Three dimensional traveltime computation using the Fast Marching Method

James A. Sethian, Alexander Mihai Popovici

Department of Mathematics, Univ. of California, Berkeley, CA

3DGeo Development, 465 Fairchild Drive, Mountain View, CA 94043

(March, 97)

ABSTRACT

We present a fast algorithm for solving the eikonal equation in three dimensions, based on the Fast Marching Method (FMM). The algorithm is of order $O(N \log N)$, where N is the total number of grid points in the computational domain. The algorithm can be used in any orthogonal coordinate system, and globaly constructs the solution to the eikonal equation for each point in the coordinate domain. The method is unconditionally stable, and constructs solutions consistent with the exact solution for arbitrarily large gradient jumps in velocity. In addition, the method resolves any overturning propagation wavefronts.

We begin with the mathematical foundation for solving the eikonal equation using the Fast Marching Method, and follow with the numerical details. We show examples of traveltime propagation through the SEG/EAGE Salt model using point source and plane wave intial conditions, and analyze the error in constant velocity media.

The algorithm allows for any shape of the initial wavefront. While a point source is the most commonly used initial condition, initial plane waves can be used for controlled illumination or for downward continuation of the traveltime field from one depth to another, or from a topographic depth surface to another. The algorithm presented here is designed for computing first arrival traveltimes. Nonetheless, since

it exploits the Fast Marching Method for solving the eikonal equation, we believe that it is the fastest of all possible consistent schemes to compute first arrivals.

INTRODUCTION

A fast, accurate and unconditionally stable 3-D traveltime algorithm is an important tool in seismic imaging. Applications of a robust traveltime computation module are not limited to 3-D Kirchhoff prestack and poststack migration, but also can be used for 3-D velocity analysis, 3-D Kirchoff modeling, 3-D Kirchhoff datuming, and 3-D variable velocity migration to zero-offset.

Traveltime computation methods have enjoyed a long history, and, in the past ten years, have seen considerable new advancements, particularly those aimed at a finite difference approach, as exploited by Vidale (1988). Previous to this work, traveltimes were mainly computed using ray tracing; while these ray tracing methods offer a high degree of accuracy, they also pose interpolation problems in shadow areas and areas where multiple caustics develop. Finite difference traveltimes solved the problem of interpolating in shadow zones but new issues ensued related to the stability of the algorithms and to the computation of the most energetic arrivals versus the first arrivals.

A broad spectrum of traveltime computation methods mushroomed in the early 90's. Vidale (1990) presented the extension of his finite difference traveltime in three dimensions. Van Trier and Symes (1991) introduced a two-dimensional explicit finite-difference method with a vectorizable inner kernel that was running efficiently on vector computer platforms, popular before the parallel computer explosion. Popovici (1991a) extended the van Trier and Symes algorithm to three dimensions, and pointed out that instability problems occur for small velocity contrasts (Popovici, 1991b). Schneider (1995) modified the algorithm to reduce the instability problems, and further improved the algorithm in several ways by using an adaptive grid, rotating the spherical coordinates by 90 degrees to remove a singularity point, and checking at run time for instability conditions.

Podvin and Lecomte (1991) extended and parallelized Vidale's algorithm, and Nichols (1996) offered a novel solution to estimate maximum energy traveltimes. Vinje et al. (1993) developed a traveltime algorithm in two dimensions using wavefront construction that propagates the traveltime wavefront by local raytracing.

At its core, the problem of computing first arrival times is equivalent to tracking an interface advancing with a speed normal to itself given by the supplied velocities. The goal in such an interface advancement is to accurately and robustly deal with the formation of cusps and corners, topological changes in the propagating interface, and stability issues in three space dimensions. In the past 15 years, a large body of work has been devoted to these issues in the more general context of tracking evolving interfaces, starting with the work of Sethian (1982, 1987), initially leading to level set methods (Osher and Sethian, 1988), and, more recently, to Fast Marching Methods (Sethian, 1996a) which are specifically aimed at the solution of the eikonal equation. Both techniques hinge on the construction of "entropy-satisfying" weak solutions, by making use of numerical schemes borrowed from the technology of hyperbolic conservation laws, and aimed at constructing the correct viscosity solution of the appropriate partial differential equations.

In this paper, we present the foundation of the Fast Marching Method (FMM), the appropriate mathematical theory, and the finite difference upwind stencil used to locally solve and advance the eikonal equation. We use this stencil in combination with a wavefront construction technique based on building a narrow band around the traveltime wavefront. The traveltime values are stored on a heap, with the minimum value at the top of the heap. The wavefront is always advanced by using the minimum traveltime value in the heap. The cost of a heap operation is $\log(N_{NB})$ where N_{NB} is the total number of traveltime values in the narrow band. This guarantees that the total number of operations needed to solve the eikonal equation on the 3-D cartesian grid is $O(N \log N_{NB})$ (where N is the total number of gridpoints), and is the key to the remarkable speed of the global algorithm.

We show traveltime results through the SEG/EAGE Salt Model and discuss the accuracy of the traveltime computations in several models with heterogeneous constant velocity blocks.

THE FAST MARCHING METHOD

The foundation for the 3-D traveltime algorithm presented here is the Fast Marching Method (FMM) introduced by Sethian (1996a, 1996b). While this is the first application of the FMM to geophysical problems, FMM has been applied to a variety of static Hamiltonian problems, including photolithography development (Sethian, 1996a), shape recovery in medical imaging (Malladi and Sethian, 1996), and the general solution of static Hamilton-Jacobi equations (Sethian, 1996c).

Advancing Interfaces and the Eikonal Equation

Imagine, for the sake of discussion, a two-dimensional closed curve propagating with speed F in a direction normal to the curve. Here, we may think of the speed $F = F(x, y)$ as the velocity at each point (x, y) in space, which is the inverse of the slowness function. We further restrict ourselves to the case in which the slowness function is finite and non-zero, in which case the velocity always exists. Let $u(x, y)$ be the time at which the curve crosses the point (x, y). The surface $u(x, y)$ then satisfies the equation

$$|\nabla u| F = 1. \tag{1}$$

Equation (1) simply says that the gradient of arrival time surface is inversely proportional to the speed of the front. This is a form of the well-known eikonal equation, which is a Hamilton-Jacobi equation, and the recasting of the problem into a stationary one is common in a variety of applications. The speed function must always be positive so that the crossing time surface $u(x, y)$ is single-valued. As detailed in the following section, the evolving curve/surface can develop corners where the evolving interface focuses, rarefaction fans where it expands, and regions where it changes topology. The key to constructing numerical schemes which correctly handle these mechanisms is to construct what are known as "entropy-satisfying" approximations to the gradient term in equation (1).

Entropy-Satisfying Approximations to the Eikonal Equation

A propagating interface can develop corners and discontinuities as it evolves (Sethian, 1982; Sethian, 1985; Sethian, 1990) which require the introduction of a weak solution in order to proceed. The correct weak solution comes from enforcing an entropy condition posed by Sethian (1985) for the propagating interface, similar to the one in gas dynamics. Furthermore, this entropy-satisfying weak solution is the one obtained by considering the limit of smooth solutions for the problem in which curvature plays a regularizing role.

