Chapter 4

Finite-difference Traveltimes

Probable-Possible, my black hen,
She lays eggs in the Relative When.
She doesn't lay eggs in the Positive Now
Because she's unable to postulate how.

– Frederick Winsor (1958), from 4.2BSD "fortune"

In previous chapters I have shown some of the complexities possible in anisotropic wave propagation. I showed both theoretical slowness and impulse-response surfaces, and finite-difference wavefields. Each of the three-dimensional finite-difference wavefield examples took from one to several hours to calculate on our computer, despite the algorithm's high efficiency and suitability to the vectorizing architecture of our computer (Etgen and Dellinger, 1989). In comparison the impulse-response surfaces were relatively cheap to calculate, at most a few minutes of CPU apiece (and that with no special care taken at all to make the algorithm efficient). Despite their extreme simplicity, the impulse-response surfaces for the most part overlayed the full-waveform finite-difference results very well. Unfortunately the algorithm I used to calculate the impulse-response surfaces uses brute force solution of the Christoffel equation and is only applicable to homogeneous media. In this chapter I demonstrate an approximate but computationally inexpensive method, finite-difference traveltimes, that allows the calculation of impulse-response surfaces in *heterogeneous* media.

Vidale (1988) first introduced a method for calculating traveltimes using a finite-difference method. His method is efficient but somewhat cumbersome to implement and not vectorizable. Vidale's work inspired Van Trier and Symes (1991) to develop a related method that also calculates finite-difference traveltimes but in a computationally streamlined way. Computational simplicity is the great strength of their method. Competitors such as ray tracing can also be very efficient, but require complex codes that are famously difficult to perfect and debug.

Neither Vidale's nor Van Trier and Symes' method includes anisotropy as published. Since Vidale's method is based directly on plane wave propagation, adding anisotropy to that algorithm is trivial theoretically. In practice it is not so simple, because numerical errors can be difficult to distinguish from the desired anisotropy (Samec, 1991). I have chosen to base my anisotropic algorithm on Van Trier and Symes' isotropic method.

4.1 Theory

I begin by summarizing the theory behind Van Trier and Symes' original method. In the process I show how the isotropic dispersion relation[1] enters their derivation, and how it can be replaced with an anisotropic one.

Van Trier and Symes' derivation begins with the isotropic eikonal equation,

$$\left(\frac{\partial \tau}{\partial x}\right)^2 + \left(\frac{\partial \tau}{\partial z}\right)^2 = s^2, \tag{4.1}$$

where $\tau(x,z)$ gives the traveltime of an event at each point in space, and $s(x,z)$ gives the slowness (inverse velocity) at each point in space.

This equation can be recast as a depth-extrapolation equation. First solve for $\partial\tau/\partial z$:

$$\frac{\partial \tau}{\partial z} = \sqrt{s^2 - \left(\frac{\partial \tau}{\partial x}\right)^2}; \tag{4.2}$$

then differentiate with respect to x:

$$\frac{\partial^2 \tau}{\partial x \partial z} = \frac{\partial}{\partial x}\sqrt{s^2 - \left(\frac{\partial \tau}{\partial x}\right)^2}. \tag{4.3}$$

[1] If you are confused by my use of the term "dispersion relation", please read the footnote on page 16.

Finally, assuming

$$\frac{\partial^2 \tau}{\partial z \partial x} = \frac{\partial^2 \tau}{\partial x \partial z} \tag{4.4}$$

and letting $u = \partial\tau/\partial x$, we obtain

$$\frac{\partial u}{\partial z} = \frac{\partial}{\partial x}\sqrt{s^2 - u^2}. \tag{4.5}$$

This is a depth-extrapolation equation for $u(x)$; $\tau(x,z)$ can be recovered from $u(x,z)$ by specifying initial conditions ($\tau = 0$ at the source) and integrating.

Equations of the form

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x}F(u) \tag{4.6}$$

are called flux-conservative equations, with F the "conserved flux" (Press et al., 1988). For the isotropic traveltime extrapolation equation (4.5), the flux is

$$F(u) = -\sqrt{s^2 - u^2}. \tag{4.7}$$

4.1.1 Deriving the eikonal equation

Our goal is to find a flux-conservative law analogous to equation (4.5) for anisotropic travel-time extrapolation. Equation (4.1), the eikonal equation, contains the physics of the problem, so I begin by rederiving it from basic principles.

Figure 4.1 shows a tiny portion of a wavefront at two closely spaced instants in time (ΔT apart). On this scale, the wavefront can be approximated as a plane wave. The plane wave is moving along in some arbitrary direction with a velocity V, so it travels a distance $V\Delta T$ between the two times. At the same time, the plane wave also moves a distance ΔX along the X axis and ΔZ along the Z axis.

The triangle in the lower right of Figure 4.1 has sides of length ΔX and ΔZ meeting at a right angle, and a height of $V\Delta T$ (if the hypotenuse is considered the base). Three applications of Pythagoras' theorem (one for each of the three right triangles involved)

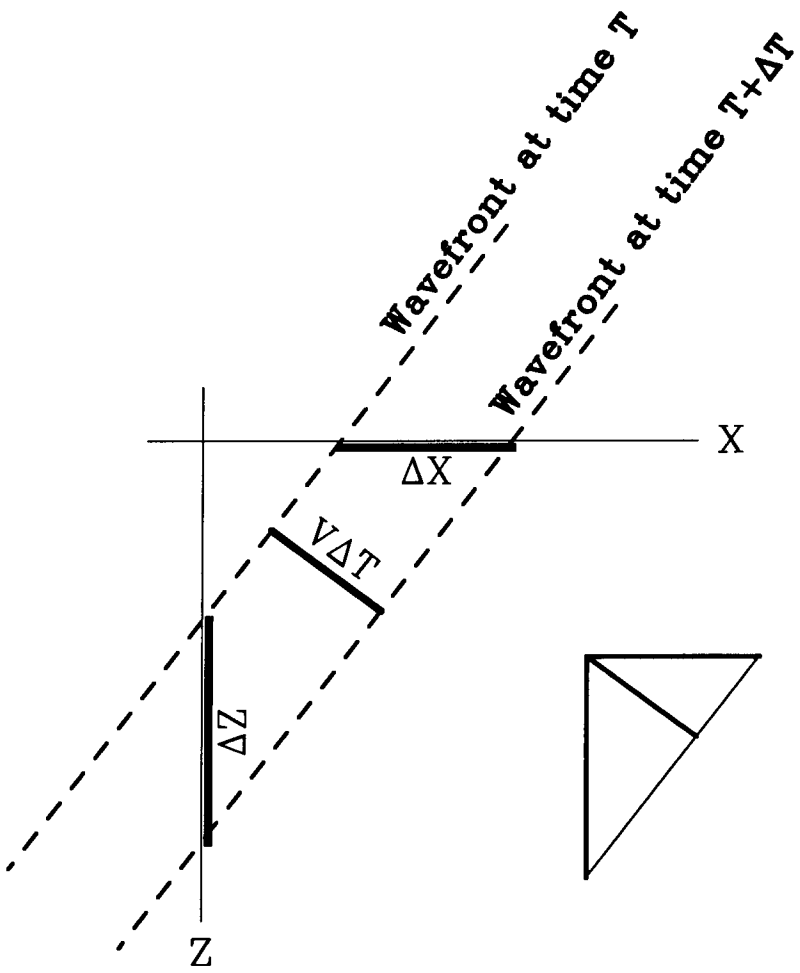

FIG. 4.1. Geometrical properties of plane-wave propagation.

and a little algebra show that

$$\left(V\Delta T\right)^2 = \frac{\Delta X^2 \Delta Z^2}{\left(\Delta X^2 + \Delta Z^2\right)}. \tag{4.8}$$

A bit more algebra and we finally arrive at

$$\left(\frac{\Delta T}{\Delta X}\right)^2 + \left(\frac{\Delta T}{\Delta Z}\right)^2 = \frac{1}{V^2} = s^2. \tag{4.9}$$

By letting $\Delta T \rightarrow 0$, we arrive at equation (4.1).

