An Adaptive Finite Difference Method for Traveltime and Amplitude

Jianliang Qian [*] *and William W. Symes* [†]

ABSTRACT

The eikonal equation with point source is difficult to solve with high order accuracy because of the singularity of the solution at the source. All the formally high order schemes turn out to be first order accurate without special treatment of this singularity. Adaptive upwind finite difference methods based on high order ENO (Essentially NonOscillatory) Runge-Kutta difference schemes for the paraxial eikonal equation overcome this difficulty. The method controls error by automatic grid refinement and coarsening based on an *a posteriori* error estimation. It achieves prescribed accuracy at far lower cost than fixed grid methods. Reliable auxiliary quantities, such as take-off angle and geometrical spreading factor, are by-products.

Introduction

Many finite difference methods have been introduced to compute the traveltime for isotropic media directly on regular grid (Reshef and Kosloff, 1986; Vidale, 1988; van Trier and Symes, 1991; Schneider et al., 1992; Qin et al., 1992; Schneider, 1995; El-Mageed et al., 1997; Fomel, 1997; Popovici and Sethian, 1997).

The traveltime field is mostly smooth, and the use of upwind differencing (in all of the cited methods) confines the errors due to singularities which develop away from the source point. The source point itself is, however, also a singularity. The truncation error of a pth order method is dominated by the product of $(p+1)$st derivatives of the time field and the $(p+1)$st power of the step(s). The $(p+1)$st derivatives of the time field, however, go like the $(-p+1)$th power of the distance to the source. Therefore near the source - when the distance is on the order of the

[*]The Rice Inversion Project, Department of Computational and Applied Mathematics, Rice University, Houston TX 77251-1892 USA, email `jlqian@caam.rice.edu`

[†]The Rice Inversion Project ,Department of Computational and Applied Mathematics, Rice University, Houston TX 77251-1892 USA, email `symes@caam.rice.edu`

step - the truncation error is quadratic in the step, i.e., first order. This inaccuracy spreads throughout the computation, and renders all higher order methods first-order convergent unless the scheme is modified near the source. The issue is not academic: the first-order error is sizeable, as we shall show. Moreover, it prevents reliable computation of auxiliary quantities such as takeoff angle and amplitude.

In this paper, we show how to use adaptive gridding concepts commonplace in the numerical solution of ordinary differential equations to resolve this difficulty. The efficiencies achieved by adaptive gridding are considerable - usually more than an order of magnitude gain in computation time for problems of typical exploration size. We also obtain dramatic improvements in the accuracy of takeoff angle computations and, therefore, for other geometric acoustics quantities as well.

Paraxial eikonal equation

Denote by (x_s, z_s) the coordinates of a source point, and by (x, z) the coordinates of a general point in the subsurface. The first arrival traveltime field $\tau(x, z; x_s, z_s)$ is the viscosity solution of the eikonal equation

$$\left(\frac{\partial \tau}{\partial x}\right)^2 + \left(\frac{\partial \tau}{\partial z}\right)^2 = s^2(x, z) \tag{1}$$

with the initial condition

$$\lim \left(\frac{\tau(x, z; x_s, z_s)}{\sqrt{(x - x_s)^2 + (z - z_s)^2}} - \frac{1}{v(x, z)} \right) = 0$$

as $(x, z) \to (x_s, z_s)$, where v is the velocity, $s = \frac{1}{v}$ is the slowness (Lions, 1982).

In some seismic applications, the traveltime field is only needed in regions where

$$\frac{\partial \tau}{\partial z} \geq s \cos \theta_{\max} > 0,$$

i.e., along downgoing, first-arriving rays making an angle $\leq \theta_{\max} < \frac{\pi}{2}$ with the vertical.

To enforce this condition, we modify the eikonal equation as an evolutionary equation in depth, as suggested by Gray and May (Gray and May, 1994):

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

where smmax is a smoothed max function:

$$\mathrm{smmax(x, a)} = \begin{cases} \frac{1}{2}a & \text{if } x < 0, \\ \frac{1}{2}a + 2\frac{x^4}{a^3}\left(1 - \frac{4}{5}\frac{x}{a}\right) & \text{if } 0 \leq x < \frac{a}{2}, \\ x + 2\frac{(x-a)^4}{a^3}\left(1 + \frac{4}{5}\frac{x-a}{a}\right) & \text{if } \frac{a}{2} \leq x < a, \\ x & \text{if } x \geq a. \end{cases}$$

This equation defines a stable nonlinear evolution in z, suitable for explicit finite difference discretization. The solution τ is identical to the solution of the eikonal equation provided that the ray makes an angle $\leq \theta_{\max} < \frac{\pi}{2}$ with the vertical.

The Advection equation for take-off angle

The amplitude satisfies the zeroth order transport equation (Cerveny et al., 1977):

$$\nabla\tau \cdot \nabla A + \frac{1}{2} A \nabla^2 \tau \quad = \quad 0. \tag{3}$$

If the traveltime field has been found by solving the eikonal equation, then equation (3) is a first order advection equation. However, we see that the Laplacian of traveltime field is involved in the transport equation, which implies that we need a third order accurate traveltime field to get a first order accurate amplitude field (Symes, 1995; El-Mageed, 1996; El-Mageed et al., 1997). To avoid this complexity, we use another approach to compute the amplitude.