As an example, consider the initial cosine curve propagating with speed $F = 1$ shown in Figure 1. As the front moves, a corner forms in the propagating front which corresponds to a shock in the slope, and a weak solution must be developed beyond this point. If the motion of each individual point is continued, the result is the swallowtail solution shown in Figure 1a, which is multiple-valued and does not correspond to a clear interface separating two regions. Instead, an appropriate weak solution is obtained by considering the associated smooth flow obtained by adding curvature κ to the speed law, that is, letting $F = 1 - \epsilon\kappa$, as in Figure 1b. The limit of these smooth solutions as ϵ goes to zero produces the weak solution shown in Figure 1c.

Another way to obtain this same solution is by enforcing an entropy condition posed by Sethian (1982), similar to the one for a scalar hyperbolic conservation law. Imagine that the front is the boundary of a propagating flame, separating a burning region below from an unburnt region above. The front at any time t is just the set of all points located a distance t from the initial front. Thus, the entropy condition may be stated briefly as "once a point burns, it stays burnt." This weak solution corresponds to a decrease in total variation of the propagating front and is irreversible (Sethian, 1985).

Since the entropy condition is similar to the one for hyperbolic conservation laws, as a numerical technique the condition suggests using this technology to solve the equations of motion (Sethian, 1987). Our goal now is to solve the eikonal equation for a front propagating with speed F, using an "entropy-satisfying" approximation to the gradient term.

Approximations to the Gradient

As motivation for approximating the gradient, we study the simpler case of an evolving curve whose position can always be described as the graph of a function. Consider the initial front given by the graph of $f(x)$, with f and f' periodic on $[0, 1]$, and suppose that the propagating front remains a function for all time. Let ψ be the height of the propagating function at time t, thus $\psi(x, 0) = f(x)$. The tangent at (x, ψ) is $(1, \psi_x)$. The change in height V in a unit time is related to the speed F in the tangent direction by

$$\frac{V}{F} = \frac{(1 + \psi_x^2)^{1/2}}{1}, \tag{2}$$

and thus the equation of motion becomes $\psi_t = F(1 + \psi_x^2)^{1/2}$.

Suppose we approximate the solution by replacing all spatial derivatives with central differences and the time derivative with a forward difference. It is easy to see that such an algorithm may not work. Let $F(\kappa) = 1$ and consider the initial value problem

$$\psi_t = (1 + \psi_x^2)^{1/2}, \qquad \psi(x,0) = f(x) = \left\{ \begin{array}{ll} 1/2 - x & x \leq 1/2 \\ x - 1/2 & x > 1/2 \end{array} \right\}. \tag{3}$$

The initial front is a "V" formed by rays meeting at $(1/2, 0)$. By our entropy condition, the solution at any time t is the set of all points located a distance t from the initial "V". To construct a numerical scheme, divide the interval $[0,1]$ into $2M - 1$ points, and form the central difference approximation to the spatial derivative ψ_x in equation (3), namely

$$\psi_t \approx \frac{\psi_i^{n+1} - \psi_i^n}{\Delta t} = [1 + [\frac{\psi_{i+1}^n - \psi_{i-1}^n}{2\Delta x}]^2]^{1/2} = [1 + [D_i^{0x}\psi]^2]^{1/2} \tag{4}$$

where in the last expression we have used standard notation for the central difference.

Since $x_M = 1/2$, by symmetry, $\psi_{M+1} = \psi_{M-1}$, thus the right-hand-side is 1. However, for all $x \neq 1/2$, ψ_t is correctly calculated to be $\sqrt{2}$, since the graph is linear on either side of the corner and thus the central difference approximation is exact. Note that this has nothing to do with the size of the space step Δx or the time step Δt. *No matter how small we take the numerical parameters, as long as we use an odd number of points, the approximation to ψ_t at $x = 1/2$ gets no better.* It is simply due to the way in which the derivative ψ_x is approximated. In Figure 2 we show results using this scheme, with the time derivative ψ_t replaced by a forward difference scheme.

It is easy to see what has gone wrong. In the exact solution, $\psi_t = \sqrt{2}$ for all $x \neq 1/2$. This should also hold at $x = 1/2$ where the slope is not defined; the Huygens construction sets $\psi_t(x = 1/2, t)$ equal to $\lim_{x \to 1/2} \psi_t$. Unfortunately, the central difference approximation chooses a different (and, for our purpose, wrong) limiting solution. It sets the undefined slope ψ_x equal to the average of the left and right slopes. As the calculation progresses, this miscalculation of the slope propagates outwards from the spike as wild oscillations. Eventually, these oscillations cause blowup in the code.

Entropy-satisfying upwind differences schemes

We focus on the gradient term $(1 + {\psi_x}^2)$ (we call this a gradient term because ψ_x is the one-dimensional gradient). Consider now the following finite difference approximation introduced by Osher and Sethian (1988)

$$ {\psi_x}^2 \approx (\max(D_i^{+x}\psi, 0)^2 + \min(D_i^{-x}\psi, 0)^2) \tag{5} $$

where we have used standard finite difference notation that

$$ D_i^{-x}\psi = \frac{\psi_i - \psi_{i-1}}{h} \qquad D_i^{+x}\psi = \frac{\psi_{i+1} - \psi_i}{h} \tag{6} $$

where ψ_i is the value of ψ on a grid at the point ih with grid spacing h.

Equation (5) is an "upwind" scheme (Sethian, 1996c); it chooses grid points in the approximation in terms of the direction of the flow of information. Intuitively, upwind means that if a wave progresses from left to right, then one should use a difference scheme which reaches *upwind* to the left in order to get information to construct the solution *downwind* to the right. If we consider our propagating "V" curve from the example above, we see that at the symmetric point, the symmetry of the scheme is changed, and a non-zero value is chosen. In Figure 3, we show what happens if we use the scheme given in equation (5). The exact answer is shown, together with two simulations. The first uses the entropy-satisfying scheme with only 20 points (Figure 3(b)), the second (Figure 3(c)) with 100 points. In the first approximation, the entropy condition is satisfied, but the corner is somewhat smoothed due to the small number of points used. In the more refined calculation, the corner remains sharp, and the exact solution is very closely approximated. Thus we see that this scheme does a correct job of satisfying the entropy-condition.

While a vast array of other upwind, entropy-satisfying schemes are available to approximate the gradient, for our purposes, the above approximation (and one small variation) will be sufficient. More details on upwind schemes, hyperbolic conservation laws, and their role in level set and fast marching methods may be found in Sethian (1996c).

