DIRECT AND INVERSE SEISMIC MODELING THREE–DIMENSIONAL COMPLEX GEOLOGY

VICTOR PEREYRA

(620 *Hansen Way, Suite* 100 *Palo Alto, California*, 94304 *USA*)

Abstract

We give a brief account of recent results from a project to develop efficient algorithms and practical computer implementations for modeling complex, 3–D geological regions, with applications to exploration and general seismology. The problem is divided into geometrical and material description and visualization, forward modeling with ray tracing and finite element elastic wave propagation, and finally, least squares inversion of travel time data.

1. Introduction

The construction of mathematical models of the Earth interior has been one of the principal activities of seismologists for many years. As measuring and computing tools have evolved, more ambitious goals have been set and achieved.

Currently, we are starting to perform interactive modeling in powerful workstations coupled with supercomputers. The aim is towards increasing accuracy, resolution, detail and visualization capabilities, within a flexible human/computer interface.

In this paper we discuss some of the mathematical and computational issues involved in developing such an interactive modeling environment. We shall concentrate on modeling Earth's elasticproperties by seismic methods, without loosing sight of the many other physical properties that are currently measured and that can therefore be modeled, contributing to a better determination of the overall properties of a given geological region. In fact, cooperative modeling of different data sets (magnetic, gravimetric, seismic, well–logs) is starting to be considered [4], and it most likely will become an area of strong future development, once each individual approach is mastered and more computer power becomes available. At that point, tools of artificial intelligence may be necessary to manage the large knowledge data bases that will result, and also to aid in collective reasoning with inferences from multiple sources [1, 5].

We first consider issues of geometrical and material model description. This will be basic for the direct and inverse modeling techniques to be discussed later.

Seismic ray tracing and finite element solutions of the elastic wave equation will be our forward modeling techniques, while inverse modeling is divided between static and dynamic modeling. In the static case, a variation of forward modeling is used only once to determine a preliminary model from measured data. This preliminary model, coupled with any additional information is used to initialize an optimization iteration that attempts to find the best fit to a parameterized model, consistent with the measured data. In this dynamic inversion loop, forward modeling is used at each step, and therefore large computational resources may be required.

2. Geometrical and Material Modeling

We consider the task of modeling a bounded volume V of the Earth's interior. For resource exploration, V will usually be a parallelepiped in Cartesian coordinates, while for other applications defferent coordinate systems and volumes maybe more appropriate.

We are particularly interested in modeling complex regions, and consequently our geometrical and material models must have sufficient generality to allow the representation and manipulation of such regions. The type of models we consider are of generalized blocky type, i.e., the volume V is subdivided in subregions $V = \bigcup_i R_i$ of smoothly varying material. Each subregion R_i is limited by curved surface patches P_{ij}, that separate abrupt (i.e.,discontinuous) changes of material. These patches in turn, are limited by spatial curved segments C_{ijk}, with end points $[v_{ijkb}, v_{ijke}]$. Adjacent patches are connected by having a boundary segment in common. Higher smoothness can be enforced if desired.

In this way, the geometry of complex material interfaces can be described starting from simple elements, by adding sufficient connectivity information. Looking ahead to the use of these models in ray tracing we require that the patches be at least twice differentiable, concentrating the discontinuities at boundary curves. In order to avoid artificial constraints, we represent the patch surfaces and the curved boundaries in parametric form, although we require that the patches be univalued with respect to at least one of the coordinate planes. This facilitates passing from the parametric to an explicit Cartesian representation, and it is sufficiently general to include all types of interesting geological interfaces.

Patch surfaces can be defined by formula from a data base of primitives, or they can be blended from their curved boundaries by transfinite interpolation [6], or they can be fit to given data. In these later cases, quadrilateral patches and parametric cubic splines boundary curves are natural basic elements. They accommodate blueprint engineering views, contour line maps, and other standard ways of describing 3–D objects. If we restrict the boundary curves to lie in planes, they can be easily input, edited, and viewed interactively on a workstation. Unfortunately, planar curves will not be general enough to describe all regions of interest, since the intersection of two arbitrary surfaces need not be planar.

An interactive system, GEOBLD [6], implementing these ideas has been written, and Figs. 1 through 4 show some models generated with it.

Material properties of interest in isotropic elastic wave propagation are the velocity of pressure and shear waves and the density. Alternatively one can consider the Lame parameters. Attenuation, anisotropy, and other more complex phenomena are starting to be modeled, but they will not be considered in this paper.

Homogeneous blocks are characterized then by three quantities: v_p, v_s, ξ; while for inhomogeneous materials these quantities will change from point to point. We will assume that this change is smooth, given by functions at least twice differentiable. Again, discontinuities will be limited to the interfaces. We see then that modeling these quantities in the inhomogeneous case is a problem akin to that of modeling the interfaces, but one dimension higher.

