}{z_C - z_B}\right) \geq 0; \\
\frac{\partial \hat{H}}{\partial \tau(A)} &= -\frac{1}{s(C)}\frac{p_1}{x_C - x_A} \leq 0; \\
\frac{\partial \hat{H}}{\partial \tau(B)} &= -\frac{1}{s(C)}\frac{p_2}{z_C - z_B} \leq 0.
\end{aligned} \tag{3.20}$$

Therefore, we prove the monotonicity. □

By the convergence theorem of Barles-Souganidis [3], Lemma 3.1 guarantees that the numerical solution of a hybrid, first-order fast sweeping scheme will converge to the viscosity solution of the eikonal equation as the mesh size goes to zero.

We summarize the new, first-order hybrid fast sweeping methods into the following algorithm.

ALGORITHM 2 (Hybrid first-order fast sweeping method (FSM)).

1. *Initial guess (enforce the boundary condition):*
   For grid points on or near the point source, their values are set to be 1 for multiplicative factors and 0 for additive factors. All other points are assigned a large value, for example infinity, initially.
2. *Gauss-Seidel iterations with alternating orderings (sweepings):*
   - *Update during each iteration: at a vertex C,*
     - *in region Ω_u, the updated value $u^{new}(C)$ at C is*

$$u^{new}(C) = \min\{u^{old}(C), u^{comp}(C)\}, \tag{3.21}$$

       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 (3.12) *and the procedure detailed as in Section 3.1. Hence $\tau^{new}(C) = \tilde{\tau}_2(C)u^{new}(C)$ or $\tau^{new}(C) = \tilde{\tau}_2(C) + u^{new}(C)$;*
     - *in region Ω_τ, the updated value $\tau^{new}(C)$ at C is*

$$\tau^{new}(C) = \min\{\tau^{old}(C), \tau^{comp}(C)\}, \tag{3.22}$$

       where $\tau^{old}(C)$ is the current value at C and $\tau^{comp}(C)$ is the value at C computed from the current given neighboring values according to (3.12) *and the procedure detailed as in Section 3.1. Hence $u^{new}(C) = \dfrac{\tau^{new}(C)}{\tilde{\tau}_2(C)}$ or $u^{new}(C) = \tau^{new}(C) - \tilde{\tau}_2(C)$.*
   - *Orderings: four alternating orderings are needed,*

$$\begin{array}{ll} (1)\ i = 1 : I; j = 1 : J; & (2)\ i = I : 1; j = 1 : J; \\ (3)\ i = I : 1; j = J : 1; & (4)\ i = 1 : I; j = J : 1. \end{array} \tag{3.23}$$

   - *Stopping criterion: given $\delta > 0$, check if $|\tau^{new} - \tau^{old}| < \delta$ in region Ω_τ and check if $|u^{new} - u^{old}| < \delta$ in region Ω_u.*

3.3. Hybrid High-Order Lax-Friedrichs Sweeping Schemes. Based on the high-order Lax-Friedrichs scheme, we present hybrid, third-order WENO based high-order Lax-Friedrichs sweeping methods by using the high-order factorizations derived above. Analogous to a first-order hybrid fast sweeping scheme, a hybrid high-order scheme is in general structured as follows:

- in the region Ω_u, we first solve the factored eikonal equations for u, and we then recover τ corresponding to the applied factorization;
- in the region Ω_τ, we solve the original eikonal equation for τ.

We summarize the hybrid high-order fast sweeping methods into the following algorithm.

ALGORITHM 3 (Hybrid high-order Lax-Friedrichs sweeping methods).

1. *Initial guess (enforce the boundary condition):*
   For grid points in a $2h \times 2h$ small domain covering the point source, their values are set to be 1 for multiplicative factors and 0 for additive factors. All other points are assigned a large value, for example infinity, initially.
2. *Gauss-Seidel iterations with alternating orderings (sweepings):*
   - *Update during each iteration: at a vertex C,*
     - *In region Ω_u, the updated value $u^{new}(C)$ at C is computed from the current given neighboring values according to* (3.12) *and the procedure detailed as in [20, 10, 39, 17, 19]. Hence $\tau^{new}(C) = \tilde{\tau}_N(C)u^{new}(C)$ or $\tau^{new}(C) = \tilde{\tau}_N(C) + u^{new}(C)$.*
     - *In region Ω_τ, the updated value $\tau^{new}(C)$ at C is computed from the current given neighboring values according to* (3.12) *and the procedure detailed as in [20, 10, 39, 17, 19]. Hence $u^{new}(C) = \dfrac{\tau^{new}(C)}{\tilde{\tau}_N(C)}$ or $u^{new}(C) = \tau^{new}(C) - \tilde{\tau}_N(C)$.*
   - *Orderings: four alternating orderings are needed,*

