Chapter 3

Finite-difference traveltime calculations

3.1 INTRODUCTION

Traveltime calculations play a central role both in the prestack migration of the seismic data and in the velocity estimation. These traveltimes typically describe the time needed for a seismic wave to travel between survey points on the surface and depth points in the subsurface. Traditionally, traveltimes are computed by ray tracing (Červený, 1987), and, because depth and surface points are usually distributed on a regular grid, the traveltimes along the rays are then interpolated onto the grid. For complicated velocity models, rays may cross each other or not penetrate shadow zones; the interpolation is thus cumbersome and computationally expensive.

Recently, several methods have been introduced to calculate traveltimes directly on a regular grid. In calculating the depth gradient of traveltimes, Reshef and Kosloff (1986) use a finite-difference approximation to the eikonal equation, which they then integrate with a Runge-Kutta method. Vidale (1988) also approximates the eikonal equation by finite differences, but he solves directly for traveltime using a planar or circular wavefront extrapolation.

Although these methods are elegant alternatives to interpolating traveltimes between rays, and are far more efficient computationally, they have some drawbacks. Reshef and Kosloff's method is unstable for models with large velocity contrasts; in regions with velocity discontinuities, traveltimes have to be calculated by alternative methods, such as ray tracing. The principal drawback in Vidale's method is that the points on the computational front must be ordered, with calculations starting at the so-called "relative minimum points". This ordering step complicates the algorithm, and, because it is not

vectorizable, makes the method computationally expensive.

Here I present an alternative scheme that overcomes the problems described above. The scheme is based on the observation that the components of the traveltime gradient satisfy hyperbolic conservation laws (Van Trier and Symes, 1990). Conservation laws are important in fluid mechanics, and a great deal of effort has been devoted to their numerical solution. Accordingly, an "off-the-shelf," finite-difference scheme can be used to compute the traveltime gradients. The scheme is an upwind finite-difference method, which is stable because it mimics the behavior of fluid flow by taking its information from upstream. Furthermore, the method involves none of the sorting implicit in Vidale's approach, and vectorizes fully.

An in-depth review of conservation laws and upwind finite-difference schemes is discussed in Van Trier and Symes, (1990). In this chapter I only describe the relevant conservation law and the upwind finite-difference scheme. I demonstrate the computations with a synthetic example. Finally, I discuss some of the limitations of the method.

3.2 EIKONAL EQUATION

The eikonal equation in two dimensions is as follows (Aki and Richards, 1980, ch.4):

$$\left(\frac{\partial t}{\partial x}\right)^2 + \left(\frac{\partial t}{\partial z}\right)^2 = s^2(x,z), \tag{3.1}$$

where $s(x,z)$ is the 2-dimensional slowness model and $t = t(x,z)$ is the traveltime field.

3.2.1 Hyperbolic conservation law

I now show that the gradient components of the eikonal equation satisfy a hyperbolic conservation law. First, use $u = \partial t/\partial x$ to rewrite equation (3.1) as

$$\frac{\partial t}{\partial z} = \sqrt{s^2 - u^2}. \tag{3.2}$$

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

$$\frac{\partial u}{\partial z} = \frac{\partial F(u)}{\partial x}, \tag{3.3}$$

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

$$F(u) \;=\; \sqrt{s^2 - u^2}. \tag{3.4}$$

$F(u)$ is called the conserved flux; if $F(u) = 0$, the rays do not "flow" downward anymore, but travel horizontally. Thus by choosing a positive sign in front of the square root in equation (3.2), and by using $u = \partial t/\partial x$ instead of $w = \partial t/\partial z$ as the substitution variable, time fields are limited to those with downward-traveling rays.

By following analogous reasoning for the z-derivative w of the time field, and by making the appropriate choices for the square roots, equations can be built for rays traveling in other directions. These other equations will have to be solved, for instance, when the field of a point source is being computed, because the rays then move in all directions.

