Modeling with Ray-Tracing in 2-D Curved Homogeneous Layered Media

V. PEREYRA*

Abstract: Algorithms for ray-tracing in a piecewise homogeneous medium with curved layer interfaces are described. Normal incidence and non-zero offset two-point and shooting are considered. Travel time and amplitude calculations are used in the generation of synthetic seismograms. Also, time-to-depth migration of seismic sections, an inverse problem, is described. The whole package has been implemented in a microcomputer, and it is integrated into a complete modeling system to aid the interpreter of seismic reflection surveys.

1. Introduction

A ray-tracing based modeling package has been developed for a micro-computer environment. In this paper we will discuss the theoretical and practical details of its implementation on an IBM-PC-XT machine. The purpose of the development was to provide an existing system, the Geological Modeling System (GMS) of Kim Tech Inc., Denver, Colorado, with a more powerful and versatile ray-tracing module. The original plan has been extended to cover some aspects of automatic inverse modeling, such as time-to-depth migration of material interfaces.

The types of media considered consist of piecewise constant velocity layers separated by curved interfaces. Pinchouts and other nonconformities are allowed in order to be able to model complex regions with interesting geological features.

The ray-tracing system is source-receiver oriented; i.e. with a two-point or global philosophy, although shooting is used to provide automatic starting and re-starting after failure capabilities. The two-point approach, coupled with appropriate continuation procedures, is considerably more efficient than conventional

* Weidlinger Associates, Palo Alto, CA 94304.

shooting-trapping techniques [1,9,10]. Parameterization in reflector offset or initial angle is used in order to make the algorithm robust and so that multiple arrivals can be obtained.

In the current implementation we have chosen to consider generalized curved layer media in order to cope with pinchouts, terminating layers, and the like. By generalized layers we mean layers that run across the whole model window, but that can be locally thin (i.e. pseudolayers). This approach simplifies noticeably the logic necessary for describing and tracing rays, since no checking is necessary when a layer disappears at a pinchout, etc.; however, it limits the generality of models that can be treated, and it will be changed in the future to one based upon connectivity information.

The interactive features already present in the menu-driven GMS allow the digitized input of models, easy input of material properties, and postprocessing of ray-traced time-amplitude data in the form of synthetic seismograms. With the addition of the ray-tracing capabilities, this provides a very versatile work station for performing many, if not all, of the most important tasks needed in the forward and, in a limited sense, inverse modeling of reflection data, in a user-friendly manner.

At present the system has options for normal incidence and non-zero offset ray-tracing. Both primary, secondary and tertiary reflections, to any or all the interfaces of the model can be requested. Also diffractions from pinchout corners can be generated. For each ray, geometrical spreading and elastic reflection-transmission coefficients are calculated. In the case of diffracted arrivals, the exact solution for the diffraction of a plane wave by a wedge [7] is used to generate the amplitude. Finally, successive time-to-depth migration of selected digitized horizons can be effected via ray-tracing, both for post- and pre-stack time sections.

The system and the hardware (especially if an IBM-AT machine with a floating-point co-processor is used) are sufficiently fast as to make this otherwise arithmetic hungry task a plausible proposition in an interactive environment. It goes without saying that these capabilities are considerably enhanced by the ease by which graphics can be integrated into the system.

2. Basic Algorithm for Two-Point Ray-Tracing

We consider two-dimensional piecewise homogeneous media separated by curved interfaces. To begin, we assume that the interfaces are non-intersecting, smooth, one-valued curves:

$$z = f_i(x) \ , \ i=1,\ldots,\text{Last} \ , \ x \in [x_m, x_M].$$

These conditions will be relaxed later on. Figure 1 depicts a typical model of this type.

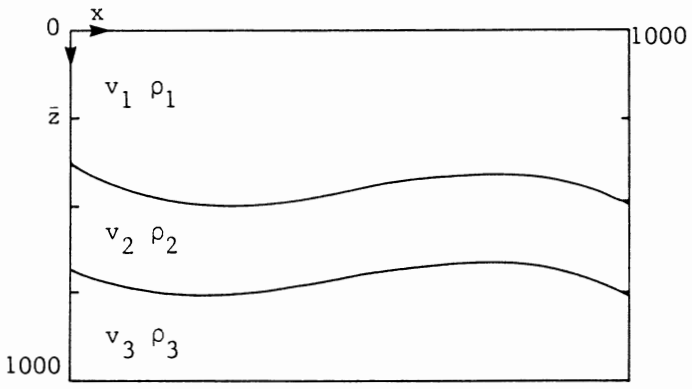

Figure 1. Typical layered model.

A geological window is chosen, and a Cartesian (x,z) system of coordinates is laid on it, with z (depth) running positive downwards. Interfaces are numbered consecutively downwards, the free surface being #1. The layers are numbered in a similar fashion, and v_i, ρ_i are, respectively, the velocity of propagation and density for layer i. Velocity can be either that corresponding to pressure or shear waves.

In Fig. 2 we see pictorially the two-point, ray-tracing problem: Given a source S and a receiver R, we want to find the ray (or rays) that join S and R, after traversing through or reflecting from a specified sequence of interfaces.

In conventional reflection surveys, both sources and receivers are usually on the free surface. However, in vertical seismic profiling, receivers, and sometimes sources, will be down a borehole, so it is of interest to consider the more general case of source and receivers in arbitrary locations.

The sequence of interfaces and regions a ray traverses is called its signature. Signatures are given in two integer arrays (ICR,IRG), whose lengths are related to the number of legs of the ray. Thus, in the example of Fig. 2, the depicted ray has signature:

$$ICR = [2,3,2,3,2,1] \quad , \quad IRG = [1,2,2,2,2,1]$$

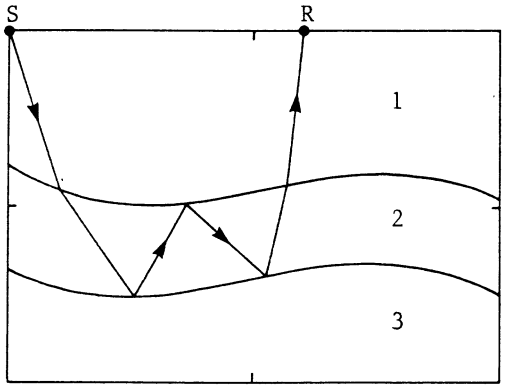

Figure 2. Two-point ray-tracing.

Rays are defined as paths of minimum (actually stationary) time, through which a point disturbance travels from S to R. In the present case, it is well-known that these paths are straight segments that break at interfaces, where the velocity of propagation varies discontinuously. The break in direction of the rays is governed by Snell's law of geometrical optics.

Let $\underset{\sim}{\nu}$ be the normal to the interface at the point of contact with the ray, and let ϕ^-, ϕ^+ be the angles that the incoming and outgoing ray segments form with $\underset{\sim}{\nu}$. Snell's law says that

$$v^+ \sin \phi^- = v^- \sin \phi^+ \qquad (1)$$

Thus the mathematical problem of calculating a ray joining a source-receiver pair, with a given signature, is reduced to finding the intersections of the crossings, so that Snell's law is satisfied. In the general case of curved interfaces, this will give a coupled system of nonlinear equations. A very clean version of the resulting system of equations has been given in H. B. Keller and D. J. Perozzi [6].

The first observation is that we can reduce the problem even further to just finding the offsets of the

intersections, since we can use the equations of the interfaces to calculate the depths.