3. Forward Wave Propagation Modeling

3.1. Ray Tracing

We have reported extensively on ray tracing elsewhere [3, 8–14, 16]. Let us just say here that two-point ray tracing is an economical way of obtaining some of the quantities that are important in many wave propagation problems. Given a source and a receiver array, ray tracing aims to calculate all the relevant trajectories that the source energy may follow to arrive to the receivers. Relevant here means that enough energy is transported,that the trajectories remain in the volume being modeled, and that the travel time falls within a window of interest. Thus, usually direct arrivals and simple reflections from the interfaces need only be considered.

Three-dimensional two-point ray tracing in a piecewise smooth heterogeneous medium involves the solution of a system of seven nonlinear ordinary differential equations, subject to multipoint boundary conditions (for each ray between a source-receiver pair). This can be a fairly intensive computational task, specially if thousands of rays are required, and great care has been put to produce efficient and robust software for it.

The use of automatic initialization, re-start and search procedures, with receiver to receiver coherence, provides a versatile and robust algorithm, capable of tracing rays in a complex, piecewise smooth inhomogeneous medium, as shown in Figs. 5 through 12. We calculate not only the trajectories, but also the travel times and amplitudes, as given by geometrical spreading and full Zoeppritz reflection/transmission coefficients.

The current version runs on a variety of computer architectures, and it can take advantage of vector processing capabilities, as well as fine and coarse grain paralellization [13, 16]. In fact, if we call task T_i that of ray tracing from a source to a receiver array after bouncing from interface i, then task T_i is completely independent of task T_j, as long as $i \neq j$, and therefroe they can be performed by different processors. Observe that no communication is necessary, except to initiate the task and to collect the results. In a standard 3–D survey, many different shots are set.

Again, ray tracing from different sources are independent tasks, so we see that there is an inherently large amount of coarse grain parallelism involved here, that can easily take advantage of loosely connected networks of computers (processors, workstations...).

3.2. Finite Elements

The elastic (vector) wave equation:

$$\xi \ddot{\underset{\sim}{u}} = \underset{\sim}{f} + (\lambda + 2\mu)\nabla(\nabla \cdot \underset{\sim}{u}) - \mu\nabla \times (\nabla \times \underset{\sim}{u})$$

with body force $\underset{\sim}{f}$, and given initial and boundary conditions, can be solved numerically in complex 2–D and 3–D media by using a finite element discretization in space, and an explicit finite difference discretization in time. Absorbing boundary conditions should be used at artificial boundaries in order to avoid spurious wave reflections. By careful subgridding according to the material properties it is possible to preserve stability and therefore take advantage of the simplicity and high vectorizability of explicit schemes [15].

A most convenient arrangement is one in which pre-processing, prototype testing and post-processing is done on a workstation, while large scale computing is done on a supercomputer with a fast communication link to the workstation.

Currently, we can generate models with GEOBLD that are understood both by the ray tracing and the finite element systems. This is important because it allows these quite different modeling techniques to interact, so they can be used where they can do the most good. For instance, time sections can be generated with the finite element solver, simulating real data on a known model, which in turn can be used by ray tracing based inversion schemes to try to recover the model in a controlled fashion. This serves to debug, test and validate the techniques in a way that would not be possible with real data, where the answer is truly nuknown!

In another application, detailed wave propagation by finite elements can be done in the near field (say to model the source), and then ray tracing can be used to propagate the energy to the far field; in this form large scale three dimensional problems can be attacked with current hardware.

4. Static Inversion

The data collected in a reflection survey consists of a large number of digitized seismograms. Many shots are set at different locations, and the reflections from buried structures are received at a geophone array. A limited time window is recorded, and the objective of the exploration interpreter is to convert pictures of this data into useful insights about the local geology.

After some preliminary processing, the interpreter starts locating interesting structures. These are often characterized by strong, coherent signals, that form a surface in (x, y, t) space, where (x, y) is the location of the sensor and t is time.

An approximate time to depth map can be effected in order to obtain a truly spatial description of the geology. A variant of the ray tracing procedure can be used in a layer by layer manner, starting from the free surface. In this way the structure is assumed known up to the current unknow reflector. A step by step algorithm has been devised to calculate the spatial position of a reflector from travel time data, and then to estimate the material properties below it from amplitude data [12, 13]. The accuracy of this method deteriorates as we proceed to deeper layers, but its main purpose is to provide an initial model to be later refined by more comprehensive methods.

