Tomographic determination of interval velocities from CDR data—preliminary results

Chuck Sword

ABSTRACT

The method of Controlled Directional Reception (CDR) can be used to determine travel times and ray parameters of waves transmitted from a given shot and received at a given geophone (Riabinkin et al., 1962; Sword, 1984a). These ray and travel-time parameters (collectively known as CDR parameters) are obtained from conventional seismic data through an automatic picking procedure, and can be inverted tomographically to give interval velocities of the medium. The inversion procedure is easily illustrated for the case of a horizontally layered medium, and synthetic data from such a medium have successfully been inverted. The inversion is more difficult when both horizontal and vertical velocity variations are permitted, but not impossible, and most of the necessary ray-tracing formulas have been derived. The inversion process is easily extended to the analysis of converted-wave data.

INTRODUCTION

In a recent paper, Bishop et al. (1985) pointed out the need for the determination of interval velocities, and they gave a concise summary of various methods that have been used in the past. They then described a tomographic approach, where picked travel-time data are inverted in order to determine interval velocities and depths of reflecting horizons.

In the present paper, I too describe a tomographic approach. One major difference between my approach and that of Bishop et al. is that rather than use travel time picks from interpreted horizons, my approach utilizes travel times and ray parameters that can be picked automatically from common-shot and common-geophone gathers. This approach is similar to those previously proposed by Harlan and Burridge (1983) and by Gray and Golden (1983). Only interval velocities are inverted for, not depths of reflectors. Because of the differences between this method and that of Bishop et al., a different approach to the inversion problem is necessary. Rather than trying to minimize the error in predicted versus measured travel time, the method to be presented here seeks to minimize the horizontal distance between the end points of each pair of downgoing and upcoming rays. This minimization problem can be linearized rather cleanly when the velocity function is laterally homogeneous, and the extension to horizontally varying velocity is fairly straightforward.

The inversion algorithm used is a variation of the Gauss-Newton algorithm for the solution of non-linear optimization problems (Gill et al., 1981; Levin, 1985), with an added damping term. An initial velocity model is chosen, and the rays that correspond to the various travel-time and ray-parameter picks are traced. The ray-tracing results are used to form a linearized least-squares inversion problem. This linear problem can be solved by the conjugate-gradient method (e.g., with the subroutine LSQR (Paige and Saunders, 1982)); the solution is used to update the velocity model. Rays are then traced in the new velocity model, and the sequence of iterations continues.

Tests on synthetic data show that this inversion scheme works when the velocity is a function of depth. It has not yet been tested on data from media whose velocity varies both horizontally and vertically. In such media the ray tracing is more difficult, as is the problem of locally linearizing the data for the linear least-squares step. At the time of writing this report, an effective algorithm for ray tracing in such media has been developed, but the mathematics necessary for the linearization are not yet fully worked out.

It would not be too difficult to solve simultaneously for two different velocity functions: one that affects only downgoing, and another that affects only upgoing, waves. This would be a useful method for determining P and S velocities from converted-wave data. It has not yet been tested. There may be difficulties since twice as many unknown velocities as in the conventional PP case must be determined.

There are several advantages to the tomographic method proposed in the present paper. No assumptions are made about the shapes of the reflectors, and data obtained from point diffractors will work as well as data from flat, dipping, or curved horizons. The picking is automated, so there is no need to interpret horizons and digitize travel times manually. The time and ray-parameter picks should occupy much less space than the original data set, producing savings in both processing time and data storage.

There are also numerous disadvantages. In contrast to the method proposed by Bishop et al., the present method is not based on the implicit assumption that reflectors are continuous; as a result, a certain amount of robustness is lost (the "focusing" of horizons can no longer be used as a criterion). Since the data are picked automatically, the present method may be adversely affected by multiples and other wave-like noise that a human interpreter could easily exclude. In contrast to Fowler's proposed method (this report), which is based to some extent on Toldi's work (1985), the present method expends a great deal of computer time picking reflections and tracing

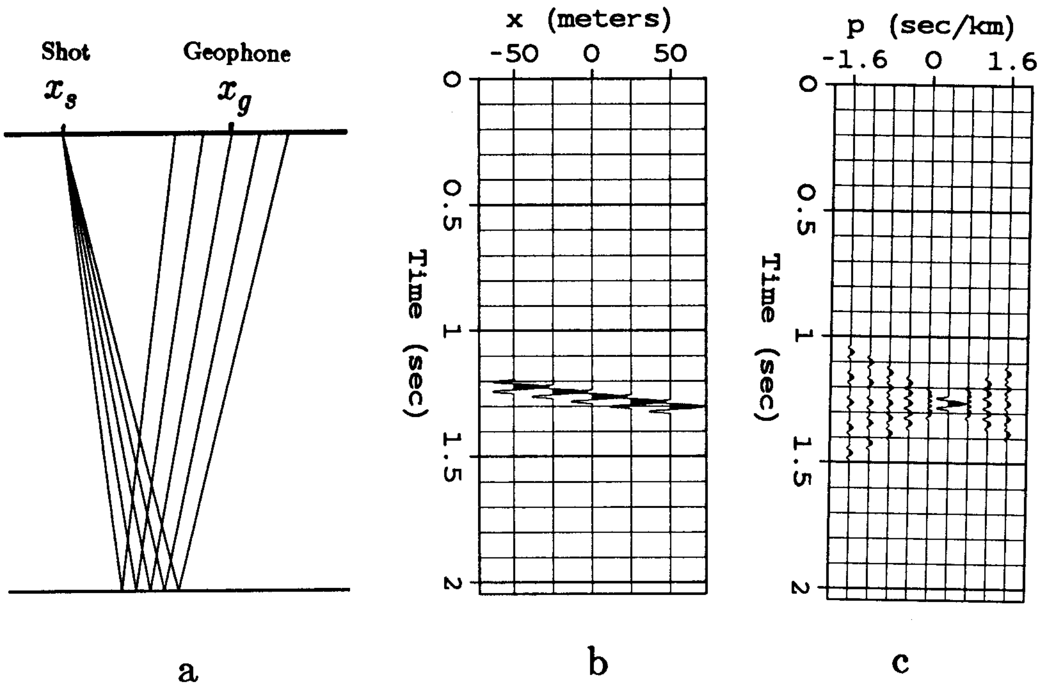

FIG. 1: Determining ray parameters. Figure 1a shows how a short-base common-shot gather might be collected in the field, while Figure 1b shows a possible outcome of this seismic experiment. Figure 1c shows the result of slant stacking this data; from the slant-stack section it is possible to determine the ray parameter p_g and travel time t.

rays. There is also the general problem inherent in any method where rays are traced—rays are fickle, and the smallest changes in velocity can cause them to shoot off wildly in all directions. Solutions to the wave equation are much more stable in this regard (Rothman, this report).

REVIEW OF CDR (CONTROLLED DIRECTIONAL RECEPTION)

The method of Controlled Directional Reception (CDR) was invented in the United States, but reached its highest development in the Soviet Union (Riabinkin et al., 1962; Sword, 1984a). The CDR approach might best be summarized by the phrase, “slant stacks over short bases”.

As illustrated in Figure 1, a portion of a common-shot or common-geophone gather can be transformed into a slant-stack section. The peaks of the slant stack can be picked, yielding the travel times and ray parameters of the various events in the section.

In practice, it is preferable to use a semblance-weighted slant stack (Stoffa et al., 1981; Kong et al., 1985), as the peaks are sharper and more easily picked, and aliased noise (such as ground roll) is better attenuated. Figure 2 shows a typical common-shot gather (a field profile). A portion of this gather was slant stacked and semblance weighted, with the result shown in Figure 3a. The peaks picked by a simple-minded automatic picking algorithm are shown in Figure 3b

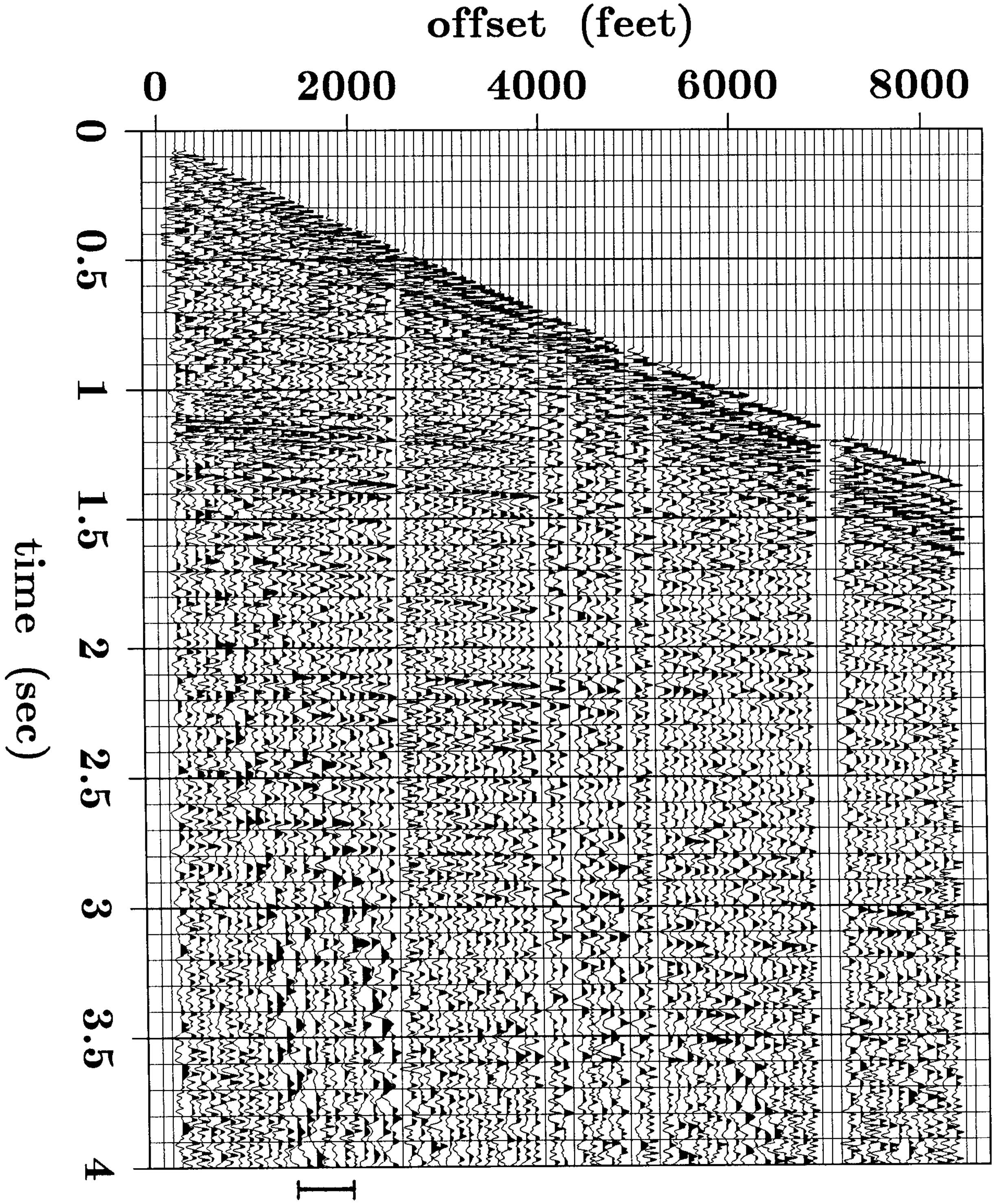

FIG. 2: A common-shotpoint gather, courtesy of an anonymous sponsor (Sword, 1985). Offsets 1400–2000 (denoted by the heavy line at the bottom of the figure) will be used to produce Figure 3. The entire gather will be used to illustrate the automated picking of CDR parameters.