In 2D isotropic media, the amplitude satisfies (Cerveny et al., 1977; Friedlander, 1958)

$$\begin{aligned} A &= \frac{v}{2\pi\sqrt{2}\sqrt{|J|}} \\ &= \frac{v}{2\pi\sqrt{2}}\sqrt{\nabla\tau \times \nabla\phi}, \end{aligned}$$

where $J(x, z; x_s, zs)$ is the Jacobian of the transformation from Cartesian coordinates (x, z) to ray coordinates (τ, ϕ), τ is the traveltime, $\phi = \phi(x, z; x_s, z_s)$ is the take-off angle from source point (x_s, z_s) to a general point (x, z) in the subsurface:

$$\begin{aligned} J &= \begin{vmatrix} \frac{\partial x}{\partial \tau} & \frac{\partial z}{\partial \tau} \\ \frac{\partial x}{\partial \phi} & \frac{\partial z}{\partial \phi} \end{vmatrix} \\ &= \begin{vmatrix} \frac{\partial \tau}{\partial x} & \frac{\partial \tau}{\partial z} \\ \frac{\partial \phi}{\partial x} & \frac{\partial \phi}{\partial z} \end{vmatrix}^{-1} \\ &= \frac{1}{\nabla\tau \times \nabla\phi}, \end{aligned}$$

where $\nabla\phi$ and $\nabla\tau$ are the gradients of take-off angle and traveltime, respectively.

Since the take-off angle ϕ is constant along any ray,

$$\nabla\tau \cdot \nabla\phi = \frac{\partial\tau}{\partial x}\frac{\partial\phi}{\partial x} + \frac{\partial\tau}{\partial z}\frac{\partial\phi}{\partial z} \quad = \quad 0. \tag{4}$$

That is, the gradient $(\frac{\partial \tau}{\partial x}, \frac{\partial \tau}{\partial z})$ is the wavefront normal which is tangential to the ray; the gradient $(\frac{\partial \phi}{\partial x}, \frac{\partial \phi}{\partial z})$ is tangential to the wavefront.

However, the gradient of the take-off angle depends on the second order derivative of traveltime, so that we need third order accurate traveltimes to get a first order accurate gradient of take-off angle. Zhang (Zhang, 1993) used this equation in polar coordinates to compute the geometrical spreading factor, but his computation of the take-off angle was based on the first order traveltime field. Consequently, the gradient of take-off angle computed by his scheme was inaccurate. Vidale (Vidale and Houston, 1990) encountered a similar difficulty.

ENO and WENO schemes for eikonal equation

There is a large number of competing finite difference and related schemes for the solution of the eikonal equation suggested in the literature; we have chosen to use the essentially nonoscillatory ("ENO") schemes of Osher and Sethian (Osher and Sethian, 1988) and Osher and Shu (Osher and Shu, 1991) and the related weighed ENO (WENO) schemes (Liu et al., 1994; Shu, 1997; Jiang and Peng, 1997) for the following reasons: (1)stable schemes of arbitrarily high order accuracy exist, permitting accurate solutions on coarse grids (which is critical to the mesh refinement or coarsenment); (2) versions exist in any dimension so that we can extend our methodology to the three-dimensional case straightforwardly (El-Mageed, 1996; El-Mageed et al., 1997).

Our adaptive scheme is based on the 2nd and 3rd order WENO difference schemes introduced by Jiang and Peng (Jiang and Peng, 1997). These are in turn extensions of 2nd and 3rd order ENO difference schemes, which we present first.

For a function f of the space variable (x, z) in the computational domain, we write

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

Let $\tau_i^k = \tau(x_i, z_k; x_s, z_s)$ and define the forward D^+ and backward D^- finite difference operators

$$D_x^{\pm} \tau_i^k = \frac{\pm[\tau_{i\pm 1}^k - \tau_i^k]}{\Delta x}.$$

The second and third order ENO refinements of $D_x^{\pm}\tau$ are

$$\begin{aligned} D_x^{\pm,2}\tau &= D_x^{\pm}\tau \mp \frac{1}{2}\Delta x\, m(D_x^{\pm} D_x^{\pm}\tau, D_x^- D_x^+\tau), \\ D_x^{\pm,3}\tau &= D_x^{\pm,2}\tau - \frac{1}{6}(\Delta x)^2\, m(D_x^{\pm} D_x^{\pm} D_x^{\pm}\tau, D_x^+ D_x^+ D_x^-\tau, D_x^+ D_x^- D_x^-\tau), \end{aligned}$$

where

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

The upwind ENO approximations for $\frac{\partial\tau}{\partial x}$ are

$$\widehat{D_x^n}\tau = \text{modmax}(\max(D_x^{-,n}\tau,0),\min(D_x^{+,n}\tau,0)),$$

for $n=2,3$, where modmax function returns the larger value in modulus.

The second order and third order ENO Runge-Kutta steps are

$$\begin{aligned}\delta_2^1\tau &= \Delta z H(\widehat{D_x^2}\tau),\\ \delta_2^2\tau &= \frac{1}{2}\left(\delta_2^1\tau + \Delta z H(\widehat{D_x^2}(\tau+\delta_2^1\tau))\right),\end{aligned} \tag{5}$$

and

$$\begin{aligned}\delta_3^1\tau &= \Delta z H(\widehat{D_x^3}\tau),\\ \delta_3^2\tau &= \frac{1}{4}\left(\delta_3^1\tau + \Delta z H(\widehat{D_x^3}(\tau+\delta_3^1\tau))\right),\\ \delta_3^3\tau &= \frac{1}{3}\left(2\delta_3^2\tau + 2\Delta z H(\widehat{D_x^3}(\tau+\delta_3^2\tau))\right),.\end{aligned} \tag{6}$$

The depth step Δz must satisfy the stability condition:

$$\Delta z \le \Delta z_{cfl} = \frac{0.9\Delta x}{\tan(\theta_{max})}.$$

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

The nth order scheme is then

$$\tau^{k+1} = \tau^k + \delta_n^n\tau^k \tag{7}$$

for $k=0,1,2,...,\ n=2,3$.

However, we have observed that the gradient of the take-off angle based on the third order ENO traveltime is too noisy to give us a smooth amplitude function. To alleviate this phenomenon, instead of ENO 3rd order refinements, we use WENO 5th order refinement (Jiang and Peng, 1997) to compute $D_x^{\pm}\tau$ in the third order Runge-Kutta step, which gives us a smooth amplitude field.