$$\begin{array}{ll} (1)\ i = 1 : I; j = 1 : J; & (2)\ i = I : 1; j = 1 : J; \\ (3)\ i = I : 1; j = J : 1; & (4)\ i = 1 : I; j = J : 1. \end{array} \tag{3.24}$$

   - *Stopping criterion: given $\delta > 0$, check if $|\tau^{new} - \tau^{old}| < \delta$ in region Ω_τ and check if $|u^{new} - u^{old}| < \delta$ in region Ω_u.*

REMARK 1. *Since the base scheme for the hybrid, high-order Lax-Friedrichs scheme amounts to applying first-order, monotone Lax-Friedrichs schemes [10] in both regions of Ω_u and Ω_τ, the base scheme is also monotone; that is, at a given point (i, j), the numerical Hamiltonian is non-deceasing at $\tau_{i,j}$ and non-increasing at the neighbors $\tau_{N\{i,j\}}$.*

REMARK 2. *If we choose $\tilde{\tau}_N$ with $N = 2$, then the initialization near the source is of second-order accuracy; hence we can expect the scheme to be second-order accurate globally when the third-order WENO Lax-Friedrichs sweeping scheme is used. If we choose $\tilde{\tau}_N$ with $N = 3$, then the initialization near the source is of third-order accuracy; consequently, we can expect the scheme to be third-order accurate globally when the third-order WENO Lax-Friedrichs sweeping scheme is used. Numerical examples in Section 4 verify these claims. In the implementation of the third-order WENO Lax-Friedrichs sweeping scheme, high order polynomial extrapolations are used for ghost points on the boundary of the computational domain; see [20, 10, 39, 17, 19].*

REMARK 3. *Intuitively, if we assume that (1) the solution for the eikonal equation with a point source is smooth except at the source point; (2) a p-th order WENO based Lax-Friedrichs sweeping scheme yields p-th order accuracy when the eikonal is smooth; and (3) an N-th order additive factorization is used in hybrid WENO based Lax-Friedrichs sweeping, then the following estimate holds:*

$$\|\tau - \tau_h\| \leq C\, h^{\min(p,N)}, \tag{3.25}$$

where τ_h is the eikonal computed by the hybrid WENO based sweeping method on a given mesh of size h, and C is a positive constant independent of h, p, and N.

This estimate can be roughly derived for the additive factorization as the following. Since τ_h is defined according to the following formula,

$$\tau_h = \begin{cases} \tilde{\tau}_N & \text{in } \Omega_\alpha = \{|\mathbf{x}| \le \alpha\, h\}; \\ \tilde{\tau}_N + u_h & \text{in } \Omega_u \setminus \Omega_\alpha; \\ \tilde{\tau}_h & \text{in } \Omega_\tau, \end{cases} \tag{3.26}$$

where u_h and $\tilde{\tau}_h$ are numerical approximations to u and τ on Ω_u and Ω_τ, respectively, we have

$$\begin{aligned} \|\tau - \tau_h\| &\le \|\tau - \tilde{\tau}_N\|_{\Omega_\alpha} + \|u - u_h\|_{\Omega_u \setminus \Omega_\alpha} + \|\tau - \tilde{\tau}_h\|_{\Omega_\tau} \\ &\le C_1\, h^N + C_2\, h^{\min(p,N)} + C_3\, h^{\min(p,N)} \\ &\le C\, h^{\min(p,N)}, \end{aligned} \tag{3.27}$$

where C_1, C_2, C_3, and C are positive constants independent of h, p, and N.

