Journal of Scientific Computing, Vol. 19, Nos. 1–3, December 2003 (© 2003)

A Slowness Matching Eulerian Method for Multivalued Solutions of Eikonal Equations

William W. Symes[1] **and Jianliang Qian**[2]

Received August 5, 2002; accepted (in revised form) November 6, 2002

Traveltime, or geodesic distance, is locally the solution of the eikonal equation of geometric optics. However traveltime between sufficiently distant points is generically multivalued. Finite difference eikonal solvers approximate only the viscosity solution, which is the smallest value of the (multivalued) traveltime ("first arrival time"). The *slowness matching method* stitches together local single-valued eikonal solutions, approximated by a finite difference eikonal solver, to approximate all values of the traveltime. In some applications, it is reasonable to assume that geodesics (rays) have a consistent orientation, so that the eikonal equation may be viewed as an evolution equation in one of the spatial directions. This *paraxial* assumption simplifies both the efficient computation of local traveltime fields and their combination into global multivalued traveltime fields via the slowness matching algorithm. The cost of slowness matching is on the same order as that of a finite difference solver used to compute the viscosity solution, when traveltimes from many point sources are required as is typical in seismic applications. Adaptive gridding near the source point and a formally third order scheme for the paraxial eikonal combine to give second order convergence of the traveltime branches.

KEY WORDS: Hamilton–Jacobi; viscosity solution; multivalued eikonal solvers; weighted essentially nonoscillatory scheme (WENO); slowness matching.

1. INTRODUCTION

Denote by $\tau(\mathbf{x}, \mathbf{x}_s)$ the time ("traveltime") taken by a wave moving at velocity $v(\mathbf{x})$ to travel from a source point $\mathbf{x}_s \in \mathbf{R}^n$ to a target point $\mathbf{x}$. For $\mathbf{x} \neq \mathbf{x}_s$ near $\mathbf{x}_s$, τ is a differentiable function of both arguments and satisfies the *eikonal equation*

$$|\nabla_{\mathbf{x}} \tau(\mathbf{x}, \mathbf{x}_s)| = \frac{1}{v(\mathbf{x})}. \tag{1}$$

Dedicated to Stanley Osher on the occasion of his 60th birthday.

[1] The Rice Inversion Project, Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77005. E-mail: symes@caam.rice.edu

[2] Institute for Mathematics and Its Applications, University of Minnesota, 207 Church St. S.E., Minneapolis, Minnesota 55455. E-mail: qian@ima.umn.edu

0885-7474/03/1200-0501/0 © 2003 Plenum Publishing Corporation

The traveltime has a well-known relation with the rays of geometric optics: the rays are the orthogonal trajectories of the wavefronts (level sets of τ) (see for example [6]).

The traveltime solves (1) only locally, but can be computed through its evolution along the rays globally (i.e., to any point $\mathbf{x}$ which can be reached by a wave which previously passes $\mathbf{x}_s$). However it generically becomes *multivalued* when source and target points are sufficiently distant and the wave velocity $v(\mathbf{x})$ varies significantly with spatial position $\mathbf{x}$. In two dimensional space, for example, denoting by σ the variance of the velocity, source and target points more distant than $O(\sigma^{-\frac{2}{3}})$ will have at least two rays connecting them, hence at least two possible values of traveltime [50]. Generally, there is a high probability for so-called transmission caustics to occur. Beyond transmission caustics, more than one ray pass over each point in space so that the traveltime is multivalued.

One way to extract a (single valued) function of position from the multiplicity of traveltimes is to assign to each point $\mathbf{x}$ the *smallest* of the (possibly many) traveltimes from $\mathbf{x}_s$ to $\mathbf{x}$. This smallest or *first arrival* traveltime is a so-called *viscosity solution* of the eikonal equation (1) [24]. The theory of viscosity solutions of Hamilton–Jacobi equations (of which the eikonal equation is an example) was developed by Crandall, Lions, and others over the last 20 years [8, 7]. Viscosity solutions are approximately computable by monotone finite-difference schemes [24, 9, 29, 30]. Finite-difference eikonal solvers developed in [48, 47, 31, 42, 41, 20, 43] and many others owe their effectiveness to this theory. On the other hand, it also explains why these methods must compute only the first arrivals.

While some applications of geometric optics need only the first arrival traveltime, others require all traveltimes. An example of the latter type is high resolution seismic imaging via integral transform, in the presence of strong seismic refraction [15, 25, 27]. It is certainly possible in principle to compute the global, multivalued traveltime field by evolution along rays ("ray tracing"). However this "Lagrangian" approach suffers from a few drawbacks. A numerical scheme will compute (approximately) a finite number of rays; when the rays diverge, as they do in strongly refracting velocity models, some regions will be visited by many rays, others by none. Controlling the ray density to achieve roughly uniform sampling of the traveltime with expensive interpolation to prescribed output points is complicated but feasible [49].

Finite difference approximation of the eikonal equation on a regular or irregular grid automatically achieves prescribed sampling of the computed traveltime field, but produces only first arrival times, as noted above. A number of authors have proposed extensions of this "Eulerian" approach to compute the complete, multivalued traveltime [2, 3, 4, 12, 13, 44, 40, 28, 34, 14].

This paper details another Eulerian approach to multivalued traveltime computation beyond transmission caustics, which we have dubbed "slowness matching" [46]. The method patches together local single-valued solutions of the eikonal equations via Fermat's principle into a global, multivalued traveltime field, using a finite-difference eikonal solver to compute the local solutions. The cost of this computation is actually comparable to that of a single-valued, first-arrival computation, when many times must be computed from many source points along a surface, as is typical in seismic migration or tomography applications. The method also retains the other advantages of finite difference eikonal solvers mentioned above.

For some seismic applications, it is also natural to assume that traveltimes of interest are carried by rays which are subhorizontal (i.e., directed downwards) everywhere. This additional assumption permits us to replace the eikonal equation with a *paraxial eikonal equation*, which defines an evolution equation in depth for (a part of) the traveltime field. A local solution of the paraxial eikonal point source problem is identical to the local single-valued traveltime at points connected to the source point by subhorizontal rays. Local solutions of the paraxial eikonal are glued together to compute all traveltimes along subhorizontal rays (without of course explicitly computing the rays!). It is this paraxial variant of slowness matching which we explain and illustrate in this paper.

The paper is organized as follows. In Sec. 2, we describe a family of adaptive finite difference schemes for local, single-valued solutions of the eikonal equation with point sources. The adaptive eikonal solver is based on second- and third-order weighted essentially non-oscillatory (WENO) Runge–Kutta schemes developed in [30, 19], and it is also used to good effect in computing other geometric optics related quantities [36]. In Sec. 3, we illustrate the multivaluedness geometrically, explain the principle for the slowness matching Eulerian approach, and describe a numerical implementation for problems in two space dimensions. In Sec. 4, we show some numerical examples to demonstrate the accuracy of the new Eulerian method. In Sec. 5, we discuss some possible extensions and open issues, including treatment of the three dimensional problem and dropping the "paraxial" restriction to sub-horizontal rays.

2. LOCAL SINGLE-VALUED SOLUTIONS: ADAPTIVE PARAXIAL SOLVERS

We suppose throughout that $\Omega \subset \mathbf{R}^n$ is open and bounded, $v \in C^K(\Omega)$ is the (positive) *velocity*, and its reciprocal *slowness* will be denoted by $s = v^{-1}$. The theory developed here requires that $K \geqslant 3$.

Rays $(\mathbf{x}(t), \mathbf{p}(t))$ from the source point $\mathbf{x}_s \in \Omega$ are solutions of *Hamilton's equations*

$$\frac{d\mathbf{x}}{dt} = \frac{\partial H}{\partial \mathbf{p}}, \qquad \frac{d\mathbf{p}}{dt} = -\frac{\partial H}{\partial \mathbf{x}} \tag{2}$$

where $H(\mathbf{x}, \mathbf{p}) = \frac{1}{2} v^2(\mathbf{x}) \, |\mathbf{p}|^2$ is the *Hamiltonian* so that the parameter t has the dimension of time. The relevant trajectories of this system have initial conditions satisfying

$$\mathbf{x}(0) = \mathbf{x}_s,$$

$$\|\mathbf{p}(0)\| = s(\mathbf{x}_s)$$

so that one ray emanates from the source point in each direction. A real $t \geqslant 0$ is a *traveltime* from $\mathbf{x}_s$ to $\mathbf{x}$ if $\mathbf{x} = \mathbf{x}(t)$ is a solution of this system. Clearly there is no reason for traveltimes to be unique, in general: several solutions may connect $\mathbf{x}_s$ to $\mathbf{x}$. However for each $\mathbf{x}_s$ there is an open neighborhood $\Omega(\mathbf{x}_s)$ of $\mathbf{x}_s$ so that a unique ray from $\mathbf{x}_s$ to $\mathbf{x}$ exists for each $\mathbf{x} \in \Omega(\mathbf{x}_s)$; this follows from, e.g., Lemma 10.2 in [26]. Therefore traveltime is well-defined as a function of $\mathrm{x} \in \Omega(\mathbf{x}_s)$, via $\tau(\mathbf{x}(t), \mathbf{x}_s) = t$; also $\mathbf{p}(t) = \nabla_{\mathrm{x}} \tau(\mathbf{x}(t), \mathbf{x}_s)$, whence τ satisfies the eikonal equation (1) with the *point source initial condition*

$$\lim_{\mathbf{x} \to \mathbf{x}_s} \frac{\tau(\mathbf{x}, \mathbf{x}_s)}{\|\mathbf{x} - \mathbf{x}_s\|} = s(\mathbf{x}_s), \qquad \tau \geqslant 0. \tag{3}$$

See [6], Chap. II for an account of this elementary Hamilton–Jacobi theory.

We shall assume throughout that each point in Ω has at least one traveltime, i.e., is connected by at least one ray to every source point under consideration. In general, many rays will do the job, as examples presented later in the paper illustrate; it is only in a neighborhood, possibly small, of each source point that the traveltime is well defined as a function of $\mathbf{x}$.