Let us assume for simplicity, that the data collected on the surface consists of travel times $T(x,y)$ and arrival directions $\underset{\sim}{u}(x,y)$, as they would be obtained by three component sensors. $T(x,y)$ corresponds to travel time from the source S to the receiver at location (x,y), after a single reflection on the unknown interface $I(x,y)$. In order to illustrate the principle, we also assume that both $T(x,y)$ and $I(x,y)$ are single valued, although this is not required in our actual implemented procedures.

With this information, we can shoot a ray (i.e., solve the ray equations with known initial conditions), backwards from the receiver, with initial direction $-\underset{\sim}{\mu}(x,y)$, through the already known structure, up to the interface immediately above the unknown reflector (see Fig. 13). This will produce then a point of intersection $\underset{\sim}{\eta}_0$ and a refracted direction $\underset{\sim}{\mu}_0$, obtained by Snell's law. For the other half of the ray we can state a two point ray tracing problem between the source S and the point $\underset{\sim}{\eta}_0$, with the provision that it must reflect at the unknown point $\underset{\sim}{\eta}$, and arrive to $\underset{\sim}{\eta}_0$ with direction $-\underset{\sim}{\mu}_0$. If we count the unknowns and conditions we will see that there is a missing condition, which allow us to introduce our additional piece of data: the total travel time $T(x,y)$.

By solving these two problems consecutively, i.e., shooting back from the receiver to $\underset{\sim}{\mu}_0$, and then two-pointing from source to $\underset{\sim}{\eta}_0$, we obtain the reflection point $\underset{\sim}{\eta}$, and also, by taking the bisector between arriving and reflected directions (in the plane they determine), we get $\underset{\sim}{v}$, the normal to the unknown interface at $\bar{\underset{\sim}{\eta}}$. Doing this for each source-receiver pair available, we obtain a set of reflection point-normals $(\bar{\underset{\sim}{\eta}}_j, \bar{\underset{\sim}{v}}_j)$ which can in turn be used to fit surface patches by least squares. In complex regions, GEOBLD can be used profitably to decide the extent of different patches, connect them, grow faults, and in general use geological know-how, together with any additional available information (like well-logs), to interactively enrich the model.

5. Dynamic Inversion

The preliminary model derived in paragraph 4 is now used as an starting model in a nonlinear least squares fit of the data. There are various ways in which this can be accomplished, but again, a stagewise approach is the most advisable, since it will allow tight human control in an interactive environment.

For this procedure we use only the interpreted travel time data. Let $\mathbf{M}(\underset{\sim}{\alpha}^0;\underset{\sim}{\beta}^0)$ be the initial parameterized model, where $\underset{\sim}{\alpha}^0$ corresponds to interface parameters, while $\underset{\sim}{\beta}^0$ corresponds to velocity parameters. In its full generality, the data set is very complex, since there will be many sources and receivers; for each source and corresponding receiver array, we will have identified travel time surfaces (potentially multivalued), corresponding (say) to simple reflections from buried reflectors. So, to be precise, our travel time data should be described by functions $T^0(S,R,I)$, where: $S =$ source, $R =$ receiver, $I =$ reflecting interface, $0 =$ observed.

For each value of S, R, I and model $\mathbf{M}(\underset{\sim}{\alpha};\underset{\sim}{\beta})$, we can calculate

$$T^c(S,R,I;\underset{\sim}{\alpha};\underset{\sim}{\beta})$$

by ray tracing. The best parameters, in the least squares sense (by no means the only norm that can be used to measure deviations between observed and calculated quantities), can be obtained by:

$$\min_{\underset{\sim}{\alpha},\underset{\sim}{\beta}} \sum_{S,R,I} (T^0(S,R,I) - T^c(S,R,I;\underset{\sim}{\alpha};\underset{\sim}{\beta}))^2 \tag{1}$$

where the sum is extended to all available data values. If this is not complicated enough, we can also introduce constraints:

$$\underset{\sim}{g}(\underset{\sim}{\alpha};\underset{\sim}{\beta}) = \underset{\sim}{0}, \tag{2}$$

$$\underset{\sim}{h}(\underset{\sim}{\alpha};\underset{\sim}{\beta}) \geq \underset{\sim}{0} \tag{3}$$

that can prevent the parameters from wandering into non-physical space. These constraints are also important in regularizing the problem, and in bringing into the formulation as much apriori information as is available. The nonlinear programming problem (1–3) can be solved by using available high quality software [2].