We may treat the multiplicative factorization similarly. We **emphasize** *that the above argument is not rigorous since it is difficult to prove the convergence order of nonlinear WENO schemes. Nevertheless, our numerical examples validate the above estimate.*

4. Numerical Examples: Multiplicative and Additive Factors. We use a few 2-D and 3-D examples to demonstrate the performance of new, hybrid schemes. We test two factorization cases:

- Case 1. the second-order factorization: $T \approx \tilde{T}_2 = T_2$;
- Case 2. the third-order factorization: $T \approx \tilde{T}_3 = T_2 + T_3$.

These two factorizations yield two classes of factorized eikonal equations: multiplicative factorizations based and additive factorization based eikonal equations.

For all the examples, we choose Ω_u to be the disk $\overline{B(\mathbf{x}_0, R)}$ with appropriate radius $R > 0$. We apply both first-order fast sweeping schemes and third-order WENO based Lax-Friedrichs schemes. Both the ℓ^∞ and ℓ^1 errors as well as corresponding convergence orders are computed. We choose the convergence parameter δ to be 10^{-12}. All the computation was carried out on a single AMD node at the Michigan State University High Performance Computing Center.

4.1. Example 1: A 2-D velocity of constant gradient. The setup is the following: the slowness s satisfies $\dfrac{1}{s} = \dfrac{1}{s_0} + \mathbf{g} \cdot (\mathbf{x} - \mathbf{x}_0)$, where the domain is $[0,\ 0.5]^2$, the source is $\mathbf{x}_0 = (0.25, 0.25)$, the constant gradient is $\mathbf{g} = (0, -1)$, $s_0 = 2$, the exact solution is known analytically [7], and $R = 0.05$.

Table 4.1 shows the results of solving the original eikonal equation by applying the usual fast sweeping method without any special treatment of the point source. Clearly, the convergence order is polluted first-order.

Tables 4.2 and 4.3 show the results of solving the eikonal equation by applying the hybrid first-order Godunov sweeping method and the third-order hybrid WENO Lax-Friedrichs sweeping method, where hybridity comes from solving the multiplicatively factorized eikonal equation in a small neighborhood of the source point.

Table 4.2 shows the results using the second-order multiplicative factorization for the eikonal equation in the source neighborhood of a disk of radius $R = 0.05$, where both the first-order Godunov sweeping and the third-order WENO Lax-Friedrichs sweeping methods are applied. As we can see, the first-order hybrid sweeping method yields clean first-order convergence in both ℓ^∞ and ℓ^1 norms, while the third-order

hybrid sweeping method yields second-order convergence in both ℓ^∞ and ℓ^1 norms. At least three interesting phenomena are worth pointing out. First, since a second-order multiplicative factorization is used in a neighborhood of the source, the corresponding higher order correction term can be computed to the second-order accuracy if a second or higher-order scheme is used to solve the corresponding eikonal equations, which is exactly epitomized in Table 4.2. Secondly, since a high order (higher than the first order) is no longer monotone, the number of iterations is increased significantly. Thirdly, in terms of accuracy measured by the ℓ^∞ norm on the same mesh, the numerical solution by the third-order hybrid scheme is at least two-digits more accurate than that by the first-order hybrid scheme; therefore, given an accuracy requirement, the gain in accuracy by using higher order schemes will compensate for the increase in the number of iterations.

Table 4.3 shows the results using the third-order multiplicative factorization for the eikonal equation in the source neighborhood of a disk of radius $R = 0.05$, where both the first-order Godunov sweeping and the third-order WENO Lax-Friedrichs sweeping methods are applied. As we can see, the first-order hybrid sweeping method yields clean first-order convergence in both ℓ^∞ and ℓ^1 norms, while the third-order hybrid sweeping method yields third-order convergence in both ℓ^∞ and ℓ^1 norms. Some remarks analogous to those for Table 4.2 can be made here as well.

Tables 4.4 and 4.5 show the results of solving the eikonal equation by applying the hybrid first-order Godunov sweeping method and the third-order hybrid WENO Lax-Friedrichs sweeping method, where hybridity comes from solving the additively factorized eikonal equation in a small neighborhood of the source point. As we can see, additive factorization based numerical schemes perform similarly to the multiplicative factorization based numerical schemes.