Upwind Schemes for the Eikonal Equation

We can use the above ideas about upwind schemes to construct appropriate schemes for the eikonal equation

$$|\nabla u(x, y, z)| = s(x, y, z), \tag{7}$$

where $u(x, y, z)$ is the traveltime field and $s(x, y, z)$ is the slowness function in the three dimensional model. Extending the above ideas of upwind approximations to the gradient to multi-dimensions, we have the scheme (Osher and Sethian, 1988):

$$\begin{aligned} |\nabla u| \approx (&\max(D^{-x}_{ijk}u, 0)^2 + \min(D^{+x}_{ijk}u, 0)^2 \\ &+ \max(D^{-y}_{ijk}u, 0)^2 + \min(D^{+y}_{ijk}u, 0)^2 \qquad (8) \\ &+ \max(D^{-z}_{ijk}u, 0)^2 + \min(D^{+z}_{ijk}u, 0)^2)^{1/2}. \qquad (9) \end{aligned}$$

The forward and backwards operators D^{-y}, D^{+y}, D^{-z}, and D^{+z} in the other coordinate directions are similar to the one defined earlier for the x direction.

A slightly different upwind scheme, given in (Rouy, 1992), which for our implementation of the fast marching is slightly more convenient, is given by

$$\begin{bmatrix} \max(D^{-x}_{ijk}u, -D^{+x}_{ijk}u, 0)^2 + \\ \max(D^{-y}_{ijk}u, -D^{+y}_{ijk}u, 0)^2 + \\ \max(D^{-z}_{ijk}u, -D^{+z}_{ijk}u, 0)^2 \end{bmatrix}^{1/2} = F_{ijk}, \tag{10}$$

where we use the same forward and backward operators D^- and D^+ and F_{ijk} is the slowness at the gridpoint ijk.

Fast marching method implementation

The central idea behind the fast marching method is to solve the eikonal equation by systematically constructing the travel times $u(x,y,z)$ in an upwind fashion. Essential to the method is the observation that the upwind difference structure of equation (10) means that information propagates "one way", that is, from smaller values of $u(x,y,z)$ to larger values. Hence, the fast marching algorithm rests on solving equation (10) by building the solution outwards from the smallest $u(x,y,z)$ value.

The algorithm is made fast by confining the "building zone" to a narrow band around the front, motivated by the narrow band technology introduced in Chopp (1993), used in recovering shapes from images in Malladi et al (1993), and analyzed extensively by Adalsteinsson and Sethian (1995). The idea is to sweep the front ahead in an upwind fashion by considering a set of points in narrow band around the existing front and to march this narrow band forward, freezing the values of existing points and bringing new ones into the narrow band structure. The key is in the selection of *which* grid point in the narrow band to update.

The algorithm proceeds as follows: First, we tag points in the initial conditions as *Accepted*. We then tag as $Close$ all points one grid point away. Finally, we tag as Far all other grid points. Therefore the inner loop consists of the following operations:

1. Begin Loop: Let $Trial$ be the point in $Close$ with the smallest value for u.

2. Add the point $Trial$ to *Accepted*; remove it from $Close$.

3. Tag as $Close$ all neighbors of $Trial$ that are not *Accepted*. If the neighbor is in Far remove it from that list and add it to the set $Close$.

4. Recompute the values of u at all neighbors according to equation (10).

5. Return to the top of the loop.

This algorithm works because the process of recomputing the $u(x, y, z)$ values at upwind neighboring points cannot yield a value smaller than any of the accepted points. Thus, we can march the solution outwards, always selecting the narrow band grid point with minimum trial value for $u(x, y, z)$, and readjusting neighbors, as shown in Figure 4.

Another way to look at this is that each minimum trial value begins an application of Huygens principle, and the expanding wave front touches and updates all others. The speed of the algorithm comes from a heapsort technique to efficiently locate the smallest element in the set $Trial$. The technique also can be extended to more general static Hamiltonians of the form

$$H(Du, x) = 0, \tag{11}$$

where Du represents the derivatives in each of the component variables $u_{x_1}, ..., u_{x_N}$. In all cases, the scheme is extremely fast; if there are N total points in the grid, then the scheme solves the equation in $O(N \log N)$.

3-D TRAVELTIME MAP EXAMPLES

We demonstrate the 3-D FMM traveltime algorithm on the 3-D SEG/EAGE salt dome velocity model (Aminzadeh et al., 1995). The salt dome model was designed to contain major complex features that are characteristic to complicated Gulf of Mexico salt structures. It includes a northwesterly plunging stock, a secondary reactivation crest southward of the stock, a low-relief eastern flank, a faulted southern flank with a toe thrust, a rounded overhang on the west flank, five sands that are gas charged (at least one contains both a gas/oil and an oil/water contact), and a shale sheath that is modeled to be geopressured. The sea floor map exhibits a counter-regional fault scarp, a bathymetric rise associated with the sill crest, and a shelf break at the southeast end of the model. The overall model size is 13.5x13.5x4.2 km on a 20 m grid.

The SEG/EAGE Salt Model has a complicated salt-to-sediments interface which creates complex wave propagation problems. Figures 6 and 8 show contour traveltimes superimposed on the velocity model and are representative for the wave propagation patterns encountered while solving the eikonal equation in the SEG/EAGE Salt Model. In Figure 6 the traveltime contours show the formation of headwaves that travel along the salt-sediment interface. Figure 8 shows the formation of cusps in the traveltime contours. Such cusps are frequent sources of instabilities in finite-difference eikonal solvers, since they include singular points where rays cross or the traveltime gradient is discontucan be multivalued. The entropy-satisfying approximation to the eikonal equation ensures that the wavefront propagation is solved without introducing numerical instabilities.

The first examples in Figures 5 and 6 show vertical slices through the model in orthogonal $x-$ and $y-$ directions. The $100 \times 100 \times 100$ 3-D grid represented in Figures 5 through 8 is sampled at $\Delta x = \Delta y = \Delta z = 40$ m and was obtained by subsampling by a factor of two the SEG/EAGE model. The point source is situated at the surface, in the middle of the X-Y plane (X=9720 m, Y=6860 m). The X-section was obtained for a constant Y=6860 m, while the Y-section was obtained for a constant X=9720 m. Figures 7 and 8 show horizontal slices through the traveltime cube, at depths of 1180 m and 1380 m respectively.

Initial conditions

Typically, the initial condition for computing Green's function in Kirchhoff migration is a point source, or a small spherical shell around the source location. The algorithm we present here offers great flexibility in selecting any surface shape for the initial traveltime condition. Figure 9 shows examples of traveltimes through the SEG/EAGE Salt Model with a horizontal plane wave initial condition. The initial condition can be specified on complicated surfaces, useful for controlled illumination or plane wave synthesis operators (Rietveld and Berkhout, 1994; Rietveld et al., 1992).

Another potential application of the flexible initial condition is for restarting traveltimes from any depth level or depth surface, in any layer-stripping algorithm. Figure ?? shows schematically how traveltimes can be computed from a surface point source to a depth surface A1. When the velocity is well known in the area above the surface A1, we compute the traveltimes to the target surface A1 only once. The velocity under surface A1 can be changed, and corresponding traveltimes can be computed by restarting the algorithm with an initial condition on the subsurface A1 (Kessler et al., 1995; Reshef, 1997).