2.1. Paraxial Eikonal Equations

The eikonal equation with the point source condition (1), (3) apparently constitutes a boundary value problem. A direct numerical attack on this problem is certainly possible [39]. However, restricting attention to traveltimes carried by rays which are oriented in a specific direction results in a subproblem which has the character of an evolution equation in a spatial coordinate. Most of the references to finite difference methods for the eikonal equation cited above make essential use of this observation, either implicitly or explicitly. Solutions to the eikonal as a spatial evolution in several directions may be combined to produce solutions of the boundary value problem (1), (3) [42, 20]. On the other hand, some applications naturally single out a particular evolution direction. For example, wave propagation in reflection seismics occurs with a generally vertical orientation. In this paper, we will adopt the point of view motivated by the reflection seismic example: that traveltimes of interest are carried by sub-horizontal rays. This subsection will explain how to modify the eikonal equation to compute accurately only traveltimes along sub-horizontal rays.

For the discussion of traveltime along sub-horizontal rays, we will restrict our attention to problems in $\mathbf{R}^2$. Accordingly we will write $\mathbf{x} = (x, z)$, $\mathbf{p} = (p, q)$ and $\mathbf{x}_s = (x_s, z_s)$ etc. "Sub-horizontal" then means "oriented in the positive z direction." Both the theory developed in this subsection and the numerical methods explained in the next two are easily extended to three spatial dimensions [20, 35, 37].

A convenient way to legislate sub-horizontal ray direction is to require that

$$\frac{\partial \tau}{\partial z} \geqslant s \cos \theta_{\max} > 0; \tag{4}$$

this is the subhorizontal condition. This inequality holds for traveltimes associated to rays making an angle $\leqslant \theta_{\max} < \frac{\pi}{2}$ with the vertical. To enforce this condition, we modify the eikonal equation to take the form of an evolution equation in depth [16]:

$$\frac{\partial \tau}{\partial z} = H\left(x, z, \frac{\partial \tau}{\partial x}\right) = \sqrt{\phi\left(s^2 - \left(\frac{\partial \tau}{\partial x}\right)^2, s^2 \cos^2 \theta_{\max}\right)}, \tag{5}$$

where ϕ is a sufficiently smooth positive function satisfying

$$\phi(x, a) = x, \qquad x \geqslant a.$$

The computations described below use the C^3 function defined for $a > 0$ by

$$\phi(x, a) = \begin{cases} \frac{1}{2} a & \text{if } \; x < 0, \\ \frac{1}{2} a + 2 \frac{x^4}{a^3} \left(1 - \frac{4}{5} \frac{x}{a}\right) & \text{if } \; 0 \leqslant x < \frac{a}{2}, \\ x + 2 \frac{(x-a)^4}{a^3} \left(1 + \frac{4}{5} \frac{x-a}{a}\right) & \text{if } \; \frac{a}{2} \leqslant x < a, \\ x & \text{if } \; x \geqslant a. \end{cases}$$

This choice of ϕ is compatible with truncation error analysis for difference schemes of up to 3rd order accuracy [33].

The remainder of this subsection presents a proof of the following assertions:

Theorem 2.1. For a suitable subset $\Omega^p(x_s, z_s) \subset \Omega(x_s, z_s)$ and an appropriate initial condition compatible with a point source located at (x_s, z_s), the solution $\tau^p(x, z, x_s, z_s)$ of (5) is identical to the traveltime $\tau(x, z, x_s, z_s)$ at $(x, z) \in \Omega^p(x_s, z_s) \subset \Omega(x_s, z_s)$ provided that the unique ray connecting the source point (x_s, z_s) to (x, z) makes an angle $\leqslant \theta_{\max} < \frac{\pi}{2}$ with the vertical *at every point*. Moreover, a point (x, z) satisfies this condition if the characteristic or *paraxial ray* for (5) through (x, z), selected by the method of characteristics, also makes an angle $\leqslant \theta_{\max}$ with the vertical, in which case it is a ray for the eikonal equation.

Remark. Thus satisfaction of the subhorizontal condition can be monitored during solution of (5). Near other points, τ has the character of a plane wave, and does not approximate the traveltime—see Fig. 3, also [38].

Remark. To generate an appropriate initial condition compatible with a point source located at (x_s, z_s), we may assume that the velocity near the source is constant and equal to the velocity at the source; see [36] for theoretical justification of such an assumption and Sec. 2.3 for more implementation details.

Proof. Note first that rays $(\mathbf{x}(t), \mathbf{p}(t))$ associated to traveltimes satisfying (4) along their entire length may be parameterized by depth z rather than time t, where $\mathbf{x}(t) = (x(t), z(t))$ and $\mathbf{p}(t) = (p(t), q(t))$. The horizontal coordinate $x(z, \theta_0)$ and the angle with the vertical $\theta(z, \theta_0) = \arcsin(v(x(z, \theta_0), z)\, p(z, \theta_0))$ completely determine the ray trajectory, and form a solution of the system

$$\frac{dx}{dz} = \tan\theta \tag{6}$$

$$\frac{d\theta}{dz} = \frac{1}{s}\left(\frac{\partial s}{\partial z} - \frac{\partial s}{\partial x}\sin\theta\right) \tag{7}$$

with initial data

$$x(z_s, \theta_0) = x_s, \qquad \theta(z_s, \theta_0) = \theta_0 .$$

[The angle θ_0 is the *takeoff angle* of the ray.] Here $|\theta_0| < \frac{\pi}{2}$. A calculation verifies that the scaled trajectory

$$x_\epsilon(z, \theta_0) = \frac{1}{\epsilon}\, x(\epsilon z, \theta_0), \qquad \theta_\epsilon(z, \theta_0) = \theta(\epsilon z, \theta_0)$$

satisfies (7) with $s(x, z)$ replaced by the scaled slowness $s_\epsilon(x, z) = s(\epsilon x, \epsilon z)$, and with the same initial conditions. As $\epsilon \to 0$, s_ϵ approaches the constant function $s_0(x, z) = s(x_s, z_s)$ in the C^k sense for any $k \leqslant K$. Therefore the right hand side of (7) converges to the constant-slowness right hand side, whence the solution converges to the constant-slowness solution

$$x_0(z, \theta_0) = x_s + (z - z_s)\tan\theta_0, \qquad \theta(z, \theta_0) = \theta_0$$

uniformly in compact sets of parameters. Note that $x_0(1, \theta_0)$ is monotone as a function of θ_0. Define

$$\theta^1_{\max} = \frac{\pi}{4} + \frac{\theta_{\max}}{2} .$$

For sufficiently small ϵ, $x_\epsilon(1, \theta_0)$ is monotone as a function of $\theta_0 \in [-\theta^1_{\max}, \theta^1_{\max}]$. Fix such an ϵ, and set

$$\begin{aligned} x_- &= \epsilon x_\epsilon(1, -\theta_{\max}) \\ x_+ &= \epsilon x_\epsilon(1, \theta_{\max}) \\ x^1_- &= \epsilon x_\epsilon(1, -\theta^1_{\max}) \\ x^1_+ &= \epsilon x_\epsilon(1, \theta^1_{\max}). \end{aligned}$$

It follows that

$$\theta_0 \mapsto x(\epsilon, \theta_0) \quad \text{is monotone on} \quad [-\theta^1_{\max}, \theta^1_{\max}] \tag{8}$$

by the scaling relation. Choose a cut-off function, $\psi \in C^\infty(\mathbf{R})$ with (1) $\psi(x)=1$, $x_- \leqslant x \leqslant x_+$, (2) $\psi(x)=0$, $x \leqslant x_-^1$ or $x \geqslant x_+^1$, and (3) $\psi'(x) \leqslant 0$, $x_+ \leqslant x \leqslant x_+^1$ and $\psi'(x) \geqslant 0$, $x_-^1 \leqslant x \leqslant x_-$. Choose positive A so that

$$A > \max\left(\sin\theta_{\max}^1 \sup_{(x,z)\in\Omega} s(x,z), \sup_{x_-^1 \leqslant x \leqslant x_-, x_+ \leqslant x \leqslant x_+^1} \frac{\tau(x, z_s+\epsilon, x_s, z_s)}{|x-x_s|}\right)$$

and set

$$\tau_0^p(x, x_s) = \psi(x)\,\tau(x, z_s+\epsilon, x_s, z_s) + A(1-\psi(x))\,|x-x_s|. \tag{9}$$

It is easy to check that

$$\left|\frac{\partial\tau_0^p(x, x_s)}{\partial x}\right| > s(x, z_s+\epsilon)\sin\theta_{\max} \qquad \text{if} \quad x > x_+ \quad \text{or} \quad x < x_-. \tag{10}$$

Now define τ^p to be the solution of (5) with initial data τ_0^p on $\{(x,z): z = z_s+\epsilon\} \cap \Omega(x_s, z_s)$.

For ϵ possibly smaller still, the method of characteristics yields a smooth solution of the Hamilton–Jacobi initial value problem (5), (9) in $\Omega^p(x_s, z_s) \equiv \Omega(x_s, z_s) \cap \{(x,z): z_s \leqslant z \leqslant z_s+\epsilon\}$.

Assume that (x, z) lies on a ray through (x_s, z_s) always making an angle $\leqslant \theta_{\max}$ with the vertical. In particular the takeoff angle θ_0 of this ray satisfies $|\theta_0| \leqslant \theta_{\max}$, so (8) implies that the ray crosses $z = z_s+\epsilon$ at $x = x_c$ with $x_- \leqslant x_c \leqslant x_+$. It follows immediately from the definition (5) that this ray is also characteristic for the paraxial eikonal, that the initial data on $z = z_s+\epsilon$ is the same as that for the traveltime field in a neighborhood of x_c, that the ray-tracing constructions of $\tau^p(x, z, x_s, z_s)$ and the traveltime at (x, z) are exactly the same, and that therefore $\tau^p(x, z, x_s, z_s)$ is actually the traveltime at (x, z).

Finally, suppose that (x, z) lies on a characteristic or paraxial ray for (5), constructed by the method of characteristics for initial condition (9), and that this paraxial ray makes an angle $\leqslant \theta_{\max}$ with the vertical at every point. [This condition can be monitored implicitly during the finite difference solution of (5).] Then the ray must meet $z = z_s+\epsilon$ at $x = x_c$ with $x_- \leqslant x_c \leqslant x_+$, since otherwise the method of characteristics together with the construction of the initial condition (9), (10) implies that the angle constraint is violated at the intersection of the paraxial ray with $z = z_s+\epsilon$. However then the paraxial ray satisfies the same initial conditions as does the ray for the eikonal equation, and also the same Hamiltonian system. Therefore the paraxial ray is a ray, and τ^p and τ are the same along it, as claimed.