An alternative approach, which I follow here, is to write the eikonal equation in polar coordinates (r, θ),

$$\left(\frac{\partial t}{\partial r}\right)^2 \;+\; \left(\frac{1}{r}\frac{\partial t}{\partial \theta}\right)^2 \;=\; s^2(r, \theta), \tag{3.5}$$

and to solve it along expanding circular fronts. The conserved-flux function in polar coordinates becomes

$$F(u) \;=\; \sqrt{s^2 - \frac{u^2}{r^2}}, \tag{3.6}$$

with $u = \partial t/\partial \theta$ satisfying

$$\frac{\partial u}{\partial r} \;=\; \frac{\partial F(u)}{\partial \theta}. \tag{3.7}$$

Solving the equation in polar coordinates has the advantage that initial conditions are easy to specify (see section 3.3.1). Also, in polar coordinates rays are less likely to become parallel to the computational fronts than they are in Cartesian coordinates. As I discuss in section 3.5, this feature can overcome some of the main limitations of the method.

Equation (3.7), like equation (3.3), now has the form of a hyperbolic conservation law equation. In computational fluid dynamics, many methods have been developed for solving these conservation laws (see Roache, 1976). The next section discusses one particular method.

3.3 FINITE-DIFFERENCE SCHEME

Although sophisticated, higher-order finite-difference schemes exist for the solution of hyperbolic conservation laws (Centrella and Wilson, 1984; Hawley et al., 1984; Harten et al., 1987), a first-order, upwind finite-difference scheme is accurate enough for tomographic applications. I implemented a first-order method described by Engquist and Osher (1980).

The first-order backward and forward finite differences in space are, respectively,

$$\begin{aligned} \Delta_- u &= u_j - u_{j-1}; \\ \Delta_+ u &= u_{j+1} - u_j, \end{aligned} \tag{3.8}$$

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

The basic upwind scheme for equation (3.7) is

$$\frac{u_j^{n+1} - u_j^n}{\Delta r} = \frac{1}{\Delta\theta} \begin{cases} \Delta_- F(u_j^n) & \text{if } F'(u_j^n) \text{ and } F'(u_{j-1}^n) \le 0; \\ \Delta_+ F(u_j^n) & \text{if } F'(u_j^n) \text{ and } F'(u_{j+1}^n) > 0, \end{cases} \tag{3.9}$$

with n the discrete-radius index of the grid, $r^n = n\Delta r$. The direction of the finite-difference operator depends on $F'(u)$: when the flux increases from left to right, flow is to the left; when it decreases, flow is to the right (see Van Trier and Symes, 1990). Note that an equivalent scheme can be set up in Cartesian coordinates, where r is replaced by z, and θ by x.

This specification is incomplete, of course. The various upwind schemes differ in the way in which the intermediate cases are handled; i.e. when the sign of $F'(u)$ changes amongst the three points of the stencil. The scheme of Engquist and Osher handles the intermediate cases via the formula:

$$u_j^{n+1} = u_j^n + \frac{\Delta r}{\Delta\theta} \left(\Delta_+ F_-(u_j^n) + \Delta_- F_+(u_j^n) \right). \tag{3.10}$$

Here

$$\begin{aligned} F_-(u) &= F(\min(u,\overline{u})) \text{ and} \\ F_+(u) &= F(\max(u,\overline{u})), \end{aligned} \tag{3.11}$$

with $\overline{u}$ the stagnation point: $F'(\overline{u}) = 0$. $\overline{u} = 0$ for the function F under consideration here. Engquist and Osher's scheme reduces to the standard first-order scheme (3.9) when u has a consistent sign, and is stable near gradient discontinuities.

Note that the specification given above is still not complete: the slowness is position-dependent, therefore so is the flux. This dependence is suppressed in the notation, but it must be respected. Some obvious ways of evaluating the flux lead to inconsistent schemes. In general it is sufficient to verify that the elementary upwind differences above are recovered when the sign of u is consistent. In my implementation, I take the slowness locally constant at its value at u_j^n.

Because an explicit scheme is used, each step (fixed by the input slowness grid) may require several partial steps to satisfy the Courant-Friedrichs-Lewy stability condition. The appropriate partial steps are determined adaptively, and the full step is always taken if it is stable. This contrasts with Vidale's code, which uses an implicit upwind scheme.

3.3.1 Initial and boundary conditions

The numerical scheme is presented in polar coordinates for the purpose of solving the point-source problem. In polar coordinates the initial condition at the source is easily specified as $u(0,\theta) = 0$. However, the same problem may be solved in Cartesian coordinates if calculations are done on a sequence of expanding rectangular fronts. These rectangular fronts are complicated to implement: one has to solve for different components of the traveltime gradient at the different sides of the rectangle (for u, the lateral gradient, on the horizontal edges, and for w, the vertical gradient, on the vertical edges). However, the rectangular approach approach saves the cost of mapping the slowness and traveltime fields to and from polar coordinates. The rectangular computational fronts are also used by Vidale (1988).