Traveltime error

The absolute error of the 3-D traveltime computations is dependent on the size of the mesh and the type of finite difference approximation to the eikonal equation. We use a first order finite difference approximation, but the algorithm can use without change a higher order finite difference approximation. For a point source in constant velocity the error is zero along the main axes of the grid, with the maximum error at 45 degrees. Figure 11a shows traveltime contours through a constant velocity medium (2000 m/s). The continuous lines represent the analytic solution while the dashed lines represent the computed solution. The computed traveltimes are always smaller than the analytic solution and are equal to the analytic solution along the main axes of the grid. Figure 11b shows the difference between the analytic solution and the computed traveltimes in a vertical slice passing through the source. Figure 12 displays the absolute values (in seconds) of the error for the case shown in Figure 11b. The absolute error decreases with depth.

While in constant velocity media the directions of zero error are aligned with the axes, in media with complex velocity, the direction of the wavefront changes and the error is distributed along all azimuths. Solving the eikonal equation on a spherical grid reduces to zero the error in the constant velocity case, but for complex velocity models it does not offer any other advantages to the cartesian grid. The overturned waves are equally resolved in both coordinate systems, and the need for complicated adaptive grid code that arises in spherical coordinates is avoided.

CONCLUSIONS

We present a very fast, accurate, and stable 3-D traveltime computation algorithm based on solving the eikonal equation using a Fast Marching Level Set formulation. The algorithm solves the eikonal equation in all the points of the 3-D cartesian or spherical grid. The finite-difference approximation to the eikonal equation is resolved to machine precision in each point of the grid, making the accuracy of the method dependent only on the grid size and the order of the finite-difference scheme used. The stability and speed of the method are ensured by following the wavefront propagation in a narrow band and solving the finite-difference eikonal stencil at points of minimum traveltime in the narrow band. The traveltimes resolved are first-arrival only, and overturned wavefronts are resolved without any additional effort.

The method can be used in many geophysical application requiring modeling wave propagation in complex geological media: 3-D Kirchhoff prestack and poststack migration, 3-D datuming, 3-D velocity analysis, 3-D Kirchhoff modeling and 3-D controlled illumination modeling. The algorithm allows for flexible implementations in either cartesian or spherical coordinates, using as initial condition a point source, a planar wave or any surface initial wavefront.

REFERENCES

Adalsteinsson, D., and Sethian, J.A., 1995, A fast level set method for propagating interfaces: Journal of Computational Physics, **118**, 269–277.

Aminzadeh, F., Burkhard, N., Kunz, T., Nicoletis, L., Rocca, F., 1995, 3-D Modeling Project: 3rd report: The Leading Edge, **14**, 125–128.

Chopp, D. L., 1993, Computing minimal surfaces via level set curvature flow: Journal of Computational Physics, **106**, 77–91.

Kessler, D., Reshef, M., Crase, E., Chan, W.-K., Tsingas, C. and Hubbard, J., 1995, Depth processing: An example: The Leading Edge, **14**, no. 9, 949–953.

Malladi, R., and Sethian, J.A., 1996, An $O(NlogN)$ Algorithm for Shape Modeling: Proceedings of the National Academy of Sciences, **93**, 9389–9392.

Malladi, R., Sethian, J. A., and Vemuri, B. C., 1993, Shape modeling with front propagation: A level set approach: Center for Pure and Applied Mathematics, Report PAM-589, Univ. of California, Berkeley.

Nichols, D.E., 1996, Maximum energy traveltimes calculated in the seismic frequency band: Geophysics, **61**, 253–263.

Osher, S., and Sethian, J.A., 1988, Fronts propagating with curvature dependent speed: algorithms based on Hamilton-Jacobi formulation: Journal of Computational Physics, **79** 12–49.

Podvin, P., and Lecomte, I., 1991, Finite-difference computation of traveltimes in very contrasted velocity models: A massively parallel approach and its associated tools: Geophys. J. Int., **105**, 271–284.

Popovici, A. M., 1991a, Finite-difference traveltime maps: Stanford Exploration Project Report **70**, 245–256.

Popovici, A. M., 1991b, Stability of finite-difference traveltime algorithms: Stanford Exploration Project Report **72**, 135–138.

Reshef, M., 1997, The use of 3-D prestack depth imaging to estimate layer velocities and reflector positions: Geophysics, **62**, no. 1, 206–210.

Rietveld, W. E. A. and Berkhout, A. J., 1994, Prestack depth migration by means of controlled illumination: Geophysics, **59**, no. 5, 801–809.

Rietveld, W. E. A., Berkhout, A. J. and Wapenaar, C. P. A., 1992, Optimum seismic illumination of hydrocarbon reservoirs: Geophysics, 57, no. 10, 1331–1345.

Rouy, E. and Tourin, A., 1992, A Viscosity Solutions Approach to Shape-From-Shading: SIAM J. Num. Anal, **29**, 867–884.

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

Sethian, J.A., 1982, An Analysis of Flame Propagation: Ph.D. Dissertation, Mathematics, University of California, Berkeley.

Sethian, J.A., 1985, Curvature and the evolution of fronts: Commun. in Math. Physics, **101**, 487–499.

Sethian, J.A., 1987, Numerical methods for propagating fronts: Variational methods for free surface interfaces, edited by P. Concus and R. Finn, Springer-Verlag, New Work,.

Sethian, J.A., 1990, Numerical algorithms for propagating interfaces: Hamilton-Jacobi equations and conservation laws: Jour. of Diff. Geom., **31**, 131–161.

Sethian, J.A., 1996a, A Fast Marching Level Set Method for Monotonically Advancing Fronts: Proc. Nat. Acad. Sci., **93**.

Sethian, J. A., 1996b, A review of the theory, algorithms, and applications of level set methods for propagating interfaces: Acta Numerica, Cambridge University Press.

Sethian, J. A., 1996c, Level Set Methods: Cambridge University Press.

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

Vidale, J.E., 1988, Finite-difference traveltime calculation: Bull. Seis. Soc. Am., **78**, 2062–2076.

Vidale, J.E., 1990, Finite-difference calculation of traveltimes in three dimensions: Geophysics, **55**, 521–526.

Vinje, V., Iversen, E., Gjoystdal, H., 1993, Traveltime and amplitude estimation using wavefront construction: Geophysics, **58**, 1157–1166.

FIGURES

FIG. 1. Cosine Curve Propagating with Unit Speed

FIG. 2. Central difference approximation to Gradient

FIG. 3. Upwind, entropy-satisfying approximations to the Gradient

FIG. 4. Upwind construction of Accepted Values.

FIG. 5. Traveltime slice through the SEG/EAGE Salt model with a point source at the surface. The traveltime contours are superimposed on the velocity model.

FIG. 6. Traveltime slice through the SEG/EAGE Salt model with a point source at the surface. The traveltime contours are superimposed on the velocity model.