The WENO 5th order schemes for $D_x^{\pm}\tau_i$ are

$$\begin{aligned}D_x^{\pm,5}\tau_i &= \frac{1}{12}\left(-D_x^+\tau_{i-2} + 7D_x^+\tau_{i-1} + 7D_x^+\tau_i - D_x^+\tau_{i+1}\right)\\ &\pm \Delta x\Phi^{WENO}\left(D_x^-D_x^+\tau_{i\pm2}, D_x^-D_x^+\tau_{i\pm1}, D_x^-D_x^+\tau_i, D_x^-D_x^+\tau_{i\mp1}\right),\end{aligned}$$

where

$$\Phi^{WENO}(a,b,c,d) = \frac{1}{3}\omega_0(a-2b+c)+\frac{1}{6}(\omega_2-\frac{1}{2})(b-2c+d)$$

with weights defined as

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

In the denominators above, we added a small positive number δ to avoid dividing by zero.

WENO schemes for advection equation

Recall the advection equation for the take-off angle,

$$\frac{\partial\tau}{\partial x}\frac{\partial\phi}{\partial x}+\frac{\partial\tau}{\partial z}\frac{\partial\phi}{\partial z} = 0. \tag{8}$$

To match with the evolutionary form of the eikonal equation in depth, we formulate the advection equation as a evolution equation in depth as well, i.e.,

$$\frac{\partial\phi}{\partial z} = -(\frac{\partial\tau}{\partial z})^{-1}\frac{\partial\tau}{\partial x}\frac{\partial\phi}{\partial x}. \tag{9}$$

To take advantage of the accuracy of traveltime provided by ENO (or WENO) Runge-Kutta third order scheme for eikonal equation, we embed the third order scheme for equation (9) into the third order scheme for the eikonal equation.

To be more precise, we introduce the approximations for x-derivative and z-derivative of τ in the above advection equation:

$$\frac{\partial\tau}{\partial x}\approx D_x[\tau] = \max(D_x^{-,3}\tau,0)+\min(D_x^{+,3}\tau,0), \tag{10}$$

$$\frac{\partial\tau}{\partial z} \approx H(\widehat{D}_x^3\tau). \tag{11}$$

However, we must be careful in defining the upwind ENO difference approximation for $\frac{\partial\phi}{\partial x}$ because the ENO choice of stencil is too sensitive to the zeros of solution

(Liu et al., 1994; Jiang and Shu, 1996). Instead, we will use weighted ENO (WENO) scheme to approximate the derivative.

Because the coefficient of the discretized advection equation has only second order accuracy, which is computed from the eikonal equation by third order scheme, we use third order WENO scheme to approximate the derivative $\frac{\partial\phi}{\partial x}$. The third order WENO scheme is based on the second order ENO stencils, it does not give rise to any new complexities in the coding.

The third order upwind WENO approximations to $\frac{\partial\phi}{\partial x}$ (Jiang and Peng, 1997) are

$$\begin{aligned}
(\frac{\partial\phi}{\partial x})_i^k \approx (D_x^{-w,3}\phi)_i^k &= \frac{1}{2}(D_x^+\phi_{i-1}^k + D_x^+\phi_i^k) - \frac{w_-}{2}(D_x^+\phi_{i-2}^k - 2D_x^+\phi_{i-1}^k + D_x^+\phi_i^k) \\
(\frac{\partial\phi}{\partial x})_i^k \approx (D_x^{+w,3}\phi)_i^k &= \frac{1}{2}(D_x^+\phi_{i-1}^k + D_x^+\phi_i^k) - \frac{w_+}{2}(D_x^+\phi_{i+1}^k - 2D_x^+\phi_i^k + D_x^+\phi_{i-1}^k)
\end{aligned}$$

where

$$\begin{aligned}
w_- &= \frac{1}{1+2r_-^2}, r_- = \frac{\epsilon + (D^-D^-\phi_{i-1}^k)^2}{\epsilon + (D^-D^+\phi_i^k)^2}, \\
w_+ &= \frac{1}{1+2r_+^2}, r_+ = \frac{\epsilon + (D^+D^+\phi_i^k)^2}{\epsilon + (D^-D^+\phi_i^k)^2}
\end{aligned}$$

and ϵ is a small positive constant to avoid the denominators from becoming zero.

Now define the upwind WENO difference for $\frac{\partial\phi}{\partial x}$ which corresponds to the upwind direction of $\frac{\partial\tau}{\partial x}$ as follows:

$$D_x^{up}\phi_i^k = \begin{cases} D_x^{-w,3}\phi_i^k & \text{if } D_x[\tau]_i^k \geq 0, \\ D_x^{+w,3}\phi_i^k & \text{else.} \end{cases} \tag{12}$$

Finally we can formulate the third order WENO Runge-Kutta scheme for the advection equation as

$$\begin{aligned}
\delta_3^1\phi &= \Delta z\Psi(\widehat{D}_x^3\tau, \phi), \\
\delta_3^2\phi &= \frac{1}{4}\left(\delta_3^1\phi + \Delta z\Psi(\widehat{D}_x^3(\tau + \delta_3^1\tau), \phi + \delta_3^1\phi)\right) \\
\delta_3^3\phi &= \frac{1}{3}\left(2\delta_3^2\phi + 2\Delta z\Psi(\widehat{D}_x^3(\tau + \delta_3^2\tau), \phi + \delta_3^2\phi)\right)
\end{aligned} \tag{13}$$

where

$$\Psi(\tau,\phi) = -\frac{D_x[\tau]D_x^{up}\phi}{H(\widehat{D}_x^3(\tau))}. \tag{14}$$

The 3rd order scheme for ϕ is then

$$\phi^{k+1} = \phi^k + \delta_3^3\phi^k \tag{15}$$

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

But this scheme is really a second order scheme because the coefficients have only second order accuracy.