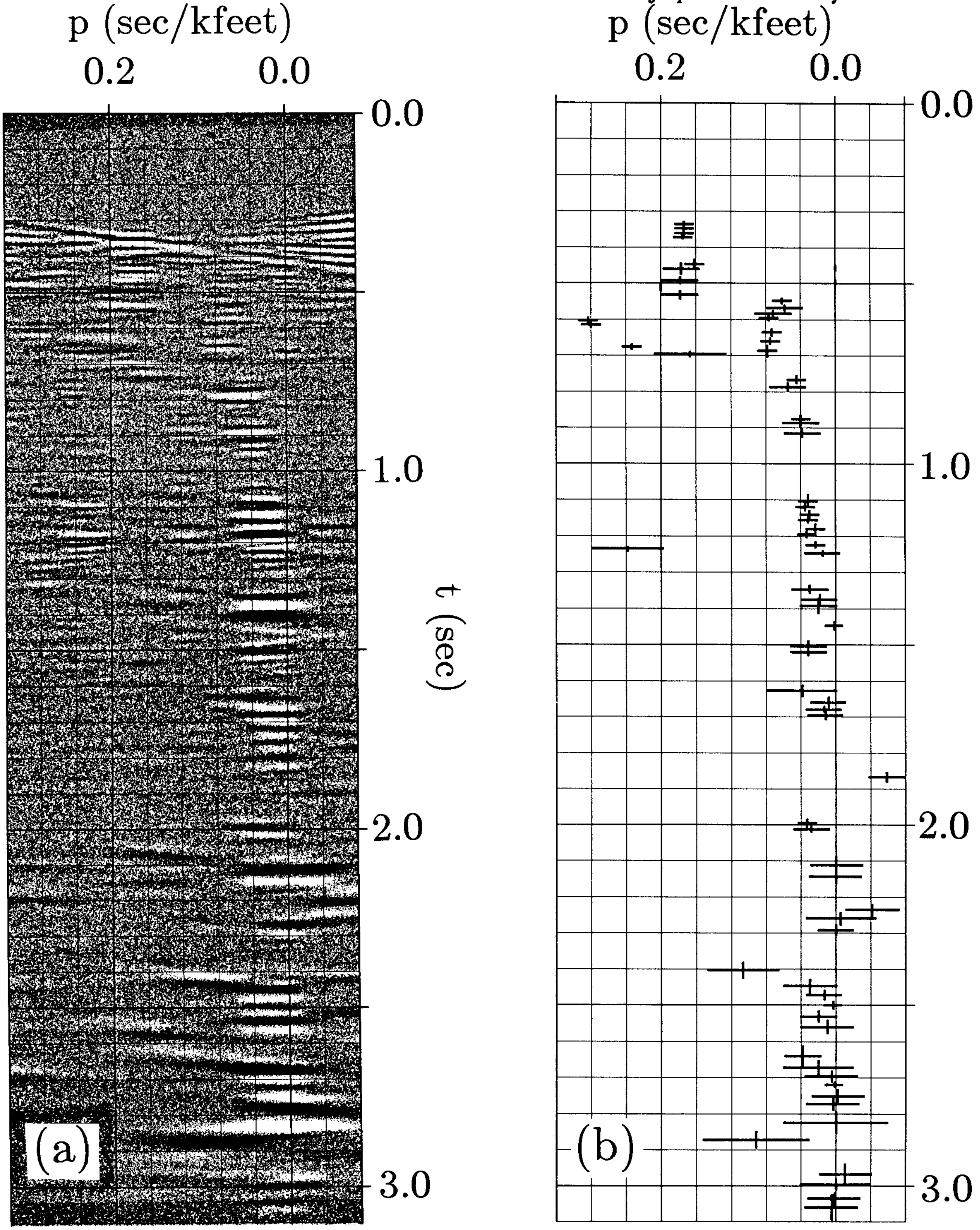

FIG. 3: Semblance slant stack and automatic picking. Offsets 1400–2000 in Figure 2 were slant stacked and semblance weighted to produce the data illustrated in Figure 3a. Figure 3b shows the peaks that were picked by an automatic picking algorithm.

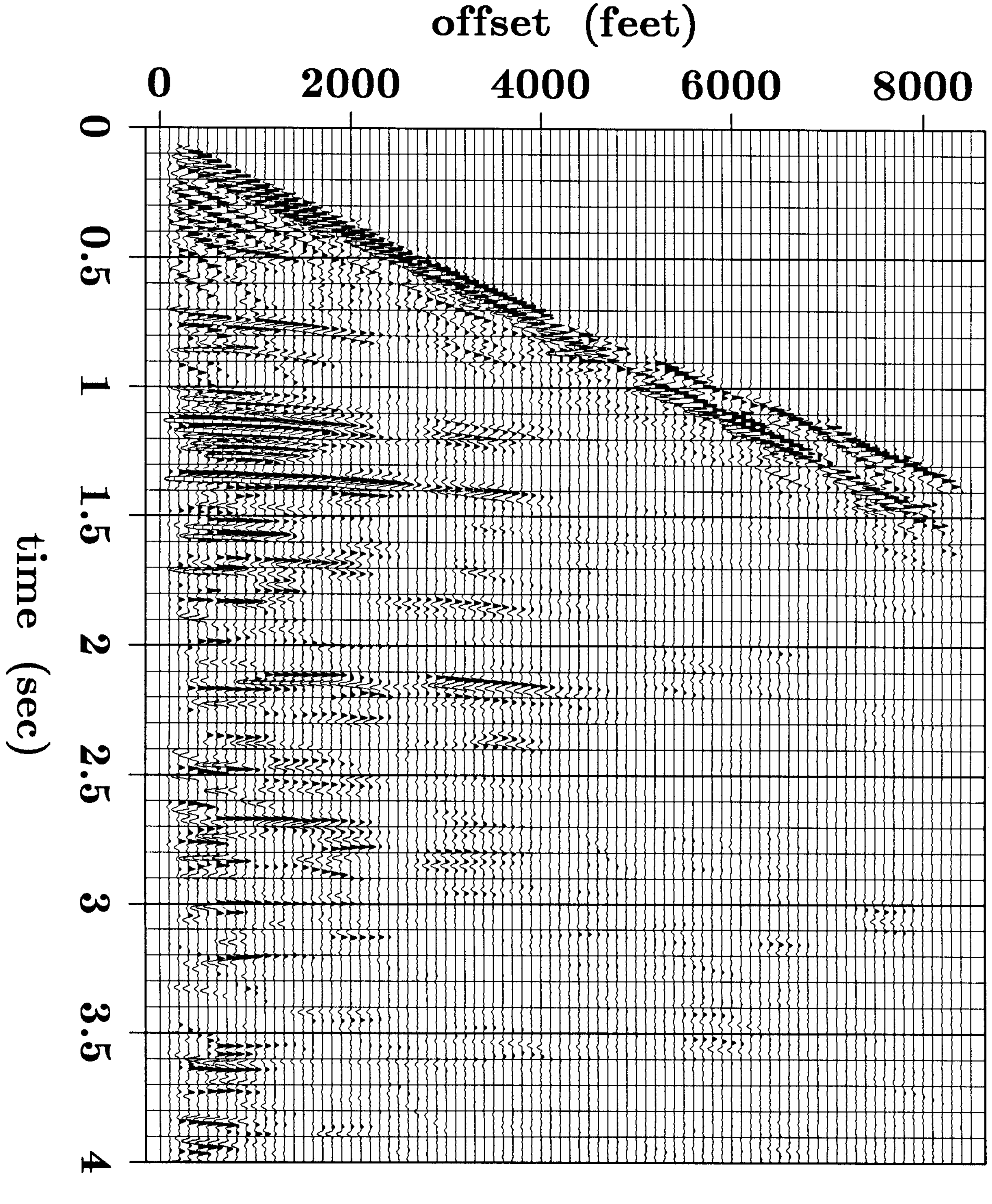

FIG. 4: A synthetic time section. The data in Figure 2 were slant stacked and automatically picked. The picks were used to reconstruct the original data, with the results shown here. The picking algorithm managed to find the main events, but it has picked up some noise as well.