□

2.2. WENO Runge–Kutta Schemes

A number of finite difference methods for Hamilton–Jacobi equations are mentioned in the introduction; others include [21, 22, 23, 18]. The work reported here is based on a class of high-order finite difference methods, so called essentially non-oscillatory schemes and their relatives, developed in [29, 30, 19].

For a function f of the space variable $\mathbf{x} = (x, z)$ in the computational domain, we write

$$f_i^k = f(x_i, z_k),$$
$$(x_i, z_k) = (x_{\min} + (i-1)\,\Delta x,\ z_{\min} + (k-1)\,\Delta z).$$

Let

$$\tau_i^k = \tau(x_i, z_k; x_s, z_s)$$

and define the forward D^+ and backward D^- finite-difference operators

$$D_x^{\pm}\tau_i^k = \frac{\pm[\tau_{i\pm1}^k - \tau_i^k]}{\Delta x}.$$

The simplest, first-order method of the family results from replacing the z-derivative in the eikonal equation by the forward difference, and the x-derivative by the *upwind* choice of first order one-sided differences:

$$\frac{1}{\Delta z}(\tau_j^{k+1} - \tau_j^k) = H(\hat{D}_x^1) \tag{11}$$

where

$$\hat{D}_x^1\tau = \text{modmax}(\max(D_x^-\tau, 0), \min(D_x^+\tau, 0)).$$

Here the modmax function returns the larger value in modulus. This scheme chooses the $D^+\tau$ to represent τ_x if both possibilities are negative, and $D^-\tau$ if both possibilities are positive. That is, the formula takes information from the direction in which the rays are coming if that direction is signaled unambiguously by the signs of both first differences. If the signs differ, one of two compromises is made. This scheme can be shown to be monotonic, have l^∞ contraction property and satisfy maximum principle. Thus it enjoys only first-order accuracy [9]. To obtain high order accurate eikonal solvers, we may use high-order ENO and WENO schemes.

The second-order ENO refinement of $D_x^{\pm}\tau$ is

$$D_x^{\pm,2}\tau_i = D_x^{\pm}\tau_i \mp \tfrac{1}{2}\,\Delta x\, m(D_x^{\pm}D_x^{\pm}\tau_i, D_x^-D_x^+\tau_i), \tag{12}$$

where

$$m(x, y) = \min(\max(x, 0), \max(y, 0)) + \max(\min(x, 0), \min(y, 0)).$$

The WENO third-order schemes for $D_x^{\pm}\tau_i$ are

$$\begin{aligned} D_x^{\pm \mathrm{W},3}\tau_i = {} & \tfrac{1}{12}(-D_x^+\tau_{i-2} + 7D_x^+\tau_{i-1} + 7D_x^+\tau_i - D_x^+\tau_{i+1}) \\ & \pm \Delta x\, \Phi^{\mathrm{W}}(D_x^-D_x^+\tau_{i\pm2}, D_x^-D_x^+\tau_{i\pm1}, D_x^-D_x^+\tau_i, D_x^-D_x^+\tau_{i\mp1}), \end{aligned}$$

where Φ^{W} is smooth and is given by

$$\Phi^{\mathrm{W}}(a,b,c,d)=\tfrac{1}{3}\,w_0(a-2b+c)+\tfrac{1}{6}\,(w_2-\tfrac{1}{2})(b-2c+d)$$

with weights defined as

$$\begin{aligned}
&w_0=\frac{\alpha_0}{\alpha_0+\alpha_1+\alpha_2}, \qquad w_2=\frac{\alpha_2}{\alpha_0+\alpha_1+\alpha_2},\\
&\alpha_0=\frac{1}{(\delta+\beta_0)^2}, \qquad \alpha_1=\frac{1}{(\delta+\beta_1)^2}, \qquad \alpha_2=\frac{1}{(\delta+\beta_2)^2},\\
&\beta_0=13(a-b)^2+3(a-3b)^2,\\
&\beta_1=13(b-c)^2+3(b+c)^2,\\
&\beta_2=13(c-d)^2+3(3c-d)^2.
\end{aligned}$$

In the denominators above, we added a small positive number δ to avoid dividing by zero. In the computation, δ is chosen to be 10^{-6}; in practice, the solution is not sensitive to the choice of δ.

The upwind second-order ENO approximation for $\frac{\partial\tau}{\partial x}$ is

$$\hat{D}_x^2\tau=\operatorname{modmax}(\max(D_x^{-,2}\tau,0),\min(D_x^{+,2}\tau,0));$$

similarly, the upwind third-order WENO approximation for $\frac{\partial\tau}{\partial x}$ is

$$\hat{D}_x^{\mathrm{W},3}\tau=\operatorname{modmax}(\max(D_x^{-W,3}\tau,0),\min(D_x^{+W,3}\tau,0)).$$

To maintain high-order accuracy, these formulae must be combined with a z-step of comparable accuracy. Here we use Runge–Kutta schemes as suggested in [30]. Thus the second- and third-order (W)ENO Runge–Kutta steps are

$$\begin{aligned}
&\delta_2^1\tau=\Delta z\,H(\hat{D}_x^2\tau),\\
&\delta_2^2\tau=\tfrac{1}{2}\,(\delta_2^1\tau+\Delta z\,H(\hat{D}_x^2(\tau+\delta_2^1\tau))),
\end{aligned} \tag{13}$$

and

$$\begin{aligned}
&\delta_3^1\tau=\Delta z\,H(\hat{D}_x^{\mathrm{W},3}\tau),\\
&\delta_3^2\tau=\tfrac{1}{4}\,(\delta_3^1\tau+\Delta z\,H(\hat{D}_x^{\mathrm{W},3}(\tau+\delta_3^1\tau))),\\
&\delta_3^3\tau=\tfrac{1}{3}\,(2\delta_3^2\tau+2\,\Delta z\,H(\hat{D}_x^{\mathrm{W},3}(\tau+\delta_3^2\tau))).
\end{aligned} \tag{14}$$

The depth step Δz must satisfy the stability condition

$$\Delta z\leqslant\Delta z_{\mathrm{cfl}}=\frac{\Delta x}{\tan(\theta_{\max})}.$$

We have typically chosen $\Delta z=0.9\Delta z_{\mathrm{cfl}}$.

The nth order scheme is then

$$\tau^{k+1} = \tau^k + \delta_n^n \tau^k \tag{15}$$

for $k = 0, 1, 2, \ldots$.

2.3. Adaptive Refinements for a Point Source

The traveltime field for the eikonal equation with a point source has an upwind singularity at the source point. This singularity generates a first-order error which propagates throughout the grid, regardless of the order of the scheme. This source-generated error negates the accuracy advantages of high-order schemes so that the computed take-off slowness and its derivatives have zeroth-order accuracy only; however, these quantities are important in the algorithm described below (and in the computation of geometric optics amplitude). Therefore, we sought another approach, that is, an adaptive gridding approach. The adaptive gridding approach was proposed in [36] which treats this singularity efficiently and achieves prescribed accuracy at a far lower cost than does a fixed grid method. To be complete, we summarize the idea of the method here.

The usual step adjustment in ODE solvers would change Δz by a factor computed from the asymptotic form of the truncation error [45, 49]. This is impractical for a PDE application because it would require an arbitrary adjustment of the spatial grid (i.e., the x-grid in the difference scheme) and, therefore, expensive interpolation. Scaling Δz by a factor of two, however, implies that the stability may be maintained by scaling Δx by the same factor. For coarsening, this means simply throwing out every other grid point, i.e., no interpolation at all, which dramatically reduces the floating point operations required. This bisection refinement is similar to the adaptive mesh refinement (AMR) proposed by Berger and Oliger [5], but we refine or coarsen the mesh globally rather than locally. Since the typical behavior of the traveltime field is to become smoother as one moves away from the source, the truncation errors tend in general to decrease. Therefore, most of the grid adjustments are coarsening and very little or no interpolation is required. Since the slowness field comes to us in gridded form, an interpolation is always required to supply estimates of slowness at the points appearing in the WENO-Runge–Kutta formula. We use a local quadratic interpolation in x and z because the third-order accuracy of which is compatible with that of the difference scheme. For traveltimes, we use a similar quadratic interpolation.

To initialize our algorithm, the user supplies a local error tolerance ϵ; σ_1 and σ_2 are two user-defined positive functions of ϵ which are used to control the coarsening and refinement. We use the 2nd and 3rd order eikonal solvers (Eqs. (13) and (14)) and estimate the truncation error of the 2nd-order scheme as $e_2 = \max |\delta_2^2 \tau - \delta_3^3 \tau|$ over the current depth. So long as $\sigma_1(\epsilon) \leqslant e_2 \leqslant \sigma_2(\epsilon)$ at every point of the current depth level, we simply proceed to the next step. When $e_2 < \sigma_1(\epsilon)$, we increase the step by a factor of two, i.e., $\Delta z \leftarrow 2\Delta z$, and we recompute the τ update and e_2. Similarly, when $e_2 > \sigma_2(\epsilon)$, we decrease the step by a factor of two. As soon as the local error is once again within the tolerance interval, we continue depth-stepping.

A very important point is that we retain the 3rd-order (a more accurate one) computation of τ at the end of each depth step as the actual update, discarding the 2nd-order computation, which is used only in step control.

Since the traveltime field is non-smooth at the source point, the truncation error analysis on which the adaptive step selection criterion is based is not valid there. Therefore, it is necessary to produce a smooth initial traveltime field. We do this by estimating the largest $z_{\text{init}} > 0$ at which the constant velocity traveltime is in error by less than $\sigma_2(\epsilon)$. Details of the z_{init} calculation are given in [36]. Having initialized τ at z_{init}, the algorithm invokes adaptive gridding. Since z_{init} is quite small, τ changes rapidly, resulting in a large number of grid refinements at the outset. However, no interpolation is performed, as τ is given analytically on $z = z_{\text{init}}$. This initially very fine grid is rapidly coarsened as the depth stepping proceeds.