Adaptive grid method and its implementation

The adaptive multigrid method has been widely used for the accurate and efficient solution of PDE (Berger and Oliger, 1984; Berger and LeVeque, 1997), which uses an adaptive local error estimate to refine or coarsen the computational grid near the singularity or discontinuity of solution. For initial value problems of ODE, most state-of-the-art software uses adaptive timestepping algorithms where the timesteps are inductively chosen so that some estimate of the local (one step) error at each step is less than some quantity related to the user-defined tolerance.

The essential principle is simple. It is based on a hierarchy of difference schemes of various orders. Presumably a higher order step is more accurate than a lower order step, so the higher order step can serve as an *ersatz* for the exact solution of the differential equation with the same data. Therefore, one can combine the step computations of two different orders to obtain a so-called *a posteriori* estimate of the truncation error for the lower order step. Since the lower and higher order truncation errors stand in a known asymptotic relation, this permits an estimate of the higher order truncation error as well. The asymptotic form of the truncation error then permits prediction of a step that will result in a lower order truncation error less than a user-specified tolerance. So long as the steps are selected to maintain this local error, standard theory predicts that the higher order global error, i.e., the actual error in the solution computed using the higher order scheme, will be proportional to the user-specified tolerance.

This straightforward idea is embedded in most modern software packages for solutions of ordinary differential equations (Gear, 1971). Its use for partial differential equations is a little more complicated because it is usually necessary to adjust the grid of the non-evolution variables along with the evolution step. The solution of the (paraxial) eikonal equation changes in a sufficiently predictable way to make grid adjustment practical.

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, for example, we can take $\sigma_1 = 0.1\epsilon$ and $\sigma_2 = \epsilon$. We use the 2nd and 3rd order eikonal solvers (equations (5) and (6)), and estimate the truncation error of the 2nd order scheme at the kth step as $e_2^k = \delta_2^2\tau^k - \delta_3^3\tau^k$. So long as $\sigma_1(\epsilon) \leq e_2^k \leq \sigma_2(\epsilon)$, we simply proceed to the next step; it is well known and explained in the references already cited that efficient adaptive stepping requires rather loose control of the local error. When $e_2^k < \sigma_1(\epsilon)$, we increase the step by a factor of two, i.e., $\Delta z \leftarrow 2\Delta z$, and recompute e_2^k. Similarly, when $e_2^k > \sigma_2(\epsilon)$, 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: we retain the 3rd order (a more accurate one) computation of τ at the end of each depth step, discarding the 2nd order computation, which is used only in step control.

The usual step adjustment in ODE solvers would change Δz by a factor computed from the asymptotic form of the truncation error. 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 eikonal scheme) and, therefore, expensive interpolation. Scaling Δz by a factor of two, however, implies that 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. Since the typical behaviour 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 coarsenings and very little or no interpolation is required.

One final detail must be supplied to produce a working algorithm. Since the traveltime field is nonsmooth at the source point, the truncation error analysis on which the adaptive step selection criterion is based is not valid there. So 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 (Belfi and Symes, 1998). 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 initial 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.

A simplified algorithm framework is as follows:

- Input: ϵ, z^0, Δz^0, Δx^0, $n = 0$, $maxref$.
- Marching step:
  - do while $z^n \leq$ target depth,
  - $z^n \leftarrow z^n + \Delta z^n$, $n \leftarrow n + 1$,
  - compute $e_2^k(\tau^n, \Delta z^n)$;
  - if $e_2^k(\tau^n, \Delta z^n) \leq \sigma_1(\epsilon)$ and $k \geq 0$,
    * $\Delta z^n \leftarrow 2\Delta z^n$,
    * $\Delta x^n \leftarrow 2\Delta x^n$,
    * $k \leftarrow k - 1$.

Table 1. Fixed grid eikonal solver: constant velocity model

dx	Abs.Err(τ, dx)(s)	$Flops$
0.01	0.001232	261,590
0.00125	0.000219	16,632,765

Table 2. Adaptive grid eikonal solver: constant velocity model

ϵ	Abs.Err(τ, dx)(s)	$Flops$
0.000025	0.001041	39,815
0.00000169	0.000160	928,770

- else if $e_2^k(\tau^n, \Delta z^n) \geq \sigma_2(\epsilon)$ and $k \leq maxref$,
  * $\Delta z^n \leftarrow \Delta z^n/2$,
  * $\Delta x^n \leftarrow \Delta x^n/2$,
  * $k \leftarrow k+1$.
- end if
- end do

Numerical Experiments

In the first example, we test our method on a constant velocity model($v = 1km/s$) with two-dimensional geometry $\{(x, z) : -0.5 \leq x \leq 0.5, 0 \leq z \leq 1.0\}$.

In this case, all the desired quantities have an obvious analytical solution to compare against the computed solutions. We compare the results obtained by two approaches, the first approach is computing all the quantities on the fixed output grid which reduces to be a first order method, the second is computing all the quantities by adaptive gridding. The ouput grid is 51×51 with $\Delta x = \Delta z = 0.02$. For adaptive grid, $maxref$ is set to be 5 with the coarsest grid 17×17, and the error tolerance is set to be 0.0001.

Table 1 shows the traveltime error and computation cost by the fixed grid, and table 2 shows the traveltime error and computation cost by the adaptive grid, where Flops denotes the number of floating point operations. From the two tables, we can see that to reach the same level of accuracy, the adaptive gridding approach saves computational cost by an order of magnitude .

Figures 1 and 2 show the traveltime contours by the two approaches, from which we cannot see any differences between two approaches. Similarly, we cannot see any differences in accuracy between two approaches from take-off angle, as shown in figures 3 and 4. This is due to visual limitations of graphics.