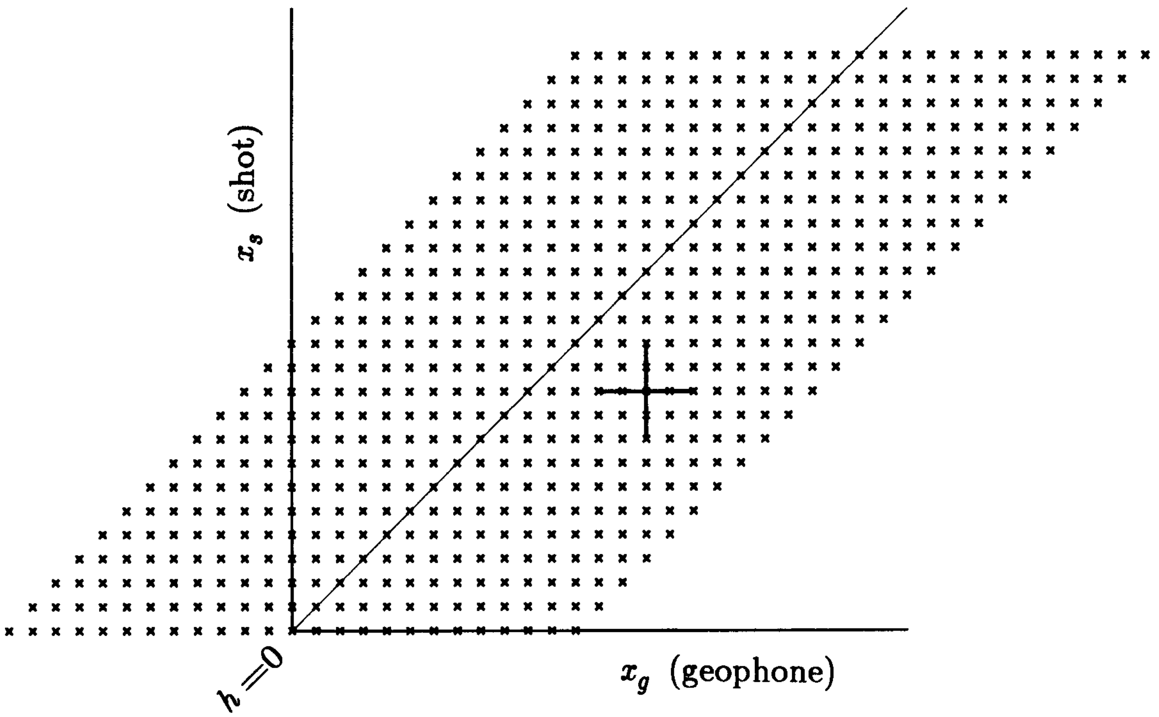

FIG. 5: A stacking chart showing the "summation bases" that might be used to determine p_s and p_g at a particular x_s and x_g. The heavy horizontal line denotes a short-base common-shot gather, whose traces can be slant stacked and picked in order to determine p_g. The heavy vertical line denotes a short-base common-geophone gather, whose traces can be slant stacked and picked in order to determine p_s.

(the picking algorithm is loosely based on ones described by Rapoport (1977)). The entire gather from Figure 2 was processed in this manner: seven adjacent traces were slant stacked, their peaks were picked, the base of summation was shifted by one trace, and the process was repeated. The use of a rolling slant stack (Ottolini, 1983) made the stacking program fairly efficient. The peaks were picked, and their corresponding ray parameters, travel times, and average shot-geophone distances stored. These stored parameters were then used to create a synthetic time section: each pick was interpreted as representing a short "dip bar" on the original time section, and a corresponding dip bar was drawn on the synthetic time section. The synthetic time section is shown in Figure 4; it agrees fairly well with the original time section, and demonstrates that the automatic picking algorithm was successful in finding the major events. Unfortunately, the algorithm was also successful at extracting non-existent events from noise. Clearly, more work remains to be done.

Once a portion of a particular common-shot gather has been picked (the portion representing data collected near a particular geophone), the reciprocal portion of the corresponding common-geophone section can be picked (see Figure 5). The picked parameters can be combined to give travel times, downgoing ray parameters, and upgoing ray parameters of all waves that were produced by that particular shot and that arrived at that particular geophone. This correlation process can be carried out for all shot-geophone pairs, yielding a fairly substantial number of picked parameters. These picked parameters are the input data for the inversion processes that will be described below.

CDR TOMOGRAPHIC INVERSION IN A LAYERED MEDIUM

Inverse problems are usually formulated in terms of a search for the minimum of an objective function, where the objective function describes the mismatch between the real data and the data predicted by a model. Typically, tomographic velocity inversions seek to minimize the discrepancy between observed and predicted travel times (see, for instance, Worthington (1984)). I have chosen a somewhat unusual objective function, which will be described and justified below. Some tomographic methods make the assumption that all rays are straight; I, like Bishop et al. (1985), have chosen to allow curvature of rays. Often it is possible to use Fermat's principle to justify using the same raypaths over several iterations, even though the velocity may have changed (Toldi, 1985); my objective function is such that Fermat's principle cannot be invoked. Finally, the non-linear inversion problem must be solved in some way; I have chosen to use a technique known as the Gauss-Newton method. Once the objective function, ray-tracing algorithms, and inversion algorithm are all implemented, tests may be run with synthetic data. The results of such tests are given below.

Possible objective functions

The input data for the inversion process consist of many sets of t, p_s and p_g picks, where t represents the observed travel time, and p_s and p_g represent the downgoing and upcoming ray parameters, of a particular wave. Each set of p_s, p_g, and t has associated with it a value of x_s, the position of the shot where p_s was measured, and x_g, the position of the geophone where p_g was measured. Of these five parameters, x_s and x_g (the shot and geophone position) are of course the most reliable, and t is almost as reliable. The remaining parameters, p_s and p_g, may not be especially accurate. If the velocity of the medium is assumed to be constant, then these five parameters, if accurate, are sufficient to determine that velocity (Sword, 1984a).

Given the five parameters and a proposed velocity model, there are several objective functions that could be used to determine how well the model fits the data. The most straightforward one is as follows: trace rays with ray parameters p_s and p_g from points x_s and x_g respectively. Keep going until the rays meet. Calculate the total predicted travel time. The objective function consists of the squared difference between the predicted and observed travel times.

This straightforward method is not the one that I chose to use. Figure 6 illustrates a couple

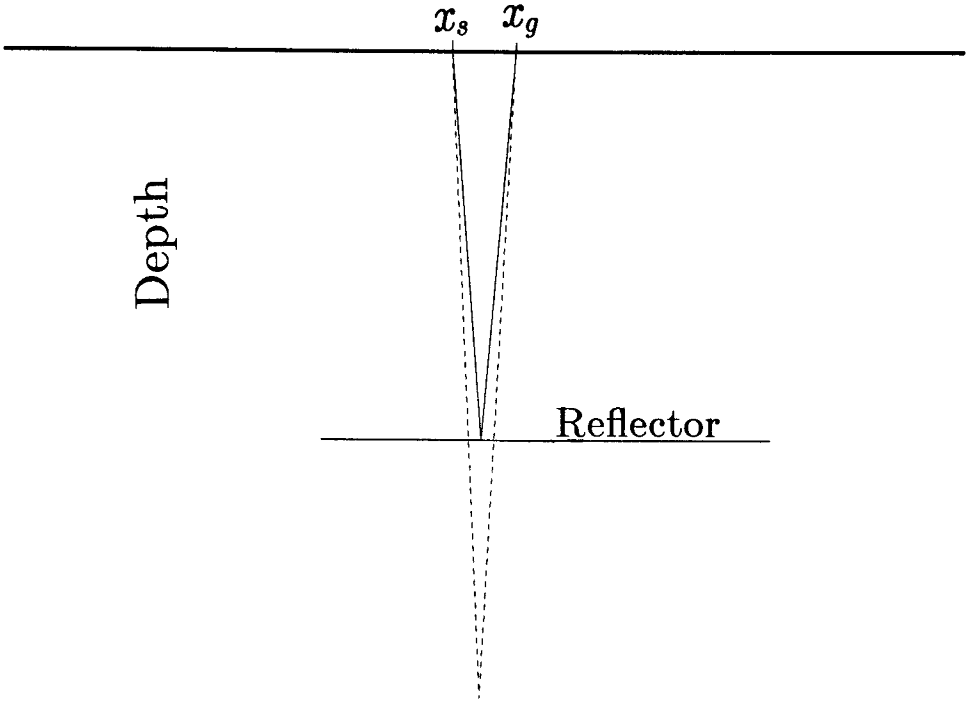

FIG. 6: The result of small errors in p_s and p_g. When x_s and x_g are near each other, small errors in the measurement of the ray parameters can lead to large errors in the predicted travel time. Here the solid line shows the actual raypath in the medium, while the dotted line shows the raypath predicted on the basis of slightly erroneous ray-parameter measurements.

of the reasons. Suppose that the velocity model happens to be correct. Since the model is correct, the objective function should be at a minimum. Suppose also that x_s and x_g are near each other, and that there are slight errors in p_s and p_g. The solid line in Figure 6 shows the actual raypath in the medium, and the dotted line shows the raypath that is predicted on the basis of the erroneous measurements of p_s and p_g. The measured travel time is based on the travel time along the solid line, while the predicted travel time is based on the travel time along the dotted line. So as a result of a small error in measuring the ray parameters, there is a large error in the predicted travel time, which in turn incorrectly suggests that there is a large error in the velocity model. The first problem, then, is that small ray-parameter errors (and these are unavoidable) can lead to large errors in velocity measurement. The second problem has to do with the way the velocity model is updated. Typically in tomographic problems the velocity will be updated along raypaths where a velocity error has been detected. This means that in the example depicted in Figure 6, the velocity will be updated not only from the surface to the reflector depth, as it should be, but also from the reflector all the way down to the point where the two erroneous rays intersect. It appears, then, that travel-time error is not a good measurement to use in the objective function.

There is another reason for not trying to minimize errors in travel time. In tomographic inversion problems, one does not find an objective function that minimizes, for instance, the error in x_s, the shot position—the shot position is assumed to be known exactly. Instead, one tries to minimize the error in another parameter (t, perhaps) that is more likely to be erroneous. In a similar fashion, when the five-parameter CDR data sets described in this paper are being used, there is no sense in trying to minimize the error in the parameters x_s, x_g, or t, which are assumed to be known relatively precisely. The objective function should be based on errors in the ray parameters p_s and p_g, not on errors in the measured versus the predicted travel times.

There are difficulties, however, in basing an objective function directly on errors in the ray parameters. I won't go into details, but it should not be too difficult to see that to measure these errors, an excessive amount of ray tracing would have to be performed. Since t would be assumed to be an exact parameter, rays would have to be traced and retraced until a pair of rays were found that intersected at a point where the travel time predicted by the ray tracing exactly matched the measured travel time.

The x-error objective function

Now I will describe the objective function I have chosen to use. This objective function is not overly sensitive to ray-parameter errors, nor is it expensive to compute.

Suppose that the upgoing and downgoing rays have both been traced back to some depth z in the velocity model (see Figure 7). The rays, in general, do not intersect at that depth. The predicted travel time at depth z can be defined as the travel time along the downgoing ray plus the travel time along the upgoing ray. Then for each pair of rays there is a depth z_e, which is defined as the depth where the predicted travel time equals the observed travel time. Once the value of z_e has been determined, the horizontal distance between the two rays at that depth can be measured; this distance is defined as x_{err}. The objective function to be minimized is x_{err}^2. The value of x_{err} is clearly zero if the ray parameters and travel time have been measured correctly and if the velocity model has been chosen correctly. If the ray parameters p_s and p_g have been measured incorrectly, x_{err} will be non-zero, but it won't vary wildly, even when the shot-geophone distance is small (compare this behavior to that of the objective function based on travel-time errors). And the rays need be computed only once per iteration; no expensive two-point ray-tracing algorithm is necessary. Another advantage of this objective function will become evident when linearization of the minimization problem is discussed.

