Stanford Exploration Project, Report 61, January 13, 1998, pages 1–50

Finite-difference calculation of traveltimes

Jos van Trier[1]

keywords: *modeling, traveltime, velocity, ray tracing*

ABSTRACT

The Eikonal equation can be transformed into a flux-conservative equation, which can then be solved with standard finite-difference techniques. A first-order upwind finite-difference scheme proves accurate enough for seismic applications. The method is limited to rays traveling in one direction, and only calculates single-valued traveltime functions. The algorithm efficiently calculates traveltimes on a regular grid, and is useful in Kirchhoff migration and modeling.

INTRODUCTION

Traveltime calculations play an important role in many methods in seismic data processing. For example, in the Kirchhoff implementation of migration and modeling of seismic data Green's functions need to be calculated that describe the traveltime between survey points on the surface and depth points in the velocity model. These traveltimes are often calculated by ray tracing, and, since depth and surface points are normally distributed on a regular grid, the traveltimes along the rays are then interpolated onto the grid. For complicated velocity models, rays may cross or not penetrate shadow zones, making the interpolation cumbersome and computationally expensive.

Vidale (1988) recently described a method that calculates traveltimes directly on a regular grid. His approach is as follows: given the initial traveltime value on a certain grid point, the traveltimes for neighbouring points are extrapolated using a planar or circular wavefront approximation. Although Vidale's method represents an elegant alternative to interpolating traveltimes between rays, it has some drawbacks. First, to ensure stability, extrapolation can only be initiated from certain points on the grid (the so-called "relative minimum points"). Second, to achieve accuracy, at each grid point the curvature of the wavefront has to be determined, and, if necessary, an expensive circular wavefront extrapolation step has to be carried out. These drawbacks not only increase the computational cost, but also make it impossible to vectorize the computations.

[1]**email:** not available

W. Symes of Rice University introduced me to Vidale's method during a recent visit to Europe, and suggested an alternative scheme that overcomes the problems described above. Symes suggests writing the Eikonal equation as a flux equation, and then solving it with a standard upwind finite-difference scheme. This leads to a method that is computationally more efficient than Vidale's scheme. Just like Vidale's method, the finite-difference calculations correctly estimate traveltimes of first arrivals in caustics and of diffractions in shadow zones. One disadvantage however, is that traveltimes can only be calculated for rays traveling in one direction, and that the method does not calculate multi-valued traveltime curves. Although this may be a serious limitation for some applications, many Kirchhoff algorithms use single-valued traveltime functions anyway, and the method is readily applicable to those algorithms.

In this paper I first describe the method and the finite-difference implementation. Then I demonstrate the calculations with some examples, where I compare the results both with analytical solutions and with Vidale's method. Finally, I discuss some of the limitations and applications of the method.

EIKONAL EQUATION

The Eikonal equation in two dimensions is as follows:

$$\left(\tau_x\right)^2 \;+\; \left(\tau_z\right)^2 \;=\; s^2(x,z), \tag{1}$$

where $s(x,z)$ is the 2-dimensional slowness model and $\tau = \tau(x,z)$ is the traveltime field. Subscripts x and z denote partial derivatives with respect to x and z, respectively.

Flux-conservative equation

Similar to the conversion of the full wave equation into a one-way wave equation, we can rewrite the eikonal equation as a one-way equation that can be solved with standard techniques.

First, use $u = \tau_x$ to rewrite equation (1) as

$$\tau_z \;=\; \sqrt{s^2 - u^2}, \tag{2}$$

By choosing a positive sign in front of the square root, and by using $u = \tau_x$ instead of $u = \tau_z$ as the substitution variable, we limit ourselves to downward-traveling rays. I will come back to these choices in one of the later sections.

Second, take the derivative of this equation with respect to x:

$$\tau_{zx} \;=\; u_z \;=\; -F_x(u). \tag{3}$$

where the function $F(u)$ is defined as

$$F(u) \;=\; -\sqrt{s^2 - u^2}. \tag{4}$$

$F(u)$ is called the conserved flux; if $F(u) = 0$, the rays do not "flow" downward anymore, but travel horizontally.

The (one-way) eikonal equation now has the form of a flux-conservative equation This is a well-known equation in computational fluid dynamics, and it can be solved in many ways (for example, see Roache, 1976). The next section discusses one particular solution.