Let $\mathbf{X} = (x_0, x_1, \ldots, x_{N+1})$ be the sequence of offsets defining the ray, where $x_0 = x_S$, $x_{N+1} = x_R$. According to the signature ICR(j), the corresponding depths will be

$$z_j = f_{ICR(j)}(x_j) \quad , \; j=1,\ldots,N \quad . \tag{2}$$

A tangent to interface ICR(j) at x_j is given by

$$\tau_j = (1, f'_{ICR(j)}(x_j))^T$$

and Snell's law (1), can be expressed by means of vector inner products as :

$$\psi_j(\mathbf{X}) = v_{IRG(j+1)} \langle \tau_j \, , \, r_j \rangle - v_{IRG(j)} \langle \tau_j \, , \, r_{j+1} \rangle = 0 \quad , \tag{3}$$

$$j=1,\ldots,N$$

where (r_j) $j=1,\ldots,N$ represent normalized ray directions; i.e. if

$$w_j = \begin{pmatrix} x_j \\ z_j \end{pmatrix} - \begin{pmatrix} x_{j-1} \\ z_{j-1} \end{pmatrix}$$

then $r_j = w_j / |w_j|$, where

$$|w_j|^2 = (x_j - x_{j-1})^2 + (z_j - z_{j-1})^2 \quad ,$$

and z_j is given by (2).

The travel time between source and receiver is easily obtained as

$$T_{SR} = \sum_{j=1}^{N+1} |r_j| / v_{IRG(j)}$$

The set (3) of N nonlinear simultaneous algebraic equations must be solved in order to find the unknown offsets $x_1,\ldots,x_N$. In vector form, (3) is represented as $\underset{\sim}{\psi}(\mathbf{X}) = \mathbf{0}$.

The method of choice for finding solutions to such systems is Newton's method. Given an initial guess $\mathbf{X}^0$, we compute corrections by means of the iteration: For i=0,... :

Solve the linear system of equations

$$A(\mathbf{X}^i)\ \delta\mathbf{X}^i = -\underset{\sim}{\psi}(\mathbf{X}^i) \tag{4}$$

where $A(\mathbf{X}) = \delta\underset{\sim}{\psi}/\delta\mathbf{X}$ is the Jacobian matrix of the system.

$$\text{Correct:}\quad \mathbf{X}^{i+1} = \mathbf{X}^i + \delta\mathbf{X}^i. \tag{5}$$

The matrix $A(\mathbf{X})$ is tridiagonal, since only three unknowns are coupled in each equation, and therefore the solution of the linear systems can be achieved in order N^2 operations.

Sometimes Newton's method has difficulties in converging; for instance, if the initial values do not belong to the domain of attraction of the desired solution. A way to prevent divergence in the early stages, or on difficult regions, is to take only a fraction of the Newton step $\delta\mathbf{X}^i$. The rationale for this strategy is that the Newton iteration is of descent for the functional $|\underset{\sim}{\psi}(\mathbf{X})|$, and therefore, if we take $t > 0$ sufficiently small

$$|\underset{\sim}{\psi}(\mathbf{X} + t\delta\mathbf{X})| < |\underset{\sim}{\psi}(\mathbf{X})|.$$

Initially, we would try to use the full Newton step, i.e. t=1, but if the norm of the residual increases with respect to the previous iteration, then t will be diminished. Since failure or difficult convergence may be an indication that no solution exists (i.e. shadow zone), we do not let our Newton search wander too much. That is also where our very effective restarting procedure comes into play.

A usual task is to calculate a whole family of rays with the same signature. Once a first ray has been successfully obtained, a natural continuation procedure is to use this ray to estimate a starting trajectory for the next one in the family. Euler continuation is proposed for this purpose in [2,6]. Using the difference of two consecutive previous rays works as well and it is considerably cheaper; thus we use Euler continuation only to start our second ray. Once this has been successfully calculated, we switch to the more economical procedure.

Either type of continuation will give erroneous starting guesses when we get close to a fold in the wave front, as in the buried focuses produced by synclines (see Fig. 3), and that will make Newton's method fail.

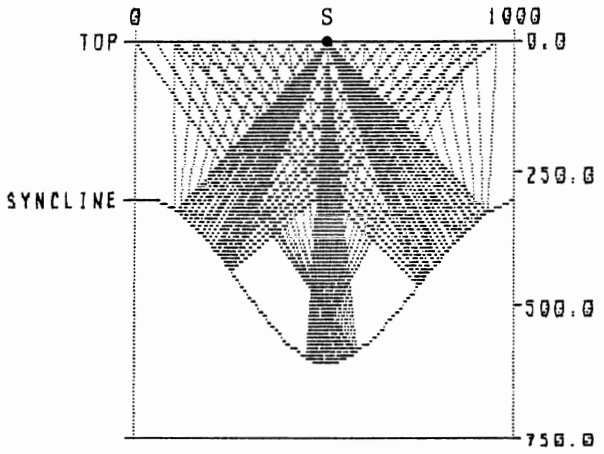

Figure 3. Triplications due to syncline.

Thus, in the presence of multiple solutions, as in the triple coverage shown above, simple continuation in receiver offset will fail to produce all the arrivals. The reason for this failure is intimately connected to continuation and the bifurcation of branches of solutions of nonlinear equations depending on parameters [5].

The problem we are considering can be stated more precisely as that of solving a family of nonlinear equations $\underset{\sim}{\psi}(\mathbf{X}, x_R) = 0$, parameterized by receiver offset.

If we plot the time $T(x_R)$ that the signal takes to travel from source to receiver, we soon realize that the familiar travel time curves (as function of offset) are also the bifurcation diagrams for this nonlinear problem (see Fig. 4, where the travel time curves are given as synthetic seismograms).

The bifurcations in this case correspond to the corners of the bowtie. A simple offset continuation procedure will follow one of the branches, namely, the one in which its initial ray lays, until it comes to the end of the branch at one of the turning points or goes off the window. Only by accident will the Euler continuation make it jump to another branch. Elaborated procedures for navigating through turning and bifurcation points

are available [5,12,13], but we have preferred to take a different tack, as explained below.

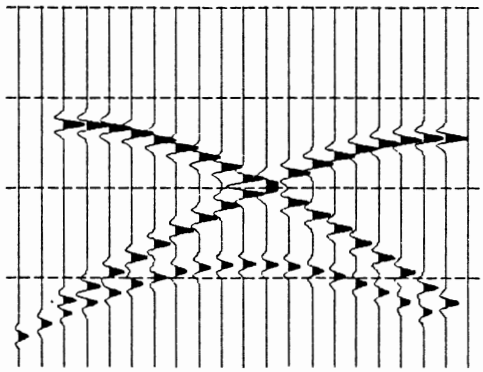

Figure 4. Travel time curves as synthetic seismograms.

What we mentioned above is just one type of possible failure of Newton's method. A robust algorithm that wants to compute as many arrivals of a given type as possible must have provisions for restarting, and also for avoiding the re-calculation of already computed rays. The way in which we have been able to solve all these problems in an unified way is to parameterize the search with respect to the initial ray angle ϕ_0. Clearly, there is a one-to-one mapping between these angles and the corresponding rays, since a given initial position and initial direction determines uniquely a ray of a given signature.

We do not give up receiver continuation, which is a useful procedure within continuous, slowly varying branches of solutions, but we supplement it with a careful bookkeeping of the initial angles searched. Start and restart after failure is provided via shooting and (fairly loose) trapping. To describe the procedure in more detail and to fix ideas, let us consider the case of a source on the free surface.

Let $[\phi_L, \phi_R] \subset [-\pi/2, \pi/2]$ be the interval of possible initial angles that will generate all the rays of the given signature, starting from S and reaching a family of evenly spaced receivers, also assumed to be on the free surface. Such an aperture can be estimated at the outset, or in the worst of cases the whole interval $[-\pi/2, \pi/2]$ can be considered.

This angle interval is then searched in a systematic fashion starting, say, from ϕ_L. The first ray is shot with this angle, and if it reaches the free surface within the receiver window, then an appropriate modified version is used to start the two-point scheme for a ray from the source to the closest receiver. If the two-pointing is successful, we record the computed initial angle, and any gaps in angle larger than a given threshold are also saved in a stack, as angle subintervals to be searched later for possible additional arrivals.