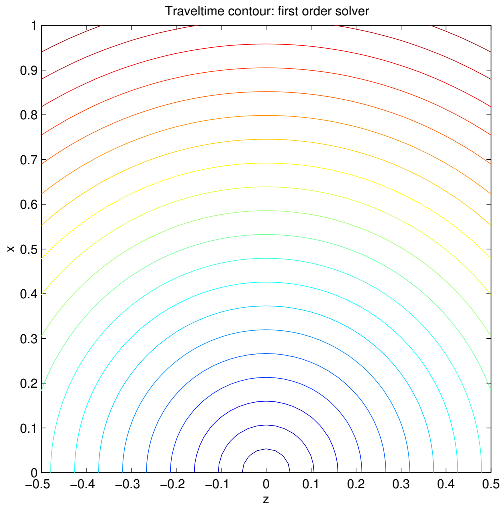

FIG. 1. Traveltime for constant velocity model: fixed grid

Figures 5 and 6 are contours of τ_x computed by two approaches. We can see that τ_x by the fixed grid approach is oscillating, but τ_x by the adaptive grid traveltime solver is convergent. Similar conclusions can be drawn for τ_z, as shown in figures 7 and 8.

Now we come to take-off angle derivatives. Figure 9 and 10 are contours of ϕ_x by two approaches. Because the coefficients in the advection equation for take-off angle depend on the traveltime gradient, any first order traveltime solver results in inaccurate $\nabla\tau$ which leads to the divergence of ϕ_x, as shown in figure 9. However, the adaptive gridding approach gives us accurate traveltime gradients, which leads to the convergent ϕ_x, as shown in figure 10. Similar conclusions can be drawn for ϕ_z, as shown in figure 11 and 12.

To illustrate further the difference of accuracy between two approaches, figures 13 and 14 show the error distributions of ϕ_x and ϕ_z.

Finally, the amplitudes based on the gradients of traveltime and take-off angle computed by the two approaches are shown in figures 15 and 16, one is divergent by the fixed grid approach, another accurate by the adaptive gridding approach.

In the second example, we use our new eikonal and amplitude solver in 2D Kirchhoff prestack migration and inversion. Figure 17 shows the impulse response by inversion with an ENO 3rd order eikonal solver. Figure 18 shows the impulse response by inversion with a WENO 5th order eikonal solver. By figures 17 and 18, we can see that the WENO 5th order solver does give us a more smooth amplitude than ENO 3rd solver.