4.1.2 What does the eikonal equation mean?

From the previous paragraphs, we can see that $\partial\tau/\partial x$ in equation (4.1) (or $\Delta T/\Delta X$ in equation (4.9)) is just the horizontal phase slowness. For a monochromatic plane wave this is equal to k_x/ω. Similarly, $\partial\tau/\partial z = k_z/\omega$. This lets us rewrite equation (4.1) as a dispersion relation:

$$\left(\frac{k_x}{\omega}\right)^2 + \left(\frac{k_z}{\omega}\right)^2 = s^2. \tag{4.10}$$

The dispersion relation is just the Fourier-transformed homogeneous wave equation. The eikonal equation is a local plane-wave approximation based on the dispersion relation, and is also valid for heterogeneous media.

Adding anisotropy

In section 4.1 we saw how $F(u)$ in equation (4.7) derives from the isotropic eikonal equation. We just saw how the isotropic eikonal equation relates to the isotropic dispersion relation. By putting these together, now we can recast $F(u)$ as a dispersion relation. $F(u)$ then has a simple interpretation: u is a horizontal phase slowness k_x/ω, and $F(u)$ is a vertical phase slowness k_z/ω. To incorporate anisotropy, replace the isotropic dispersion relation with an anisotropic one; the result should be a traveltime extrapolation equation valid for the specified anisotropic medium.

4.1.3 What is conserved?

Should we expect any details of the numerical implementation of the anisotropic algorithm to differ from the isotropic case? We still have a flux-conservative equation, only F has changed; but what "flux" does our new equation preserve?

In the isotropic case it is tempting to think of some sort of "flow" out from the source as being conserved, such as that shown in Figure 4.2. Here u is interpreted as a horizontal flow, and $F(u)$ as a vertical flow. This makes intuitive sense; we know u represents horizontal phase slowness and $F(u)$ represents vertical phase slowness.

Figure 4.3 shows a strongly anisotropic version of the same picture. Since the medium is homogeneous the picture should have a scale invariance about the source, and any "flow" must be along straight radial lines emanating from the source. This was true in

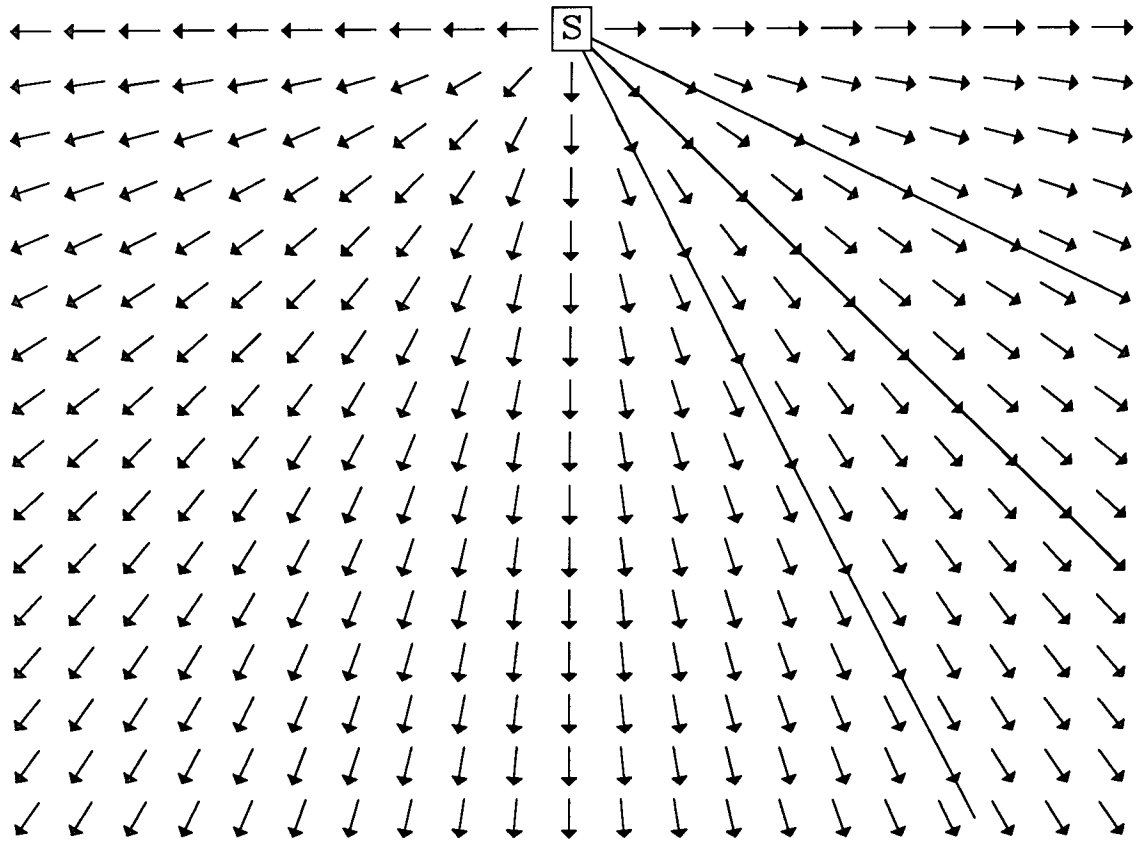

FIG. 4.2. Adding u as horizontal flux and $F(u)$ as vertical flux together as vectors produces an intuitively appealing picture that apparently shows an isotropic "flux" flowing away from a point source along radial lines. Unfortunately it is not consistent with the mathematics, as we can see by looking at the anisotropic version of this picture, Figure 4.3.

the isotropic picture, but it isn't true here. What is wrong? To answer this question we must examine the principles underlying the flux-conservative equation.

Flux-conservative equations

The canonical example of a flux-conservative equation considers the case of current flowing down a wire. For this case we extrapolate in time, not depth, and $F(u) = vu$, where v is the velocity, u gives the charge density of electrons at each point on a one-dimensional wire, and $F(u)$ gives the current flow. Equation (4.6) repeated here for convenience,