We also remark that given an accuracy requirement, a high-order scheme (higher than first order) is more efficient than a first-order scheme; this point can be appreciated from Tables 4.1, 4.2, 4.3, 4.4, and 4.5. For example, assuming that the pointwise accuracy requirement is taken to be $\epsilon = 3.5 \times 10^{-4}$ in the ℓ^∞ norm, Tables 4.1 and 4.2 indicate that when the original eikonal equation is used, the first-order scheme requires a mesh of 3201×3201 and the CPU running time of 11.52 seconds (both mesh and time obtained by extrapolation) while the third-order (actually the second-order) hybrid Lax-Friedrichs scheme only needs a mesh of 101×101 with the CPU running time of 0.64 seconds; furthermore, by taking into account the numbers of iterations and flop operations on different meshes, we can conclude that to achieve a given accuracy requirement, the overall computational cost of the third-order scheme is much less than that of the first-order scheme.

Tables 4.6 and 4.7 show the results of solving the factored equations with the hybrid schemes with $R = 0.1$. Similar results are obtained as in the case of $R = 0.05$.

4.2. Example 2: A 3-D velocity of constant gradient. The setup is the following: the slowness function s satisfies $\dfrac{1}{s} = \dfrac{1}{s_0} + \mathbf{g} \cdot (\mathbf{x} - \mathbf{x}_0)$, where the domain is $[0,\ 0.5]^3$, the source is $\mathbf{x}_0 = (0.25, 0.25, 0.25)$, the constant gradient is $\mathbf{g} = (0, -1, 0)$, $s_0 = 2$, the exact solution is known analytically [7], and $R = 0.1$.

Table 4.8 shows the results of solving the eikonal equation by applying the first-order hybrid Godunov sweeping method and the third-order hybrid WENO Lax-Friedrichs sweeping method, where hybridity comes from solving the multiplicatively factorized eikonal equation in the small neighborhood of the source of a disk of radius $R = 0.1$. The convergence order and accuracy behaviors of the hybrid schemes are

Original eikonal equation with FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.75E-2	9.87E-3	5.52E-3	3.06E-3
Order of convergence	——	0.826	0.838	0.851
l_1 Error	2.00E-3	1.16E-3	6.16E-4	3.78E-4
Order of convergence	——	0.786	0.913	0.705
# Iter	8	8	8	8
CPU time (second)	0.01	0.03	0.18	0.72

TABLE 4.1
Example 1: original eikonal equation with first-order FSM.

Multiplicatively factored eikonal equation with $R = 0.05$				
$\tilde{T}_2 = T_2$: first-order hybrid FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.12E-2	5.59E-3	2.79E-3	1.40E-3
Order of convergence	——	1.003	1.003	0.995
l_1 Error	9.04E-4	4.45E-4	2.21E-4	1.10E-4
Order of convergence	——	1.023	1.010	1.007
# Iter	14	14	14	14
CPU time (second)	0.03	0.17	0.90	3.03
$\tilde{T}_2 = T_2$: third-order hybrid Lax-Friedrichs scheme				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	2.86E-4	7.11E-5	1.77E-5	4.62E-6
Order of convergence	——	2.008	2.006	1.938
l_1 Error	4.49E-5	1.15E-5	3.04E-6	8.32E-7
Order of convergence	——	1.965	1.919	1.869
# Iter	150	246	428	786
CPU time (second)	0.64	5.05	40.04	285.38

TABLE 4.2
Example 1: hybrid schemes for multiplicatively factored eikonal equation with $\tilde{T}_2 = T_2$.

similar to those in the corresponding two-dimensional cases. The additive factorization 3-D schemes behave similarly, and we will not show results here.

5. Conclusion. We proposed to factorize the eikonal into two multiplicative or additive factors, one of which is specified to approximate the eikonal up to arbitrary order of accuracy near the source, and the other of which serves as a higher-order correction term. We have developed recursive formulas to compute the approximate eikonal up to arbitrary order of accuracy near the source. Furthermore, we have designed two types of hybrid, new, high-order fast sweeping schemes for the point-source eikonal equation. We also showed that the first-order hybrid fast sweeping schemes are monotone and consistent so that they are convergent in computing viscosity solutions. 2-D and 3-D numerical examples demonstrated that a p-th order numerical scheme yields desired, clean p-th order convergence by using a p-th order factorization.