The goal of the algorithm is to make sure that within a given maximum resolution, i.e. a minimum angle gap $\Delta\phi$, the whole angle aperture is checked for possible ray solutions. Apparently this same objective could be achieved by shooting systematically $(\phi_R - \phi_L)/\Delta\phi$ rays, and picking those that land within a prescribed tolerance from a receiver. However, examination shows that the sensitivity of arrival offset with respect to initial angle can be made arbitrarily high, which will in turn require a very small $\Delta\phi$ in order for this alternative procedure to be successful. The main advantage of two-pointing and receiver continuation is that a much more economical procedure results, since the gap in ϕ corresponding to two consecutive receivers need not be checked for further arrivals, regardless of its size.

The algorithm is recursive, insofar as the stack of angle subintervals to be revised varies dynamically, and the process ends when this stack is empty, meaning that the whole initial search interval has been visited, within the given resolution. Whenever a multiple arrival is computed, its initial angle and travel time are compared with those of previous arrivals to the same station (and of the same family, in case multiple families are being considered simultaneously), and if a close match occurs, then the new arrival is declared a repetition and it is disregarded.

3. Basic Algorithm for Shooting Rays

The shooting of rays through layered structures, such as those considered earlier, still requires equation solving to obtain the intersection of ray segments with the curved interfaces. With the same notation as in Section 2, the problem to be considered here is to shoot a ray from a given source position S (x_0, y_0), with a given initial angle ϕ_0, through a curved layered structure, according to the signature given in the arrays ICR, IRG (see Fig. 5).

Let $S_i = \begin{pmatrix} x_i \\ z_i \end{pmatrix}$, i=1, ..., N+1, be the unknown intersections with the interfaces. The straight line segment that starts at S_i, with direction ϕ_i is parameterized as

$$L_i(\lambda) = S_i + \lambda P_i \ , \ \lambda > 0 \ ,$$

where

$$P_i = \begin{pmatrix} \sin\phi_i \\ \cos\phi_i \end{pmatrix}, \quad L_i(\lambda) = \begin{pmatrix} L_{ix}(\lambda) \\ L_{iz}(\lambda) \end{pmatrix} \ .$$

For each leg i, we want to solve the scalar nonlinear equation

$$\rho(\lambda) = L_{iz}(\lambda) - f_{ICR(i+1)}(L_{ix}(\lambda)) = 0 \qquad (6)$$

for $\hat{\lambda}_i$. Then $S_{i+1} = L_i(\hat{\lambda}_i)$.

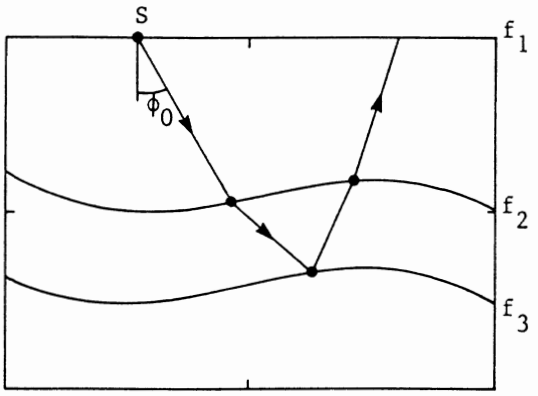

Figure 5. Shooting a ray.

We could use Newton's method to solve these scalar equations, but a more robust, and usually just as efficient, procedure is ZEROIN [3], a combination of bisection, the chord method and cubic extrapolation. We have enhanced the standard algorithm by adding a search to ensure that the interval in λ given to ZEROIN is such that the function $\rho(\lambda)$ changes sign at the end points, as required but not enforced by the original code. Another reason for preferring this algorithm instead of a Newton type one, is that in the way we have implemented it, no initial guesses are necessary, and except for very extreme geometries, an initial interval with a change of sign is rapidly found. After that

ZEROIN guarantees the return of a root with the desired accuracy.

Once an intersection S_{i+1} is found, Snell's law provides a mean to obtain the direction P_{i+1} of the next leg of the ray. In fact, we have that if

$$\tau_i = (1, f'_{ICR(i)} (L_{ix} (\hat{\lambda}_i)))/(1+f'^2_{ICR(i)})^{1/2}$$

and $v_i = v_{IREG(i)}$, then P_i and P_{i+1} must satisfy

$$\bar{v}_{i+1} < \tau_i, P_i > = \bar{v}_i < \tau_i, P_{i+1} >. \qquad (7)$$

Since τ_i, P_i, P_{i+1} are co-planar, we must have

$$P_i = P_{i+1} + \beta \tau_i$$

for some scalar β. Taking inner product with τ_i, we get

$$< \tau_i, P_i > = < \tau_i, P_{i+1} > + \beta ,$$

or

$$\beta = [< \tau_i, P_i > - < \tau_i, P_{i+1} >] ,$$

and replacing the value of $< \tau_i, P_i >$ from (7), we get

$$\beta = [\bar{v}_i/\bar{v}_{i+1} - 1] < \tau_i, P_{i+1} > .$$

Therefore, the new normalized direction P_{i+1} is:

$$P_{i+1} = (P_i - \beta \tau_i) / | P_i - \beta \tau_i | ,$$

and we can go back to (6), with i increased by 1. The algorithm terminates when P_{N+1} and the corresponding last intersection S_{N+1} are generated.

This type of shooting algorithm provides an automatic way of starting the two-point solver for the first time and also, through the use of our search stack, to restart it upon failure. This may occur in the neighborhood of a folding of the travel time curve, or when the rays get close to abrupt changes in the interface geometry or material properties.

4. Normal Incidence Ray-Tracing

Various types of processing are performed on measured seismic reflection data in order to improve its quality and simultaneously reduce the volume of data that has to be manipulated.

One of these processes consists of taking the signal corresponding to a coincident source-receiver pair, and adding to it those of nearby receivers, with appropriate time corrections. This process is referred to as stacking, and if performed and displayed for every source, it produces a normal incidence or stacked section.

The name normal incidence stems from the fact that the only way for a ray to go from a point on the free surface, bounce in a deep horizon (say #L), and return to the same point following the same path, is that the angle of incidence with horizon L be 90 degrees (see Fig. 6). Of course, with complex curved layers it is possible to have nonnormal incident rays with zero-offset, but those will only be considered as special cases of our non-zero offset calculations.

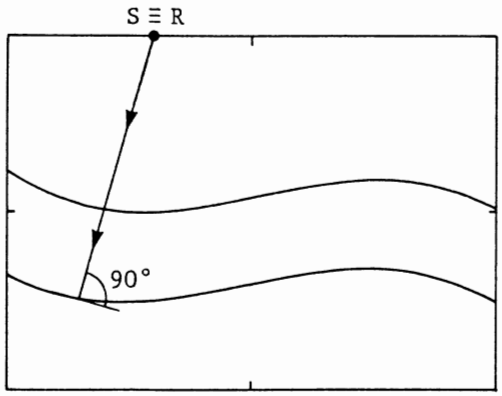

Figure 6. Normal incidence ray-tracing.

The algorithms for both two-point and shooting normally incident rays are just small modifications of the ones for the general problem. Of course, since the path to the deep horizon coincides with the return path, only half of the work needs to be done. For two-pointing, equations (3) are set for all the interfaces up to the L-1 and, as the last equation we add the normal incidence condition

$$< \underset{\sim}{\tau}_L , \mathbf{r}_L > = 0 \quad . \tag{8}$$

Travel time will be recorded as the two-way travel time.

The type of shooting that we will be interested in doing requires that the ray start normally from an arbitrary point on the deep reflector, and travel to

the free surface, say, with the given signature. This type of ray will provide good starting guesses for the normal incidence two-point calculation.