$$\frac{\partial u}{\partial t} = -\frac{\partial}{\partial x} F(u), \tag{4.6}$$

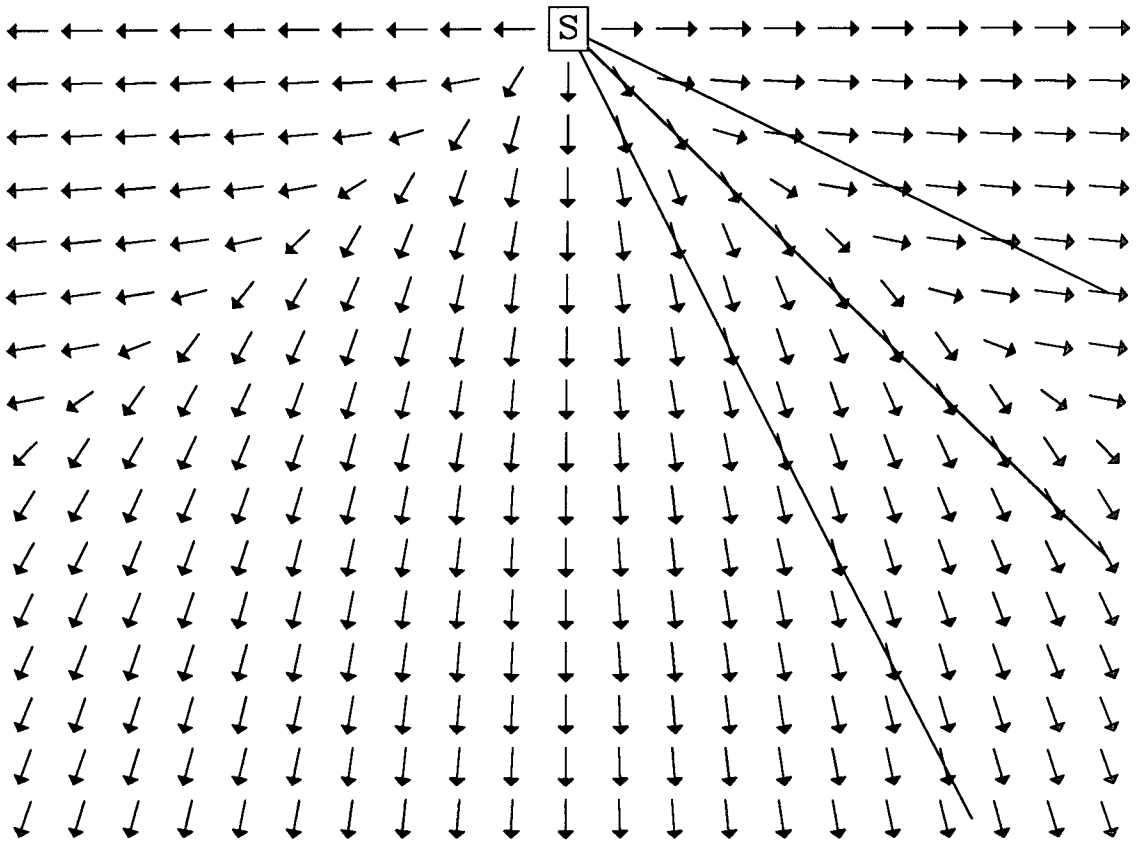

FIG. 4.3. An anisotropic version of Figure 4.2. Since the medium is homogeneous, the picture should be scale-invariant about the source, and any "flow" from the source should follow straight radial lines. The vectors here do not point along the three sample radial lines shown, so they cannot represent the true conserved flux. (The figure is indeed scale-invariant: the deviation of the "flow" vectors from the radial lines is constant with distance, except for a slight numerical error near the top.) Figure 4.4 shows the correct (although counterintuitive) answer.

then simply says that electric charge is neither created nor destroyed. If the current flow into some region is less than the flow going out (i.e., $\partial F(u)/\partial x > 0$), then the amount of charge in the region must be decreasing by the same amount ($\partial u/\partial t < 0$).

We can also apply the same flux-conservative equation to cases of two-dimensional time-independent fluid flow, in which case the equation is written

$$\frac{\partial u}{\partial z} = -\frac{\partial}{\partial x} F(u). \tag{4.11}$$

In this equation u gives the vertical component of the fluid flow, and $F(u)$ gives the horizontal component. This flux-conservative equation states that any fluid that flowed

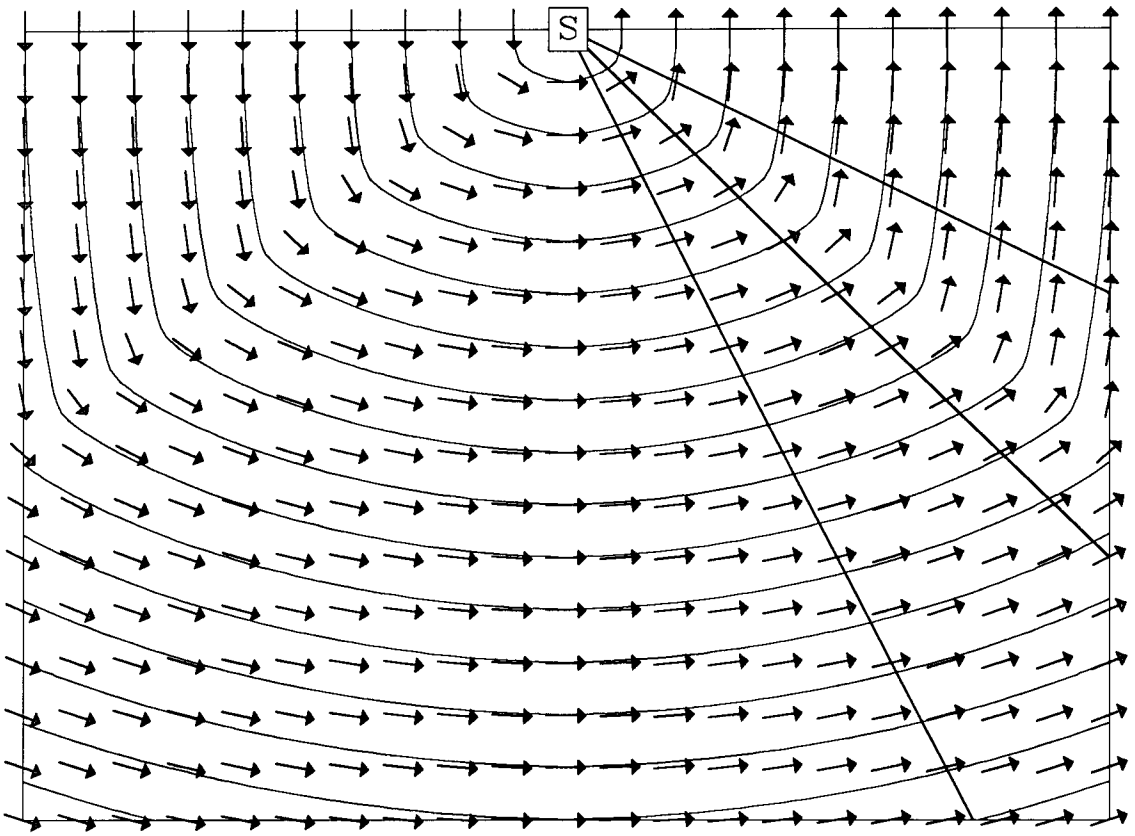

FIG. 4.4. The conserved flux in finite-difference traveltimes. The flux points along tangents to the contours of constant traveltime; the contours function as the flow lines of the conserved fluid.

in to some region from the left but didn't flow out again on the right must have flowed out the bottom or top (depending on the sign of u).

Even though for our traveltime extrapolation problem u represents a horizontal phase slowness, mathematically it represents the vertical component of the conserved flux. Similarly, although $F(u)$ represents a vertical phase slowness, it represents the horizontal component of the conserved flux. Figure 4.4 shows the correct "fluid flow" for our problem: the contour lines of constant traveltime are the flow lines of the conserved "fluid"!

Van Trier and Symes (1991) discuss in their paper that their method is only applicable for finding the first arrival. Here we have a simple explanation of why this should be so. The contour lines of constant traveltime are the conserved flow lines. The conserved fluid can neither be created nor destroyed, and so the contours of constant τ can also neither be created nor destroyed. This is only true if τ is continuous, a condition that is always

met by the first-arrival field.

4.2 Implementation

In the previous section, I showed that theoretically at least creating an anisotropic traveltime extrapolation algorithm is as simple as replacing the isotropic $F(u)$ with an anisotropic dispersion relation. In this section I will show how to implement this as an algorithm, again following Van Trier and Symes (1991) as closely as possible. In practice, the anisotropic algorithm proves to be considerably more cumbersome to implement than the isotropic one.

4.2.1 The Engquist-Osher finite-difference scheme

There are many possible finite-difference techniques for solving flux-conservative equations like Equation (4.6); they are all valid if the medium is homogeneous, so F depends only on u and not also on x and z. Heterogeneities are a problem because at such points the traveltime field $\tau(x,z)$ has a discontinuous first derivative, creating discontinuities in u and $F(u)$ and violating the smoothness assumption implicit in equation (4.4).

Mathematically the solution is to add a viscosity term to the equation to "smooth out" the discontinuities. In practice the discretization errors of the various possible finite-difference schemes correspond to some sort of extra term in the equation being solved. The numerical viscosity term in the Engquist-Osher scheme (Engquist and Osher, 1980) used by Van Trier and Symes allows it to remain stable and reasonably accurate near derivative discontinuities of τ. No other first-order method I have yet tried does this.