In our current implementation, we maintain a data structure for the computational grid that is independent of the output grid; the desired quantities are calculated on the computational grid and interpolated back to the output grid. As a safeguard against pathological program behaviors, we specify a maximum number of permitted grid refinements.

To avoid unnecessary computations, we update τ only within the triangle

$$\{(x, z): |x - x_s| \leqslant |z - z_s| \tan \theta_{\max}\}.$$

All rays with takeoff angles less than $\theta_{\max}$ must lie inside this triangle, and it is only along such rays that the paraxial eikonal equation produces correct first-arrival times. Output points outside the triangle are assigned a very large number so that constructed ray paths will never reach those places.

Note that rays must point outward at the boundary of the computational domain for this scheme to succeed: otherwise, traveltimes inside depend on (unknown) traveltimes outside. A robust implementation monitors the sign of τ_x at the boundary.

This scheme gives extremely accurate results in smooth models. Many tests with smooth migration velocity models have shown errors of $\ll 1\%$ on modest grids, when compared with accurate ray tracing. It can be combined with similar finite difference transport solvers to give asymptotic Green's functions for modeling and migration [36].

In common with standard ODE adaptive timestepping methods, this adaptive eikonal solver is not optimal as the a posteriori error estimators on which it is based are only asymptotically accurate. To obtain an optimal adaptive eikonal solver, or in general, an optimal adaptive method for Hamilton–Jacobi equations, one has to use a posteriori error estimators, such as those developed in [1].

3. GLOBAL MULTI-VALUED SOLUTIONS: SLOWNESS MATCHING

3.1. Illustration of Multi-Valued Solutions

The sinusoidal waveguide velocity field depicted in Fig. 1 is motivated by a computation in [49]. The velocity field is given by

$$v(x, z) = 1 + 0.2 \sin 0.5\pi z \sin 3\pi x.$$

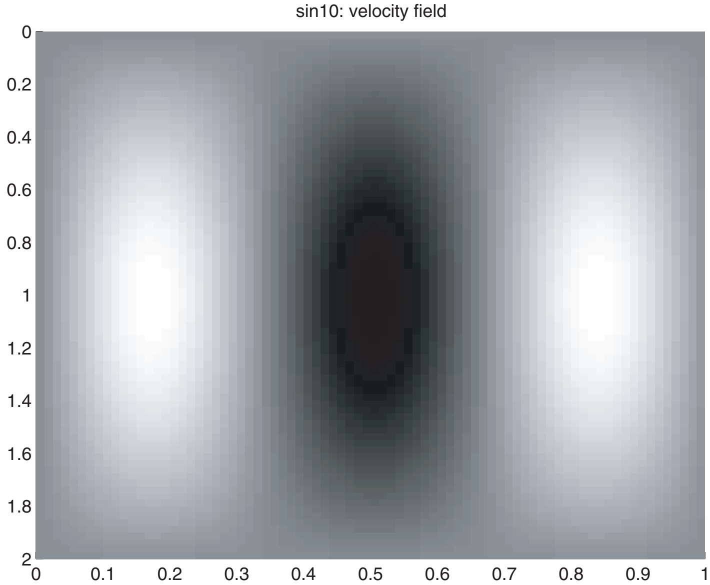

Fig. 1. Grey scale plot of velocity used in traveltime computations (product of sinusoids). Maximum excursion is approximately 30% from mean. Horizontal and vertical axes in km.

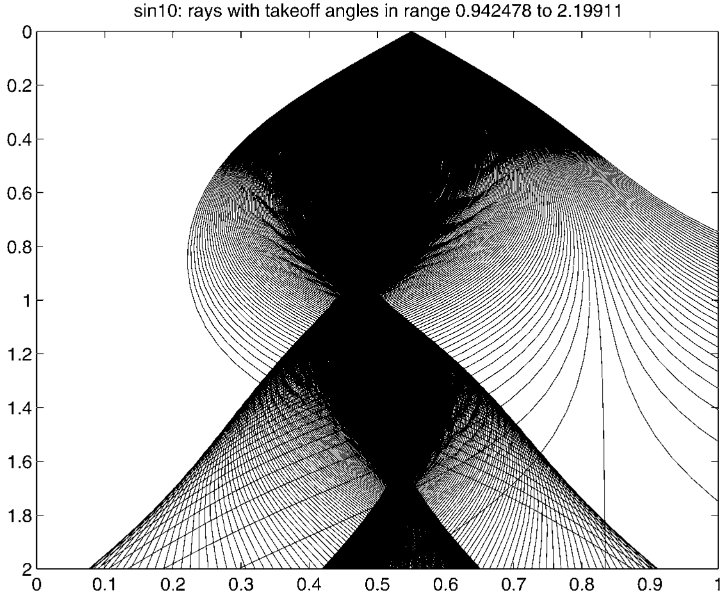

Fig. 2. Rays traced through model of Fig. 1. Note that five rays pass over some points at bottom. Caustics (ray envelopes) are clearly visible.

Figure 2 shows rays traced from a source point (0.55, 0) near the center of the top of the model. Slow regions form lenses and create crossing rays, imperfect foci, and caustics (ray envelopes). Clearly one cannot assign single-valued traveltimes globally in this model.

The method explained in the last section computes the first-arrival time field with no difficulty; contours of this field appear as Fig. 3. However much has been missed: in particular the most energetic parts of the wave field emanating from the point source are actually carried on later arriving convergent bundles of rays. [15] discusses the serious consequences of omitting these arrivals from consideration in migration of reflection data, for instance.

Thus the challenge is to devise an algorithm which retains the efficiency of modern first arrival solvers and at the same time compute all arrival traveltimes. To arrive at such an algorithm, we introduce the *slowness matching principle*.

3.2. Slowness Matching Principle

Suppose that $P = \{(\mathbf{x}(t'), \mathbf{p}(t')): 0 \leqslant t' \leqslant t\}$ is a ray connecting $\mathbf{x}_s \in \Omega$ to $\mathbf{x} \in \Omega$, i.e., $\mathbf{x}(0) = \mathbf{x}_s$ and $\mathbf{x}(t) = \mathbf{x}$. The *takeoff slowness vector* of P is the initial value $\mathbf{p}_s = \mathbf{p}(0)$, and the *arrival slowness vector* is the final value $\mathbf{p} = \mathbf{p}(t)$.

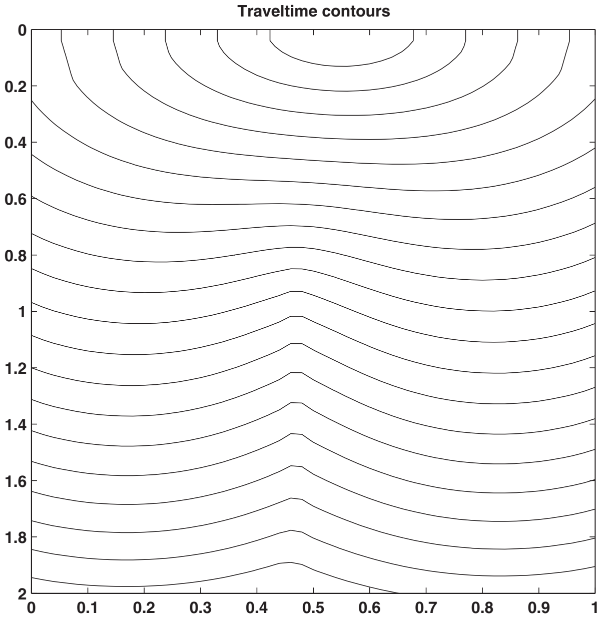

Fig. 3. Traveltime contours computed with WENO adaptive paraxial eikonal solver. Note that contours are planar outside of aperture (near top, sides of figure), where rays subtend angles $> \theta_{\max}$ with vertical and solution does not approximate traveltime.

Recall from Sec. 2 that under standing assumptions ($v \in C^K(\Omega)$, $K \geqslant 3$, and strictly positive), for every $\mathbf{x}_s$ there is a neighborhood $\Omega(\mathbf{x}_s)$ so that for each $\mathbf{x} \in \Omega(\mathbf{x}_s) - \{\mathbf{x}_s\}$ there exists a unique ray connecting $\mathbf{x}_s$ and $\mathbf{x}$ lying within $\Omega(\mathbf{x}_s)$. Also the traveltime $\tau(\cdot, \mathbf{x}_s) \in C^2(\Omega(\mathbf{x}_s) - \{\mathbf{x}_s\})$ is simply the time along the unique ray connecting $\mathbf{x}_s$ and $\mathbf{x}$, and satisfies the eikonal equation (1) in $\Omega(\mathbf{x}_s) - \{\mathbf{x}_s\}$. A kind of reciprocity principle is easy to establish: if $\mathbf{x} \in \Omega(\mathbf{x}_s)$ then $\mathbf{x}_s \in \Omega(\mathbf{x})$ and $\tau(\mathbf{x}_s, \mathbf{x}) = \tau(\mathbf{x}, \mathbf{x}_s)$. Therefore, the arrival slowness for the ray from $\mathbf{x}_s$ to $\mathbf{x}$ is $\nabla_{\mathbf{x}} \tau(\mathbf{x}_s, \mathbf{x})$, and the takeoff slowness for the ray is $-\nabla_{\mathbf{x}_s} \tau(\mathbf{x}, \mathbf{x}_s) = -\nabla_{\mathbf{x}_s} \tau(\mathbf{x}_s, \mathbf{x})$ when $\mathbf{x} \neq \mathbf{x}_s$. This basically says that arrival and takeoff slowness vectors can be represented by gradients of traveltimes with respect to either current point $\mathbf{x}$ or source point $\mathbf{x}_s$.

Traveltimes are *additive*, in this sense:

Theorem 3.1. If t_0 is the traveltime from $\mathbf{x}_0$ to $\mathbf{x}$ along ray $P_0 = \{(\mathbf{x}_0(t), \mathbf{p}_0(t)) \colon 0 \leqslant t \leqslant t_0\}$, and t_1 is the traveltime from $\mathbf{x}$ to $\mathbf{x}_1$ along ray $P_1 = \{(\mathbf{x}_1(t), \mathbf{p}_1(t)) \colon 0 \leqslant t \leqslant t_1\}$, then $t_0 + t_1$ is a traveltime from $\mathbf{x}_0$ to $\mathbf{x}_1$ provided that the arrival slowness vector for P_0 is equal to the takeoff slowness vector for P_1. In that case the concatenation of P_0 and P_1 is the ray for $t_0 + t_1$; its takeoff slowness vector is the takeoff slowness vector of P_0, and its arrival slowness vector is the arrival slowness vector of P_1.