If one assumes outgoing rays at the boundaries, one-sided finite-differences can be used at the left and right sides of the model. These boundary conditions are of the same order as the scheme inside the model, and do not cause any problems. (Problems occur only

when the traveltime gradient is pointing inward at the boundary, in which case the solution cannot be computed anyway because it depends on information outside the computational domain.) After u has been computed, traveltimes are found from an integration of $\partial t/\partial r$ over r with a simple trapezoidal rule, where $\partial t/\partial r$ is found from equation (3.5).

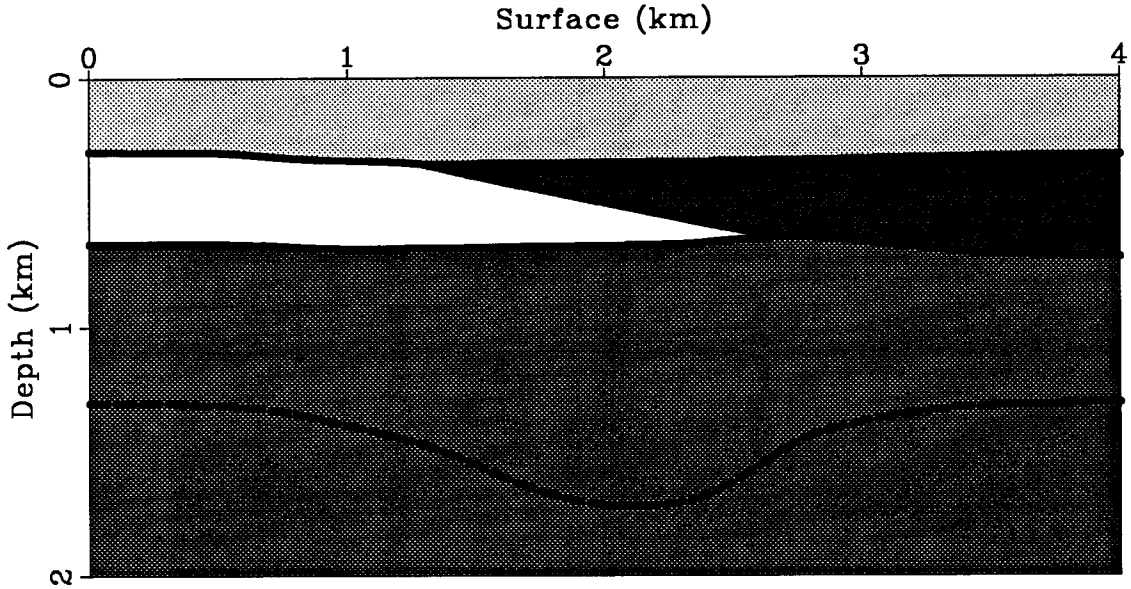

FIG. 3.1. Wedge model. The grid spacing is 10 × 10 m. Low intensities denote low velocities. The interfaces between the different layers are represented by solid lines in the figure. The dashed line denotes an imaginary reflector in the bottom layer. Figure 4.3 shows reflection events that correspond to these reflectors.

3.4 EXAMPLE

The example illustrates the traveltime calculations for a structural model. The model is shown in Figure 3.1; 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 layer's 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 3.2 shows the result of tracing rays through a smoothed version of the model. The smoothing causes the rays to bend or turn in regions with a large velocity gradient.