The Engquist-Osher scheme as used by Van Trier and Symes is

$$u(x,z+\Delta z) = u(x,z) + \frac{\Delta z}{\Delta x}\Big((F^-_{+1} - F^-_{+0}) + (F^+_{-0} - F^+_{-1})\Big), \tag{4.12}$$

where

$$F^-_{+0} = \begin{cases} F(u(x,z),x,z), & \text{if } u(x,z) \le \bar{u}(x,z); \\ F(\bar{u}(x,z),x,z), & \text{if } u(x,z) > \bar{u}(x,z); \end{cases},$$

$$F^-_{+1} = \begin{cases} F(u(x+\Delta x,z),x+\Delta x,z), & \text{if } u(x+\Delta x,z) \le \bar{u}(x+\Delta x,z); \\ F(\bar{u}(x,z),x,z), & \text{if } u(x+\Delta x,z) > \bar{u}(x+\Delta x,z); \end{cases},$$

$$F^+_{-0} = \begin{cases} F(u(x,z),x,z), & \text{if } u(x,z) \geq \bar{u}(x,z); \\ F(\bar{u}(x,z),x,z), & \text{if } u(x,z) < \bar{u}(x,z); \end{cases},$$

$$F^+_{-1} = \begin{cases} F(u(x-\Delta x,z),x-\Delta x,z), & \text{if } u(x-\Delta x,z) \geq \bar{u}(x-\Delta x,z); \\ F(\bar{u}(x,z),x,z), & \text{if } u(x-\Delta x,z) < \bar{u}(x-\Delta x,z); \end{cases},$$

and $\bar{u}$ is the solution of the equation

$$\frac{\partial}{\partial u} F(u(x,z),x,z) = F'(u(x,z),x,z) = 0 \quad .$$

In the isotropic case this algorithm is easy to code, and vectorizes with very little special care required.

The meaning of $\bar{u}$

The Engquist-Osher algorithm is complicated looking, but it becomes simpler to understand once one understands the physical meaning of "$\bar{u}$".

Figure 4.5 shows a graph of $F(u)$ versus u; from the discussion in section 4.1.2, we know this figure can also be interpreted as the lower half of a dispersion-relation plot (slowness surface). Since the group-velocity vector is always perpendicular to the phase-slowness surface (see section B.2), the sign of $F'(u)$ determines whether the x component of the group-velocity vector points to the left or right. When $u = \bar{u}$, $F' = 0$ and the group-velocity direction is straight down.

How the Engquist-Osher scheme works

To apply equation (4.6) with a first-order approximation of the term $\partial/\partial x$, we could use either the right difference

$$\Big(F(u(x+\Delta x)) - F(u(x))\Big)/\Delta x \tag{4.13}$$

or the left difference

$$\Big(F(u(x)) - F(u(x-\Delta x))\Big)/\Delta x. \tag{4.14}$$

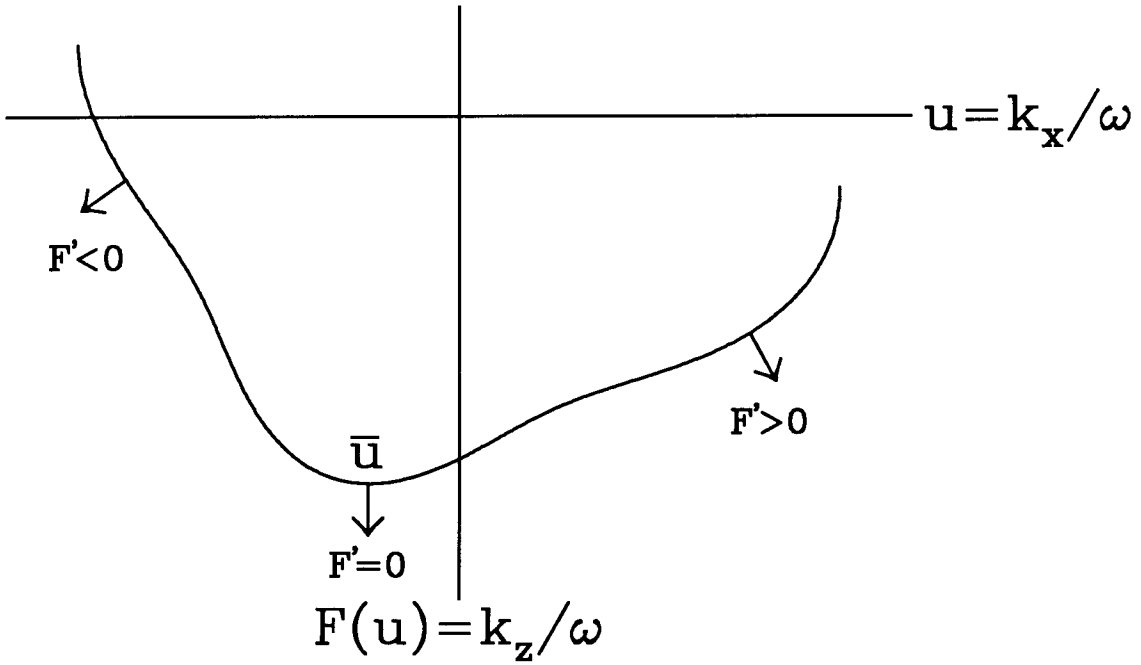

FIG. 4.5. A plot of $F(u)$ versus u, or equivalently k_z/ω versus k_x/ω. The arrows show the group-velocity direction, which is perpendicular to the phase-slowness surface. The x component of the group-velocity direction has the same sign as $F'(u)$. Since $F'(\bar{u}) = 0$, when $u = \bar{u}$ the group-velocity direction points straight down. Note that in the anisotropic case neither the sign of u nor F is a reliable indicator of the group direction!

Extrapolation must be done causally to be stable, i.e. in the same direction as the group-velocity direction. Given that the z component of the group velocity is always down (we choose F to be the lower half of a dispersion relation just so that is guaranteed true), the sign of the x component is the critical factor. The inequalities testing u against $\bar{u}$ in the Engquist-Osher scheme are switches that flip between the left and right differences (equations (4.14) and (4.13)) according to whether the x component of the group-velocity vector points to the right or left. It is easy to verify that if the direction flow is consistent across the three points of the finite-difference stencil the Engquist-Osher upwind scheme uses the causal choice.

Figure 4.6 shows a more interesting case where the direction is not consistent across the stencil. In the figure we wish to find u at the next z-level at the central (shown by a large $\times$) x gridpoint (call it x_0). The x component of the group direction switches sign at $u(x) = \bar{u}$, while the direction we need to extrapolate in switches sign at $x = x_0$. The thin dotted portion of the $F(u(x))$ curve at the center of the plot must be avoided: the group

direction and the extrapolation direction are contradictory there. The fat solid portions of the $F(u(x))$ curve correspond to causal extrapolation and so are OK.

Given that u and $F(u)$ are only defined at the gridpoints (indicated by the little squares), how can the Engquist-Osher scheme perform the difference across just the right part in between the gridpoints? The trick is that the Engquist-Osher scheme defined by equation 4.12 replaces

$$\Big(F(u(x+\Delta x)) - F(u(x))\Big)/\Delta x \tag{4.15}$$

by

$$\Big(F(u(x+\Delta x)) - F(\bar{u})\Big)/\Delta x, \tag{4.16}$$

which simultaneously forms the difference across exactly the causal portion of the wavefront and weights according to the amount of the wavefront used.

4.2.2 Why transverse isotropy

It appears the Engquist-Osher scheme is general enough to handle anisotropic dispersion relations. The main difficulty in implementing the anisotropic version of the algorithm lies in calculating the quantities $F(u)$ and $\bar{u}$ required by the method. To show how it works, I begin with an example of the simplest non-trivial kind of anisotropy, qP and qSV waves in transverse isotropy with a vertical axis of symmetry.