FIG. 7. Horizontal traveltime slice through the SEG/EAGE Salt model with a point source at the surface. The traveltime contours are superimposed on the velocity model.

FIG. 8. Horizontal traveltime slice through the SEG/EAGE Salt model with a point source at the surface. The traveltime contours are superimposed on the velocity model.

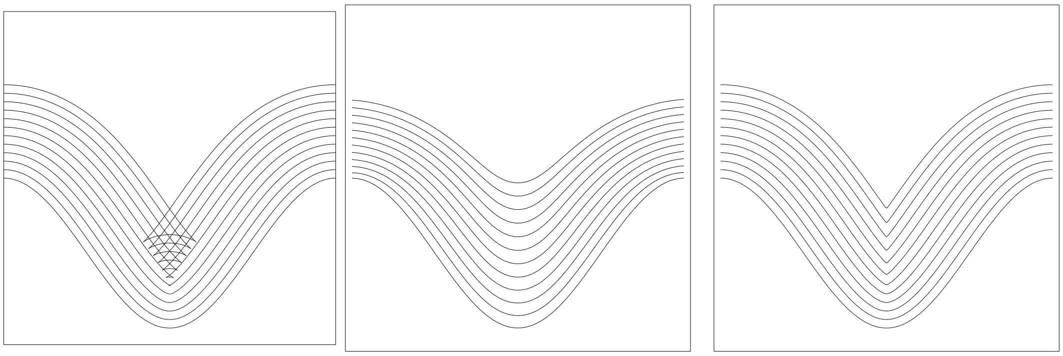
FIG. 9. Traveltime slices through the SEG/EAGE Salt model with a horizontal plane wave source at the surface. The traveltime contours are superimposed on the velocity model.

FIG. 10. Traveltimes can be computed down to a surface in depth (A1) and restarted

from the surface A1 for different velocity models.

FIG. 11. Depth contours of the analytic and computed traveltimes in a constant velocity medium. a. Horizontal slice. b. Vertical slice.

FIG. 12. Absolute error between the analytic and the computed traveltimes, in a vertical slice through the source.

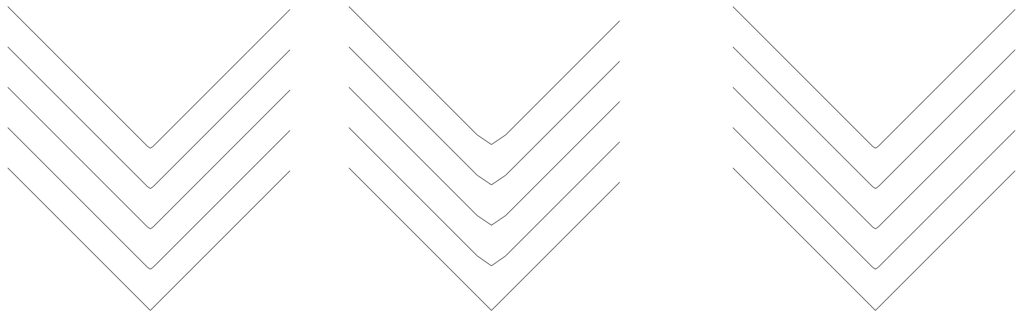


$Swallowtail(F = 1.0)$ $F = 1. - 0.25\kappa$ $Entropy\ Solution(F = 1.0)$

FIG. 1. Cosine Curve Propagating with Unit Speed

Exact Solution Central Differences $\Delta t = .005$ Central Differences $\Delta t = .0005$

FIG. 2. Central difference approximation to Gradient



(a) Exact Solution (b) Scheme with 20 Points (c) Scheme with 100 Points

FIG. 3. Upwind, entropy-satisfying approximations to the Gradient

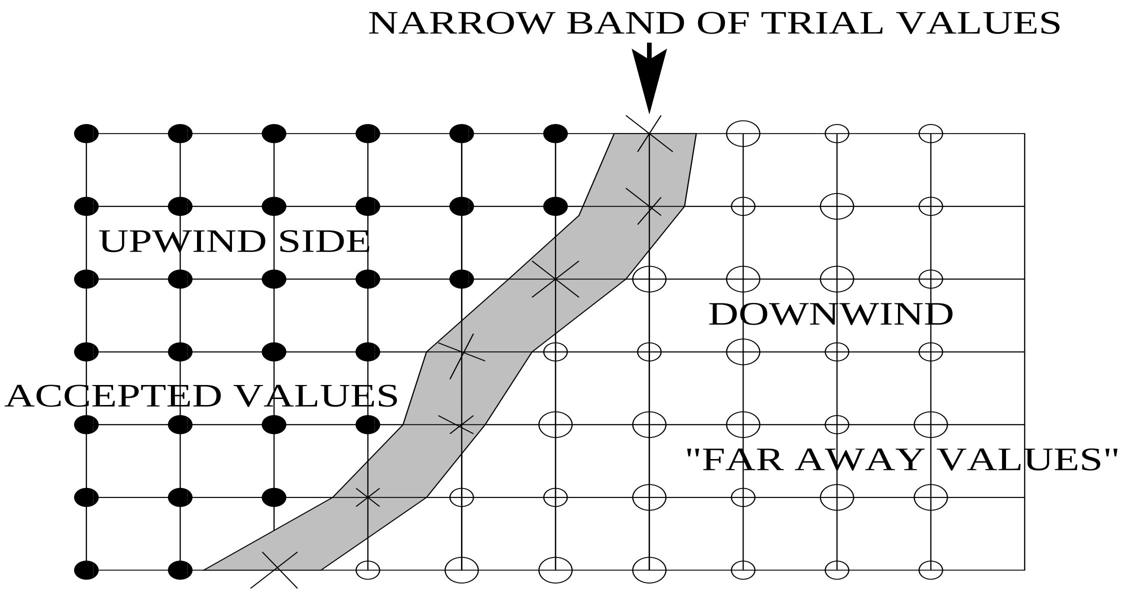


FIG. 4. Upwind construction of Accepted Values.