Proof. The equality of arrival slowness for P_0 and takeoff slowness for P_1 is simply the statement that $\mathbf{p}_0(t_0) = \mathbf{p}_1(0)$. Since $\mathbf{x}$ is the final point of P_0 and the starting point of P_1, i.e., $\mathbf{x}_0(t_1) = \mathbf{x}_1(0) = \mathbf{x}$, the assumptions imply that the data for P_0 at $t = t_1$ coincide with the data for P_1 at $t = 0$. It follows that the concatenation

$$P_2 = \{(\mathbf{x}_2(t), \mathbf{p}_2(t)) \colon 0 \leqslant t \leqslant t_0 + t_1\}$$

defined by $\mathbf{x}_2(t) = \mathbf{x}_0(t)$ if $0 \leqslant t \leqslant t_0$, else $\mathbf{x}_2(t) = \mathbf{x}_1(t - t_0)$, similarly for $\mathbf{p}_2$, is a C^1 curve and a solution of Hamilton's equations, with initial and final data as stated. □

Corollary 3.2. Suppose that

1. t is a traveltime from $\mathbf{x}_0$ to $\mathbf{x}$ with takeoff slowness vector $\mathbf{p}_0$ and arrival slowness vector $\mathbf{p}$;
2. $\mathbf{x}_1 \in \Omega(\mathbf{x})$ and $\tau = \tau(\cdot, \mathbf{x})$ is the local smooth solution of the eikonal equation in $\Omega(\mathbf{x})$ with source at $\mathbf{x}$;
3. $\mathbf{p} + \nabla_{\mathbf{x}} \tau(\mathbf{x}_1, \mathbf{x}) = \mathbf{0}$ (slowness matching condition).

Then $t + \tau(\mathbf{x}, \mathbf{x}_1)$ is a traveltime from $\mathbf{x}_0$ to $\mathbf{x}_1$ with takeoff slowness vector $\mathbf{p}_0$ and arrival slowness vector $\mathbf{p}_1 = \nabla_{\mathbf{x}_1} \tau(\mathbf{x}_1, \mathbf{x})$.

Conversely, suppose that t_1 is a traveltime from $\mathbf{x}_0$ to $\mathbf{x}_1$ with associated ray P_1, that $(\mathbf{x}, \mathbf{p})$ is a point on P_1 with $\mathbf{x} \in \Omega(\mathbf{x}_1)$ (or equally well $\mathbf{x}_1 \in \Omega(\mathbf{x})$), and that t is the time along P_1 from $\mathbf{x}_0$ to $\mathbf{x}$. Then $t_1 = t + \tau(\mathbf{x}, \mathbf{x}_1)$ and the slowness matching condition (3) holds.

Proof. The first part follows immediately from the theorem. For the second, note that $\mathbf{x} \in \Omega(\mathbf{x}_1)$ implies that only one ray connects $\mathbf{x}$ and $\mathbf{x}_1$ in $\Omega(\mathbf{x}_1)$, and its time is $\tau(\mathbf{x}_1, \mathbf{x})$, its slowness given by $\nabla_{\mathbf{x}} \tau(\mathbf{x}_1, \mathbf{x})$. □

The condition (3) in the corollary means the arrival slowness matches the takeoff slowness at the point $\mathbf{x}$, hence the name *slowness matching condition* [46]. The point $\mathbf{x}$ is the *slowness matching point*.

The computations to be presented in the next section use a version of the slowness matching principle (Corollary 3.2) adapted to the paraxial eikonal and the sub-horizontal assumption. Recall that solutions of the paraxial eikonal equation are identical to solutions of the eikonal equation, and thus with the traveltime, at points connected to the source point by unique sub-horizontal rays (Theorem 2.1). As before, we state this modified slowness matching principle explicitly for 2D problems. Generalization to 3D is straightforward.

Corollary 3.3. Suppose that $0 \leqslant \theta_{\max} < \frac{\pi}{2}$, and

1. t is a traveltime from (x_0, z_0) to (x, z) with takeoff slowness vector (p_0, q_0) and arrival slowness vector (p, q);
2. the ray $\{(x(t'), z(t'), p(t'), q(t')) \colon 0 \leqslant t' \leqslant t\}$ associated with t is *downgoing*, i.e., $|q(t')| \geqslant s(x(t'), z(t')) \cos \theta_{\max}$, $0 \leqslant t' \leqslant t$;
3. $(x_1, z_1) \in \Omega^p(x, z)$ and $\tau^p = \tau^p(\cdot, x, z)$ is the local smooth solution of the paraxial eikonal equation in $\Omega^p(x, z)$ with source at (x, z);
4. the (unique) ray connecting (x, z) and (x_1, z_1) is downgoing;
5. the slowness matching condition holds at the point (x, z):

$$p = -\frac{\partial \tau^p}{\partial x}(x_1, z_1, x, z). \tag{16}$$

Then $t + \tau^p(x_1, z_1, x, z)$ is a traveltime from (x_0, z_0) to (x_1, z_1) with takeoff slowness vector (p_0, q_0) and arrival slowness vector

$$p_1 = \frac{\partial \tau^p}{\partial x_1}(x_1, z_1, x, z), \qquad q_1 = \sqrt{s(x_1, z_1)^2 - p_1^2}.$$

Conversely, suppose that t_1 is a traveltime from (x_0, z_0) to (x_1, z_1) with associated ray P_1, that P_1 is downgoing in the sense specified above, that (x, z, p, q) is a point on P_1 with $(x_1, z_1) \in \Omega^p(x, z)$, and that t is the time along P_1 from (x_0, z_0) to (x, z). Then $t_1 = t + \tau^p(x_1, z_1, x, z)$ and the slowness matching condition (16) holds.

Proof. Follows directly from Theorem 2.1 and Corollary 3.2. □

The above corollary leads us to apply slowness matching principle at some intermediate depth z, which fits well with the evolution in depth of the paraxial eikonal solver that we have already proposed. The key point is this: even when the traveltime is single-valued above and below the intermediate depth z, *there may be more than one solution of the slowness matching condition* at z, leading to multiple traveltimes from (x_0, z_0) to (x_1, z_1). Moreover, *all traveltimes along sub-horizontal (downgoing) rays may be found in this way.* To make the traveltime single-valued, we may choose the depth strip to be small enough as we argued above. Figure 4 illustrates these contentions: it is possible that there are two rays from (x_0, z_0) to (x_1, z_1) through the depth z, and they must both satisfy the slowness matching

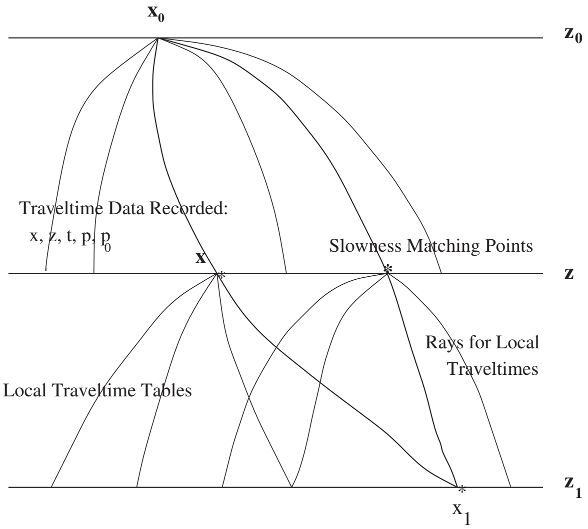

Fig. 4. Illustration of slowness matching principle: when $t+\tau(x, z; x_1, z_1)$ is a traveltime.

condition (16) at depth z. Moreover, each solution of (16) for x-coordinate is necessarily the location at which a ray crosses at depth z. Therefore one finds in this way all downgoing rays from (x_0, z_0) to (x_1, z_1) and their associated traveltimes.

This procedure applies recursively by dropping the requirement that the traveltime in the depth interval from z_0 to z be single valued. It only requires that the horizontal ray slowness p be available at each point on depth z, and this function may be multivalued which renders a dynamic data structure to be necessary. In this way, the slowness matching procedure extends a (possibly multivalued) traveltime field through the depth interval from z to z_1.

3.3. Numerical Implementation

Corollary 3.3 suggests an obvious recursive algorithm to determine all traveltimes along downgoing rays, requiring only the ability to compute local single-valued traveltime fields in perhaps narrow depth strips. These traveltime fields need only be accurate at points along downgoing rays. A local paraxial eikonal solver of the type explained above provides such aperture-limited traveltime fields with an optimal operation count, i.e., the number of floating point operations required is proportional to the number of gridpoints visited. Numerical solution of the slowness matching condition can be done by enumerating zero crossings of

$$p+\frac{\partial\tau^p}{\partial x}(x, z; x_1, z_1)$$

and interpolating or by any other convenient root finding technique. This completes the algorithm.

Algorithm 1.