$F(u)$ for transversely isotropic qP and qSV waves extrapolated in z is

$$F_{\mathrm{P}}(u) = -\sqrt{(-\sqrt{A+B^2}+B)/2}$$

for qP waves and

$$F_{\mathrm{S}}(u) = -\sqrt{(\sqrt{A+B^2}+B)/2}$$

for qSV waves, where

$$A = \frac{-4(C_{11}C_{44}u^4 - (C_{11}+C_{44})u^2 + 1)}{C_{33}C_{44}} \tag{4.17}$$

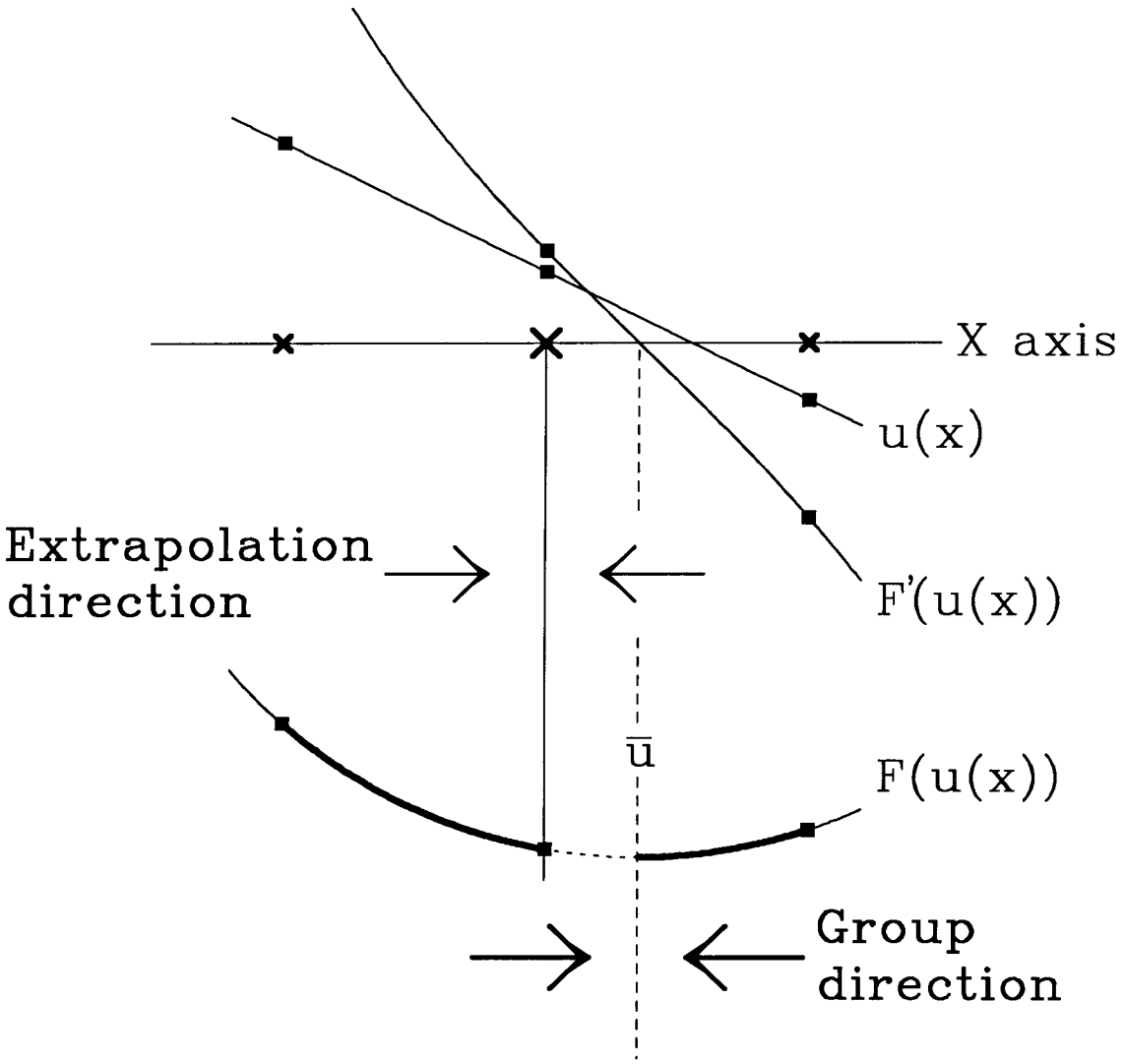

FIG. 4.6. The Engquist-Osher upwind finite-difference scheme. A set of three adjacent x gridpoints are shown, marked by the "×" symbols. A smooth continuous $u(x)$ has been fit through these three points, and the resulting F and F' have been plotted. The dotted portion of the F curve should not be used to extrapolate the new value of u at the central gridpoint, since that extrapolation would be anti-causal and unstable. Extrapolation must not be done against the group direction.

and

$$B = \frac{C_{13}^2 u^2 - C_{11}C_{33}u^2 + 2C_{13}C_{44}u^2 + (C_{33} + C_{44})}{C_{33}C_{44}}.$$

This is as "simple" as $F(u)$ gets for anything more complex than elliptical anisotropy. (I leave deriving the elliptically anisotropic case as an exercise for the reader.) Because of the high symmetry in this problem only a quadratic in k_x^2 and k_z^2 needed to be solved to find $F(u)$; solving for k_z as a function of k_x is much more difficult for more general anisotropic symmetry systems. In the most general case of arbitrary 3-dimensional anisotropy it requires solving a sixth order polynomial! This is not a very serious drawback. General anisotropy, with such unavoidable effects as mode-mode coupling and complex waveforms to contend with, certainly requires more sophisticated modeling methods than finite-difference traveltimes in any case.

4.2.3 Why Cartesian coordinates

Van Trier and Symes (1991) do their extrapolation in polar coordinates. Polar coordinates have several advantages. If the origin of the coordinate system is placed at the source, calculating the initial conditions becomes trivial. Extrapolation in z becomes inaccurate for waves with energy traveling nearly horizontally, and breaks down completely for overturning waves. Waves rarely overturn to the extent of heading back towards the source, so polar coordinates mostly avoids this problem. $F(u)$ in the isotropic case is only barely more complicated, too, becoming $-\sqrt{s^2 - (u/r)^2}$. The only drawback of polar coordinates is that the input velocity model and output traveltime field must be mapped between the Cartesian model grid and the polar coordinates used by the calculation.

In principle, there is no theoretical obstacle to anisotropic extrapolation in polar coordinates. Anisotropy implies direction-dependence, so we must expect F to depend on r and θ as well as u. (Note that as far as the extrapolation operator is concerned, even the homogeneous anisotropic case becomes "heterogeneous" in polar coordinates, because the function F varies with position (r, θ).) Unfortunately, rotating the anisotropy breaks whatever symmetry might have been present, and we are launched into a more general case even for models that only contain "simple" anisotropy. For the case of transverse isotropy, for example, equation (4.17) in polar coordinates becomes a fourth-order equation, far more cumbersome. The simplicity of the isotropic $F(u)$ in polar coordinates is a special