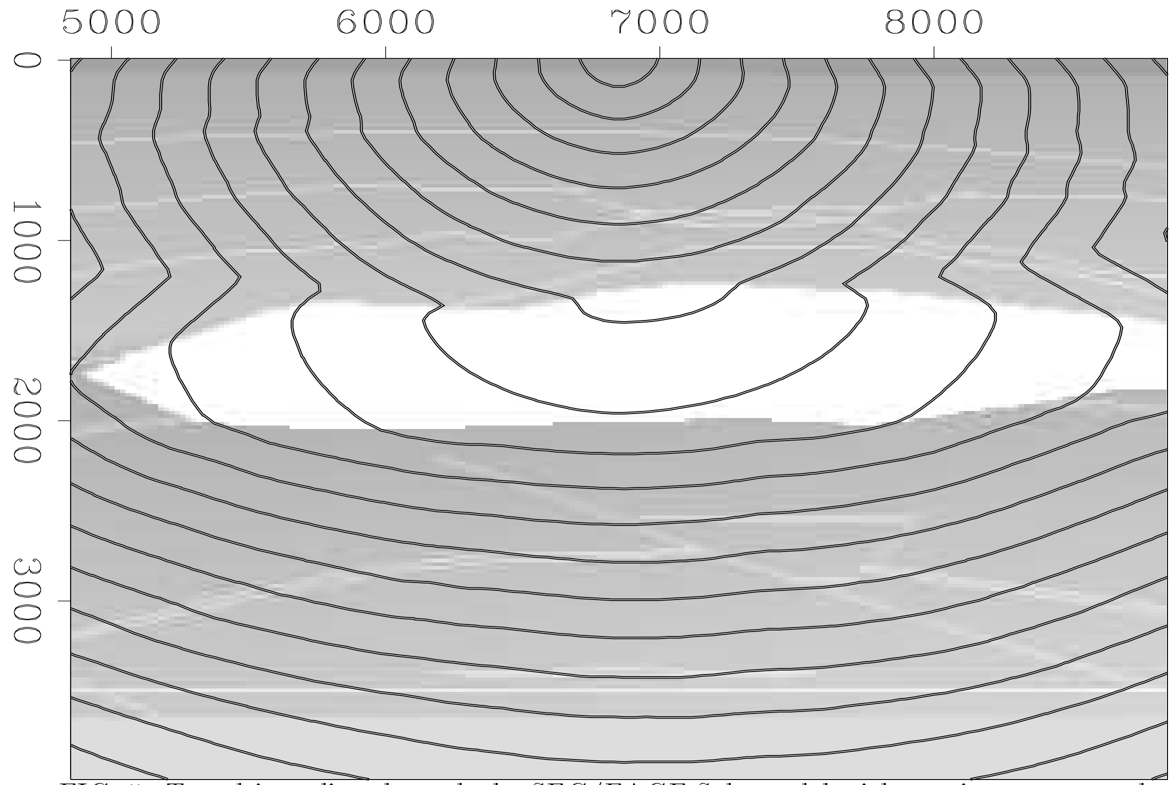


FIG. 5. Traveltime slice through the SEG/EAGE Salt model with a point source at the surface. The traveltime contours are superimposed on the velocity model.

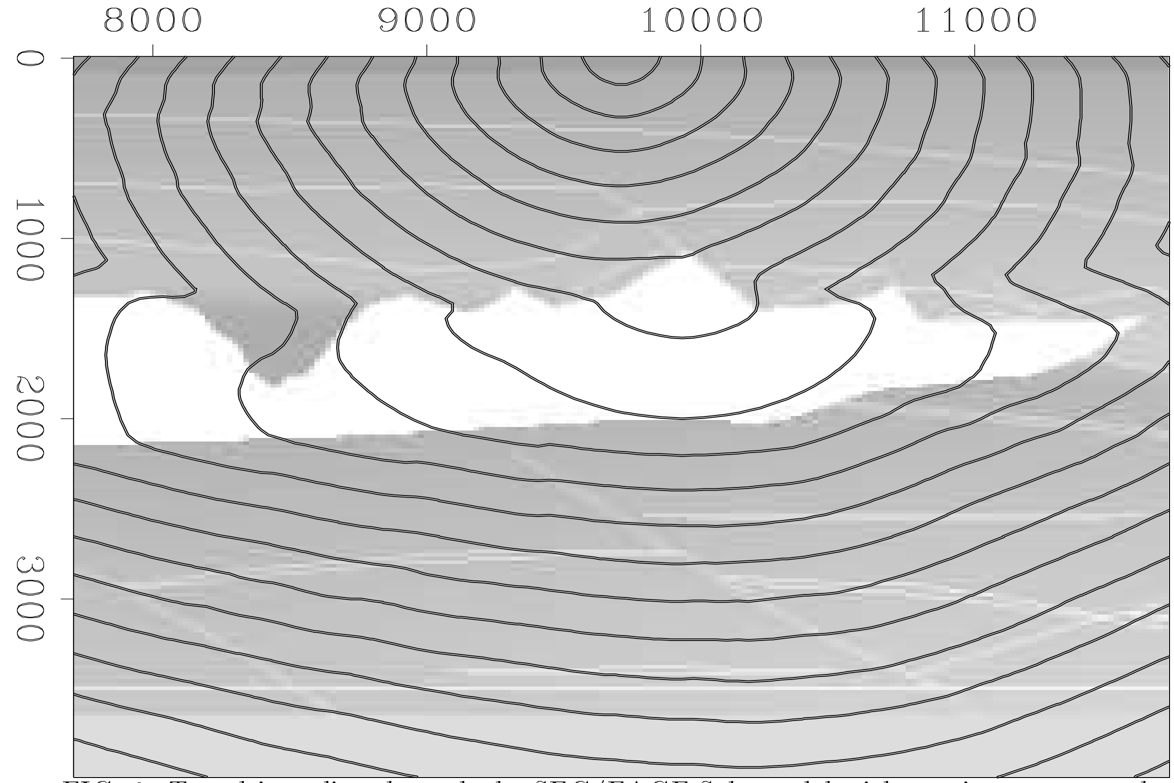


FIG. 6. Traveltime slice through the SEG/EAGE Salt model with a point source at the surface. The traveltime contours are superimposed on the velocity model.