Conclusion

In this paper we formulated a paraxial eikonal equation with depth as evolu-

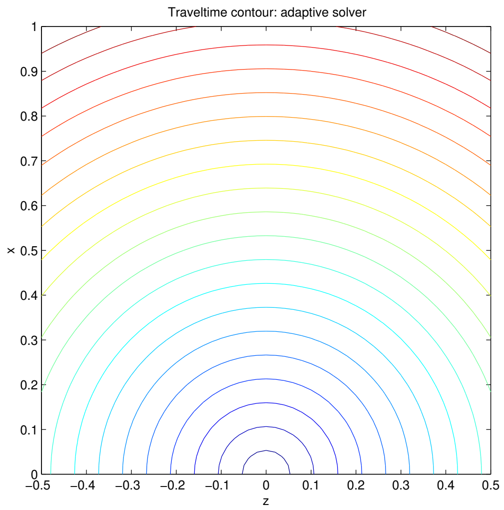

FIG. 2. Traveltime for constant velocity model: adaptive grid

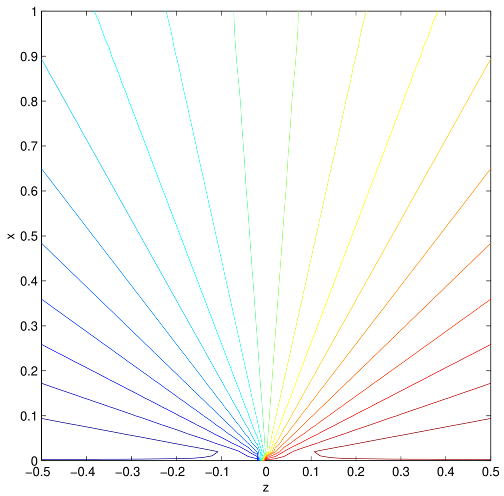

FIG. 3. Takeoff angle for constant velocity model: fixed grid

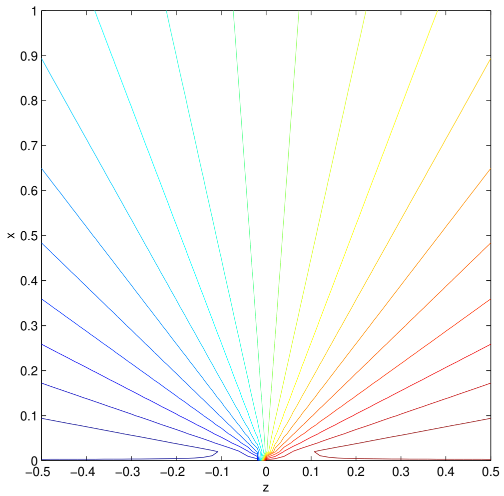

FIG. 4. Takeoff angle for constant velocity model: adaptive grid

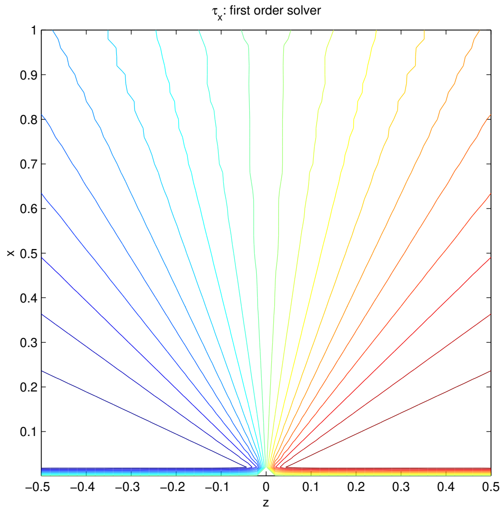

FIG. 5. Traveltime x derivative τ_x for constant velocity model: fixed grid

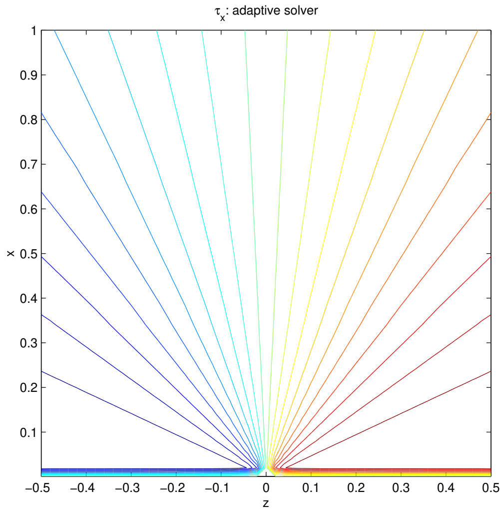

FIG. 6. Traveltime x derivative τ_x for constant velocity model: adaptive grid

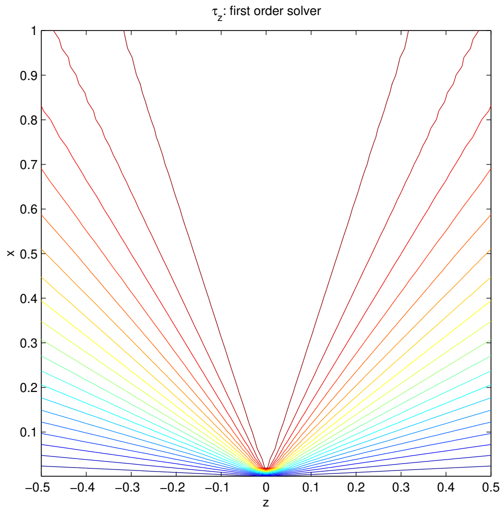

FIG. 7. Traveltime z derivative τ_z for constant velocity model: fixed grid

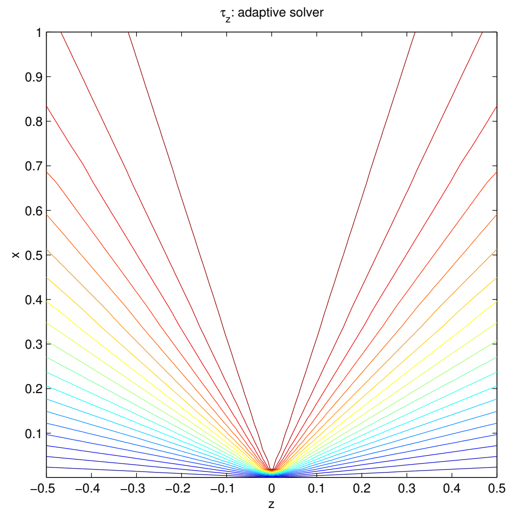

FIG. 8. Traveltime z derivative τ_z for constant velocity model: adaptive grid

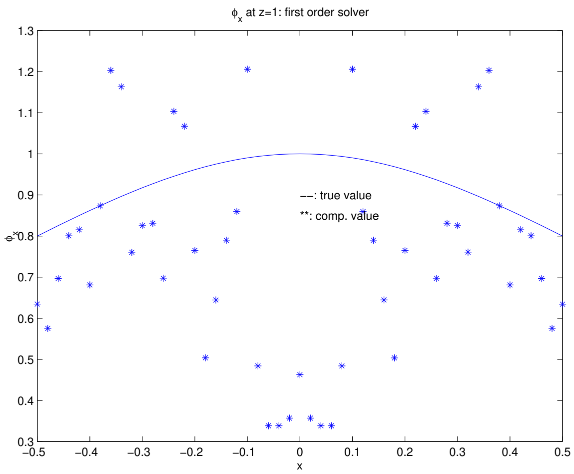

FIG. 9. Takeoff angle x derivative ϕ_x at $z = 1$ for constant velocity model: fixed grid

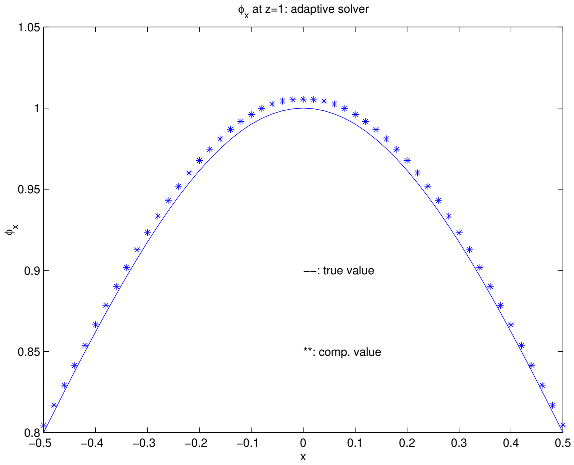

FIG. 10. Takeoff angle x derivative ϕ_x at $z = 1$ for constant velocity model: adaptive grid

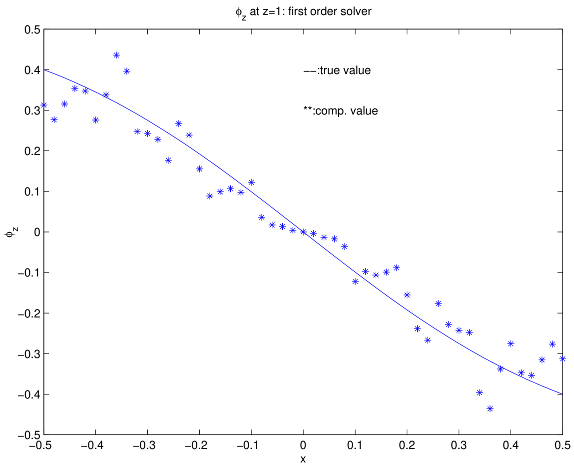

FIG. 11. Takeoff angle z derivative ϕ_z at $z = 1$ for constant velocity model: fixed grid