Also this points out to the correct parameterization to ensure complete coverage of all possible arrivals. Namely, we will use offset along the deep reflector as the search parameter, just as we used initial ray angle at the source for non-zero offset bookkeeping. Clearly, provided that the deep reflector is a one-to-one function of x, this will give a one-to-one correspondence between the parameter space and all the possible normal incidence arrivals. Therefore, it is an adequate parameterization for continuation through travel time curve foldings, as opposite to simple receiver offset.

Again, the normal incidence shooting algorithm is similar to the one we described in Section 3 for the non-zero offset case. The only two differences are:

a) The ray is traced from a point (x_o, y_o) on the deep horizon L to the free surface (say).
b) The initial angle is predetermined as

$$\phi_o = \cos^{-1} (\underset{\sim}{\tau}_L) \quad ,$$

where $\underset{\sim}{\tau}_L$ is the normalized tangent vector to interface L at x_o.

Just as in the case of non-zero offset, this procedure does not require starting guesses, and therefore is ideally suited to provide starting and restarting rays for the two-point process. A stack of unsearched subintervals of deep reflector offsets is kept, and the search ends only when this stack is empty. The only additional care we take is to scale the search step, to take into account the interface slope. If $z=f(x)$ is the equation for the interface, we scale the basic search step Δx, that measures the attainable resolution, as

$$\tilde{\Delta}x = \Delta x \; / \; \mathrm{sqrt}(1 + (f')^2) \quad .$$

So, the steeper the interface the smaller will be the actual search step. This amounts to an arc-length based equidistribution of the density of points that will be examined.

5. Pinchouts and Other Nonconformities

As promised, in this section we explain how to handle pinchouts, and some apparently multivalued interfaces, like those necessary to model reverse faults. A

pinchout occurs when a layer evanesces within our geological window, as shown in Fig. 7.

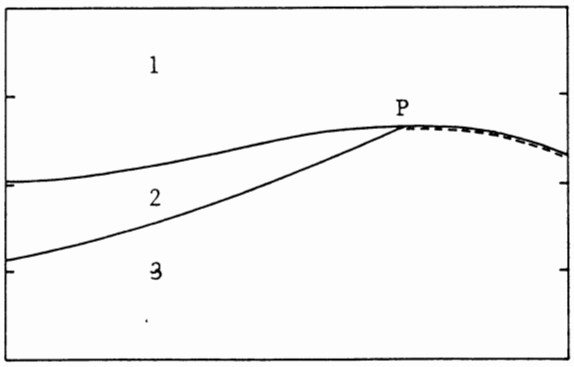

Figure 7. A pinched-out layer.

Curved, or even straight, but dipping interfaces imply lateral variations of velocity, which cause problems in many of the standard geophysical data processes. Pinchouts provide even more extreme examples of those types of difficulties, and additionally, pinchout tips produce diffracted arrivals that complicate the picture even further.

To be able to include this and other nonconforming situations, we follow and idea of Cerveny et al. [1], where the introduction of fictitious thin layer segments is suggested. Thus, we would convert the truncated layer 2 in Fig. 7 into a complete layer, by extending interface 3 parallel to I_2, from the pinchout tip P to the right end of the model window. In doing this, we actually create a thin layer of non-zero width, which will essentially be transparent to all the ray-tracing algorithms (and the user). We also keep enough information about the original, unmodified model, so as to know if an interface used to be shorter, and thus effect the correct processing in a number of situations that require special care. Thus, for all practical purposes, we are back in the full length case. Multiple pinchouts are treated in a similar form, and the actual extension of the layers is done automatically in a preprocessing step.

A more challenging situation arises when we want to make a model of a reverse fault, as shown in Fig. 8. A somewhat artificial solution to the problem of converting the model of Fig. 8 to a complete layer model is indicated in Fig. 9. We have numbered interfaces and

regions in the usual way, used (P_i, Q_i) to indicate the end points of the original segments, and R_2, R_5 to indicate the end points of the added segments.

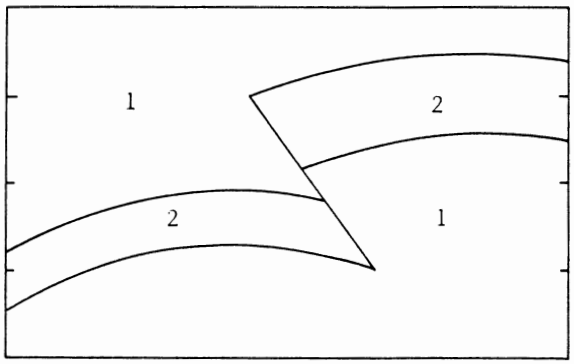

Figure 8. Reverse fault.

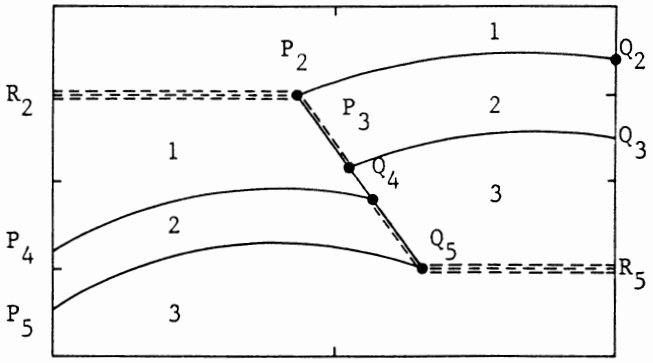

Figure 9. Layered form of reverse fault.

In short, we complete interface 2 $[P_2, Q_2]$ to the left, with the straight line segment $[R_2, P_2]$, and we take as new interfaces

$$I_3 = [R_2, P_2, P_3, Q_3];$$

$$I_4 = [P_3, Q_4, Q_5, R_5];$$

$$I_5 = [P_4, Q_4, Q_5, R_5];$$

$$I_6 = [P_5, Q_5, R_5].$$

Of course, in deeper interfaces, parallel segments get displaced by a small amount. The addition of segments $[R_2, P_2]$, $[Q_5, R_5]$, and the judicious input of the actual model segments in the order and form indicated above, will make it look like a collection of pinchouts that can then be converted automatically to a regular layered media as before.

6. Amplitude Calculation

Together with ray paths and travel times, we can also obtain from geometrical ray theory, amplitudes, polarities and phases, which are good high frequency approximations. This amplitude, considered as a peak amplitude, is used to scale a given wavelet, chosen by the user, and placed at the time of arrival in order to generate so-called synthetic seismograms (see Fig. 10). These are time histories recorded at the various stations that try to mimick the way in which the actual collected data is presented in the form of time sections. They provide a visual, qualitative way of matching computed and recorded time histories, thus proving or disproving the adequacy of our model.

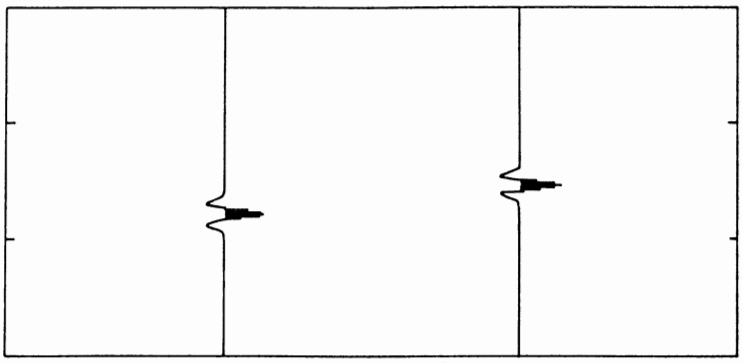

Figure 10. Part of a synthetic seismogram showing one arrival at two stations.