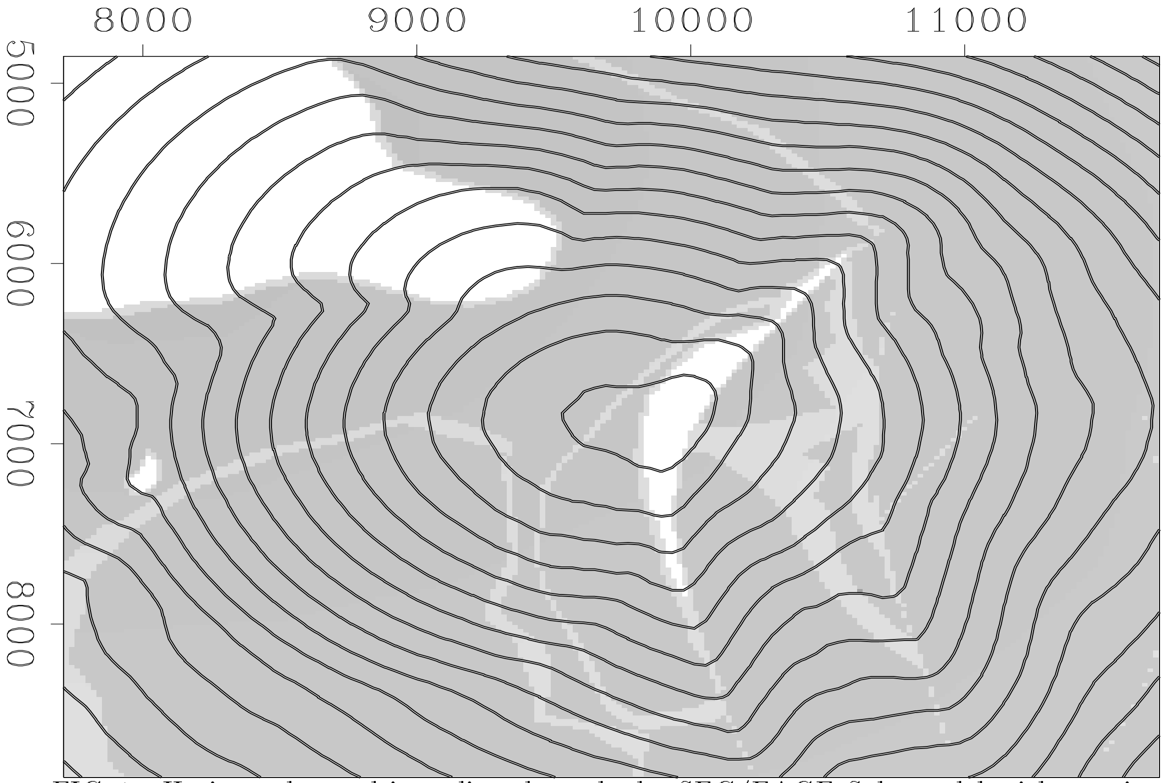


FIG. 7. Horizontal traveltime slice through the SEG/EAGE Salt model with a point source at the surface. The traveltime contours are superimposed on the velocity model.

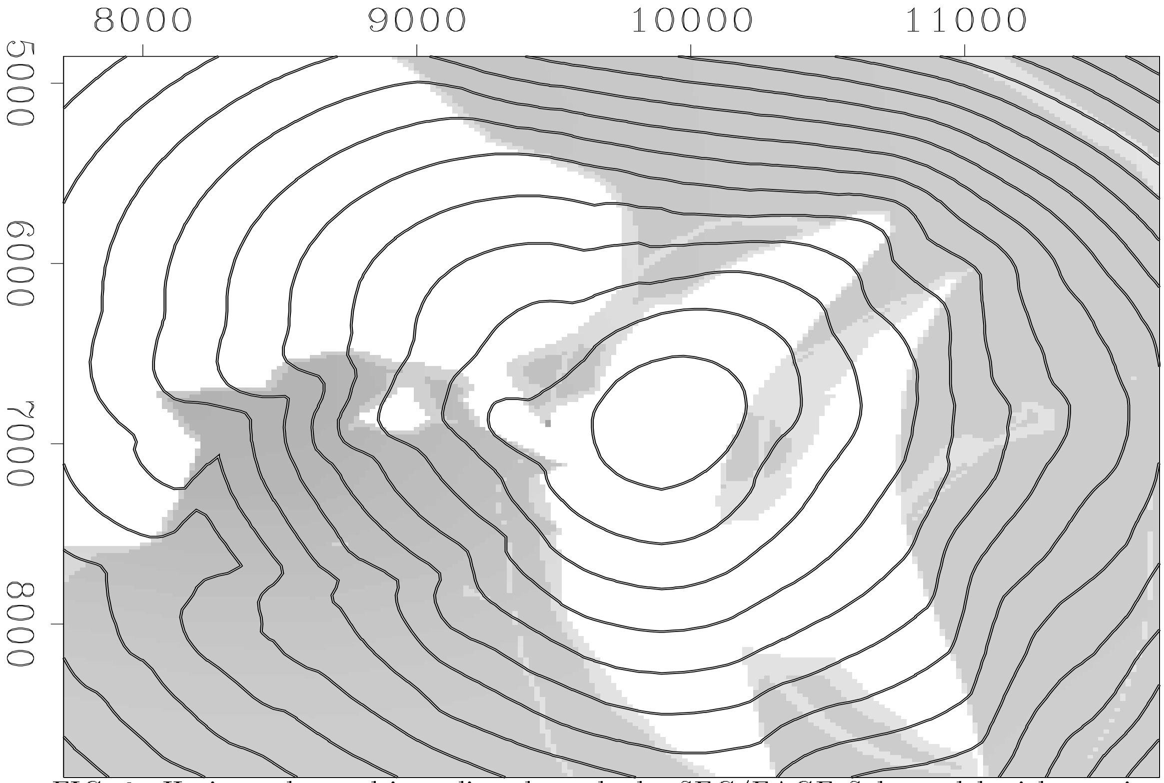


FIG. 8. Horizontal traveltime slice through the SEG/EAGE Salt model with a point source at the surface. The traveltime contours are superimposed on the velocity model.