Ray tracing

Ray tracing in a laterally homogeneous medium is rather easy. If p is the ray parameter and

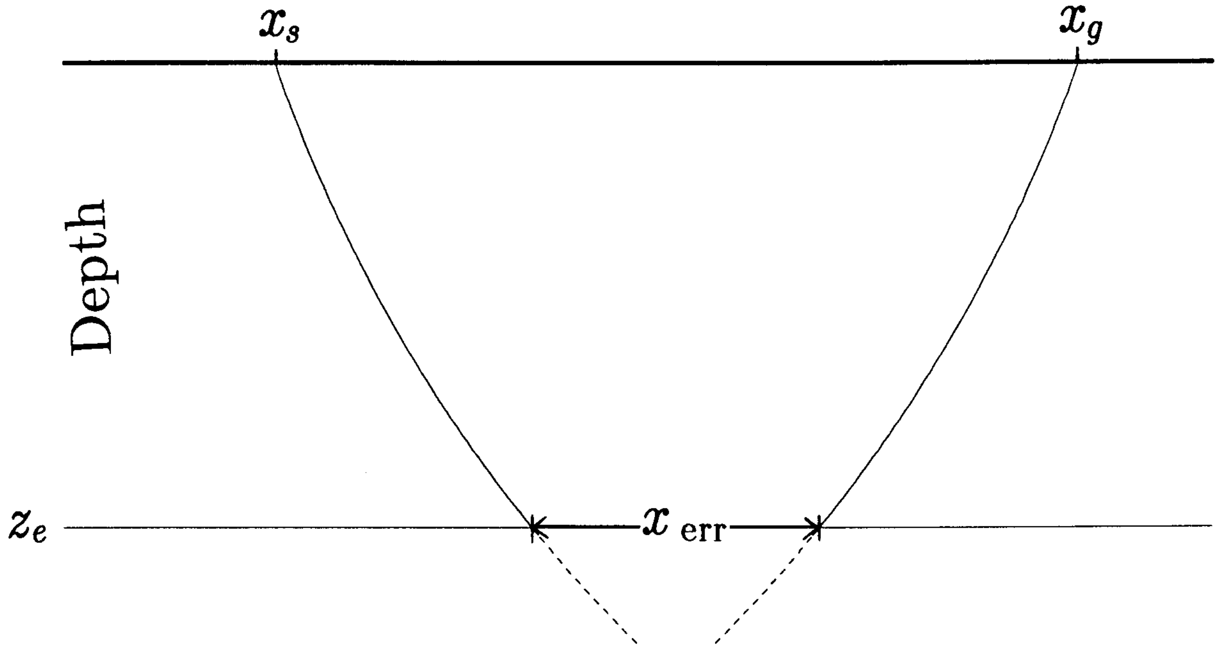

FIG. 7: A better objective function. Two rays are traced down to a depth z_e, the depth where their combined predicted travel time equals the measured travel time. The distance between the rays at this depth is defined to be x_{err}. This objective function is reasonably stable in the presence of errors in the measured ray parameters.

z is depth, then

$$\sin\theta(z) = p\,v(z), \tag{1}$$

where $\theta(z)$ is the angle, measured from the vertical, of the ray at depth z, and $v(z)$ is the velocity of the medium. The ray parameter p is constant in a horizontally layered medium. If a particular layer has a thickness Δz, then the ray will travel within that layer a horizontal distance (defined as Δx) of

$$\Delta x = \Delta z \tan\theta(z). \tag{2}$$

Similarly, travel time (Δt) through that layer is

$$\Delta t = \frac{\Delta z}{v(z)\cos\theta(z)}. \tag{3}$$

It is assumed that velocity v (and thus θ as well) remains constant over the distance Δz. These three equations, plus some well-known trig identities ($\cos\theta = \sqrt{1-\sin^2\theta}$, for instance) are sufficient for finding x_{err}, given a velocity model and the parameters x_s, x_g, p_s, p_g, and t.

Minimizing the objective function

Now that an objective function has been defined, it is necessary to devise a method of minimizing it. First, however, the minimization problem must be stated somewhat more precisely than it has been up to this point.

First, a definition of the model: the model is defined to consist of NZ layers, each of thickness Δz. The top of layer i lies at depth $i\Delta z$; thus the top layer is layer 0. The velocity in layer i, then, can be written as v_i, implying that $v(z) = v_i$ in the range $i\Delta z \leq z < (i+1)\Delta z$.

Next, a definition of the data: the data consist of M sets of CDR parameters; the lth set of parameters contains $p_{s(l)}$, $p_{g(l)}$, $x_{s(l)}$, $x_{g(l)}$, and $t_{(l)}$.

Finally, a definition of the objective function: for the lth set of CDR parameters, and dependent on the velocity model $\mathbf{v}$ (the vector composed of $v_0, v_1, \ldots, v_{NZ-1}$), there is an associated error term $x_{\mathrm{err}(l)}$. This term is found by the ray-tracing methods described above, and it is conveniently written as $x_{\mathrm{err}(l)}(\mathbf{v})$. The vector composed of all M of the error terms can be written as $\mathbf{x}_{\mathrm{err}}$. The objective function f, then, can be written as

$$f(\mathbf{v}) \equiv \sum_l x^2_{\mathrm{err}(l)}(\mathbf{v}) = \|\mathbf{x}_{\mathrm{err}}(\mathbf{v})\|^2. \tag{4}$$

The problem that is to be solved is now easily stated: find the value of $\mathbf{v}$ that minimizes $f(\mathbf{v})$. This is a non-linear minimization problem; out of the many possible ways to solve such problems (Gill et al., 1981), I have chosen to use a modified Gauss-Newton method. It is not surprising that Bishop et al. also chose this method—it makes efficient use of the information provided by ray tracing, without requiring too many relatively expensive ray-tracing steps. It is an iterative method.

This method works as follows. First, let $\mathbf{v}^k$ be the current velocity model, the input to the kth iteration. The desired new velocity model is $\mathbf{v}^{k+1}$, which is to be found according to the formula

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \Delta\mathbf{v}. \tag{5}$$

In the standard Gauss-Newton method, the problem that is solved is:

$$\underset{\sim}{\mathbf{A}}^k \Delta\mathbf{v} = -\mathbf{x}^k_{\mathrm{err}}, \tag{6}$$

where $\mathbf{x}^k_{\mathrm{err}} \equiv \mathbf{x}_{\mathrm{err}}(\mathbf{v}^k)$, $\underset{\sim}{\mathbf{A}}^k$ is the Jacobian matrix derived from $\mathbf{x}^k_{\mathrm{err}}$ (this matrix will be described in more detail below), and $\Delta\mathbf{v}$ is the difference between the current velocity model and the desired model. Equation (6) represents a sparse overdetermined system of linear equations; it can be solved by the conjugate gradient algorithm LSQR (Paige and Saunders, 1982).

There are a few details that must be added to this general description. It is necessary to add a damping term; otherwise the solution goes rapidly out of control. It is common to employ a damping term which tries to keep the values of $\Delta\mathbf{v}$ as low as possible. This is what Bishop et al. used, and the use of this term changes the Gauss-Newton inversion into a Levenberg-Marquardt inversion (Gill et al., 1981). For the purposes of the present problem, however, it turned out to be more helpful to follow Toldi (1985) and employ a damping term that tries to keep differences between adjacent Δv_i's as low as possible. Suppose a matrix $\underset{\sim}{\mathbf{D}}$ is defined such that $(\underset{\sim}{\mathbf{D}}\Delta\mathbf{v})_i = \eta_v(\Delta v_{i+1} - \Delta v_i)/\Delta z$, where η_v might be termed the "vertical damping coefficient". Then the minimization problem no longer takes the form of equation (6), but instead is expressed as

$$\begin{pmatrix} \underset{\sim}{\mathbf{A}} \\ \underset{\sim}{\mathbf{D}} \end{pmatrix} \Delta\mathbf{v} = \begin{pmatrix} \mathbf{x}_{\text{err}} \\ \mathbf{0} \end{pmatrix} . \tag{7}$$

This equation, too, can be solved numerically by the LSQR algorithm.

An additional complication is that it is often desirable to write equation (5) in the form

$$\mathbf{v}^{k+1} = \mathbf{v}^k + \lambda\Delta\mathbf{v}, \tag{8}$$

where λ is the scalar which minimizes $f(\mathbf{v}^k + \lambda\Delta\mathbf{v})$. If $\mathbf{v}^k$ is not too far from the value of $\mathbf{v}$ which minimizes f, λ will be close to 1. In other cases, however, setting λ equal to 1 may not produce good results. In practice, it is very expensive to find λ exactly, since ray tracing must be performed every time f is calculated. It is relatively cheap, however, to find an approximate value of λ, if $f(\mathbf{v}^k)$, $f(\mathbf{v}^{k+1?})$, $\underset{\sim}{\mathbf{A}}^k$, and $\underset{\sim}{\mathbf{A}}^{k+1?}$ are known (here $\mathbf{v}^{k+1?}$ is a trial value for $\mathbf{v}^{k+1}$, found by setting λ equal to 1 in equation (8), and $\underset{\sim}{\mathbf{A}}^{k+1?}$ is the corresponding Jacobian matrix). Without going into details, I will note that given such information, an approximate value of λ can be found through the Cubic Fit line-search method (Luenberger, 1984, pp. 205-206).

It remains to be shown how the Jacobian matrix $\underset{\sim}{\mathbf{A}}$ is derived. This matrix is defined according to the formula

$$A_{ij}^k \equiv \left(\frac{\partial x_{\text{err}(j)}}{\partial v_i}\right)_{\mathbf{v}=\mathbf{v}^k} . \tag{9}$$

The resulting formula for $\underset{\sim}{\mathbf{A}}$, as well as its derivation, has been relegated to Appendix A. One interesting result, however, is that A_{ij} in general depends only on $p_{s(j)}$, $p_{g(j)}$, v_i, and $v_{I(j)}$, where $v_{I(j)}$ is the velocity at layer $I(j)$, the layer containing the endpoints of the jth pair of rays. This result helps make the problem a little bit more linear, since it means that A_{ij} does not depend on v_l if $l \neq (i \text{ or } I(j))$.