REFERENCES

[1] S. Ahmeda, S. Bak, J. R. McLaughlin, and D. Renzi, *A third order accurate fast marching*

Multiplicatively factored eikonal equation with $R = 0.05$				
$\tilde{T}_3 = T_2 + T_3$: first-order hybrid FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.12E-2	5.59E-3	2.80E-3	1.40E-3
Order of convergence	——	1.003	0.997	1.000
l_1 Error	9.05E-4	4.45E-4	2.21E-4	1.10E-4
Order of convergence	——	1.024	1.010	1.007
# Iter	14	14	14	14
CPU time (second)	0.02	0.15	0.78	2.73
$\tilde{T}_3 = T_2 + T_3$: third-order hybrid Lax-Friedrichs scheme				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.33E-5	2.90E-6	3.76E-7	4.68E-8
Order of convergence	——	2.197	2.947	3.006
l_1 Error	1.69E-6	3.53E-7	4.53E-8	5.65E-9
Order of convergence	——	2.259	2.962	3.003
# Iter	145	231	405	758
CPU time (second)	0.62	4.53	34.78	275.47

TABLE 4.3
Example 1: hybrid schemes for multiplicatively factored eikonal equation with $\tilde{T}_3 = T_2 + T_3$.

Additively factored eikonal equation with $R = 0.05$				
$\tilde{T}_2 = T_2$: first-order hybrid FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.06E-2	5.36E-3	2.69E-3	1.35E-3
Order of convergence	——	0.984	0.995	0.995
l_1 Error	7.93E-4	3.94E-4	1.97E-4	9.82E-5
Order of convergence	——	1.009	1.000	1.004
# Iter	12	12	10	10
CPU time (second)	0.03	0.15	0.80	3.34
$\tilde{T}_2 = T_2$: third-order hybrid Lax-Friedrichs scheme				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	2.96E-4	7.40E-5	1.83E-5	4.57E-6
Order of convergence	——	2.000	2.016	2.002
l_1 Error	5.28E-5	1.31E-5	3.26E-6	8.14E-7
Order of convergence	——	2.011	2.007	2.002
# Iter	163	253	430	788
CPU time (second)	0.69	5.03	38.71	272.26

TABLE 4.4
Example 1: hybrid schemes for additively factored eikonal equation with $\tilde{T}_2 = T_2$.

method for the eikonal equation in two dimensions, SIAM J. Sci. Comput. **33** (2011), 2402–2420.

[2] S. Bak, J. R. McLaughlin, and D. Renzi, *Some improvements for the fast sweeping method*, SIAM J. Sci. Comput. **32** (2010), 2853–2874.

[3] G. Barles and P. E. Souganidis, *Convergence of approximation schemes for fully nonlinear second order equations*, Asymptotic Analysis **4** (1991), 271–283.

[4] J. D. Benamou, *An introduction to Eulerian geometrical optics (1992 - 2002)*, J. Sci. Comp. **19** (2003), 63–93.

Additively factored eikonal equation with $R = 0.05$				
$\tilde{T}_3 = T_2 + T_3$: first-order hybrid FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.04E-2	5.23E-3	2.63E-3	1.32E-3
Order of convergence	——	0.992	0.992	0.995
l_1 Error	7.94E-4	3.95E-4	1.97E-4	9.85E-5
Order of convergence	——	1.007	1.004	1.000
# Iter	12	12	10	10
CPU time (second)	0.03	0.18	0.78	2.94
$\tilde{T}_3 = T_2 + T_3$: third-order hybrid Lax-Friedrichs scheme				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.17E-5	1.89E-6	2.43E-7	3.07E-8
Order of convergence	——	2.630	2.959	2.985
l_1 Error	1.56E-6	2.59E-7	3.36E-8	4.23E-9
Order of convergence	——	2.591	2.946	2.990
# Iter	147	233	404	756
CPU time (second)	0.65	4.92	35.19	267.13

TABLE 4.5

Example 1: hybrid schemes for additively factored eikonal equation with $\tilde{T}_3 = T_2 + T_3$.