FINITE-DIFFERENCE SCHEME

Although sophisticated higher-order finite-difference schemes exist for the solution of the flux-conservative equation (Centrella and Wilson, 1984; Hawley et al., 1984), a first-order upwind finite-difference scheme is generally accurate enough for geophysical applications. I implemented a first-order method described by Engquist and Osher (1980).

Depending on the flow direction, upwind finite-difference schemes use a backward or forward finite-difference approximation to the derivative operator. The idea is that, although centered approximations are mathematically more accurate, the underlying physics may dictate otherwise. In other words, it is better to make a big error that is not transported, than to make a small one that blows up. The first-order forward and backward finite differences in space are, respectively:

$$\Delta_+ u \;=\; u_{j+1} - u_j, \qquad \Delta_- u = \; u_j - u_{j-1}, \tag{5}$$

where u_j is the discrete representation of $u(x)$ on the grid $x_j = j\Delta x$.

Using the above finite-difference scheme, equation (3) is approximated by:

$$\frac{u_j^{n+1} - u_j^n}{\Delta z} \;=\; \begin{cases} -\dfrac{\Delta_+ \, F(u_j^n)}{\Delta x} & \text{if } u_j^n \le 0; \\[2ex] -\dfrac{\Delta_- \, F(u_j^n)}{\Delta x} & \text{if } u_j^n > 0, \end{cases} \tag{6}$$

with n the discrete-depth index of the grid: $z^n = n\Delta z$. However, the resulting scheme is unstable when u is discontinuous (when the field is going through a "shock"). The discontinuity occurs at the source: going from left to right through the source, u changes sign from $-s$ to $+s$, i.e., rays change direction from left to right.

Engquist and Osher solve the unstability problem in the shock region with the following scheme:

$$u_j^{n+1} = u_j^n \;-\; \frac{\Delta z}{\Delta x}\left(\Delta_+ F_-(u_j^n) \;+\; \Delta_- F_+(u_j^n)\right), \tag{7}$$

where

$$F_+(u) \;=\; F(\max(u,\overline{u})), \qquad F_-(u) \;=\; F(\min(u,\overline{u})), \tag{8}$$

and $\overline{u}$ is the stagnation point, i.e., $F'(\overline{u}) = 0$. $\overline{u} = 0$ for the function F under consideration here. Engquist and Osher's scheme reduces to the standard first-order scheme (6) away from the source, and is nonlinearly stable near the source.

Initial and boundary conditions

Equation (8) is used to extrapolate u downward in z. Initial values for u have to be specified at the surface. In many cases the velocity model is constant at the surface, and the initial conditions can be calculated analytically for a given source position. For models with non-constant surface velocities, traveltimes can be computed along the surface with a simple 1-D finite-difference approximation. The derivative of the traveltime function with respect to x then gives u.

When one assumes outgoing rays at the boundaries, one-sided finite differences can be used at the left and right side of the model. These boundary conditions are of the same order as the scheme inside the model, and generally do not cause any problems. After u has been computed, traveltimes are calculated by integrating u over x using Simpson's rule.

EXAMPLES

In the first example I calculate traveltimes for a constant-velocity medium with a velocity of 2.5 km/s. The grid is evenly sampled in depth and laterally, with a sample interval of 10 m. Figure **??** shows the traveltime function for a source on the surface at the middle of the model, and Figure **??** displays the difference with the analytical solution. Errors accumulate as depth increases, and are largest on the outer sides of the model, where rays travel at angles closer to the horizontal. Since the finite-difference scheme is designed for downward traveling rays, this behavior is expected. The maximum error is about .6 ms, which is one order of magnitude smaller than the standard time sampling interval of 4 ms (see Figure **??**). Finite-difference traveltime calculations of one shot on a 100×100 grid take about .1 s in CPU time on the Convex C-1.

Figure **??** shows the difference between Vidale's scheme and the analytical solution. I have only implemented the plane wave extrapolation method, and errors can probably be reduced if a combination of plane and circular wave extrapolation is used. The errors are largest away from the vertical, diagonal and horizontal direction, where the plane wave approximation breaks down. The errors at the bottom of the model are of the same order of magnitude as the errors in the method described here (see again Figure **??**). However, Vidale's scheme does not vectorize, and on the Convex C-1 the finite-difference calculations are about 5-10 times faster than his method.

The next example illustrates the calculations for a more complicated model. The model is shown in Figure **??**; it consists of 3 layers and a wedge intrusion. The velocity in the top layer is 2 km/s, the middle layer has a velocity of 1.75 km/s, and the bottom velocity is 2.5 km/s. The velocity in the wedge that intrudes the middle layer from the right is 2.75 km/s. Figure **??** shows the result of tracing rays from the surface downwards; because of the velocity contrasts, rays cross and shadow zones are apparent.