Once a ray connecting a source and a receiver has been obtained, we compute two different contributions to peak amplitudes: geometrical spreading and reflection-transmission coefficients. The first contribution quantifies the natural spread of energy with distance, and also the geometrical focusing and de-focusing of ray beams due to the curved interfaces. A good formulation of the geometrical spreading calculation for our type of models can be found in Section 3.6 of Cerveny et al. [1]. The second contribution comes from the partition of energy at interfaces. When a ray intersects a material discontinuity, four other rays can be generated: one

reflected and one transmitted of the same type, and also reflected and transmitted converted rays (P to S or vice versa). This partition of energy depends upon the angle of incidence of the ray and the values of velocities and densities on each side of the interface. For the given ray signature, a reflection-transmission coefficient and a phase are generated according to the general formulation of Zoeppritz, as implemented by Young and Braile [11].

The product of the two contributions, the phase and the arrival time are then used to generate a wavelet in the seismogram corresponding to the receiver. Multiple arrivals to the same receiver will appear as additional wavelets, which may interfere with each other to give more complicated wave forms if the arrival times are close together.

7. Diffractions

Interface discontinuities produce a different kind of phenomenon called diffraction. When a wave front intersects an interface at a discontinuity, like a pinchout corner or a break in direction, a diffracted wave front is produced. Because of the discontinuity, standard Snell's law is not valid at such points, and the type of rays we have discussed so far cannot be continued. In 1958, J. B. Keller [8] published a theory that permits to extend the use of rays to this more general situation.

The first point to make is that a ray impinging on a corner or point discontinuity reflects energy in all directions (see Fig. 11). Amplitudes in this case are calculated by considering local, exact solutions to the wave equation for given geometries.

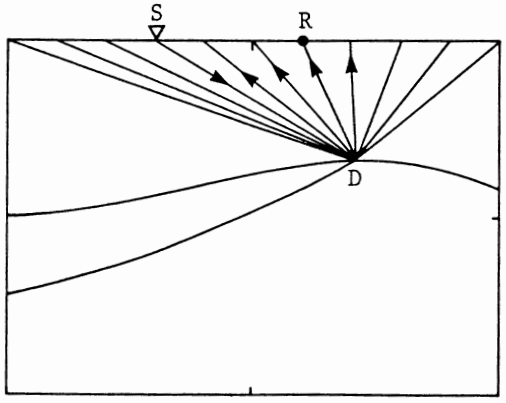

Figure 11. Diffracted ray field.

The data for computing a diffracted ray-path is, as usual, the source and receiver positions, the ray signature and additionally the diffractor position. If the source and receiver are coincident (zero offset), then the problem is a two-point one, with one end point at the source receiver location, and the other at the diffractor. In this case there is nothing new.

Amplitudes are computed in the usual way except at the diffractor. For the diffracted amplitude field at pinchout corners, we have implemented the exact solution for diffraction by a wedge given by J. Keller and Blank [7]. Of course, the wedges considered there are made of straight lines, so we use the interface tangents at the diffractor as an approximation.

In Fig. 12 we show a synthetic seismogram corresponding to the diffractor of Fig. 11.

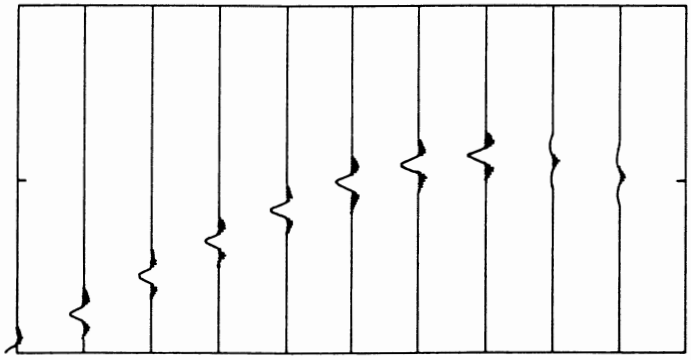

Figure 12. Synthetic seismograms for diffractors.

For the non-zero offset diffractions we solve two independent two-point problems, one for the downgoing [S,D], and one for the upgoing path [D,R], respectively.

8. Time-to-Depth Migration of Normal Incidence Data

A first step in achieving one of the main goals of inverse modeling of the Earth subsurface is to convert time section data into depth section images. Assuming that the material properties are known, we can describe this process in our terminology as:

> Given a time versus offset curve corresponding to the reflections from a deep horizon, find the position of the horizon, i.e. produce a depth versus offset image.

This process is called time-to-depth migration.

Travel time curves are picked from recorded sections collected in the field. This is a primary task of the interpreter of seismic reflection data. If the raw data is utilized, then the arrivals will correspond to non-zero offset reflections. Usually, raw data is too noisy for direct interpretation. In order to improve its quality, data processing is performed, producing a stacked section. This type of section simulates a set of normal incidence data.

Unfortunately, stacking often assumes physical conditions that are not valid, like horizontal stratification. Thus, in the presence of lateral variations, as those produced by dipping or generally curved interfaces, stacking introduces errors (distorting the information contained in the data) and then migration, after stacking does not produce correct results [4,9].

Naturally, normal incidence sections can be obtained from the original data by just taking the arrivals to the geophones that coincide with the source, or stacking can be limited to only a few nearby receivers. In any case, the inverse modeling of such sections, even if it produces incorrect results, may serve to point defects in the processing, which can in turn be corrected.

The depth migration that we will describe assumes the knowledge of all the velocities and horizons above the desired reflector, and also that a time curve associated with the reflector can be picked from the data. Let this time curve be $T(x_0)$, where x_0 is receiver offset and, for the time being, let us assume that it is single valued and differentiable. Multiple valued, or discontinuous time curves will be handled in terms of their single valued smooth segments.

If $P(s) = (x(s), z(s))$ represents the ray path, then

$$d\,T(x_0)/dx_0 = \sin \phi_0 / v(x_0) \,, \tag{9}$$

where ϕ_0 is the ray angle at x_0 measured with respect to the z axes. Therefore,

$$\phi_0 = \sin^{-1} [v(x_0)\, d\,T(x_0)/dx_0] \,, \tag{10}$$

and from the measured time curve, we can infer the value of the initial angle ϕ_0. Thus all we need to do is shoot a ray, with initial angle ϕ_0 through the structure, and stop when the time $T(x_0)/2$ has elapsed. At that moment we will have hit the reflector and, if we

consider the normal to the ray, then we have both a point and the tangent to the interface.

Of course, real data contains errors and time curves will only be available at a finite number of points, the actual receiver locations. Thus, both $d\,T(x_o)/dx_o$ and $T(x_o)$ itself will only be known approximately. For instance, $d\,T(x_o)/dx_o$ can be approximated by differences, or a spline can be passed through the discrete data set and the derivative taken as that of the spline. In either case an interpolation error will occur, which will be more or less severe, depending upon the form of the travel time curve. Thus, travel time curves with rapidly varying curvature will tend to produce larger errors than slowly varying ones.

A different question is how errors in the initial angle propagate through the structure or, in other words, how stable is the position of the end point of the shot ray with respect to variations in the initial angle. Actually, the geometrical spreading is a measure of that sensibility, since it quantifies how much an infinitesimal ray tube will spread when propagated through the structure. Thus we can use geometrical spreading to weight the information produced by our computed rays. These weights can be used **a posteriori** in the least-squares generation of an analytic representation of the interface, say by cubic splines or other appropriate parameterization.

Notice that diffracted arrivals should be properly collapsed to a point, this being one of the standard measures of a migrated section's quality. In Figs.13 and 14 we show synthetic data and its recovery via the above procedure.