Results of tests on synthetic data

The CDR tomographic inversion method has been tested on synthetic data. These data consisted of sets of CDR parameters that corresponded to reflections off of layers ranging in depth from .2 to .7 (the units aren't important). The model velocity varied according to the formula

$$v(z) = v_0 + az \tag{10}$$

(a linear velocity gradient). Analytic ray tracing in such a velocity model is fairly straightforward, so constructing a synthetic data set was not too difficult. It should be noted that the ray

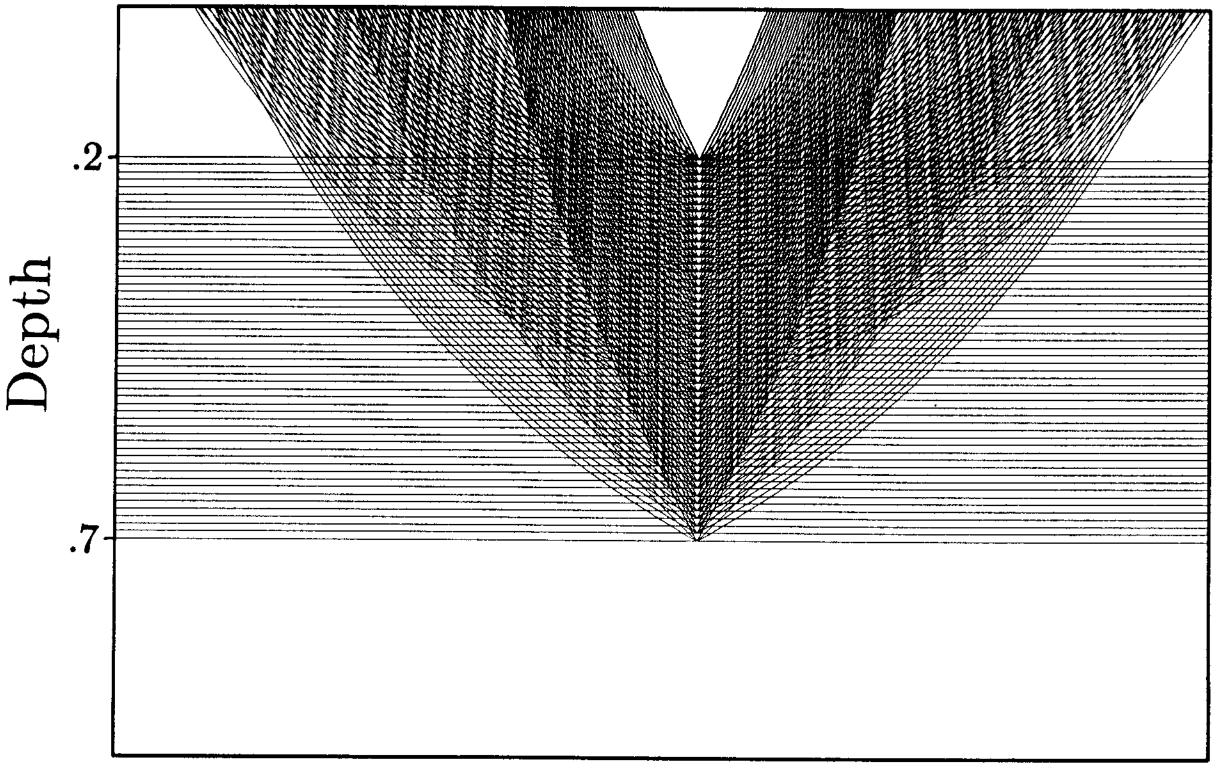

FIG. 8: Synthetic rays. A velocity model was created based on equation (10), with parameters $v_0 = 1.0$ and $a = 1.5$. Horizontal lines show the reflectors in the model, while the curved lines show the 205 raypaths that were used to create a synthetic data set.

parameters, p_s and p_g, were determined directly as part of the synthetic modeling program—the automatic picking algorithm described previously was not used, and thus there were no picking errors in the synthetic ray parameters. A model was chosen with the parameters $v_0 = 1.0$ and $a = 1.5$ (see equation (10)). Figure 8 shows the raypaths that were generated during the modeling. The surface-position parameters, x_s and x_g, were rather arbitrary; they depended only on where the rays happened to intersect the surface. A total of 205 sets of CDR parameters were generated.

These synthetic data were used as input to an inversion program. The inversion program's task was to find the velocity model that would best fit these data. The model was a flat-layered medium with layers a distance of $\Delta z = .05$ apart; the total depth of the model was 1.0. Again, the units are unimportant, but the reader is free to choose any sets of units that seem reasonable (distance in tens of kilometers and time in seconds, for instance). A vertical damping coefficient of $\eta_v = 1.0$ was used, and the initial velocity model was a constant velocity everywhere of 0.8. Figure 9 shows the results of the first 8 iterations on a scale of velocity versus depth, and Figure 10 shows the results after 8 iterations, on a scale of percent velocity error versus depth (the error is measured against the original velocities that were used to create the synthetic data). The results

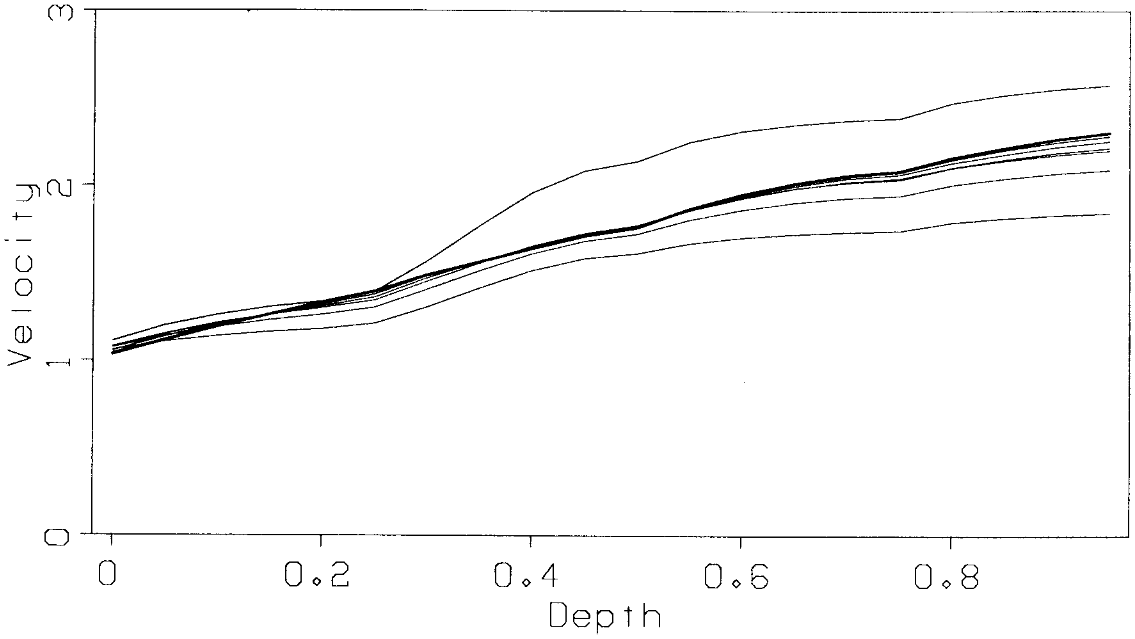

FIG. 9: Iterative inversion. The synthetic data generated by the rays in Figure 8 were inverted by the method described in this paper. The results of the first 8 iterations are show here; the result of the 8th iteration is drawn with a thicker line.

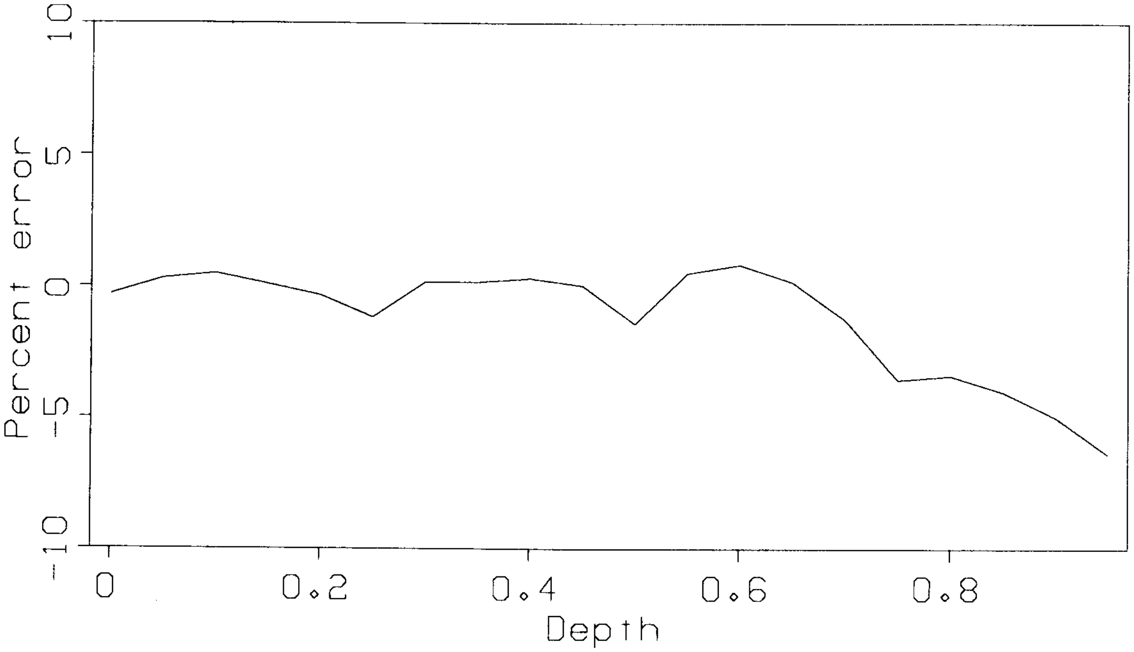

FIG. 10: Velocity error. The results of the 8th iteration of the velocity inversion (see Figure 9) were compared to the original velocity model. The deepest data came from a depth of .7, so below that depth the inversion was unconstrained.