As is obvious from Figure 3.2, interpolating traveltimes from the rays onto the grid is not easy for this model; some parts of the model are not illuminated by rays, and in

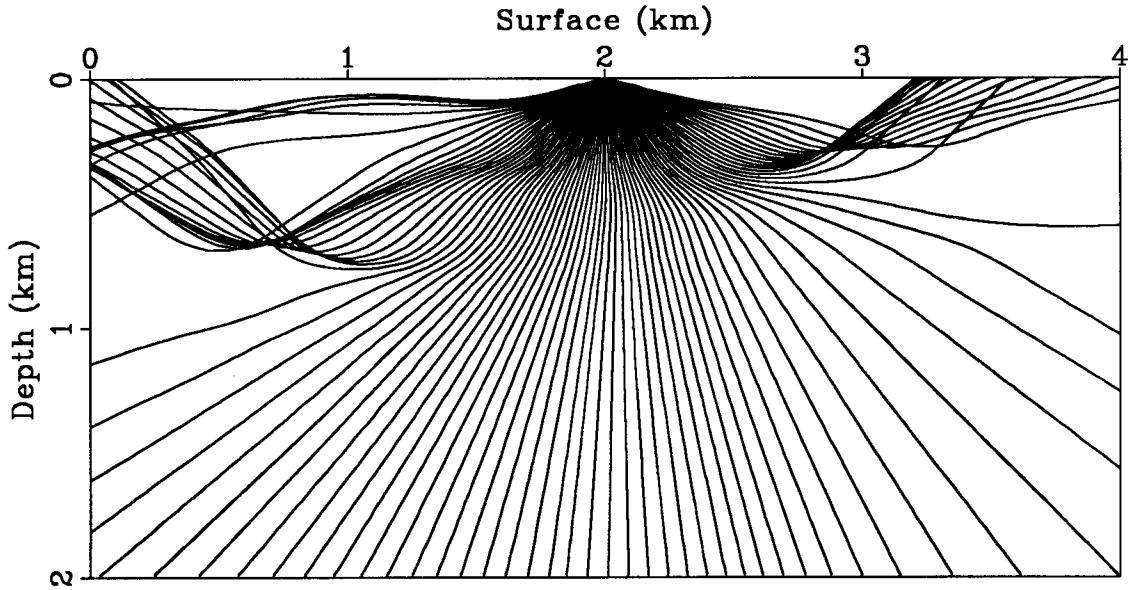

FIG. 3.2. Rays traced through a smoothed version of the model in Figure 3.1.

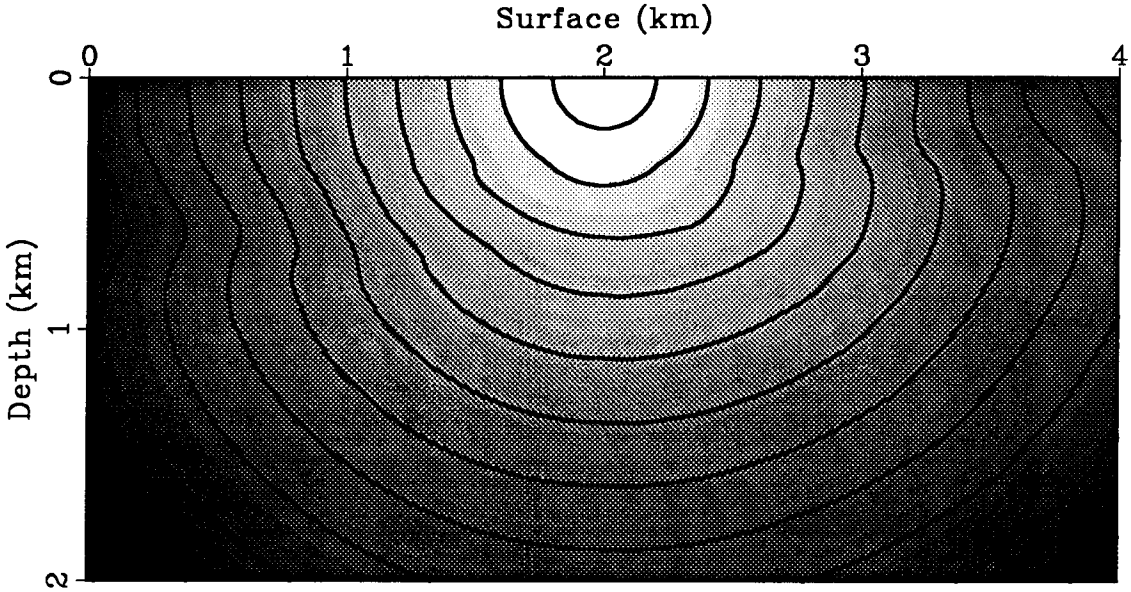

FIG. 3.3. Finite-difference traveltimes calculated for the model of Figure 3.1. Overlain on the figure are contour lines of the traveltime field. The contour interval is .1 s.