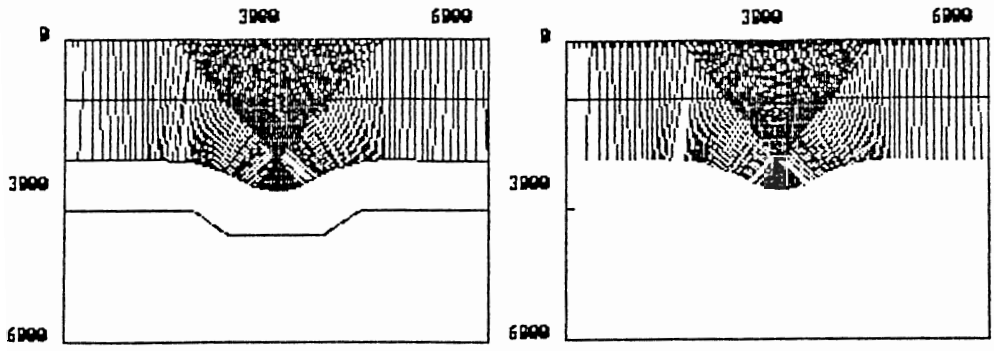

Figure 13. Synthetic migrated data.

The process will work its way from the free surface down, and therefore horizons will be migrated in succession. If a good velocity model is not known, then additional procedures (not considered in this paper) can be invoked to improve velocities and other material properties, simultaneously or alternating with the migration process.

Finally, in Fig. 15 we show a more ellaborated model, in which some overshooting is apparent when recovering a highly curved piece of horizon.

9. Migration of Non-Zero Offset Data

As we described before, non-zero offset data corresponds to noncoincident source-receiver pairs. Outside of this, the problem is the same as for normal incidence data, although its solution is somewhat more complicated.

Again, from the picked travel time curve $T(x_o)$ we can calculate the ray angle ϕ_o upon arrival to the station located at $(x_o,0)$. However, shooting backward from $(x_o,0)$ with angle ϕ_o beyond the last known interface will only provide half of the desired path, since the path from the deep reflector to the source will now be different. In any case, shooting backward from the receiver will produce a unique direction ψ_R from interface L-1 downwards (see Fig. 16).

At this point we do not know where the break point P, where the ray should bounce back from the unknown horizon, is located. What we do know is the total travel time that is available for the trip from S to R.

Our solution to the problem consists of shooting with initial angle ϕ_S from S; after traveling through the known structure and traversing horizon L-1, another direction ψ_S will be produced. It is then a simple matter to calculate the intersection of the two directions ψ_R, ψ_S: if this intersection occurs in the half space below horizon, L-1, then it will be an acceptable candidate for P.

Obviously, we can also compute the time $T^c(x_o; \phi_S)$ necessary to travel from S to R following this path:

if $T^c(x_o; \phi_S) = T(x_o)$ then we are finished.

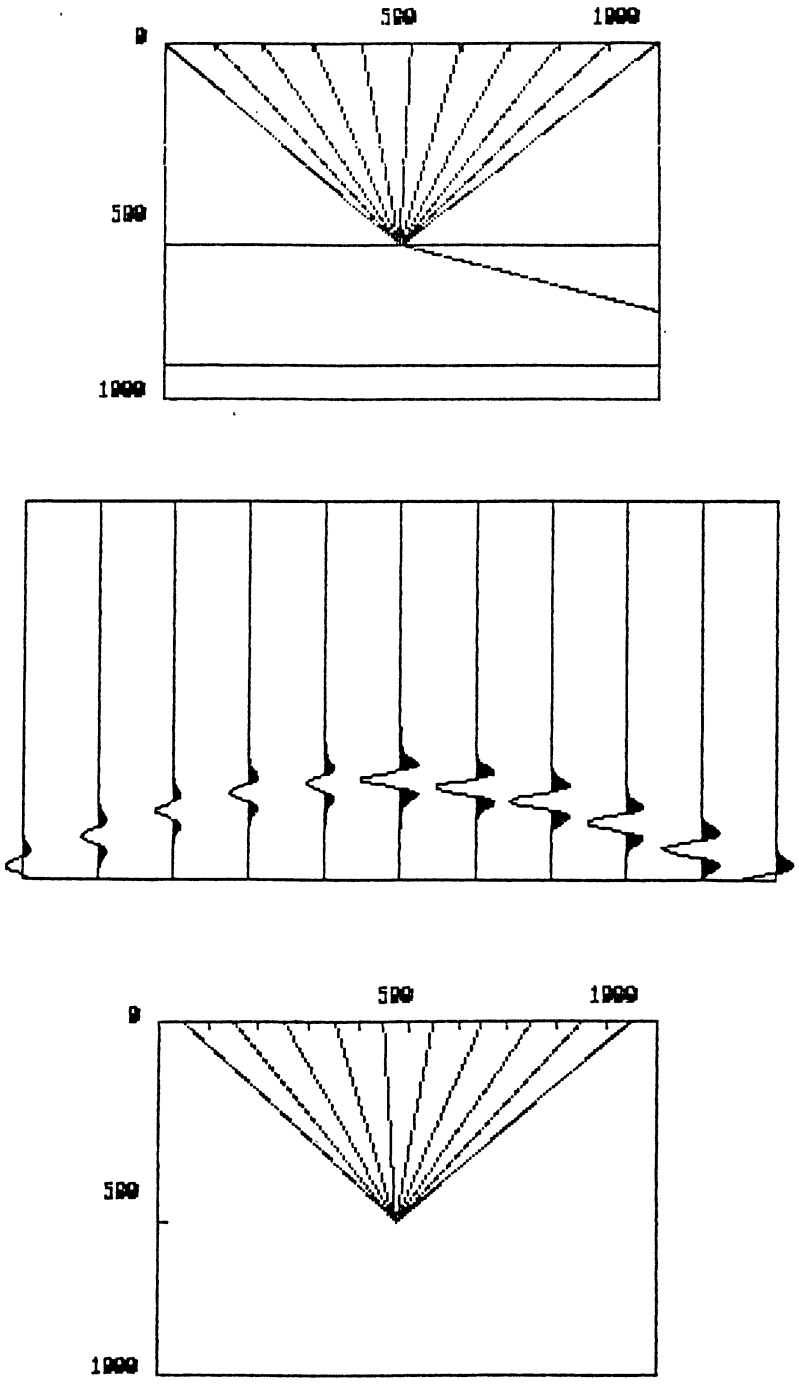

Figure 14. Synthetic and migrated diffracted data.