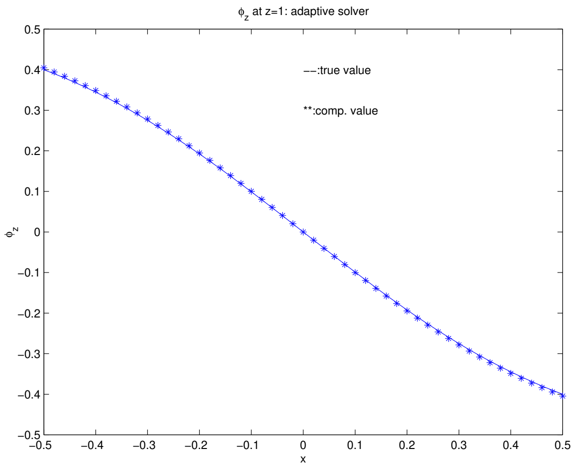

FIG. 12. Takeoff angle z derivative ϕ_z at $z = 1$ for constant velocity model: adaptive grid

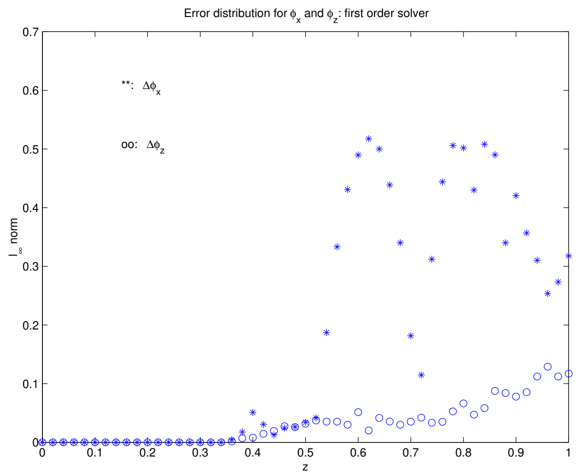

FIG. 13. Error distribution in takeoff angle derivatives phi_x and phi_z: fixed grid

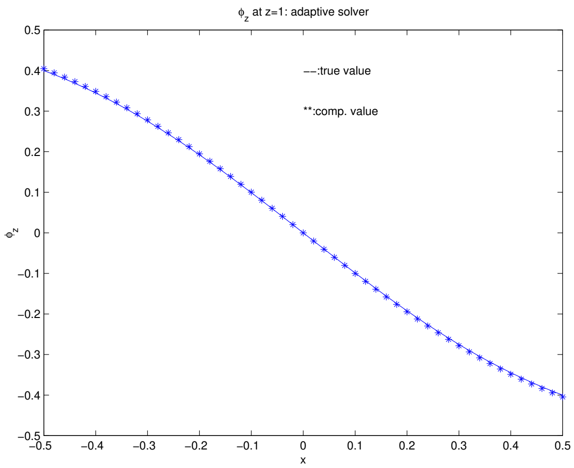

FIG. 14. Error distribution in takeoff angle derivatives phi_x and phi_z: adaptive grid