As is obvious from Figure **??**, interpolating traveltimes from the rays onto the grid is not easy for this model. However, the finite-difference calculation correctly fills in the problem areas as can be seen in Figure **??**: the contour lines in the plot reveal the correct curvature of the wave fronts in the high- and low-velocity regions. Figure **??** provides a more quantitative check; it compares interpolated traveltimes of the rays at the bottom of the model with the finite-difference traveltimes. Since the rays are calculated for a spline model that is fitted to the grid model (Van Trier, 1988), some discrepancies may be expected, especially if one realizes that a slight change in velocity contrast can drastically change the direction of the ray. Except for the region near the shadow zone, the traveltime curves match reasonably well.

LIMITATIONS

The flux-conservative form of the Eikonal equation is essentially a one-way equation. This means that rays can only be propagated in one direction, and that only single-valued traveltime curves are computed. So far, I have been discussing downward traveling rays, but if $u = \tau_z$ instead of $u = \tau_x$ is used in equation (2), one can calculate traveltimes of rays that travel sideways. Initial conditions then have to be specified along a vertical line.

In the downward-propagation finite-difference scheme, the numerical errors get large if $F\Delta z \ll \Delta x$, which occurs when rays travel almost horizontally. This numerical noise can be reduced by decreasing the stepsize in z when rays have large propagation angles at the current depth level, and increasing it when the propagation angles get smaller. If greater accuracy is still required, an other alternative is to use higher-order finite difference schemes.

Since the method does not calculate ray paths, it is only of limited use in tomographic methods. It is possible, however, to calculate ray directions on the grid from the traveltime gradient. One component of the slowness vector, τ_x, is already available as u, the other one, τ_z, is calculated using $\tau_z = \sqrt{s^2 - u^2}$.

APPLICATION IN KIRCHHOFF METHODS

My main use for the method has been in Kirchhoff migration and modeling. Although ray tracing in itself may be faster than finite-difference calculations, the interpolation of traveltimes along rays onto the grid can be quite cumbersome and expensive: special care is needed in handling crossing rays and shadow zones. The finite-difference method does not handle multi-valued traveltime functions, however, and ray tracing is still necessary if one wants to correctly include triplications in the migration.

The Green's functions that are used in Kirchhoff methods describe traveltimes between depth points in the model and survey points on the surface. One can precompute these traveltime maps with the described method for densely spaced surface points, store them on disk (if enough disk space is available), and then read them when needed in the Kirchhoff summation. If a map for a certain surface point is not on disk, a simple linear interpolation of traveltimes in two neighboring maps is accurate enough for most purposes.

The amplitudes terms in the Green's functions can be approximately estimated from the traveltime function: the geometrical spreading and obliquity terms are approximate functions of traveltime or its gradient. Ray propagation angles are readily available from the gradient vector of the traveltime field (see previous section), and can be used accordingly in amplitude calculations.

CONCLUSIONS

I have presented a method that greatly simplifies traveltime calculations on a regular grid. The algorithm has the advantage over Vidale's method that all points can be treated equally, i.e., no minimum traveltime points have to be found at each extrapolation step, and no choices based on local wavefront curvature have to be made. The computations are vectorizable, and allow efficient calculation of Greens's functions in Kirchhoff methods.

ACKNOWLEDGMENTS

I would like to thank Bill Symes of Rice University for bringing this topic to my attention, and for the interesting discussions we had on velocity analysis. This paper is mostly due to his suggestions.

REFERENCES

Centrella, J., and Wilson, J., 1984, Astrophysical Journal Supplement, **54**, 229-249.

Engquist, B., and Osher, S., 1980, Stable and entropy satisfying approximations for transonic flow calculations: Math. Comp., **34**, no. 149, 45-75.

Hawley, J., Smarr, L., and Wilson, J., 1984, Astrophysical Journal Supplement, **55**, 211-246.

Roache, P., 1976, Computational fluid dynamics: Hermosa, Albuquerque, New Mexico.

Van Trier, J., 1988, Vectorized initial-value ray tracing for arbitrary velocity models: SEP–**57**, 279-288.

Vidale, J., 1988, Finite-difference calculation of travel times: Bull., Seis. Soc. Am., **78**, no. 6, 2062-2076.