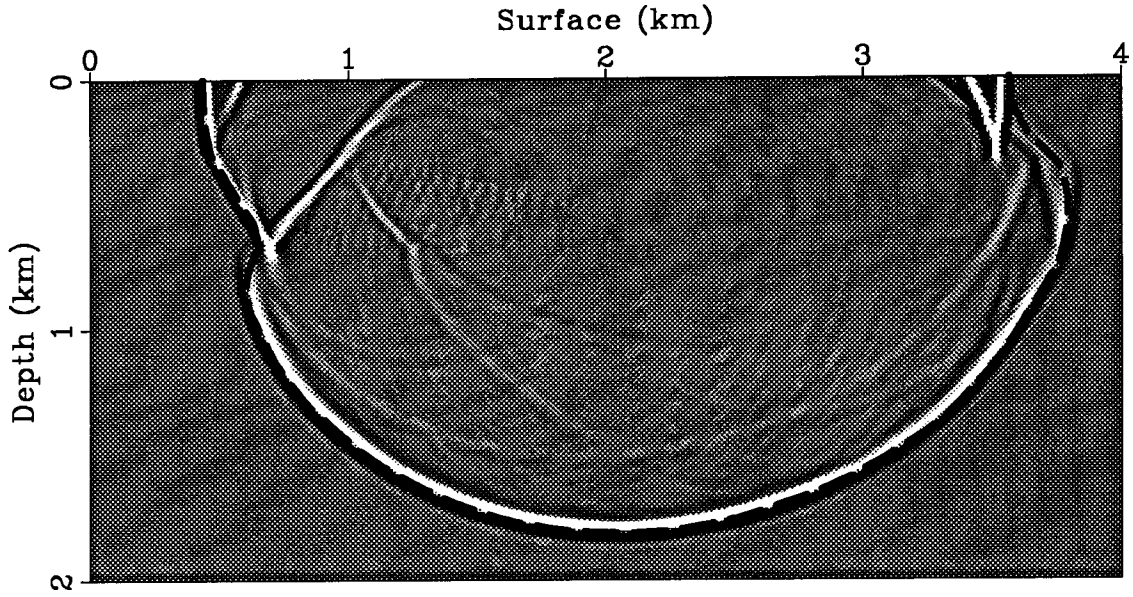

FIG. 3.4. Comparison of the wave field computed by wave-equation modeling with the traveltime field calculated by upwind finite-differences. The intensity plot shows a snapshot of the wave field at .7s; the overlain dashed curve is the .7s-contour of the traveltime function (see Figure 3.3).

some other parts rays cross. However, the finite-difference calculation apparently fills in the problem areas correctly, as can be seen in Figure 3.3: the contour lines in the plot reveal the correct curvature of the wave fronts in the high- and low-velocity regions.

This result is verified in Figure 3.4, which shows the result of finite-difference modeling of acoustic waves propagating through the model of Figure 3.1. The figure displays a snapshot of the wave field at .7 s. Also shown in the figure is the .7s-contour line of the traveltime function (Figure 3.3). Barring some discrepancies due to the limited bandwidth and dispersion of the source wavelet, the contour exactly follows the first-arrival wave front. In particular note the match between wave field and traveltime function in the upper-right corner of the model, where the refracted wave travels in front of the direct wave.

3.5 LIMITATIONS

The main limitation of the method was noted before: it computes only the first-arrival field, as that is the unique, stable solution to the conservation law described in section 3.2.1 (Van Trier and Symes, 1990). However, this restriction does not mean that reflection

events computed with the finite-difference traveltime calculations cannot be multi-valued. Section 4.3 presents an example of reflection events corresponding to the reflectors in Figure 3.1, where one of the events shows a triplication.

The choice of the computational fronts poses another limitation: the solution evolves in time, but it is computed by circular or rectangular fronts that expand in space. If the time field to be computed does not have an outward-pointing gradient at each front—i.e., if the time gradient becomes parallel to the computational front (a turned ray)—then the square root in the flux function F (equations (3.4) and (3.6)) becomes imaginary and the calculation stops. It is easy to construct examples that cause any a-priori prescribed family of computational fronts to fail to have this essential outflow property. Therefore this code cannot be relied upon completely; it succeeds only when the user has general knowledge of the shape of the time field.