A method for computing traveltimes for an arbitrary velocity model

*Valentine Meshbey, Dan Kosloff Yevgeny Ragoza, Oleg Meshbey *, Uzi Egozy and Jim Cozens; Paradigm Geophysical*

Summary

A method for computing traveltimes for a complex velocity model is presented. The method is based on Fermat's principle and is used to perform 3-D prestack migration. Results from 3-D prestack depth migration obtained using this algorithm are presented. Improved image quality is demonstrated in comparison with 3-D poststack depth migration.

Introduction

The development of fast algorithms for computing traveltimes in an arbitrary complex 2-D/3-D medium has recently become particularly important. These algorithms are especially interesting in the context of performing fast prestack depth migration in two and in three dimensions. It is well known that the sensitivity of prestack migration to the accuracy of the velocity model enables the creation of an iterative procedure for improving the velocity model. This iterative process uses tomography (Kosloff et. al, 1996) to update the velocity model after prestack depth migration. The procedure improves the accuracy of the velocity model and thus improves the quality of the final depth sections, slices and volumes.

In the 2D case, computing the traveltimes by ray tracing can be quite fast. Nonetheless, ray tracing methods have several drawbacks. Examples are the difficulty in taking into account all reflection and diffraction effects, the existence of "blind zones" on the traveltime maps, and the difficulty in sending a ray that will arrive at a specific point. Solution to these problems require an increase in algorithm complexity and consequently an increase in computation time. Other traveltime computation methods are often based on direct solution of the Eikonal Equation (Gray, 1986, Reshef and Kosloff, 1986, Vidale, 1988, Van Trier and Symes, 1991, Nichols, 1996). These methods usually involve the evaluation of traveltimes on a fine grid.

The requirements on CPU time are more severe in 3-D. This induced us to search for new approaches for generating traveltime tables. In this work we present such an approach based on a shortest path method. The algorithm is described in the following sections and a field data example of 3-D prestack depth migration is used to demonstrate the method.

Algorithm

The traveltime from a given shot or receiver position to the subsurface can be computed as a solution to an extremal problem using dynamic programming (Bellman, 1957). This approach was first discussed by Meshbey et. al (1979 and 1980). To illustrate the algorithm let us consider the 2-D case first.

Let the velocity model be defined on a rectangular grid with intervals Δx and Δz, where Δx is constant and Δz may increase with the depth. A velocity model is defined on a fine grid. This definition enables us to consider horizontal and vertical velocity variations within a layer. Let us assume that the traveltimes from a given source point on the surface to all the grid points on depth **level** j-1 (i.e., all the points (x_i, z_{j-1}), $i = 1,2,3...M$) are known. Then, according to Fermat's principle, the traveltime $\left(t_k^j\right)$ from this point on the surface to a point k on level j must meet the condition that

$$t_k^j = \min_{k-M<i<k+M}\left\{t_i^{j-1} + \frac{1}{v_{ik}}\left[\left(x_i^{j-1} - x_k^j\right)^2 + \left(\Delta z_{j-1,j}\right)^2\right]^{1/2}\right\},$$

Where $i=k+m$, $m = 0,\pm1,\pm2,...\pm M$, and $v_{i,k}$ is the average velocity along the straight line connecting points i and k. In other words, the traveltime at point k on level j is the minimum time evaluated by considering all possible rays traveling from the surface point through a point i on level j-1 to the point k on level j (Figure 1). In the 2-D case, the search for the ray with minimum time proceeds as follows. First we define the searched zone, L_0, that is relevant for prestack migration: $L_0=A+l_{max}$, *whereA* is the migration aperture and l_{max} is maximum source-receiver offset. This zone is defined around the source position. Now we proceed to evaluate the traveltime for the first grid point $(k=1)$ within this zone. For this first point we search for the

Downloaded 05 Jun 2012 to 72.20.129.98. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

minimum time ray using all the grid points within the interval L_O. Let grid point i_o on level j-l correspond to the minimum time $\left(\tilde{t}^{j}_{k,i_0}\right)$. For the following grid points we limit the search to a smaller region around i_o.