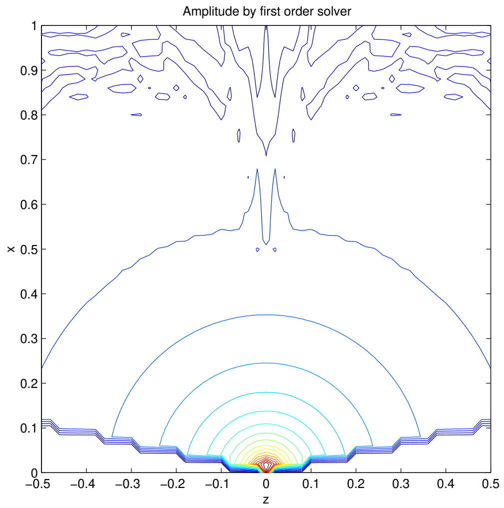

FIG. 15. 2D amplitude for constant velocity model: fixed grid

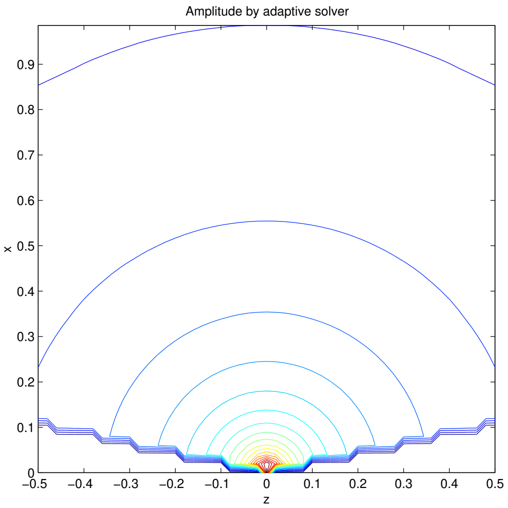

FIG. 16. 2D amplitude for constant velocity model: adaptive grid

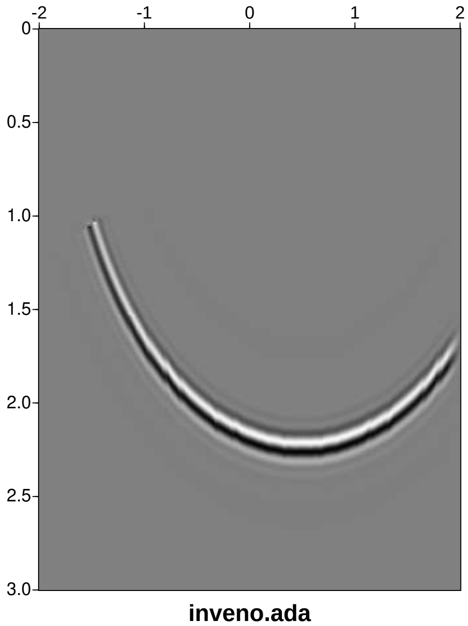

FIG. 17. The impulse response by inversion with ENO 3rd adaptive eikonal solver

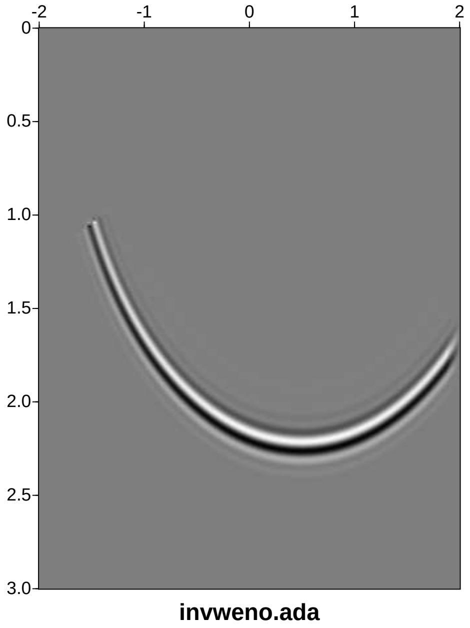

FIG. 18. The impulse response by inversion with WENO 5th adaptive eikonal solver

tion direction. Then we presented high order ENO difference schemes to solve the eikonal equation for traveltime and the advection equation for take-off angle. To deal with the singularity of point source, we proposed a new adaptive traveltime eikonal solver and detailed the implementation. Numerical experiments showed that the new method is not only accurate, but also gives us efficiency gain of more than an order of magnitude in computational time. The extension to 3D isotropic media is straightforward.

REFERENCES

Belfi, C. and Symes, W. W. (1998). An adaptive ENO algorithm for the eikonal equation. In *Annual Report, The Rice Inversion Project*. Rice University. (http://www.trip.caam.rice.edu/).

Berger, M. and LeVeque, R. J. (1997). Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. In *Preprint*. Courant Institute.

Berger, M. and Oliger, J. (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *J. Comput. Phys.*, 53:484–512.

Cerveny, V., Molotkov, I. A., and Psencik, I. (1977). *Ray Method in seismology*. Univerzita Karlova press.

El-Mageed, M., Kim, S., and Symes, W. W. (1997). 3-D Kirchhoff migration using finite difference traveltimes and amplitudes. In *Annual Report. The Rice Inversion Project*. Rice University. (http://www.trip.caam.rice.edu/).

El-Mageed, M. A. (1996). *3D First arrival traveltimes and amplitudes via Eikonal and Transport Finite differences solvers*. PhD thesis, Department of Computational and Applied mathematics, Rice University, Houston, TX77251-1892.

Fomel, S. (1997). A variational formulation of the fast marching eikonal solver. In *SEP95*. Stanford Exploration Project, Stanford University.

Friedlander, F. (1958). *Sound Pulses*. Cambridge University Press.

Gear, C. W. (1971). *Numerical initial value problems in ordinary differential equations*. Englewood Cliffs, N. J.: Prentice-Hall.

Gray, S. and May, W. (1994). Kirchhoff migration using eikonal equation traveltimes. *Geophysics*, 59:810–817.

Jiang, G. S. and Peng, D. P. (1997). Weighted ENO schemes for Hamilton-Jacobi equations. In *CAM97-29, UCLA*.

Jiang, G. S. and Shu, C. W. (1996). Efficient implementation of weighted ENO schemes. *J. Comput. Phys.*, 126:202–228.

Lions, P. L. (1982). *Generalized solutions of Hamilton-Jacobi equations*. Pitman Advanced Publishing Program.

Liu, X. D., Osher, S. J., and Chan, T. (1994). Weighted Essentially NonOscillatory schemes. *J. Comput. Phys.*, 115:200–212.

Osher, S. and Shu, C. W. (1991). High-order Essentially NonOscillatory schemes for Hamilton-Jacobi equations. *SIAM J. Num. Anal.*, 28:907–922.

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

Popovici, A. M. and Sethian, J. (1997). Three-dimensional traveltime computation using the fast marching method. In *67th SEG Annual Internat. Mtg. Expanded Abstracts*, pages 1778–1781. Soc. Expl. Geophys.

Qin, F., Luo, Y., Olsen, K. B., Cai, W., and Schuster, G. T. (1992). Finite difference solution of the eikonal equation along expanding wavefronts. *Geophysics*, 57:478–487.

Reshef, M. and Kosloff, D. (1986). Migration of common shot gathers. *Geophysics*, 51:324–331.

Schneider, W. A. J. (1995). Robust and efficient upwind finite-difference traveltime calculations in three dimensions. *Geophysics*, 60:1108–1117.

Schneider, W. A. J., Ranzinger, K., Balch, A., and Kruse, C. (1992). A dynamic programming approach to first arrival traveltime computation in media with arbitrarily distributed velocities. *Geophysics*, 57:39–50.

Shu, C. (1997). Essentially non-oscillatory and weighted essentially non-oscillatory schemes for hyperbolic conservation laws. In *ICASE No.97-65, NASA Langley Research Center*.

Symes, W. W. (1995). Mathematics of reflection seismology. In *Annual Report, The Rice Inversion Project*. Rice University. (http://www.trip.caam.rice.edu/).

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

Vidale, J. (1988). Finite-difference calculation of travel times. *Bull., Seis. Soc. Am.*, 78:2062–2076.

Vidale, J. E. and Houston, H. (1990). Rapid calculation of seismic amplitudes. *Geophysics*, 55:1504–1507.

Zhang, L. (1993). *Imaging by the wavefront propagation method*. PhD thesis, Stanford University, Stanford, CA94305.