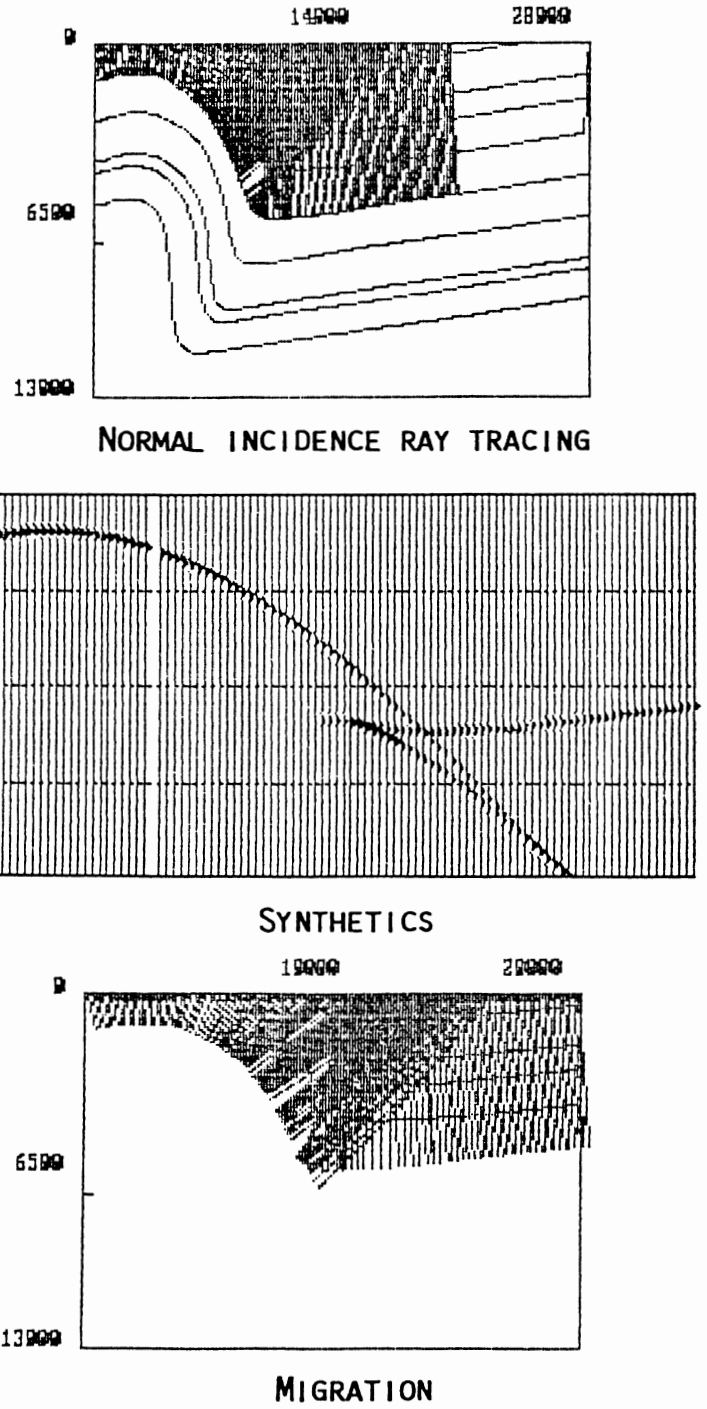

Figure 15. Migration of normal incidence data.

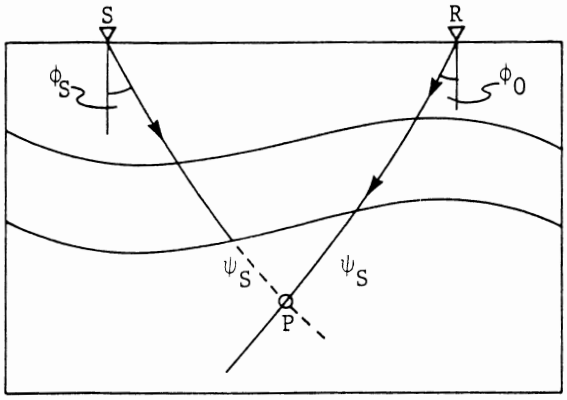

Figure 16. Schematic migration procedure.

Thus our problem has been reduced to solving the scalar equation

$$T^c(x_o;\ \phi_s) - T(x_o) = 0\ , \qquad (11)$$

as a function of ϕ_S, which can be acomplished with a further judicious use of ZEROIN.

In Figs. 17 and 18 we show an example of the application of this procedure using three different sources and a set of 60 receivers for the same model and interface of Fig. 13, where normal incidence was used. We see some discrepancies, but we must also recall that with non-zero offset data there will be considerable redundancy.

We would like to point out here that we could as well have set up a "two-point problem" to migrate this data. In fact, keeping the first part, i.e. shooting from the receiver, we can formulate a two-point problem from the source to the unknown point P, using as the last equation the travel time matching (11).

Unfortunately, (11) will couple all the points, apparently losing the main appeal of the two-point formulation, namely the fact that the resulting system of equations is tridiagonal. However, if we consider the matrix of our linearized system in block form

$$\begin{pmatrix} T & \underset{\sim}{u} \\ \underset{\sim}{v} & \alpha \end{pmatrix}$$

we see that we can solve the system with two bidiagonal back-solves and a few additional operations.

We have chosen the shooting approach for expediency, since it is simpler and the scalar equation solver is very robust but, of course, the curse of ill-conditioning looms in the background.

10. Comments on Microcomputer Implementation

The description of the system contained in Sections 2 through 9 makes very little reference to the computer environment in which it was developed. This in itself tells some of the story, since the system could as well have been developed on a minicomputer or a mainframe, and only a few years back that would have been the only choice available.

What was different? What influence did the microcomputer environment have in the development and in the actual performance of the system? What characteristics are especially valuable in this environment? In this section we will try to answer these questions based on our current experience.

Since we were adding to an existing system, we had initial constraints that may not be present in other projects. Besides the obvious constraint of communication between the existing and new modules, the main other issue was the fact that GMS was written in BASIC. When this project was initiated there were no other good quality compilers available, that would have allowed us to do as much as we did with BASIC. As with everything else, that has changed considerably in a short period of time. As a consequence, the whole system is now being translated into C. BASIC has obvious disadvantages for developing large systems, mainly its lack of real subroutines and the global nature of its variables. However, on a PC it has also a number of advantages, noticeably: the graphic and string instructions, the interpreted mode, and the possibility to access extended memory.

The principal concept that comes across in evaluating the system is that it constitutes an economical, self-contained, personal work station for nontrivial geophysical modeling that can be used in a production mode for rapid, interactive work, or for training future interpreters and wave analyzers.

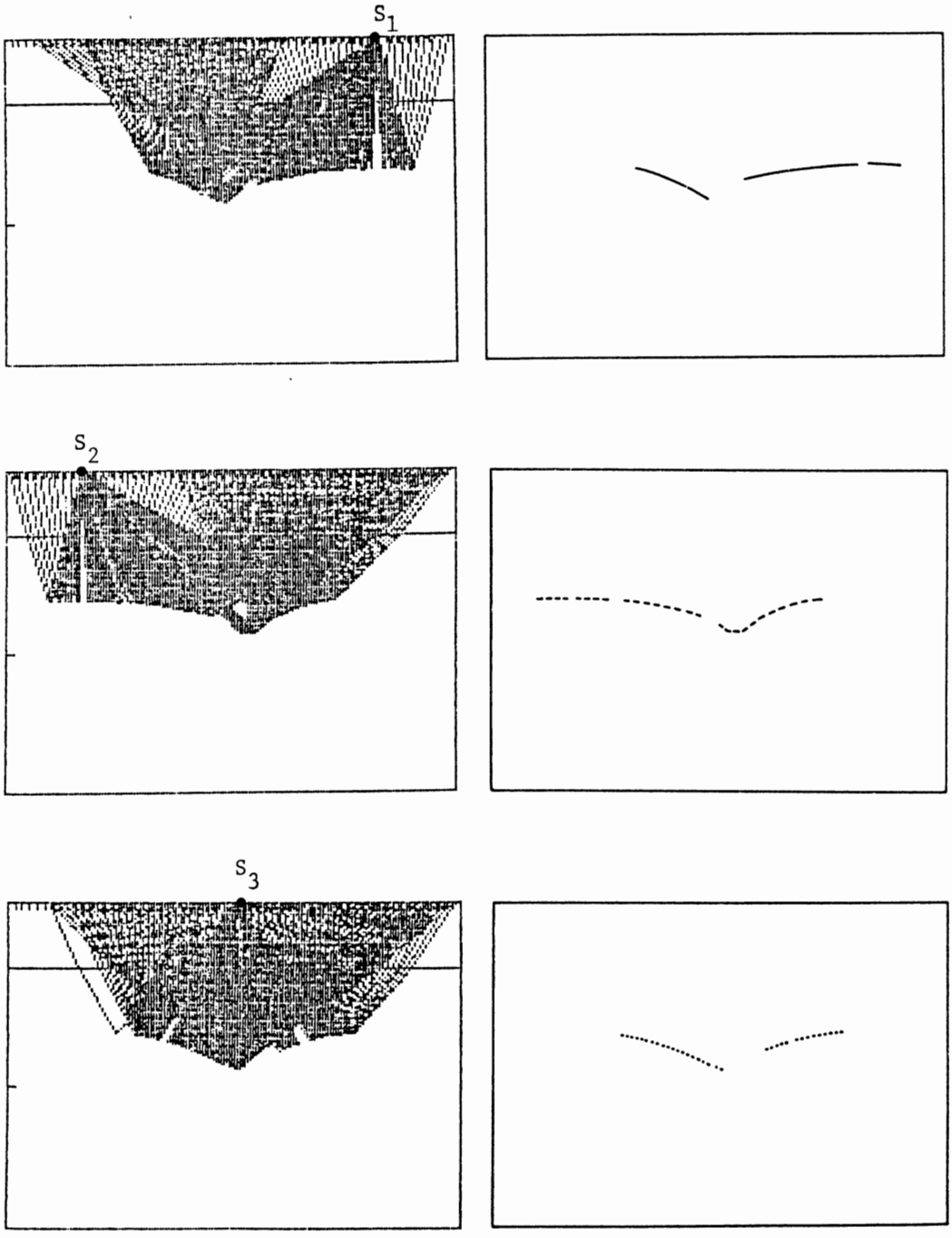

Figure 17. Recovery of a synclinal from non-zero offset data (3 sources).