- Initialization:
  - Choose δz so that the modified eikonal equation with an appropriate initial datum on $z = z_0$ has smooth solution for $z_0 \leqslant z \leqslant z_0 + \delta z$ where δz can be either an *a priori* estimate or a guess via trial and error.
  - Set $z_n = z_0 + n\delta z$, $n = 0, 1, 2, \ldots$: slowness matching depths. Choose step Δx for grid on slowness matching depths and enumerate grid points x_n^m, $m = 1, \ldots, N$.
  - Solve the modified eikonal equation (5) with the appropriate initial datum at $z = z_0$ in $z_0 \leqslant z \leqslant z_1$. Record the following data

    $$x(1, m) = x_1^m, \qquad t(1, m) = \tau,$$

    $$p(1, m) = \frac{\partial \tau}{\partial x}(x_1^m, z_1; x_0, z_0), \qquad p_0(1, m) = -\frac{\partial \tau}{\partial x_0}(x_1^m, z_1; x_0, z_0)$$

    evaluated at $(x_0, z_0; x_1^m, z_1)$ in a linked list, in order of increasing p_0.
- Repeated steps for depth n. At depth n, we have list

  $$\{x(n, \cdot), t(n, \cdot), p(n, \cdot), p_0(n, \cdot)\}.$$

  To advance to depth $n+1$:
  - For each grid point x_n^l, compute local traveltime table

    $$\tau(x_n^l, z_n; x_{n+1}^m, z_{n+1}) \qquad (m = 1, 2, \ldots, N)$$

    as well as takeoff and arrival slownesses as defined before.
  - Initialize counter $k = 0$ for a new list.
  - For each grid-point x_{n+1}^m (each m):
    - * For each item $\{x(n, j), t(n, j), p(n, j), p_0(n, j)\}$ in level n list (each j): if

      $$\left(p(n, j) + \left.\frac{\partial \tau(x, z_n, x_{n+1}^m, z_{n+1})}{\partial x}\right|_{x = x(n, j)} \right) \times \left(p(n, j+1) + \left.\frac{\partial \tau(x, z_n, x_{n+1}^m, z_{n+1})}{\partial x}\right|_{x = x(n, j+1)} \right) < 0,$$

      then estimate zero crossing for x by linear interpolation. Use resulting convex combination to interpolate $t \leftarrow t + \tau$, $p = \tau_x$, and $p_0 =$ interpolate of p_0's from level n. Insert the new item into the list as $x(n+1, k) = x_{n+1}^m$, $t(n+1, k)$, $p(n+1, k)$, and $p_0(n+1, k)$, in order of increasing takeoff slowness p_0; increment $k \leftarrow k + 1$;
    - * Next j;
  - Next m;
- Next n;

Remark. We actually only need to record "in aperture" times, i.e., those for which rays make an angle $\leqslant \theta_{\max}$ with depth axis. The necessary angle information is a by-product of the local paraxial eikonal solver.

Remark. In the algorithm the initial datum compatible with a point source is generated by assuming that the velocity near the source is constant and equal to the velocity at the source. This assumption naturally introduces some traveltime errors, but we can compute an *a priori* estimate of the initial step so that the adaptive eikonal solver is furnished with accurate, smooth initial data; see Sec. 2.3 for implementation and see also [36] for more details.

Remark. In the algorithm we need to choose δz in every depth strip so that the solution is locally smooth. At the initial strip we might obtain an *a priori* estimate of δz or guess an initial δz by trial and error. However, for depth strips afterward it is possible to choose δz adaptively so that locally the solution is still smooth. On the other hand, our computational experience (see next section) indicates that inserting unnecessary depth strips degrades the numerical results; therefore one has to be careful in choosing δz in the computation. Fortunately, the adaptive eikonal solver based on WENO Runge–Kutta schemes is robust enough to always yield some predictable approximation of viscosity solutions of the eikonal equation even though the theoretical solution might not be smooth. This implies that for an arbitrary δz the algorithm will produce numerical solutions of the paraxial eikonal equation, which in turn justifies the choice of fixed δz in the slowness matching algorithm.

The complexity of the algorithm for a grid of traveltime output points of size $n_x \times n_z$ is dominated by the cost of the local traveltime solves. Suppose that the total depth interval is divided into m roughly equal layers. Because of the aperture limitation just mentioned, each local traveltime solve costs $O((n_x/m)(n_z/m))$ flops, even accounting for grid adaptation near each source point. This estimate presumes that the output grid is eventually the computational grid, after a step-doubling phase near the source point, and that is indeed the behavior of the adaptive grid algorithm explained above. Since there are n_x gridpoints at the top of each layer, and m layers, the total cost of local traveltime solves is $O(n_x^2 n_z/m)$. The cost of the slowness matching computations depends on the root-finding algorithm used. The algorithm just described searches exhaustively the previous slowness match layer for each point on the current one. For a fixed bound on the number of traveltime branches, which in turn is bounded by a function of the C^2 norm of $\log s$, a reasonable estimate is proportional to the number of gridpoints visited, i.e., $O(n_x^2 m)$, which is negligible compared to the cost of the local traveltime solves (as we observe in practice). The overall cost compares unfavorably to the $O(n_x n_z)$ cost of a single first arrival computation using one of the WENO paraxial algorithms. However, most of the flops go into computing local tables, which are reusable across sources (x_s, z_s). Therefore, if multiple $(O(n_x))$ multi-arrival traveltime tables are required, i.e., one for each gridpoint along the top of the grid, then the cost of the slowness matching multi-arrival computations is of roughly same as that of computing $O(n_x)$ first arrival tables. Thus, for applications such as Kirchoff migration of densely sampled reflection seismic data, the slowness matching algorithm becomes competitive with other approaches.

Corollary 3.3 guarantees that this procedure actually computes all of the traveltimes along downgoing rays. Concerning numerical accuracy, note that the adaptive eikonal solver described in the last section computes τ with effectively third-order accuracy, so difference approximation will compute $\nabla\tau$ with second-order accuracy as well. It follows that the solution of the slowness matching condition, hence the multiple traveltimes and slownesses derived from it, will have second-order accuracy essentially, apart from the errors introduced by numerical root finding. Given accurate local tables, main source of error is linear interpolation which is of $O((\Delta x)^2)$. These latter are consistent in order of accuracy; the upshot is that the computed multiple traveltimes should be of second-order accuracy essentially in the grid spacing.

4. NUMERICAL EXPERIMENTS

The slowness matching code used in the following experiments combines a Fortran implementation of the adaptive WENO eikonal solver and a driver module written in C++. It stores output traveltimes in linked lists to accommodate their *a priori* unknown number.

The waveguide example (Fig. 1) has an analytic slowness field, and the ordinary differential equation solver MATLAB™ `ODE45` computes trajectories of Hamilton's equations essentially to machine precision, thus giving "ground truth" against which to compare the output of the slowness matching algorithm.

Trial and error showed that $\delta z = 0.25$ km in Algorithm 1 is a sufficiently small depth interval to keep local traveltimes single-valued in this example. Figure 5 compares the ray trace and slowness matching traveltimes at depth $z = 2.0$ km using $\Delta x = 0.05$ km. The slowness matching and ray-tracing times differ somewhat, but the development of the two triplications is clear. To illustrate the convergence theory sketched above, Fig. 6 shows the same data, but this time using $\Delta x = 0.025$ km in the slowness matching algorithm. Since an adaptive eikonal solver is used in the algorithm, it is hard to measure the convergence order exactly; but in the figure the near second-order convergence is clearly apparent, and the triplications unfold nicely.

The next example is Marmousi model from the 1996 INRIA Workshop on Multi-arrival Traveltimes. The calibration data used here were computed by Dr. Klimés.

Figure 7 is the calibration for first arrivals only by using the adaptive gridding WENO eikonal solver. Figure 8 is the calibration for multi-arrivals by the slowness matching algorithm, where $\delta z = 1$ km. Figure 9 shows the details of the center section. Since the convex combinations of traveltimes computed by the algorithm described above is *always* an overestimate (Fermat's principle!), inserting (unnecessary) slowness matches degrades the result. Figures 10 and 11 show the comparison $\delta z = 0.5$ km vs. $\delta z = 1$ km.

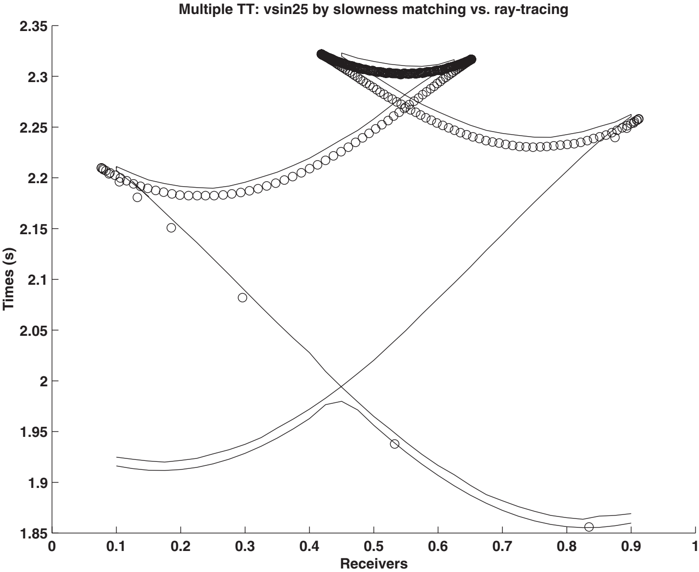

Fig. 5. Traveltime at depth $z = 2.0$ km. Dots are ray arrivals, uniformly sampled in angle at source. Line is result of slowness matching algorithm with $\Delta x = 0.05$ km used to compute local traveltime fields via WENO adaptive eikonal solver. Note that the lower line corresponds to the first-arrival traveltime field.

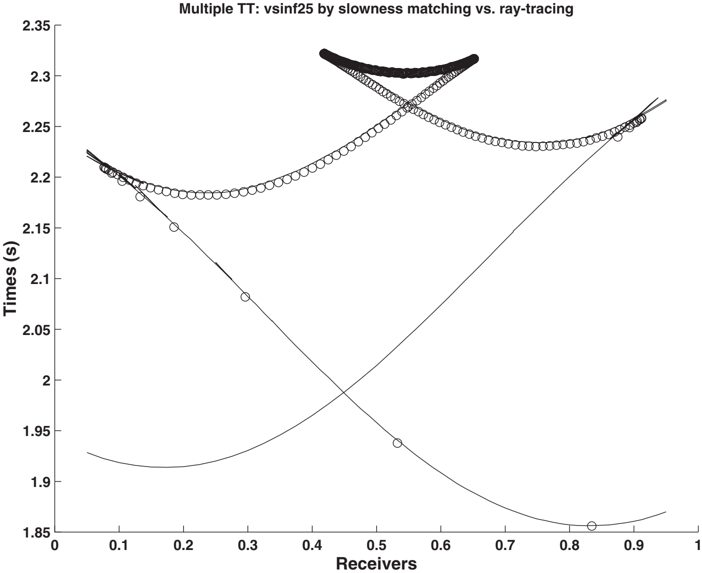

Fig. 6. Traveltime at depth $z = 2.0$ km. Dots are ray arrivals, uniformly sampled in angle at source. Line is result of slowness matching algorithm with $\Delta x = 0.025$ km used to compute local traveltime fields via WENO adaptive eikonal solver. Comparison with Fig. 5 shows near 2nd-order convergence.