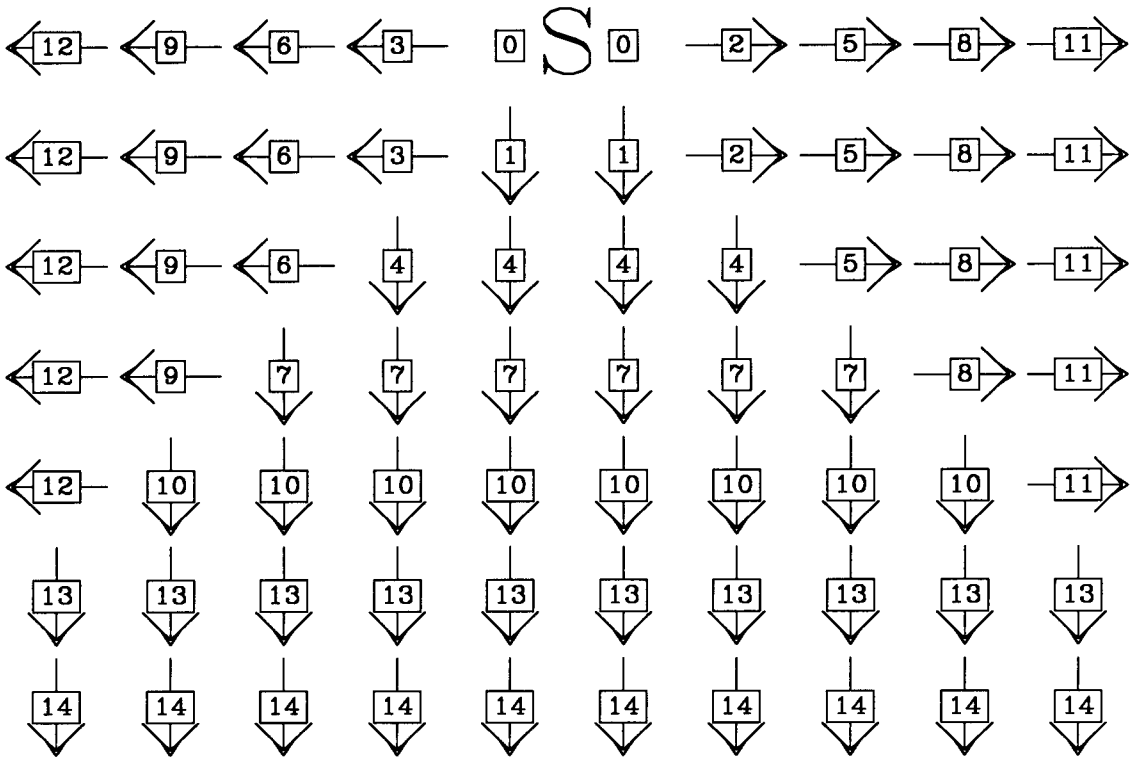

FIG. 4.7. Expanding rectangular computational wavefronts that could be used to extrapolate traveltimes away from a point source, shown by an "S". The boxed numbers show the order of calculation, and the arrows show the direction of extrapolation. This method can abort if the group-velocity direction does not point along the extrapolation direction or if the group-velocity direction points "in" at one of the computational edges (in which case the Engqusit-Osher scheme demands knowledge of u at a gridpoint that hasn't been reached yet).

case; isotropy is the only symmetry system that can be rotated through an arbitrary angle without changing.

Calculating $F(u)$ is not the worst problem. The real drawback of a complicated $F(u)$ becomes apparent when one tries to solve the equation $F'(\bar{u}) = 0$ for $\bar{u}$. This equation probably cannot even be solved analytically in polar coordinates for anisotropy more complicated than elliptical anisotropy.

Perhaps there is some way around this problem? It may be possible to find $F(u)$ and $\bar{u}$ numerically, or to find and use other finite-difference approximations that aren't so dependent on the nature of F. (Crandall and Majda (1980) list possible alternative schemes that might be adaptable for this purpose, although I have not yet tried implementing them to find out.)

Some of the advantages of polar coordinates can be recovered by using Symes' "down and out" method, as shown in Figure 4.7. This method is more cumbersome to code than a polar-coordinate method, and must be aborted if an "inflow" condition occurs at a computational grid edge. When this happens usually the order of extrapolations left, right, and down can be modified and the extrapolation begun again successfully, but the necessity of doing this can make the method somewhat annoying to use.

The extrapolations left and right carry no extra computational burden, because $u = \tau_x$ and $F(u) = \tau_z$ (which both had to be calculated anyway even for downward extrapolation) neatly exchange roles. For the case of transverse isotropy with a vertical axis, we can also take advantage of the symmetry noted in footnote 5 on page 21 to use the same subroutine for calculating F in both extrapolation directions. (Switching between the two cases entails interchanging pairs of elastic constants.)

4.3 Examples

For all the examples in this thesis, I will use Engquist-Osher on a Cartesian grid with rectangular "down and out" computational fronts. I will only allow transverse isotropy aligned with the Cartesian axes. This keeps $F(u)$ reasonably simple and allows me to set $\bar{u} = 0$ everywhere. Even so, the anisotropic algorithm is several times slower than the isotropic one.

4.3.1 Homogeneous examples

To test the accuracy of the method, I begin with four homogeneous examples for which the correct result can be calculated analytically (although not explicitly).

The top two plots in Figure 4.8 (note you need to turn the page sideways) show how the algorithm works for the isotropic case. The result is exact for waves propagating parallel to either of the Cartesian axes, but is about 1.5% too slow for waves propagating at 45°.

This error is due to numerical anisotropy. Polar Engquist-Osher extrapolation such as used by Van Trier and Symes (1991) has a similar numerical anisotropy, but it will not be detected by the typical test case of circular or near-circular wavefronts. (The same way plane waves propagating nearly straight down or to the left or right are not a good test of the errors in Cartesian extrapolation.) There is also a noticeable non-circularity of the

contours nearest the source. This error is caused by the rapid variation of the traveltime gradient near the source, which causes the finite-difference approximation to break down.

It should be possible to minimize both these sources of error, although I will not attempt to do so here. The first could be attacked by characterizing the numerical anisotropy and adding an appropriate anisotropy in F to counteract it. The second can be reduced by calculating the traveltime-gradient field analytically for a block of gridpoints around the source. This is easy to do in the isotropic case (Symes does it already in his code).

The lower two plots in Figure 4.8 (labeled "Borderline triplication on qSV") show the results for an anisotropic medium. For the qP wavefronts on the left the discrepancy between the calculated and analytical results is about the same as in the isotropic case. The qSV wavefronts on the right show a much larger error; the analytical wavefronts have a sharp "kink" that the calculated wavefronts do not follow. The "kink" is an incipient triplication.

Figure 4.9 shows two more extreme test cases. The top two plots (labeled "Very Anisotropic") show a case where the qP wavefronts are very noncircular (although nearly elliptical), and the qSV wavefronts are strongly triplicating. Again, the error for the smooth qP wavefronts is about the same as in the isotropic case. For the qSV wave it is impossible for the finite-difference method to follow the triplication, since it cannot become multi-valued; instead it just cuts off the cusp of the triplication.

The bottom two plots in Figure 4.9 show what happens when there is a triplication aligned with the extrapolation directions. This should cause trouble for the Engquist-Osher finite-difference scheme being used, because it assumes that $F(\bar{u} = 0)$ is the single global minimum of $F(u)$. Here there are two minima and $\bar{u} = 0$ is a relative maximum. Somehow the scheme still performs reasonably well.

From these homogeneous examples, it appears we can expect at most a numerical anisotropy of up to about 1.5% in any case where the wavefronts are not too sharply curved. If there are "kinks" or triplications in the wavefront the method smooths them out, which is at least a graceful way of failing.

4.3.2 Heterogeneous example

Figure 4.10 shows a heterogeneous test model. To avoid discontinuities in $u = \partial\tau/\partial x$, the elastic constants do not change abruptly but "ramp" linearly from one set of values to the next.