[5] J. D. Benamou, S. Luo, and H.-K. Zhao, *A Compact Upwind Second Order Scheme for the Eikonal Equation*, J. Comput. Math. **28** (2010), 489–516.

[6] M. Boué and 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.

[7] S. Fomel, S. Luo, and H.-K. Zhao, *Fast sweeping method for the factored eikonal equation*, J. Comput. Phys. **228** (2009), no. 17, 6440–6455.

[8] P. A. Gremaud and C. M. Kuster, *Computational study of fast methods for the eikonal equations*, SIAM J. Sci. Comput. **27** (2006), 1803–1816.

[9] G. S. Jiang and D. Peng, *Weighted ENO schemes for Hamilton-Jacobi equations*, SIAM J. Sci. Comput. **21** (2000), 2126–2143.

[10] C. Y. Kao, S. Osher, and J. Qian, *Lax-Friedrichs sweeping schemes for static Hamilton-Jacobi equations*, J. Comput. Phys. **196** (2004), 367–391.

[11] ———, *Legendre transform based fast sweeping methods for static Hamilton-Jacobi equations on triangulated meshes*, J. Comput. Phys. **227** (2008), 10209–10225.

[12] C. Y. Kao, S. Osher, and Y. H. Tsai, *Fast sweeping method for static Hamilton-Jacobi equations*, SIAM J. Numer. Anal. **42** (2005), 2612–2632.

[13] S. Leung and J. Qian, *An adjoint state method for three-dimensional transmission traveltime tomography using first-arrivals*, Comm. Math. Sci. **4** (2006), no. 1, 249–266.

[14] F. Li, C.-W. Shu, Y.-T. Zhang, and H.-K. Zhao, *A second-order discontinuous Galerkin fast sweeping method for eikonal equations*, J. Comp. Phys. **227** (2008), 8191–8208.

[15] S. Luo, *A uniformly second order fast sweeping method for eikonal equations*, J. Comput. Phys. **241** (2013), 104–117.

[16] S. Luo, S. Leung, and J. Qian, *An adjoint state method for numerical approximation of continuous traffic congestion equilibria*, Comm. in Computational Physics **10** (2011), 1113–1131.

[17] S. Luo and J. Qian, *Factored singularities and high-order Lax-Friedrichs sweeping schemes for point-source traveltimes and amplitudes*, J. Comput. Phys. **230** (2011), 4742–4755.

[18] S. Luo and J. Qian, *Fast sweeping methods for factored anisotropic eikonal equations: multiplicative and additive factors*, J. Sci. Comput. **52** (2012), 360–382.

[19] S. Luo, J. Qian, and H.-K. Zhao, *Higher-order schemes for 3-D first-arrival traveltimes and amplitudes*, Geophysics **77** (2012), T47–T56.

[20] S. Osher and C.-W. Shu, *High-order essentially nonoscillatory schemes for Hamilton-Jacobi equations*, SIAM J. Numer. Anal. **28** (1991), no. 4, 907–922.

[21] S. J. Osher and J. A. Sethian, *Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulations*, J. Comput. Phys. **79** (1988), 12–49.

Multiplicatively factored eikonal equation with $R = 0.1$				
$\tilde{T}_2 = T_2$: first-order hybrid FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	8.70E-3	4.34E-3	2.17E-3	1.08E-3
Order of convergence	——	1.003	1.000	1.007
l_1 Error	5.18E-4	2.58E-4	1.25E-4	6.20E-5
Order of convergence	——	1.006	1.045	1.012
# Iter	20	20	20	19
CPU time (second)	0.05	0.21	1.13	4.45
$\tilde{T}_2 = T_2$: third-order hybrid Lax-Friedrichs scheme				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	2.86E-4	7.11E-5	1.77E-5	4.60E-6
Order of convergence	——	2.008	2.006	1.944
l_1 Error	4.49E-5	1.15E-5	3.04E-6	8.33E-7
Order of convergence	——	1.965	1.919	1.868
# Iter	150	248	430	788
CPU time (second)	0.63	4.72	35.73	282.76
$\tilde{T}_3 = T_2 + T_3$: first-order hybrid FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	8.70E-3	4.34E-3	2.17E-3	1.08E-3
Order of convergence	——	1.003	1.000	1.007
l_1 Error	5.21E-4	2.54E-4	1.26E-4	6.24E-5
Order of convergence	——	1.036	1.011	1.014
# Iter	20	20	20	18
CPU time (second)	0.05	0.20	0.99	3.98
$\tilde{T}_3 = T_2 + T_3$: third-order hybrid Lax-Friedrichs scheme				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	9.27E-6	1.25E-6	1.58E-7	1.96E-8
Order of convergence	——	2.890	2.984	3.011
l_1 Error	1.22E-6	1.62E-7	2.03E-8	2.53E-9
Order of convergence	——	2.913	2.996	3.004
# Iter	143	232	405	758
CPU time (second)	0.60	4.06	35.44	257.18