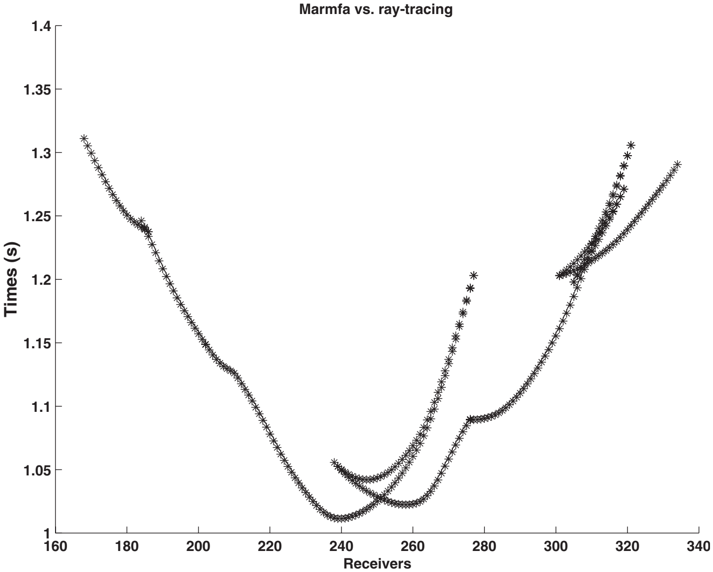

Fig. 7. Traveltime at depth $z = 2.904$ km for Marmousi model. Stars are ray arrivals, uniformly sampled in angle at source. Line is result of first-arrivals via WENO adaptive eikonal solver with $\Delta x = 0.024$ km.

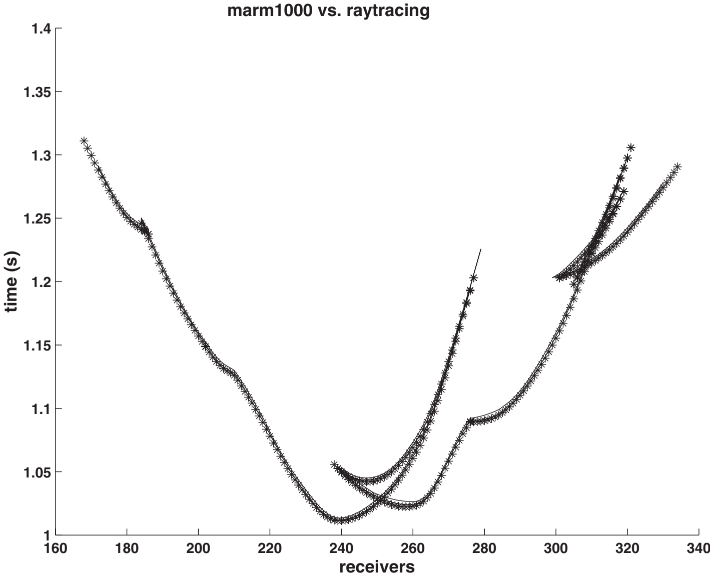

Fig. 8. Traveltime at depth $z = 2.904$ km. Stars are ray arrivals, uniformly sampled in angle at source. Line is result of slowness matching algorithm with $\delta z = 1$ km. The local traveltime fields is computed with $\Delta x = 0.024$ via WENO adaptive eikonal solver.

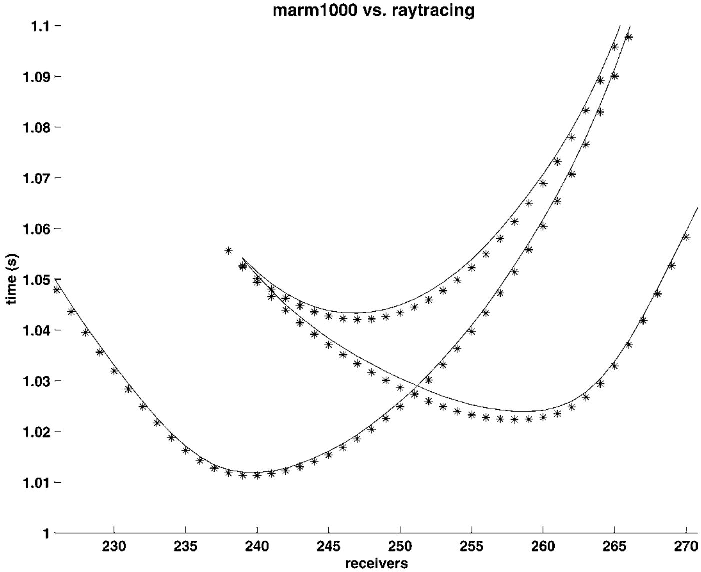

Fig. 9. Details for traveltime at depth $z = 2.904$ km for Marmousi model. Stars are ray arrivals, uniformly sampled in angle at source. Line is result of slowness matching algorithm with $\delta z = 1$ km. The local traveltime field is computed with $\Delta x = 0.024$ via WENO adaptive eikonal solver.

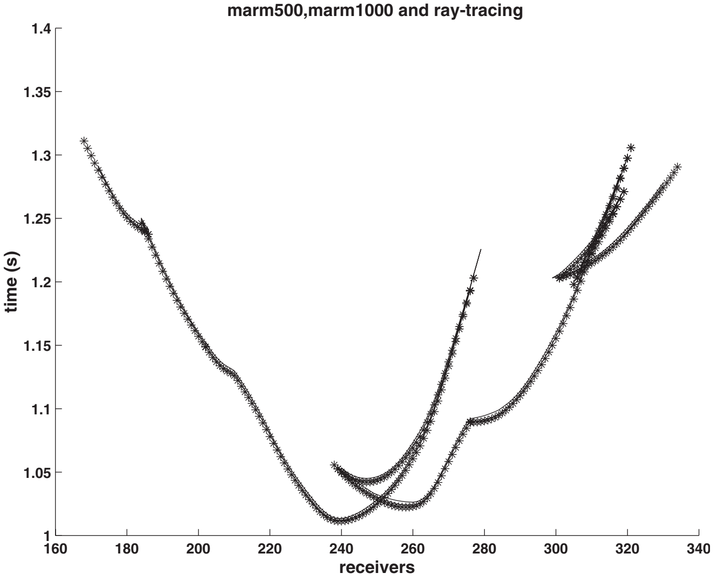

Fig. 10. Traveltime at depth $z = 2.904$ km. Stars are ray arrivals, uniformly sampled in angle at source. Lines are results of slowness matching algorithm with $\delta z = 1$ km and $\delta z = 0.5$ km. The local traveltime field is computed with $\Delta x = 0.024$ via WENO adaptive eikonal solver.

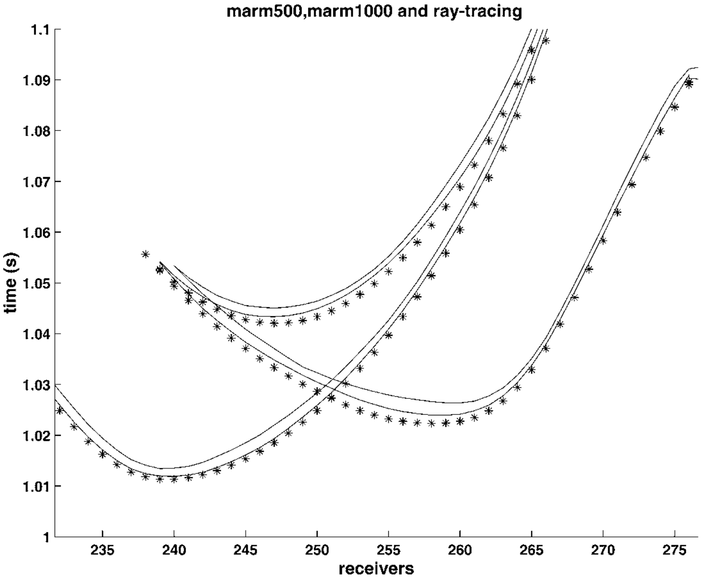

Fig. 11. Details for traveltime at depth $z = 2.904$ km. Stars are ray arrivals, uniformly sampled in angle at source. Lines are results of slowness matching algorithm with $\delta z = 1$ km (lower curve, more accurate) and $\delta z = 0.5$ km (upper curve, less accurate). The local traveltime field is computed with $\Delta x = 0.024$ via WENO adaptive eikonal solver.

5. EXTENSIONS

The algorithm described in the preceding section yields complete traveltime information at the intermediate depth levels selected in the computation and of course at the bottom. This is sufficient for modeling of transmitted waves, but some applications, for example, Kirchhoff migration of reflection data, require time fields throughout the grid, not just at a few levels. A simple modification of the algorithm generates the global multi-arrival traveltime field on a full grid. In fact, one simply takes $\delta z = \Delta z$ in Algorithm 1, hence $m = n_z$. Remarkably, the procedure remains economical: for each point on each interface, the aperture limitation implies that only $O(1)$ points need be searched, not $O(n_x)$. Thus the total cost of the slowness matching phase becomes $O(n_x^2 n_z)$, which is comparable to the cost of the local traveltime solves so that the overall cost estimate remains the same.

The enumerative approach to locating roots of the slowness matching condition is unnecessarily inefficient. A simple derivative free search, such as Brent's method [11, 32], should reduce the cost of the search considerably. More accurate local traveltimes or an appropriate averaging procedure would make the traveltime second derivative accessible numerically, permitting Newton's method or a relative to be used [11].

Dropping the restriction to times along downgoing rays requires more serious modification. A variety of methods have been suggested for aperture-unlimited local traveltime computation: see for example [47, 31, 42, 10, 43, 20]. Then replace the depth range strips of the slowness matching algorithm described here by rectangular patches, in which local traveltimes are single valued, and perform slowness matching at their common boundaries. In contrast to the down-going ray construction, in which there is a preferred direction, this isotropic algorithm must sweep through the network of rectangular patches in some order, and almost surely multiple sweeps will be required. Thus arises an interesting combinatorial problem: what is the minimum number of sweeps guaranteed to catch all travel-times, and how should such an optimal program be organized? For discussion of a similar question in the context of first arrival computation, see [42]. Assuming a reasonable bound on the number of sweeps, the economics of this algorithm would compare favorably to global traveltime solves when a dense array of surface source points is required.