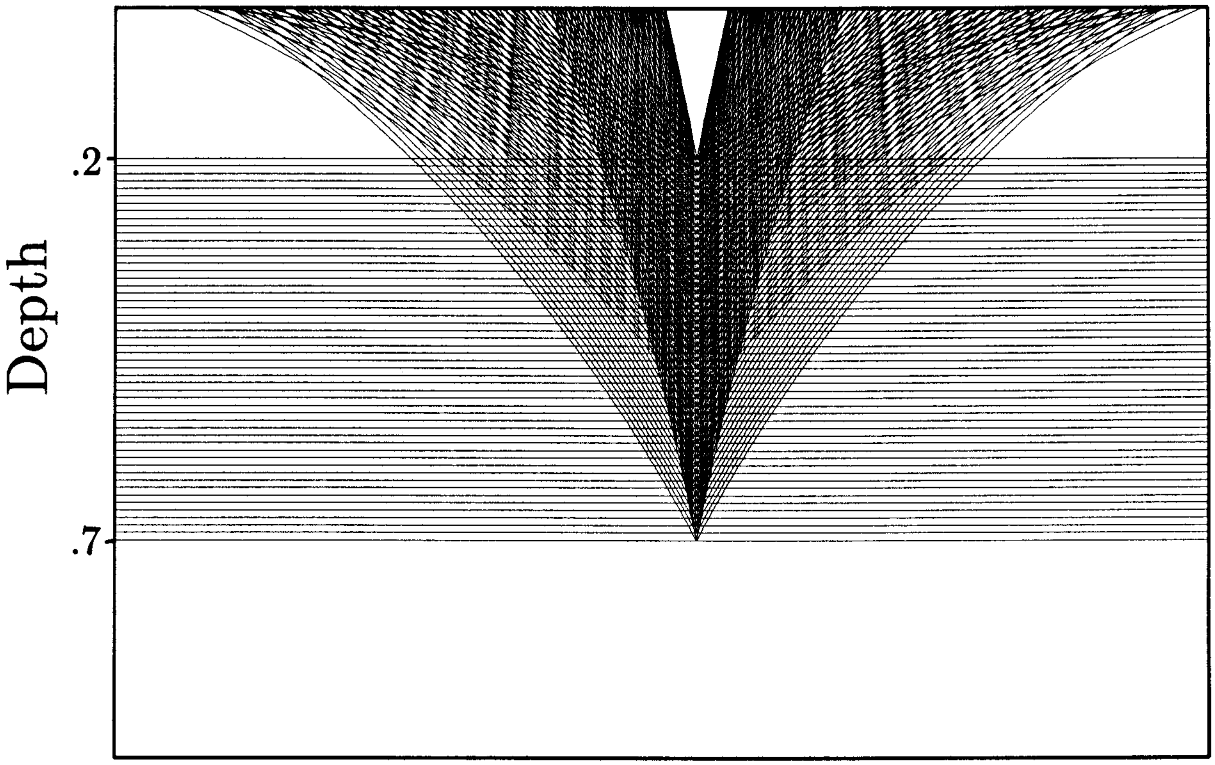

FIG. 11: Synthetic rays. This Figure is analogous to Figure 8, but now $v_0 = 2.5$, $a = -1.5$, and the number of rays is 168.

are encouraging. It is clear from Figure 10, however, that the damping was not as good as it might be. Recall that the algorithm described previously uses constraints that encourage $\Delta\mathbf{v}$, the velocity update, to vary smoothly, but no such constraints are placed on $\mathbf{v}$ itself.

Figure 11 shows the raypaths in a second model, whose velocity again varied according to equation (10), but this time with $v_0 = 2.5$ and $a = -1.5$ (not a typical geological model). There were 168 sets of CDR picks. The same initial guess (a constant velocity of .8) was used in the inversion program, and a vertical damping coefficient of $\eta_v = 1.0$ was again used. Figure 12 shows the results of the first 8 iterations. Figure 13 shows the percentage error after 8 iterations. Once more the velocity function is close to the original velocity, but again it is less smooth than is desirable.

INVERSION IN A 2-D INHOMOGENEOUS MEDIUM

While velocity analysis in a laterally homogeneous medium is somewhat useful, it has been done before, in very many ways. More useful would be a tomographic inversion of laterally and vertically inhomogeneous media. Fortunately, the method described in this paper may be extended in a straightforward way to the fully inhomogeneous case. The objective function remains the

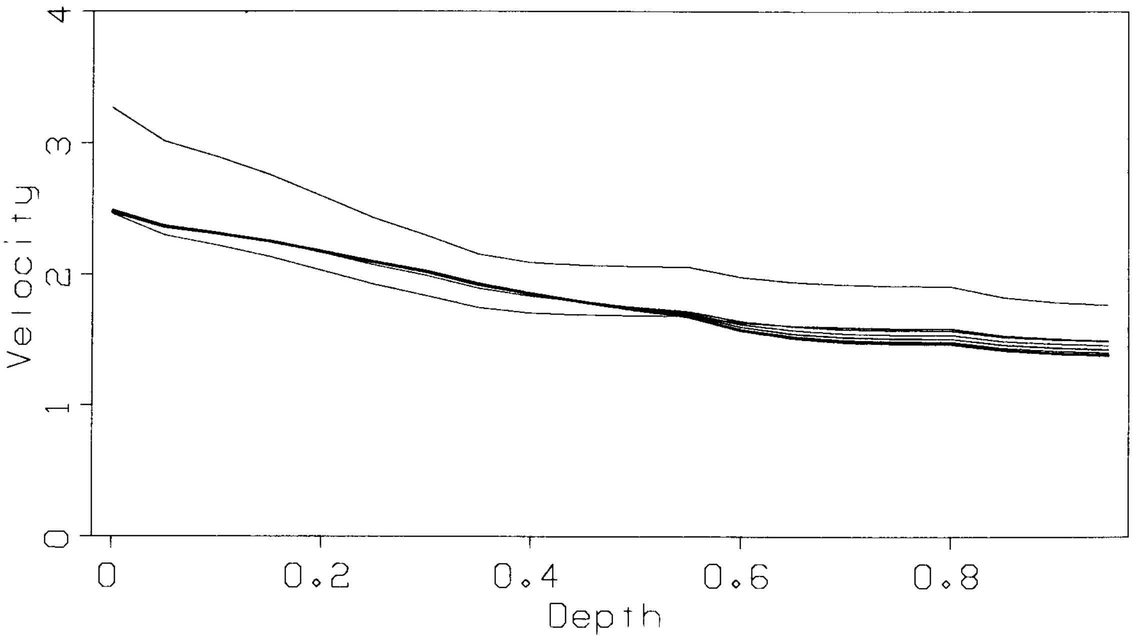

FIG. 12: Iterative inversion. This Figure is analogous to Figure 9, but now a different synthetic data set (see Figure 11) is being inverted. The heavy line denotes the result of the last (8th) iteration.

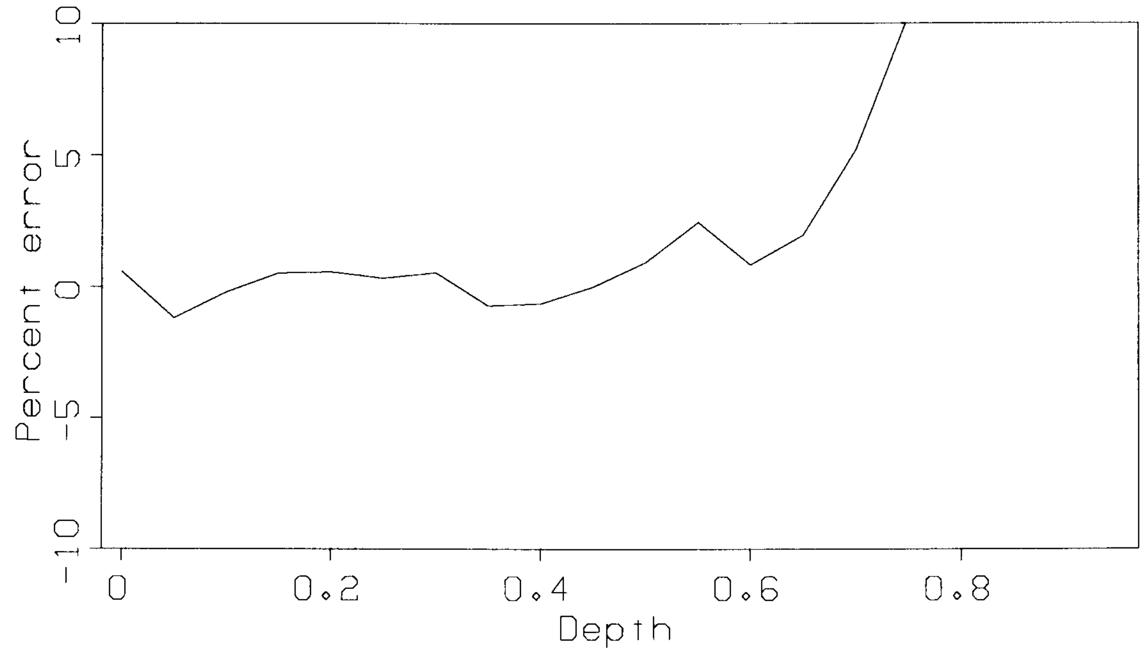

FIG. 13: Velocity error. This Figure is analogous to Figure 10, although now the results shown in Figure 12 are analyzed. Again, the deepest data came from a depth of .7, so the velocity below that depth is unconstrained.

same. The ray tracing becomes more complicated, however, and the Jacobian matrix $\underset{\sim}{\mathbf{A}}$ is not as easy to determine as before. These are not, however, insuperable obstacles.

Ray tracing in two dimensions

When velocity varied only vertically, so that $v = v(z)$, it was possible to discretize the velocity model such that the velocity remained constant within each layer. However, if velocity varies horizontally as well, it is not desirable simply to discretize the model into little boxes, with the velocity constant within each box. It is preferable to discretize the model into boxes while allowing a horizontal velocity gradient to exist within each box. That is, if v_i is the velocity associated with a particular box, a_i is the value of the associated velocity gradient, and x_i is the horizontal position of the center of the box, then

$$v(x,z) = v_i + a_i(x - x_i) \tag{11}$$

within the box. The values of a_i are chosen in such a way to make $v(x,z)$ a piecewise-continuous function in x, even at the boundaries between boxes. Note, however, that $v(x,z)$ will not be a continuous function in z, just as it was not a continuous function in z in the horizontally layered case.