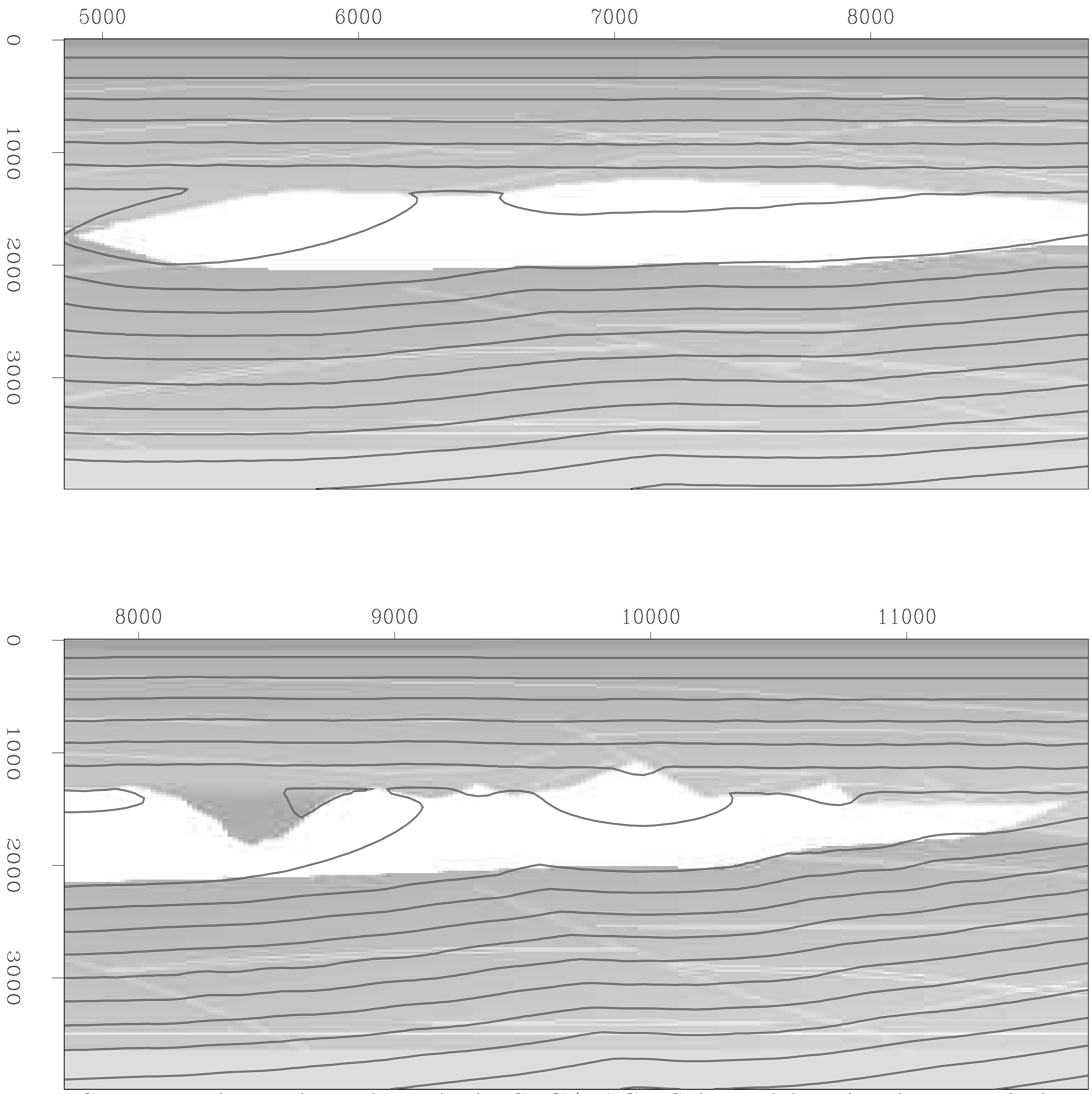


FIG. 9. Traveltime slices through the SEG/EAGE Salt model with a horizontal plane wave source at the surface. The traveltime contours are superimposed on the velocity model.

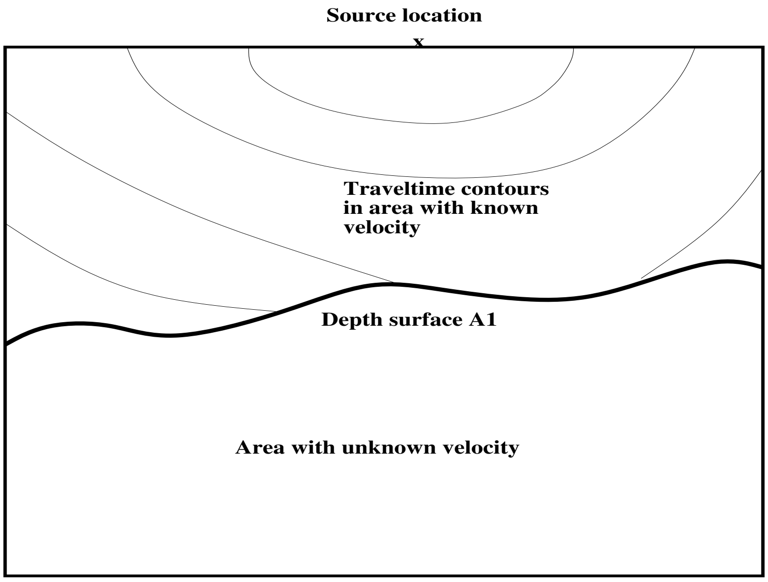


FIG. 10. Traveltimes can be computed down to a surface in depth (A1) and restarted from the surface A1 for different velocity models.

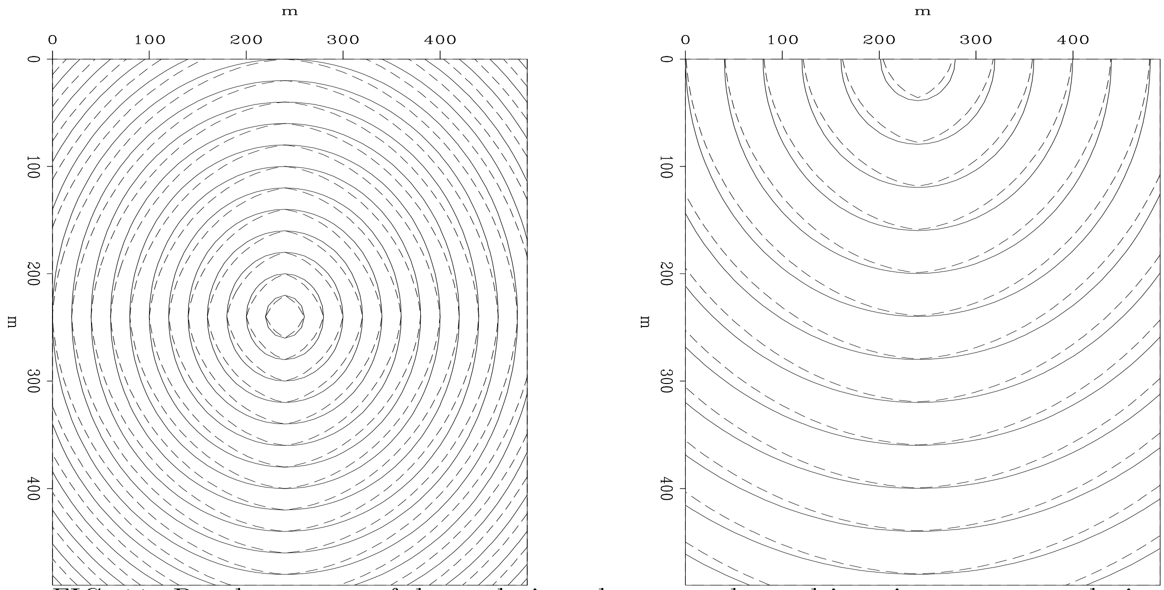


FIG. 11. Depth contours of the analytic and computed traveltimes in a constant velocity medium. a. Horizontal slice. b. Vertical slice.

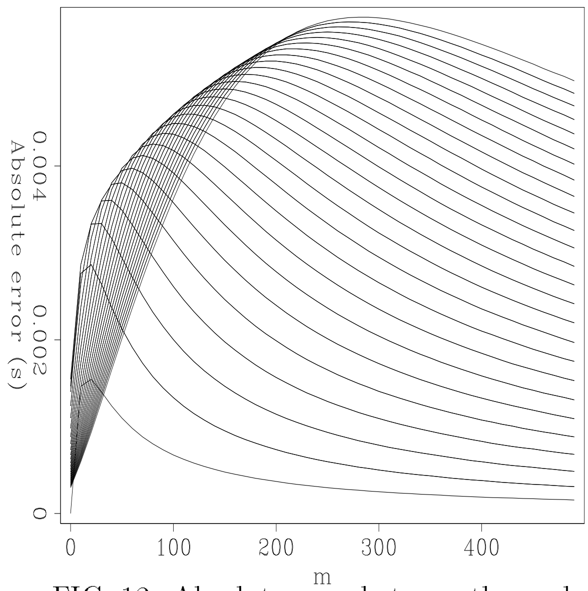


FIG. 12. Absolute error between the analytic and the computed traveltimes, in a vertical slice through the source.