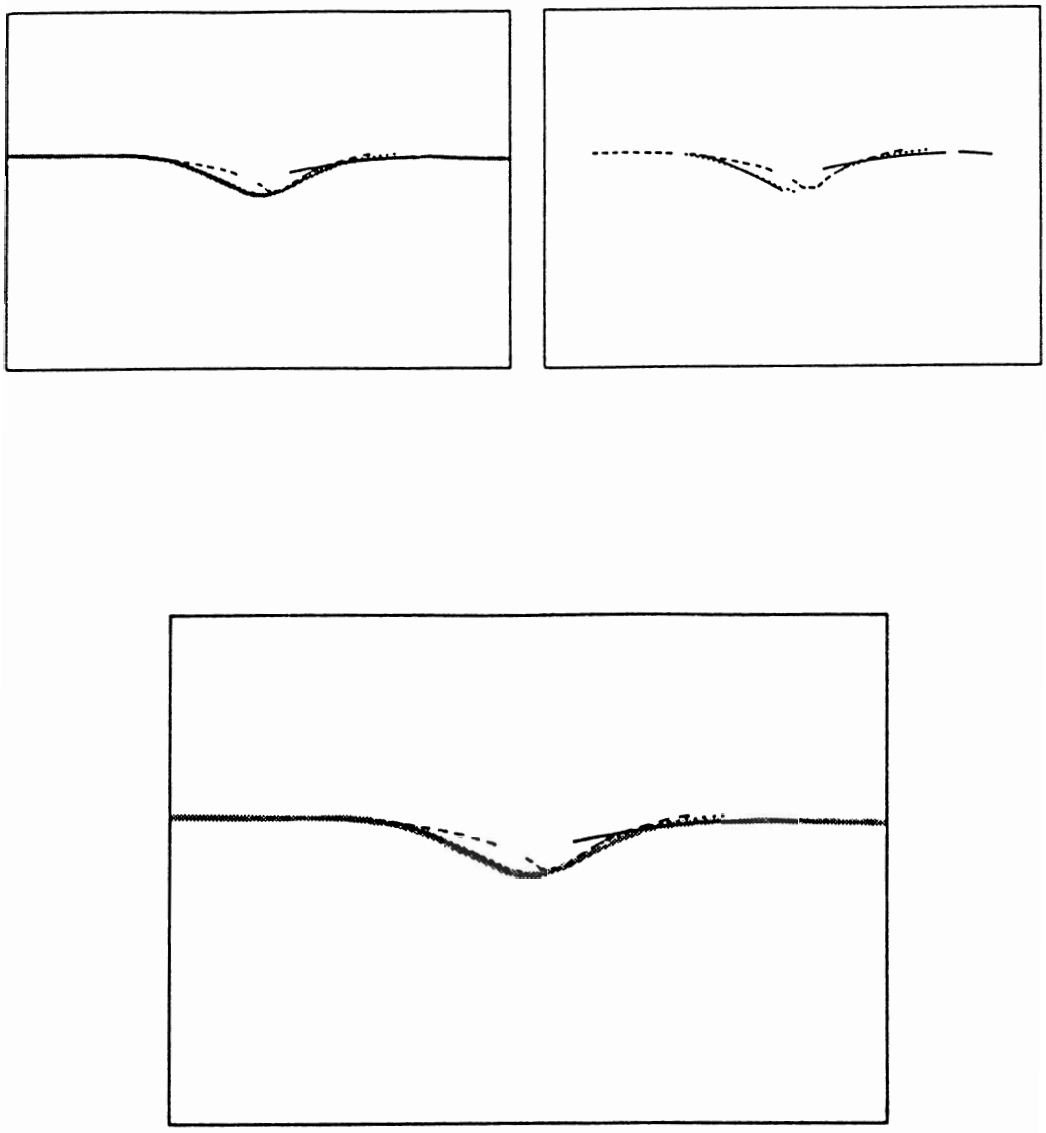

Figure 18. Comparison of exact and recovered horizon.

When the basic configuration (say an IBM-AT) is enhanced with a digitizing tablet, good quality printer, maybe a high resolution color monitor for graphics, a

separate monitor for interactive input-output, and a floating-point co-processor, it puts at your finger tips a fairly powerful and versatile machine that runs at constant speed.

Obviously, there will always be tasks that go beyond the capabilities of a PC, but then a communication package is all one needs to connect it to the world. For instance, we have developed a three-dimensional ray tracing code for complex media that would not (as yet) transport gracefully to a PC. However, we plan to do a good deal of our interactive 3-D modeling on a PC, and then shift files back and forth, and initiate tasks from it in our mainframe.c

It is important to point out that this is all standard equipment, both hard and soft, that can be purchased from your neighborhood computer store, and therefore is available to the public at large. With more specialized, expensive, and/or experimental equipment, much more can be done right now. But, it will still take some time for these devices to become generally available.

Acknowledgment: This work benefitted from many conversations with my colleague, G. Wojcik. His help and early work in the calculation of diffracted amplitudes is especially appreciated.

References

[1] V. CERVENY, I. A. MOLOTKOV and I. PSENCIK (1977), Ray Method in Seismology, Univ. Karlova, Praha.

[2] P. DOCHERTY (1985), "A fast ray tracing routine for laterally inhomogeneous media," Manuscript.

[3] G. FORSYTHE, M. MALCOLM and C. MOLER (1977), Computer Methods for Mathematical Computations, Prentice Hall, Englewood Cliff, NJ.

[4] L. HATTON, K. L. LARNER and B. S. GIBSON (1981), "Migration of seismic data from inhomogeneous media," Geophysics 46.

[5] H. B. KELLER (1978), "Global homotopies and Newton methods," Recent Advances in Numerical Analysis (Ed. C. de Boor and G. H. Golub), Academic Press, NY.

[6] H. B. KELLER and D. J. PEROZZI (1983), "Fast seismic ray tracing," SIAM J. Appl. Math. 43, pp. 981-992.

[7] J. B. KELLER and A. A. BLANK (1951), "Diffraction and reflection of pulses by wedges and corners," Comm. Pure Appl. Math. 4, pp. 75-94.

[8] J. B. KELLER (1958), "A geometric theory of diffraction," Calculus of Variations and its Applications, Mc Graw Hill, NY, pp. 27-52.

[9] K. L. LARNER, L. HATTON, B. S. GIBSON and I-C. HSU (1981), "Depth migration of imaged time sections," Geophysics 46, pp. 734-750.

[10] B. T. MAY and J. D. COVEY (1981), "An inverse ray method for computing geologic structures from seismic reflections--Zero offset case," Geophysics 46, pp. 268-287.

[11] G. B. YOUNG and L. W. BRAILE (1976), "A computer program for the application of Zoeppritz's amplitude equations and Knott's energy equations," Bull. Seism. Soc. America 66, pp. 1881-1885.

[12] E. J. DOEDEL and J. P. KERNEVEZ (1985), "Software for continuation problems in ODE's with applications," to appear in SIAM J. Sci. & Stat. Comp.

[13] R. SEYDEL (1983), "BIFPACK--A program package for calculating bifurcations," Buffalo, NY.