Ray tracing is, of course, more complicated in the presence of a horizontal gradient. Suppose a ray, with ray parameter p_0, crosses into a new box at $x = 0$, $z = 0$. Let the velocity at that point be v_0, let the velocity within the box be described by the formula $v_0 + ax$, and define $k \equiv -a/v_0$. Then

$$\sin\theta = v_0 p_0, \tag{12}$$

where θ is the angle of the ray from the vertical immediately *after* it has entered the box. Suppose the ray emerges from the box at the point $x = \Delta x$, $z = \Delta z$, and that just before it emerges, it has an angle from the vertical of $\theta + \Delta\theta$. Then the following two formulas (adapted from Bishop et al.) hold:

$$\cos(\theta + \Delta\theta) = (1 - k\Delta x)\cos\theta, \tag{13}$$

and

$$\sin(\theta + \Delta\theta) = \sin\theta + k\Delta z\cos\theta. \tag{14}$$

Once $\sin\theta$ has been determined from equation (12), $\cos\theta$ may be obtained from a well-known trig identity, and Δz is known (it is the thickness of the layer, if the ray travels all the way from the top of the layer to the bottom). Then $\sin(\theta + \Delta\theta)$ may be obtained from equation (14), $\cos(\theta + \Delta\theta)$ obtained by a trig identity, and equation (13) solved to yield

$$\Delta x = \frac{1}{k}\left(1 - \frac{\cos(\theta + \Delta\theta)}{\cos\theta}\right). \tag{15}$$

Noted that when k is close to zero, equation (2) should be used in place of this equation. If Δx is known, and Δz is the unknown quantity (this could happen if the ray runs into the boundary between two horizontally adjacent boxes), equations (13) and (14) can be solved to yield a similar formula for Δz as a function of Δx. I will not give it here.

Another important parameter that must be determined is p_1, the ray parameter at the point of emergence. It is easily shown that $\sin(\theta+\Delta\theta) = (v_0+a\Delta x)p_1$, and this equation, in conjunction with equation (13) and the definition of k, can be solved to yield

$$p_1 = \frac{\cos\theta}{v_0}\tan(\theta+\Delta\theta). \tag{16}$$

Since velocity discontinuities occur only with changes in z, p_1 remains unchanged as the ray crosses the interface into the next box, and it can be used as the input ray parameter, p_0, when the ray is traced through this next box.

The final parameter that must be calculated is Δt, the amount of time the ray spends in a particular box. This quantity can be found from the integral

$$\Delta t = \int_s \frac{ds}{v(s)}, \tag{17}$$

where s represents the path that the ray follows within the box. After a fair amount of algebra, this integral can be transformed into a more useful formula:

$$\Delta t = \frac{1}{kv}\log\left(\frac{\cos\theta\,(1+\sin(\theta+\Delta\theta))}{\cos(\theta+\Delta\theta)\,(1+\sin\theta)}\right). \tag{18}$$

When k is close to zero, it is preferable to use equation (3).

These ray-tracing formulas have been implemented in a computer program; Figure 14 shows the results of one test of this program. The dotted lines in this figure represent the path the rays would have taken if there had been a sharp, sloping, 2:1 velocity contrast between the two layers. The solid lines show the paths taken by rays traveling through a discretized model (the grid is composed of 60 horizontal × 20 vertical boxes). In place of a sharp contrast, a smoother velocity gradation has been used (its boundaries are marked by heavy dotted lines).

The Jacobian in two dimensions

Finding the Jacobian matrix $\underset{\sim}{\mathbf{A}}$ is not an easy task when v is a function of both x and z. Besides the expected difficulties of computing derivatives in a layer containing a horizontal gradient, there is an additional problem. Recall that in a previous section it was noted that in laterally homogeneous media, A_{ij} depends only on v_i and $v_{I(j)}$, where $I(j)$ is the index of the layer containing the endpoints of the jth ray. When the medium is laterally inhomogeneous, however, A_{ij} depends on v_i, v_{i+1}, $\ldots$, $v_{I(j)}$. As a consequence, computing A_{ij} is more expensive than for the flat-layer case, and it will probably be necessary to introduce some advanced ray-tracing

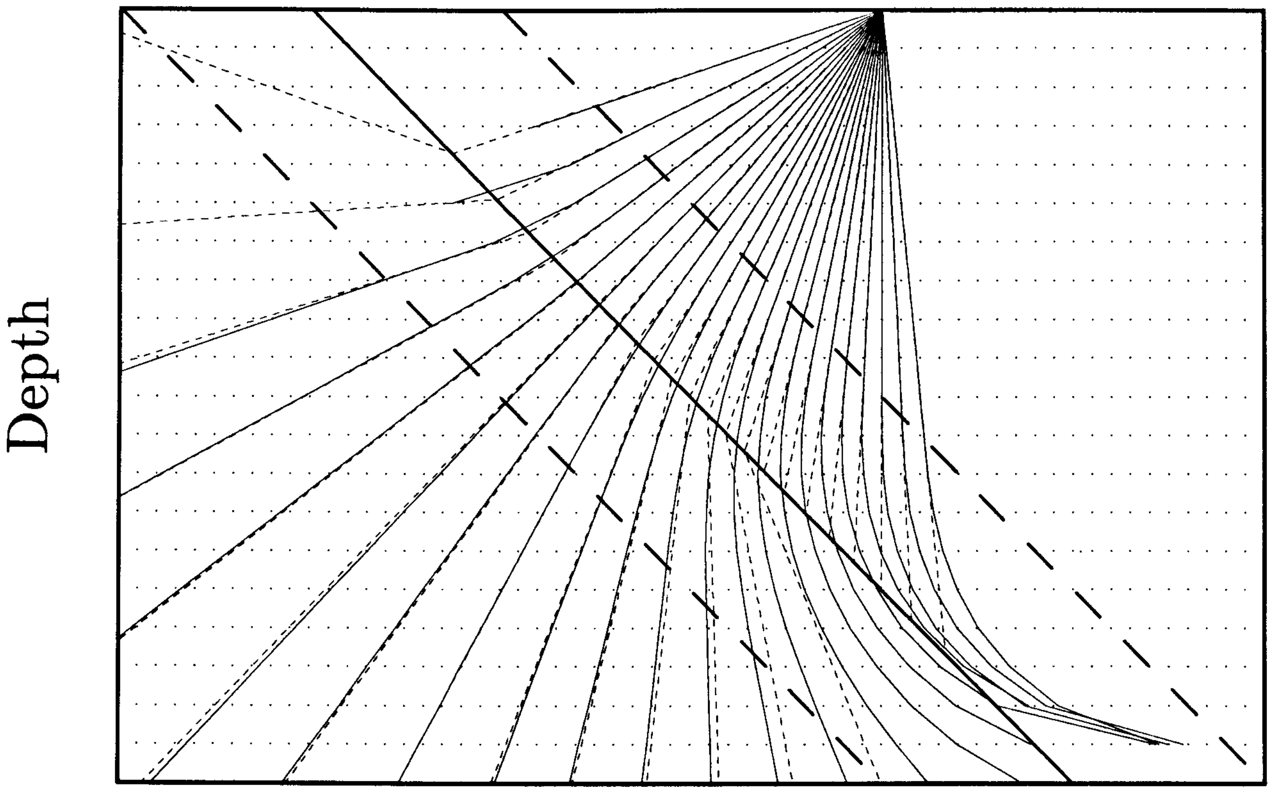

FIG. 14: A comparison of synthetic ray tracing techniques. The dotted lines show the result of analytic ray tracing in a medium with a sharp velocity contrast. The medium has a velocity of 1.0 to the right of the heavy solid line, and a velocity of 2.0 to the left. The solid lines show the results of ray-tracing with the numerical techniques discussed in this paper. The dots show the corners of the boxes used in this technique, and the heavy dashed lines denote a velocity gradient: the velocity is 1.0 to the right of the right-hand line, and 2.0 to the left of the left-hand line. The analytic and numerical results agree well except when the rays are super-critical or nearly so.

techniques, such as the Paraxial Ray Approximation (Červený and Pšenčík, 1984). The gruesome details of calculating $\underset{\sim}{\mathbf{A}}$ for the two-dimensional case are contained in Appendix B.

CONVERTED WAVES

In principle, there is no reason why separate velocity models could not be used for upgoing and downgoing waves. There would be twice as many unknowns as when a conventional velocity model is used, but this problem could be obviated by making the horizontal or vertical grid spacing twice as large. The reason for using separate upgoing- and downgoing-velocity models is that they would make it possible to analyze converted-wave data, where the upgoing and downgoing waves are of different types. This sort of velocity analysis would have an advantage over certain previously proposed methods (Sword, 1984b) in that it would not require any a priori assumptions about the ratio of P-wave to S-wave velocity. It was noted previously that

CDR tomographic inversion in the laterally homogeneous case is not worthwhile, since there are so many other ways of performing velocity analyses in such a medium. The velocity analysis of converted waves may be an exception to that statement.

CONCLUSIONS

CDR tomographic inversion appears to be a viable method of determining interval velocities. It has been shown that an algorithm can be constructed to pick CDR parameters (p_s, p_g, x_s, x_g, and t) from conventionally recorded seismic data. CDR tomographic inversion in a laterally homogeneous medium has been demonstrated to work on a synthetic data set, although the resulting velocity is not as smooth as it could be at depths where it is unconstrained.

More work remains to be done. The method must be extended so that it works when $v = v(x, z)$; it would also be worthwhile to explore further the question of converted waves.

ACKNOWLEDGMENTS

This paper has benefited greatly from useful discussions: with Bill Harlan, Jeff Thorson, and Will Gray about the inversion of CDR data; with Stew Levin about non-linear inversion techniques; with Rick Ottolini about slant stacks; with Paul Fowler about all sorts of things; and especially with Dan Rothman and Ivan Pšenčík about dynamic ray tracing and the paraxial ray approximation. I would also like to thank Jon Claerbout for bringing the LSQR conjugate-gradient algorithm to my attention.

REFERENCES

Bishop, T.N., Bube, K.P., Cutler, R.T., Langan, R.T., Love, P.L., Resnick, J.R., Shuey, R.T., Spindler, D.A., and Wyld, H.W., 1985, Tomographic determination of velocity and depth in laterally varying media: Geophysics, **50**, 903–923.

Červený, V., and Pšenčík, I., 1984, Gaussian beams in elastic 2-D laterally varying layered structures: Geophys. J. R. astr. Soc., **78**, 65–91

Fowler, P., 1985, Migration velocity analysis by optimization: linear theory: SEP–44 (this volume).

Gill, P.E., Murray, W., and Wright, M.H., 1981, Practical optimization: Academic Press Inc.

Gray, W.C., and Golden, J.E., 1983, Velocity determination in a complex earth: 53rd Annual International SEG Meeting, September 15, Las Vegas, Nevada.

Harlan, W., and Burridge, R., 1983, A tomographic velocity inversion for unstacked data: SEP–37, 1–7.

Kong, S.M., Phinney, R.A., and Chowdhury, K.R., 1985, A nonlinear signal detector for enhancement of noisy seismic record sections: Geophysics, **50**, 539–550.

Levin, S.A., 1985, Newton trace balancing: SEP–42, 69–80.

Luenberger, D.G., 1984, Linear and nonlinear programming (2nd edition): Addison-Wesley Publishing Company Inc.

Ottolini, R., 1983, Signal/noise separation in dip space: SEP–37, 143-149.