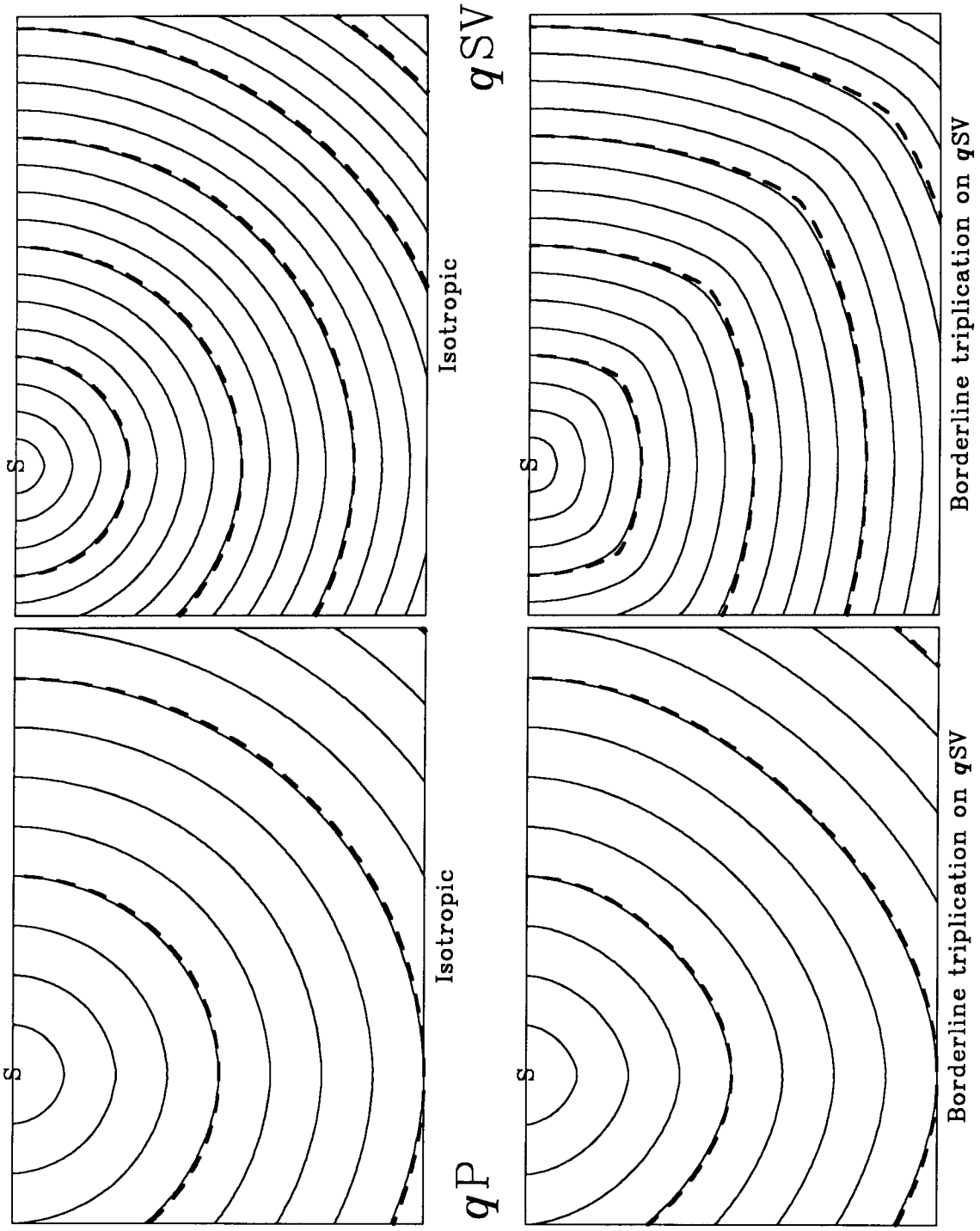

FIG. 4.8. Two homogeneous finite-difference travetime examples. (Please look at this figure from the right.) The "S" marks the point source location. The thin lines show contours of constant traveltime. The thick dashed lines show the analytical solutions for every fourth contour. The discrepancy between the fat and thin curves shows the error of the finite-difference traveltime method. The two plots on the top correspond to the same isotropic case, with qP on the left and qSV on the right. The two plots on the bottom correspond to an anisotropic case chosen so that the qSV wave is just triplicating.

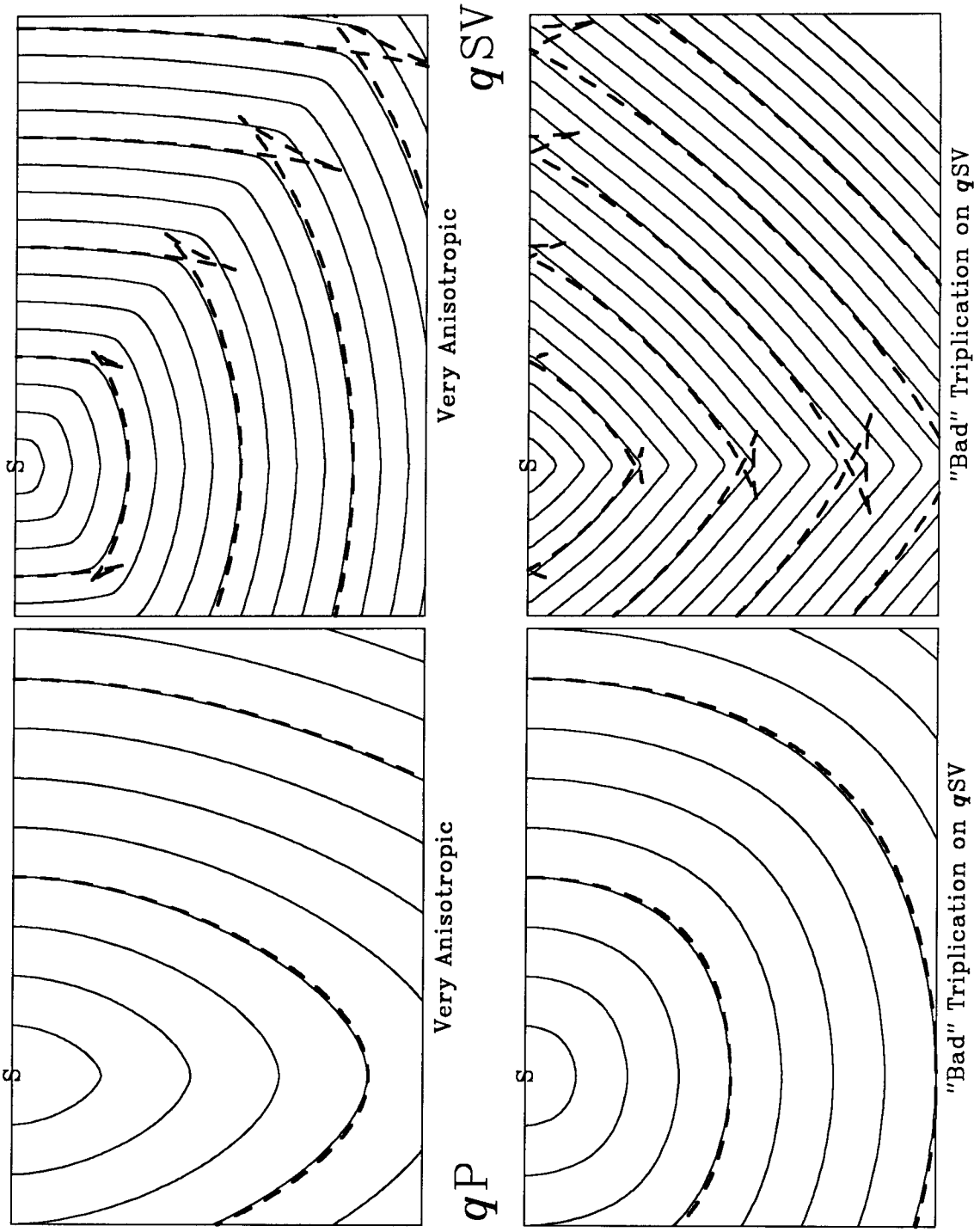

FIG. 4.9. Two more homogeneous finite-difference traveltime examples. (Please look at this figure from the right.) The "S" marks the point source location. The thin lines show contours of constant traveltime. The thick dashed lines show the analytical solutions for every fourth contour. The discrepancy between the two curves shows the error of the finite-difference traveltime method. The two plots on the top show an extremely anisotropic case. The qP wavefronts are decidedly noncircular, and the qSV wavefronts strongly triplicate. The bottom two plots show another extremely anisotropic case, where the qSV wave triplicates in the direction of the coordinate axes.