TABLE 4.6
Example 1: hybrid schemes for multiplicatively factored eikonal equation.

[22] A. Pica, *Fast and accurate finite-difference solutions of the 3D eikonal equation parametrized in celerity*, 67th Ann. Internat. Mtg, Soc. of Expl. Geophys., 1997, pp. 1774–1777.

[23] J. Qian and W. W. Symes, *Paraxial eikonal solvers for anisotropic quasi-P traveltimes*, J. Comput. Phys. **173** (2001), 1–23.

[24] ———, *Adaptive finite difference method for traveltime and amplitude*, Geophysics **67** (2002), 167–176.

[25] ———, *Finite-difference quasi-P traveltimes for anisotropic media*, Geophysics **67** (2002), 147–155.

[26] ———, *Paraxial geometrical optics for quasi-P waves: theories and numerical methods*, Wave Motion **35** (2002), 205–221.

[27] J. Qian, Y.-T. Zhang, and H.-K. Zhao, *A fast sweeping methods for static convex Hamilton-Jacobi equations*, J. Sci. Comput. **31(1/2)** (2007), 237–271.

[28] ———, *Fast sweeping methods for eikonal equations on triangulated meshes*, SIAM J. Numer. Anal. **45** (2007), 83–107.

Additively factored eikonal equation with $R = 0.1$				
$\tilde{T}_2 = T_2$: first-order hybrid FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.06E-2	5.36E-3	2.69E-3	1.35E-3
Order of convergence	——	0.984	0.995	0.995
l_1 Error	7.93E-4	3.94E-4	1.97E-4	9.82E-5
Order of convergence	——	1.009	1.000	1.004
# Iter	20	20	20	20
CPU time (second)	0.06	0.21	1.27	4.92
$\tilde{T}_2 = T_2$: third-order hybrid Lax-Friedrichs scheme				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	2.95E-4	7.35E-5	1.83E-5	4.57E-6
Order of convergence	——	2.005	2.006	2.002
l_1 Error	5.27E-5	1.31E-5	3.26E-6	8.14E-7
Order of convergence	——	2.008	2.007	2.002
# Iter	163	253	430	788
CPU time (second)	0.69	5.17	38.41	276.87
$\tilde{T}_3 = T_2 + T_3$: first-order hybrid FSM				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	1.04E-2	5.23E-3	2.64E-3	1.32E-3
Order of convergence	——	0.992	0.986	1.000
l_1 Error	7.94E-4	3.95E-4	1.97E-4	9.85E-5
Order of convergence	——	1.007	1.004	1.000
# Iter	20	20	20	18
CPU time (second)	0.05	0.22	1.27	3.74
$\tilde{T}_3 = T_2 + T_3$: third-order hybrid Lax-Friedrichs scheme				
Mesh	101×101	201×201	401×401	801×801
l_∞ Error	7.89E-6	1.04E-6	1.30E-7	1.63E-8
Order of convergence	——	2.923	3.000	2.996
l_1 Error	1.04E-6	1.35E-7	1.69E-8	2.13E-9
Order of convergence	——	2.946	2.998	2.988
# Iter	147	233	404	756
CPU time (second)	0.62	4.16	31.98	255.92

TABLE 4.7
Example 1: hybrid schemes for additively factored eikonal equation.

[29] S. Serna and J. Qian, *A stopping criterion for higher-order sweeping schemes for static Hamilton-Jacobi equations*, J. Comput. Math. **28** (2010), 552–568.