Three dimensional generalization is in principle straightforward, though issues of numerical efficiency remain to be settled. The slowness matching procedure requires finding solutions of two equations in two unknowns, which could be approached in the same way as in the two dimensional case. Obviously the benefit of the more efficient search strategies would be more pronounced in the three dimensional case.

To prove the convergence of the algorithm, there needs some more thinking. First our algorithm computes multivalued traveltimes rather than caustics. Secondly, any convergence theorem for our algorithm will have to account for the changing number of the traveltimes computed at points near caustics as the grid is refined. Our numerical evidence suggests that: if there are n traveltimes in a neighborhood of a point (this excluding caustic points), then for a small enough Δx we compute n traveltimes, and they are in error $O(\Delta x^2)$.

For applications in wave propagation, such as high frequency asymptotic modeling and migration of seismic reflection data, amplitudes and phase factors (Maslov indices, [17]) are needed. The local computations are already in hand. Extension of the slowness matching requires computation of a curvature factor to combine amplitudes along ray segments, which is possible by using the adaptive eikonal solver designed for both traveltimes and amplitudes [36]. The Maslov index is immediately available in two dimensions, because of the ordering step in the algorithm outlined here—the index increments for each sign change of the arrival slowness, ordered by increasing takeoff slowness.

ACKNOWLEDGMENT AND DISCLAIMER

This work was partially supported by the Office of Naval Research under Grant Number N00014-96-1-0156 and The Rice Inversion Project. TRIP Sponsors for 2001 are Amerada Hess, Conoco Inc., Exxon/Mobil Production Research Co., Landmark Graphics, Shell International Research and Western Geco. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the Office of Naval Research, or the U.S. Government.

REFERENCES

1. Albert, S., Cockburn, B., French, D., and Peterson, T. (2002). A posteriori error estimates for general numerical methods for Hamilton–Jacobi equations. Part I: The steady state case. *Math. Comp*. **71**, 49–76.
2. Benamou, J. D. (1996). Big ray tracing: Multivalued travel time field computation using viscosity solutions of the eikonal equations. *J. Comput. Phys*. **128**, 463–474.
3. Benamou, J.-D. (1999). Direct solution of multi-valued phase-space solutions for Hamilton–Jacobi equations. *Comm. Pure Appl. Math*. **52**, 1443–1475.
4. Benamou, J.-D., and Solliec, I. (2000). A Eulerian method for capturing caustics. *J. Comput. Phys*. **162**, 132–163.
5. Berger, M., and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys*. **53**, 484–512.
6. Courant, R., and Hilbert, D. (1962). *Methods of Mathematical Physics*, Vol. II, Wiley.
7. Crandall, M. G., Evans, L. C., and Lions, P. L. (1984). Some properties of viscosity solutions of Hamilton–Jacobi equations. *Trans. Am. Math. Soc*. **282**, 487–502.
8. Crandall, M. G., and Lions, P. L. (1983). Viscosity solutions of Hamilton–Jacobi equations. *Trans. Am. Math. Soc*. **277**, 1–42.
9. Crandall, M. G., and Lions, P. L. (1984). Two approximations of solutions of Hamilton–Jacobi equations. *Math. Comp*. **43**, 1–19.
10. Dellinger, J., and Symes, W. W. (1997). *Anisotropic Finite-Difference Traveltimes Using a Hamilton–Jacobi Solver*, 67th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Soc. Expl. Geophys., pp. 1786–1789.
11. Dennis, Jr., J. E., and Schnabel, R. B. (1983). *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, Prentice–Hall, Englewood Cliffs.
12. Engquist, B., and Runborg, O. (1996). Multi-phase computations in geometrical optics. *J. Comput. Appl. Math*. **74**, 175–192.
13. Engquist, B., Runborg, O., and Tornberg, A.-K. (2002). High frequency wave propagation by the segment projection method. *J. Comp. Phys*. **178**, 373–390.
14. Fomel, S., and Sethian, J. (2002). Fast phase space computation of multiple traveltimes. *Proc. Nat. Aca. Sci*. **99**, 7329–7334.
15. Geoltrain, S., and Brac, J. (1993). Can we image complex structures with first-arrival traveltime? *Geophysics* **58**, 564–575.
16. Gray, S., and May, W. (1994). Kirchhoff migration using eikonal equation traveltimes. *Geophysics* **59**, 810–817.
17. Guillemin, V., and Sternberg, S. (1979). *Geometric Asymptotics*, American Mathematical Society, Providence.
18. Hu, C., and Shu, C. W. (2000). A discontinous Galerkin finite element method for Hamilton–Jacobi equations. *SIAM J. Sci. Comput*. **21**, 666–690.
19. Jiang, G. S., and Peng, D. (2000). Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM J. Sci. Comput*. **21**, 2126–2143.
20. Kim, S., and Cook, R. (1999). 3-D traveltime computation using second-order ENO scheme. *Geophysics* **64**, 1867–1876.
21. Kossioris, G., Makridakis, Ch., and Souganidis, P. E. (1999). Finite volume schemes for Hamilton–Jacobi equations. *Numer. Math*. **83**, 427–442.
22. Kurganov, A., and Tadmor, E. (2000). New high-resolution semi-discrete central schemes for Hamilton–Jacobi equations. *J. Comput. Phys*. **160**, 720–742.
23. Lin, C. T., and Tadmor, E. (2000). High-resolution nonoscillatory central schemes for Hamilton–Jacobi equations. *SIAM J. Sci. Comput*. **21**, 2163–2186.
24. Lions, P. L. (1982). *Generalized Solutions of Hamilton–Jacobi Equations*, Pitman Advanced Publishing Program.
25. Liu, Z., and Bleistein, N. (1995). Migration velocity analysis: Theory and an iterative algorithm. *Geophysics* **60**, 142–153.
26. Milnor, J. (1973). *Morse Theory*, Annals of Math., No. 51, Princeton University Press.
27. Operto, S., Xu, S., and Lambare, G. (2000). Can we image quantitatively complex models with rays. *Geophysics* **65**, 1223–1238.

28. Osher, S., Cheng, L.-T., Kang, M., Shim, H., and Tsai, Y.-H. (2002). Geometrical optics in a phase space based level set and Eulerian framework. *J. Comput. Phys.* **179**, 622–648.
29. Osher, S. J., and Sethian, J. A. (1988). Fronts propagating with curvature dependent speed: Algorithms based on Hamilton–Jacobi formulations. *J. Comput. Phys.* **79**, 12–49.
30. Osher, S. J., and Shu, C. W. (1991). High-order essentially non-oscillatory schemes for Hamilton–Jacobi equations. *SIAM J. Num. Anal.* **28**, 907–922.
31. Podvin, P., and Lecomte, I. (1991). Finite difference computation of traveltimes in very contrasted velocity models: A massively parallel approach and its associated tools. *Geophys. J. Int.* **105**, 271–284.
32. Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T. (1986). *Numerical Recipes*, Cambridge University Press, New York.
33. Qian, J., Belfi, C. D., and Symes, W. W. (1999). *Adaptive Finite Difference Method for Traveltime and Amplitude*, 69th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Soc. Expl. Geophys., pp. 1763–1766.
34. Qian, J., Cheng, L.-T., and Osher, S. J. (2001). A level set based Eulerian approach for anisotropic wave propagations. Submitted to *Wave Motion*.
35. Qian, J., and Symes, W. W. (2001). Paraxial eikonal solvers for anisotropic quasi-P traveltimes. *J. Comput. Phys.* **173**, 1–23.
36. Qian, J., and Symes, W. W. (2002). Adaptive finite difference method for traveltime and amplitude. *Geophysics* **67**, 167–176.
37. Qian, J., and Symes, W. W. (2002). Finite-difference quasi-P traveltimes for anisotropic media. *Geophysics* **67**, 147–155.
38. Qian, J., Symes, W. W., and Dellinger, J. A. (2001). *A Full-Aperture Anisotropic Eikonal Solver for Quasi-P Traveltimes*, 71st Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Soc. Expl. Geophys., pp. 129–132.
39. Rouy, E., and Tourin, A. (1992). A viscosity solutions approach to shape-from-shading. *SIAM J. Num. Anal.* **29**, 867–884.
40. Ruuth, S. J., Merriman, B., and Osher, S. J. (1999). A fixed grid method for capturing the motion of self-intersecting interfaces and related PDEs. *J. Comput. Phys.* **151**, 836–861.
41. Schneider, Jr., W. A. (1995). Robust and efficient upwind finite-difference traveltime calculations in three dimensions. *Geophysics* **60**, 1108–1117.
42. Schneider, Jr., W. A., Ranzinger, K. A., Balch, A. H., and Kruse, C. (1992). A dynamic programming approach to first arrival traveltime computation in media with arbitrarily distributed velocities. *Geophysics* **57**, 39–50.
43. Sethian, J. A., and Popovici, A. M. (1999). 3-D traveltime computation using the fast marching method. *Geophysics* **64**, 516–523.
44. Steinhoff, J., Fan, M., and Wang, L. (2000). A new Eulerian method for the computation of propagating short acoustic and electromagnetic pulses. *J. Comput. Phys.* **157**, 683–706.
45. Stoer, J., and Bulirsch, R. (1992). *Introduction to Numerical Analysis*, 2nd edn., Springer-Verlag, New York.
46. Symes, W. W. (1998). *A Slowness Matching Finite Difference Method for Traveltimes Beyond Transmission Caustics*, 68th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, Soc. Expl. Geophys., pp. 1945–1948.
47. van Trier, J., and Symes, W. W. (1991). Upwind finite-difference calculation of travel-times. *Geophysics* **56**(6), 812–821.
48. Vidale, J. (1988). Finite-difference calculation of travel times. *Bull., Seis. Soc. Am.* **78**, 2062–2076.
49. Vinje, V., Iversen, E., and Gjoystal, H. (1993). Traveltime and amplitude estimation using wavefront construction. *Geophysics* **58**, 1157–1166.
50. White, B. S. (1984). The stochastic caustic. *SIAM J. Appl. Math.* **44**, 127–149.