For the 3-D case the procedure is similar. The formula that describes the traveltime from the source point to a point (k,n) on depth level j according to Fermat's principle is given by

$$t^{j}_{k,n} = \min_{\substack{k-M<i<k+M \\ n-N<p<n+N}} \left\{ t^{j-1}_{i,p} + \frac{1}{v_{i,p.k,n}} \left[\left(x^{j-1}_{i} - x^{j}_{k} \right)^2 + \left(\Delta z_{j-1,j} \right)^2 \right]^{1/2} \right\},$$

where
$i=k+m,\ m = 0,\pm1,\pm2 ,... \pm M,$
$p=n+q,\ q = 0,\pm1,\pm2 ,... \pm N$,
$v_{i,p,k,n}$ is the average velocity along the straight ray that connects the point (i,p) on layer j-1 with the point (k,n) on layer j and,
$t^{j-1}_{i,p}$ is the traveltime to point (i,p) on layer j-1.

In the 3-D case the searched zone is a rectangle but the searching algorithm is very similar. The minimum time value for the first point $(k=1,n=1)$ on level j is obtained by searching all the rays that pass through all the points (i,p) lying inside the square surrounding this point. For the next point the searched zone is limited to a small square $q\Delta x \times q\Delta y$, centered around (i_0,p_0) which is the location of minimum time found for the previous point.

The procedure described above is repeated from one depth level to the next one. The Az intervals increases with depth from approx. 0.1 km near the surface to approx. 0.4 *km* at the depth of 7-8 km. We assume straight rays within the intervals and use the average velocity along the ray for computing the traveltimes. Once the traveltimes are computed from all points on the surface to a point in the subsurface we can use any pair of rays and assume that one of the points is a source point and the other is a receiver point. The sum of the traveltimes is obviously the traveltime of the diffracted wave that is used by the Kirchhoff migration.

Example

The traveltime computation method presented here was implemented in a 3-D prestack depth migration program and the results from this program are used here to illustrate the method. Figure 2 presents a depth section obtained from 3-D prestack depth migration of a Texaco Gulf Coast onshore 3-D survey. The task of the migration here is of course to image the large salt body at the center of this survey. For comparison, the same section after 3-D poststack depth migration is presented in Figure 3. A better image of the steep flanks of the salt dome is achieved by the prestack process.

Conclusions

We presented here an algorithm for computing traveltimes in a complex velocity model. Our experience indicates that the accuracy of this method is sufficient for 3-D prestack migration. This method is especially suitable for "target oriented" applications of 3-D prestack migration because each grid point is computed separately and does not involve computation on a full 3-D grid. The algorithm has many advantages over the conventional ray tracing approach. First of all, the method described here is inherently a two-point procedure that does not require interpolation. Also, the process fills the traveltime maps and does not contain "empty zones". The computation of *dt/dx,* which is a problematic point with regular ray tracing, is not required in our method. The existence of Head Waves in the traveltime maps is another known problem that is usually associated with Eikonal methods. With our approach the traveltimes correspond, in most cases, to body waves rather than head waves because the method is essentially an upward shooting ray tracing approach. Finally, the fact that the traveltimes obtained by our method are suitable for 3-D prestack depth migration is demonstrated by the field data results.

Downloaded 05 Jun 2012 to 72.20.129.98. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

Acknowledgment

We wish to thank Texaco Exploration and Production inc., for providing us with the 3-D dataset presented here and for granting us the permission to publish the results.

References

Bellman, R., 1957, Dynamic Programming, Princeton University Press, p.339.

Gray, S., 1986, Efficient traveltime calculation for Kirchhoff migration, Geophysics, v. 5 1, 564-575.

Kosloff, D., Sherwood, J., Koren, Z., Machet, E., and Falkovitz, Y., 1996, Velocity and interface depth determination by tomography of depth migrated gathers, Submitted to Geophysics.

Meshbey, V., Glogovski, V.M., and Bogdanov G.A., 1979, Study of CDP time-distance curves of primary reflections in non - homogeneous strata: Exploration Geophysics, no. 86, p. 12, (in Russian).

Meshbey, V., Bogdanov G.A., and Tertitski L.M., 1980, Generation of amplitude depth sections for interval velocity model: Exploration Geophysics, no. 88, p. 7, (in Russian).

Nichols, D.E., 1996, Maximum energy traveltimes calculated in the seismic frequency band, Geophysics, v.61,253-263.

Reshef, M., and Kosloff, D., 1986, Migration of common shot gathers, Geophysics, v.51,324-331.

Van Trier, J., and Symes, W., 1991, upwind finite differences calculation of traveltimes, Geophysics v.56,812-821.