[30] J. A. Sethian and A. M. Popovici, *3-D traveltime computation using the fast marching method*, Geophysics **64** (1999), 516–523.

[31] W. W. Symes and J. Qian, *A slowness matching Eulerian method for multivalued solutions of eikonal equations*, J. Sci. Comp. **19** (2003), 501–526.

[32] C. Taillandier, M. Noble, H. Chauris, and H. Calandra, *First-arrival traveltime tomography based on the adjoint-state method*, Geophysics **74** (2009), WCB1–WCB10.

[33] Y.-H. Tsai, L.-T. Cheng, S. Osher, and H.-K. Zhao, *Fast sweeping algorithms for a class of Hamilton-Jacobi equations*, SIAM J. Numer. Anal. **41** (2003), 673–694.

[34] J. N. Tsitsiklis, *Efficient algorithms for globally optimal trajectories*, IEEE Trans. Auto. Control **40** (1995), 1528–1538.

[35] J. van Trier and W. W. Symes, *Upwind finite-difference calculation of traveltimes*, Geophysics **56** (1991), 812–821.

Multiplicatively factored eikonal equation with $R = 0.1$			
$\tilde{T}_2 = T_2$: first-order hybrid FSM			
Mesh	$51 \times 51 \times 51$	$101 \times 101 \times 101$	$201 \times 201 \times 201$
l_∞ Error	2.56E-2	1.27E-2	6.35E-3
Order of convergence	——	1.011	1.000
l_1 Error	1.13E-3	5.34E-4	2.59E-4
Order of convergence	——	1.081	1.044
# Iter	24	24	24
CPU time (second)	13.05	103.29	826.87
$\tilde{T}_2 = T_2$: third-order hybrid Lax-Friedrichs scheme			
Mesh	$51 \times 51 \times 51$	$101 \times 101 \times 101$	$201 \times 201 \times 201$
l_∞ Error	1.41E-3	3.50E-4	8.70E-5
Order of convergence		2.010	2.008
l_1 Error	8.41E-5	2.12E-5	5.61E-6
Order of convergence		1.988	1.918
# Iter	223	306	518
CPU time (second)	16.66	196.03	2969.00
$\tilde{T}_3 = T_2 + T_3$: first-order hybrid FSM			
Mesh	$51 \times 51 \times 51$	$101 \times 101 \times 101$	$201 \times 201 \times 201$
l_∞ Error	2.56E-2	1.27E-2	6.35E-3
Order of convergence	——	1.011	1.000
l_1 Error	1.13E-3	5.35E-4	2.60E-4
Order of convergence	——	1.079	1.041
# Iter	24	24	24
CPU time (second)	13.27	103.98	832.34
$\tilde{T}_3 = T_2 + T_3$: third-order hybrid Lax-Friedrichs scheme			
Mesh	$51 \times 51 \times 51$	$101 \times 101 \times 101$	$201 \times 201 \times 201$
l_∞ Error	7.00E-5	8.48E-6	1.23E-6
Order of convergence	——	3.045	2.785
l_1 Error	4.31E-6	5.06E-7	7.42E-8
Order of convergence	——	3.091	2.770
# Iter	185	288	496
CPU time (second)	13.54	181.22	2806.00

TABLE 4.8
Example 2: hybrid schemes for multiplicatively factored eikonal equation.

[36] J. Vidale, *Finite-difference calculation of travel times*, Bull. Seis. Soc. Am. **78** (1988), 2062–2076.
[37] L. Zhang, J. W. Rector, and G. M. Hoversten, *Eikonal solver in the celerity domain*, Geophysical Journal International **162** (2005), 1–8.
[38] Y. T. Zhang, S. Chen, F. Li, H.-K. Zhao, and C.-W. Shu, *Uniformly accurate discontinuous Galerkin fast sweeping method for eikonal equations*, SIAM J. Sci. Comput. **33** (2011), 1873–1896.
[39] Y.-T. Zhang, H.-K. Zhao, and J. Qian, *High order fast sweeping methods for static Hamilton-Jacobi equations*, J. Sci. Comput. **29** (2006), 25–56.
[40] H.-K. Zhao, *A fast sweeping method for eikonal equations*, Math. Comput. **74** (2005), 603–627.