Paige, C.C., and Saunders, M.A., 1982, LSQR: an algorithm for sparse linear equations and sparse least squares: ACM Transactions on Mathematical Software, **8**, 43-71.

Rapoport, M.B., 1977, Determination of wave parameters through the CDR summation of seismic traces, *in* Riabinkin, L.A., Ed., Digital processing of reflection seismic data: Transactions of the Gubkin Institute of Petrochemical and Gas Production (Moscow), **120**, 17–22 (in Russian).

Riabinkin, L.A., Napalkov, Yu.V., Znamenski, V.V., Voskresenski, Yu. N., and Rapoport, M.B., 1962, Theory and practice of the CDR seismic method: Transactions of the Gubkin Institute of Petrochemical and Gas Production (Moscow), **39** (in Russian).

Rothman, D., 1985, Velocity estimation by simulated annealing: problems and prospects: SEP–44 (this volume).

Stoffa, P.L., Buhl, P., Diebold, J.B., and Wenzel, F., 1981, Direct mapping of seismic data to the domain of intercept time and ray parameter—A plane wave decomposition: Geophysics, **46**, 255–267.

Sword, C., 1984a, The method of controlled directional reception: SEP–41, 369–393.

Sword, C., 1984b, Approximating the kinematics of converted waves: SEP–41, 347–368.

Sword, C., 1985, Analysis of a two-component data set: SEP–42, 177–188.

Toldi, J.L., 1985, Velocity analysis without picking: Ph.D. thesis, Stanford Univ. (also SEP–43).

Worthington, M.H., 1984, An introduction to geophysical tomography: First Break, **2**, 20–26.

APPENDIX A

DETERMINING THE JACOBIAN WHEN $v = v(z)$

Recall from equation (9) that $\underset{\sim}{\mathbf{A}}^k$, the Jacobian matrix at the kth iteration, is defined according to the formula

$$A_{ij}^k \equiv \left(\frac{\partial x_{\mathrm{err}(j)}}{\partial v_i} \right)_{\mathbf{v}=\mathbf{v}^k} . \tag{A-1}$$

Some new definitions are now necessary: Δz_i is the vertical distance that the jth ray travels in the ith layer (here, as in later definitions, the j subscript has been omitted for the sake of clarity); Δx_i is the horizontal distance that this ray travels in the ith layer; and Δt_i is the travel time of this ray within the layer. Then it is possible to write:

$$\frac{\partial x_{\mathrm{err}(j)}}{\partial v_i} = \frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta x_i} \cdot \frac{d\Delta x_i}{dv_i} + \frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta t_i} \cdot \frac{d\Delta t_i}{dv_i} . \tag{A-2}$$

Now the problem is reduced to one of finding the value of each term.

First, however, there is a point that needs to be clarified. Recall that Δx_i has just been defined to be the horizontal distance traveled by the jth ray in the ith layer. This definition ignores the fact that two rays are being traced simultaneously in order to find $x_{\mathrm{err}(j)}$, and that $x_{\mathrm{err}(j)}$ is thus dependent on both of these rays. That is,

$$x_{\mathrm{err}} = x_{ge} - x_{se}, \tag{A-3}$$

where x_{se} is the horizontal position of the endpoint of the jth downgoing (shot) ray and x_{ge} is the horizontal position of the endpoint of the jth upgoing (geophone) ray. In practice, although a velocity change in a given layer will produce changes in both rays, it is most convenient to consider the effect of the change on only one of the rays at a time, and sum the results later into A_{ij}. This is the approach that will be used in this Appendix.

The first term in equation (A-2) is easily determined, after a moment of thought. Since v is a function only of z, x_{se} (or x_{ge}) varies exactly the same amount as Δx_i varies, independently of the velocity structure of the model. As a consequence of this result and of equation (A-3), and depending on which ray is under consideration,

$$\frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta x_i} = -1 \quad \text{(downgoing);} \qquad \text{otherwise,} \quad \frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta x_i} = 1 \quad \text{(upgoing).} \tag{A-4}$$

The second term in equation (A-2) can be determined by differentiating equation (2). The result is:

$$\frac{d\Delta x_i}{dv_i} = \frac{p\Delta z_i}{\cos^3 \theta_i}, \tag{A-5}$$

where θ_i is the angle of the ray in layer i as measured from the vertical.

The third term in equation (A-2) is somewhat more complicated to derive. To begin with,

$$\frac{\partial x_{\text{err}(j)}}{\partial \Delta t_i} = \frac{dx_{\text{err}}}{dz_e} \cdot \frac{dz_e}{d\Delta t_i}, \tag{A-6}$$

where z_e is the depth of the endpoints of the jth pair of rays. From equation (A-3),

$$\frac{dx_{\text{err}}}{dz_e} = \frac{dx_{ge}}{dz_e} - \frac{dx_{se}}{dz_e}, \tag{A-7}$$

which through simple trigonometry becomes

$$\frac{dx_{\text{err}}}{dz_e} = \tan\theta_{ge} - \tan\theta_{se}, \tag{A-8}$$

where θ_{ge} and θ_{se} are the angles at depth z_e of the upgoing and downgoing rays respectively. These angles can be derived from p_s and p_g through the rule that $vp = \sin\theta$.

The second part of equation (A-6) must be approached indirectly. First, note that even though travel time Δt_i within a layer may vary, the overall travel time t must remain constant. Thus, as Δt_i increases, $t - \Delta t_i$ decreases. As a result,

$$\frac{dz_e}{d\Delta t_i} = -\frac{dz_e}{dt}. \tag{A-9}$$

Next, it is useful to note that

$$\frac{dz_e}{dt} = \frac{1}{dt/dz_e}. \tag{A-10}$$

Then

$$\frac{dt}{dz_e} = \frac{dt_s}{dz_e} + \frac{dt_g}{dz_e}, \tag{A-11}$$

where t_s and t_g are the total travel times of the downgoing (shot) ray and the upgoing (geophone) ray, respectively. An application of the chain rule yields

$$\frac{dt}{dz_e} = \frac{dt_s}{dl_s}\frac{dl_s}{dz_e} + \frac{dt_g}{dl_g}\frac{dl_g}{dz_e}, \tag{A-12}$$

where l_s and l_g are the total path lengths of the downgoing (shot) ray and the upgoing (geophone) ray, respectively. Through the application of some trigonometry, and thanks to the fact that $v = dl/dt$,

$$\frac{dt}{dz_e} = \frac{1}{v_{se}}\frac{1}{\cos\theta_{se}} + \frac{1}{v_{ge}}\frac{1}{\cos\theta_{ge}}, \tag{A-13}$$

where $v_{se} \equiv v(x_{se}, z_e)$, and $v_{ge} \equiv v(x_{ge}, z_e)$.

Now it is possible to combine equations (A-6), (A-8), (A-9), (A-10), and (A-13) and write:

$$\frac{\partial x_{\text{err}(j)}}{\partial \Delta t_i} = -v_e \frac{\sin\theta_{ge}\cos\theta_{se} - \sin\theta_{se}\cos\theta_{ge}}{\cos\theta_{se} + \cos\theta_{ge}}, \tag{A-14}$$

where $v_e \equiv v(z_e)$.

The fourth term in equation (A-2) can be found by differentiating equation (3), with the result:

$$\frac{d\Delta t_i}{dv_i} = -\frac{\Delta t_i}{v_i} + \frac{p\Delta z_i \sin\theta_i}{v_i \cos^3\theta_i}. \qquad \text{(A-15)}$$

The final result, found by substituting equations (A-4), (A-5), (A-14) and (A-15) into equations (A-1) and (A-2), is:

$$A_{ij}^k = \pm\frac{p\Delta z_i}{\cos^3\theta_i} - v_e \frac{\sin\theta_{ge}\cos\theta_{se} - \sin\theta_{se}\cos\theta_{ge}}{\cos\theta_{se} + \cos\theta_{ge}} \left(-\frac{\Delta t_i}{v_i} + \frac{p\Delta z_i \sin\theta_i}{v_i\cos^3\theta_i}\right). \qquad \text{(A-16)}$$

The ± sign is positive when the ray under consideration is the upgoing (geophone) ray; otherwise it is negative.

APPENDIX B

DETERMINING THE JACOBIAN WHEN $v = v(x, z)$

When the medium is no longer assumed to be laterally homogeneous, equations (A-1) and (A-2) still hold, although there are some differences in the meaning of the notation (whereas, for instance, the index i previously referred to the ith layer, it now refers to the ith box). It is necessary, however, to go through equation (A-2) and find new values for each of the four terms. Three of these values can be determined in a fairly straightforward way. The other is more difficult to find.

The second term of equation (A-2) is found by differentiating equation (15), and simplifying. The differentiation is performed under the assumption that although v_i is varying, k_i, defined as $-a_i/v_0$ (see equation (10), and recall that v_0 is the velocity at the point where the ray enters the ith box), remains constant. The result is

$$\frac{d\Delta x_i}{dv_i} = \frac{\Delta z_i}{1 - k_i \Delta x_i} \cdot \frac{p_i}{\cos^3 \theta_i}, \tag{B-1}$$

where p_i is the ray parameter of the ray as it enters the ith box, and θ_i is the angle of the ray from the vertical immediately after it enters the box.

The third term of equation (A-2) is derived in the same way as equation (A-14), but without the assumption that $v_{se} = v_{ge}$:

$$\frac{\partial x_{\mathrm{err}(j)}}{\partial \Delta t_i} = -v_{se} v_{ge} \frac{\sin \theta_{ge} \cos \theta_{se} - \sin \theta_{se} \cos \theta_{ge}}{v_{se} \cos \theta_{se} + v_{ge} \cos \theta_{ge}}. \tag{B-2}$$

The fourth term of equation (A-2) can be obtained by differentiating equation (18), and simplifying. This differentiation is also performed under the assumption that although v_i is varying, k_i remains constant. The result is:

$$\frac{d\Delta t_i}{dv_i} = -\frac{\Delta t_i}{v_i} + \frac{p_i \Delta z_i \sin(\theta + \Delta\theta)_i}{v_i \cos^2(\theta + \Delta\theta)_i \cos \theta_i}, \tag{B-3}$$

where $(\theta + \Delta\theta)_i$ is the angle of the ray immediately before it emerges from the ith box.

The first term in equation (A-2) has been left for last. The reason is simple: an expression for it has not yet been derived. The behavior of this term depends in a complex way on the v and k values in all of the boxes that the ray travels through, starting from the ith box and going to the endpoint of the ray. The method of the Paraxial Ray Approximation (Červený and Pšenčík, 1984) offers the best hope of finding a way of numerically evaluating this term.