Vidale, J., 1988, Finite differences calculation of traveltimes, Bull. Seis. Soc. Am., v. 78,2062-2076.

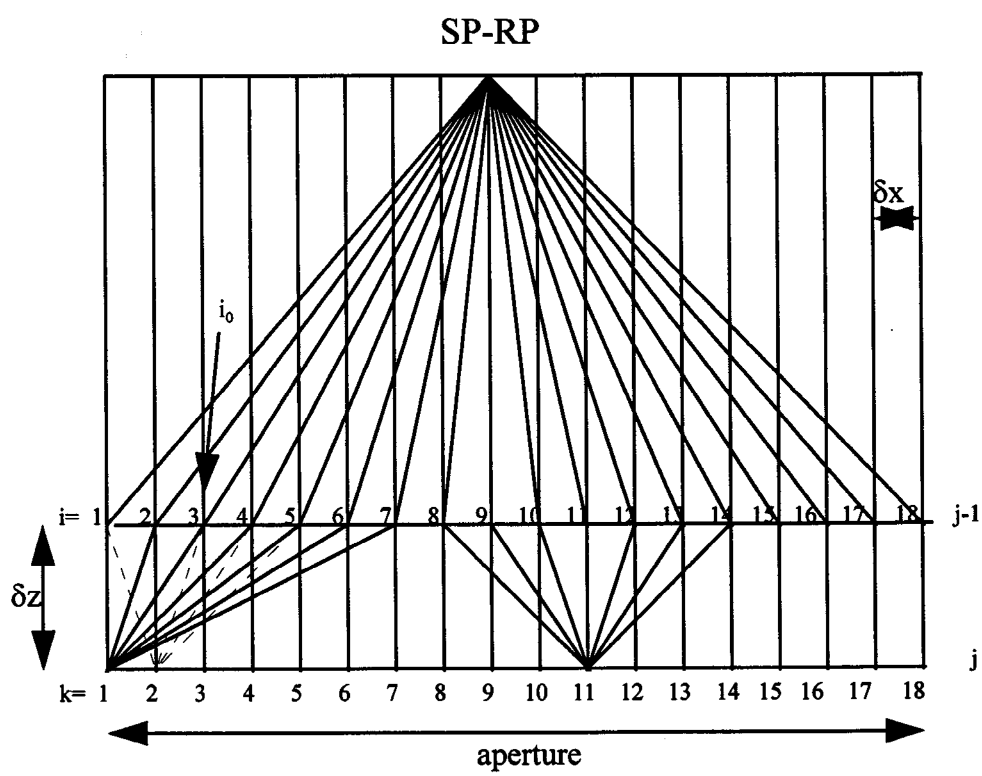

Fig 1: Schematic description of the ray path construction

Downloaded 05 Jun 2012 to 72.20.129.98. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/