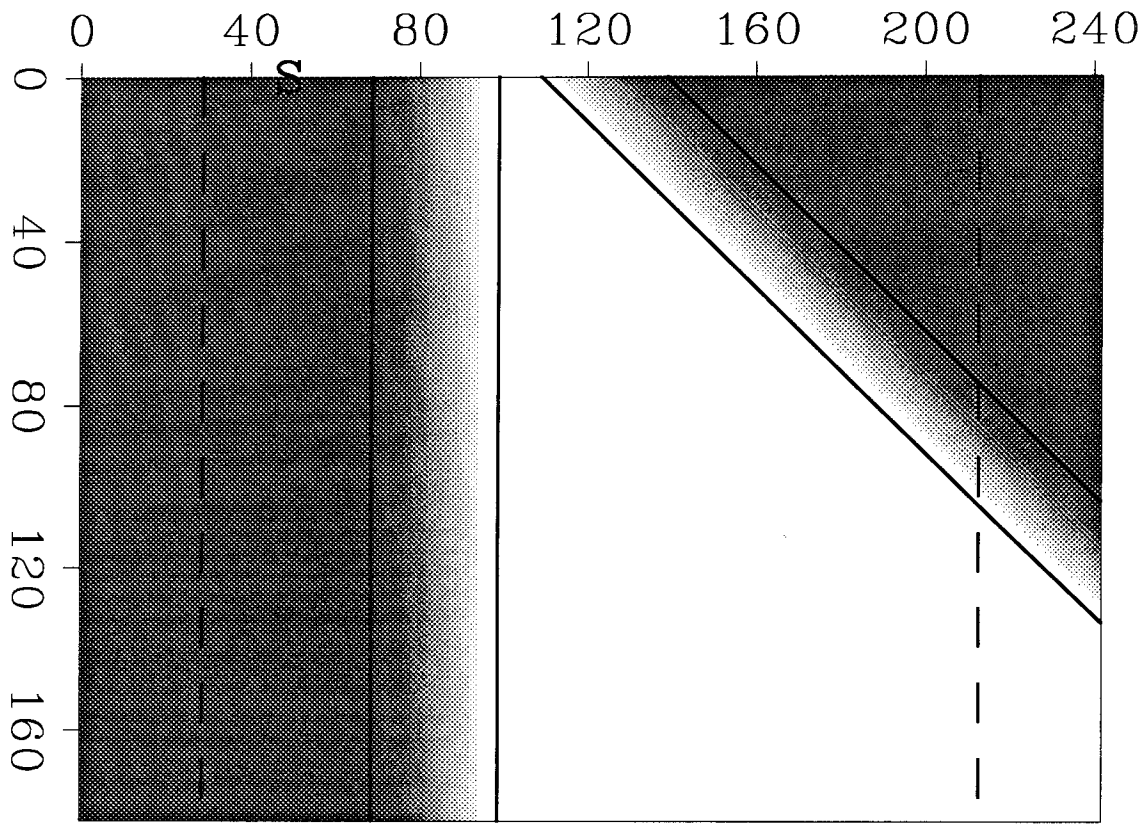

FIG. 4.10. A heterogeneous test model, 243 × 243 gridpoints (part of the bottom of the model is not shown). The source position is marked by the "S". The left edge and upper right corner of the model (shown grey) are isotropic. The triangular region in between (shown white) has a much higher velocity and is strongly anisotropic. (The elastic constants are listed in Table C.1.) The elastic constants vary linearly from one set of constants to the other across boundary zones 30 gridpoints wide (shown shaded between white and grey). The solid lines show the limits of the various model regions. The coarsely dashed vertical lines show the limits of the absorbing boundaries built in to the modeling program.

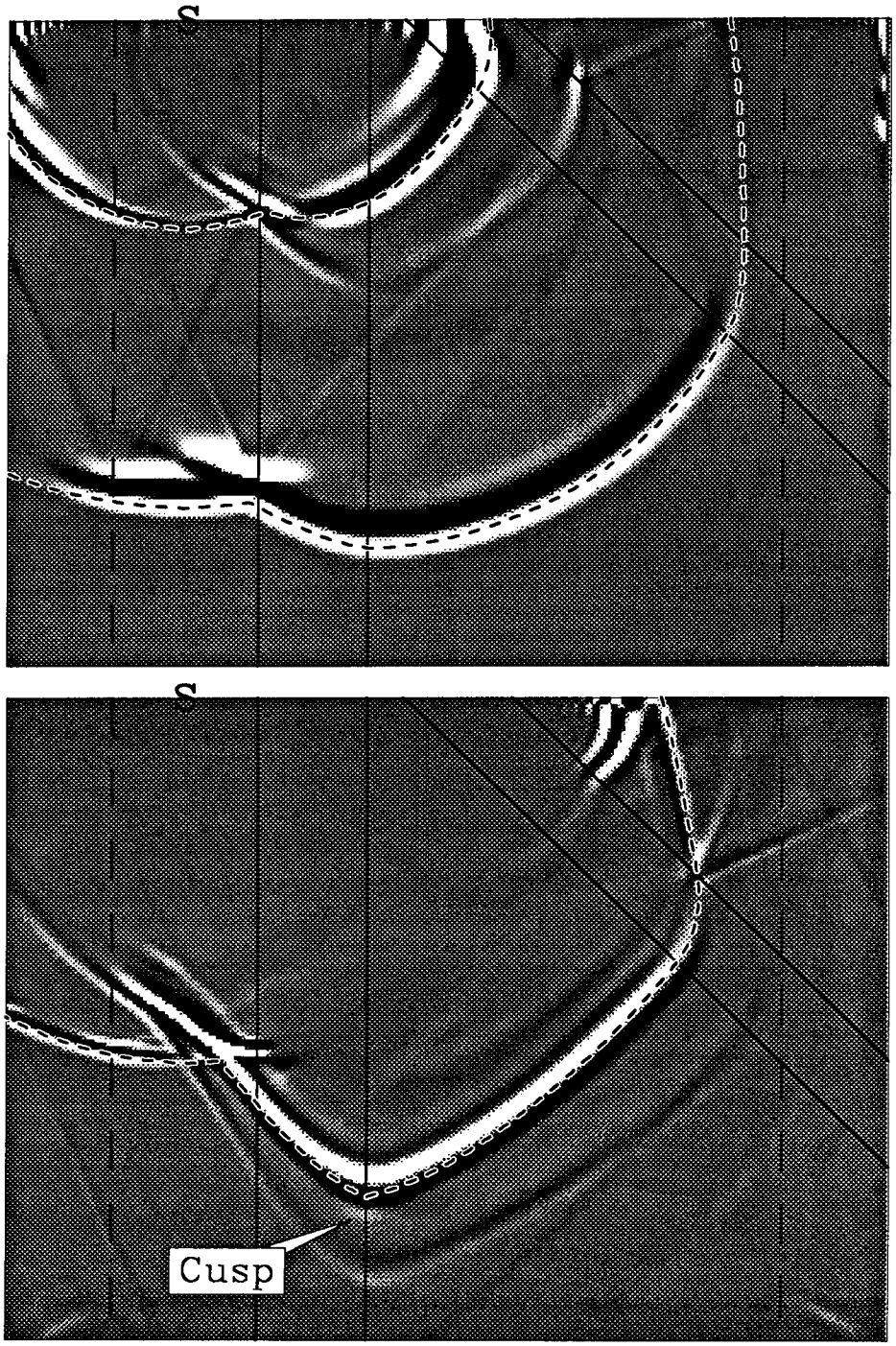

FIG. 4.11. Two finite-difference snapshots calculated for the heterogeneous model shown in Figure 4.10. The top plot shows the z component, and the bottom plot the x component at a later time. Corresponding finite-difference traveltime contours (finely dashed) have been overlayed. Ideally the contours should overlay the first peak of the wavelet. The sharp bend in the qSV wavefront at the lower center of the plot is *not* a refraction; it is an incipient triplication. Note the beginnings of a cusp on the leading edge of the wavefront there. Also note how the finite-difference wavefronts reflect and convert at the "ramp" boundaries, even though the elastic parameters are continuous there (although their first derivatives are not).

Figure 4.11 shows two snapshots calculated for this model. The appropriate traveltime contours are plotted over the snapshots. The Engquist-Osher finite-difference traveltime method appears to trace the first-arriving unconverted qP and qSV wavefronts correctly, despite the strong heterogeneities and anisotropy. The errors seem consistent with those noted in the previous homogeneous examples.

Unfortunately, despite the good results the finite-difference traveltime extrapolation for this model proved disturbingly sensitive to the values of Δx and Δz used. Simply put, it went unstable too easily. The fault lies with the background model. It is smooth enough for the differential equation, but not for the difference equation. The background model should at least have continuous first derivatives for the method to work reliably. This is a minor annoyance since doing it right requires some sort of elastic interpolation, but such interpolation has been done successfully before (Muir, Dellinger, Etgen, and Nichols, submitted). The benefits of the method seem to outweigh the minor difficulties.