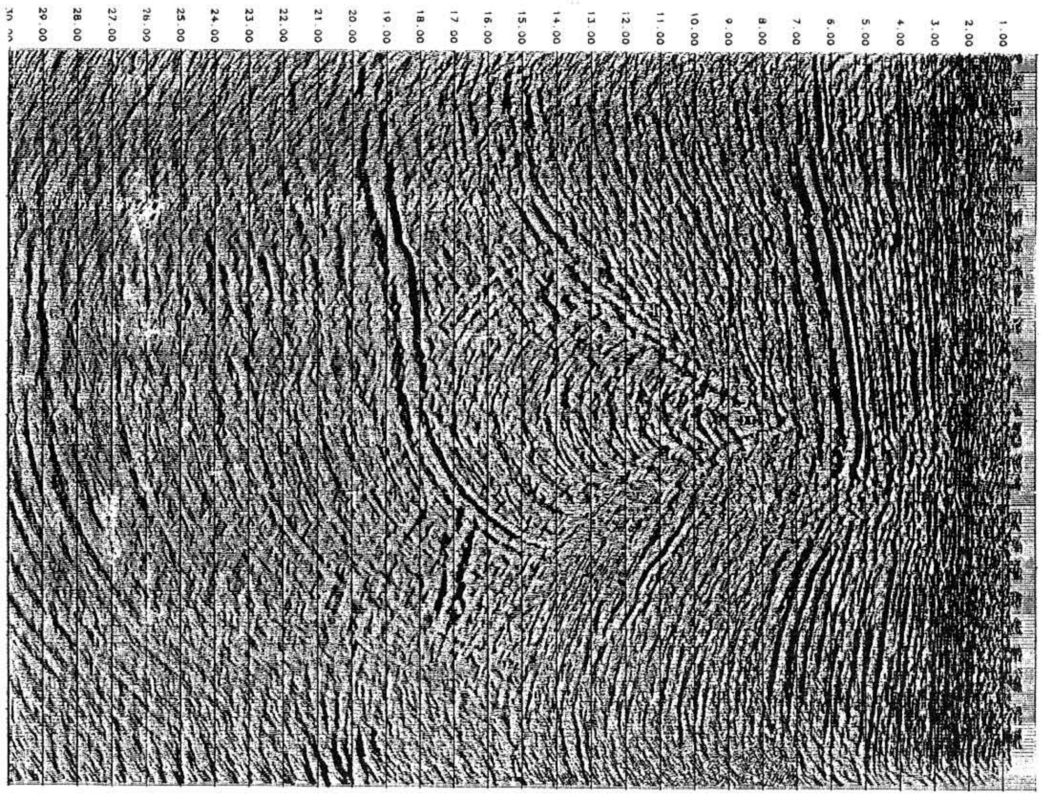

Fig. 2: Depth section after 3-D prestack depth migration.

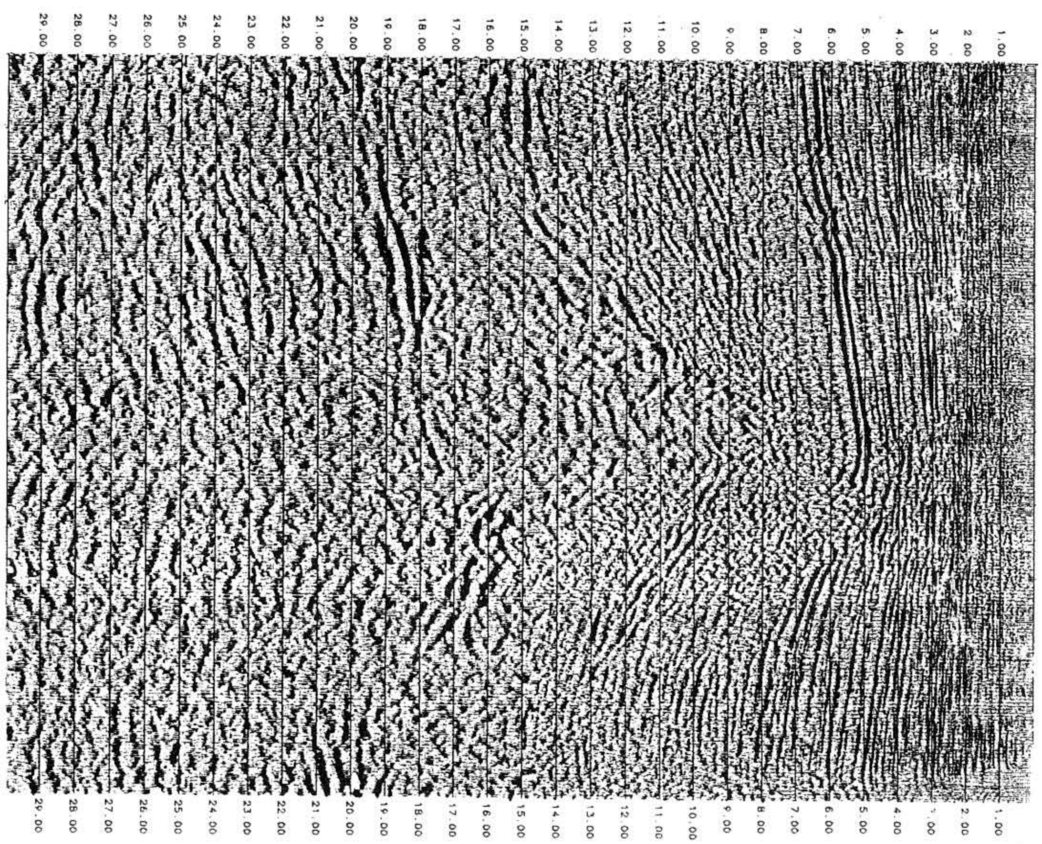

Fig. 3: Depth section after 3-D poststack depth migration.

Downloaded 05 Jun 2012 to 72.20.129.98. Redistribution subject to SEG license or copyright; see Terms of Use at http://segdl.org/