We can, of course, include all the data at once, and try to improve upon all the model parameters $(\underset{\sim}{\alpha}^0;\underset{\sim}{\beta}^0)$ simultaneously. However, a more cautious approach is to proceed from the free surface down, one region at a time, using only the relevant data and parameters. This will result in similar problems, but of smaller size. Once the model has been refined in this way, one can attempt a more global fit. Ideally, the system should be sufficiently flexible to allow the choice of submodels and relevant data sets in a simple manner, providing good visualization of the optimization results, so that local decisions can be made by the operator interactively.

We have shown elsewhere [7, 13, 14], how this procedure can be implemented with our ray tracing approach, in 2–D and 3–D, but only in the unconstrained case. We hope to start considering the constrained case in the near future, and also to consider full wave inversion, by using the finite element forward modeling system within the optimization loop.

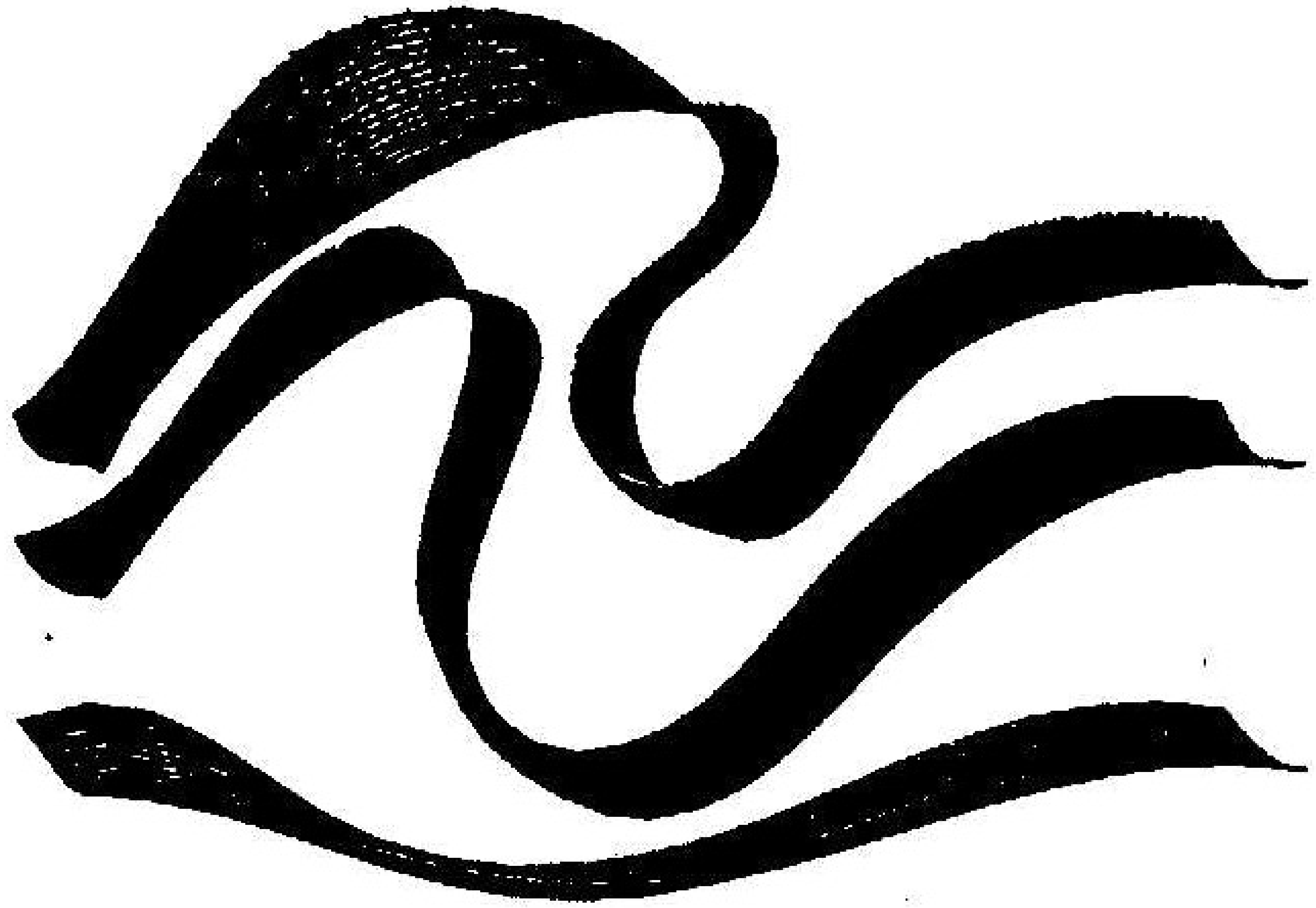

Figure 1. A recumbent fold, featuring multi-valued surfaces.

Figure 2. A reverse fault, involving multi-valued, discontinuous surfaces.

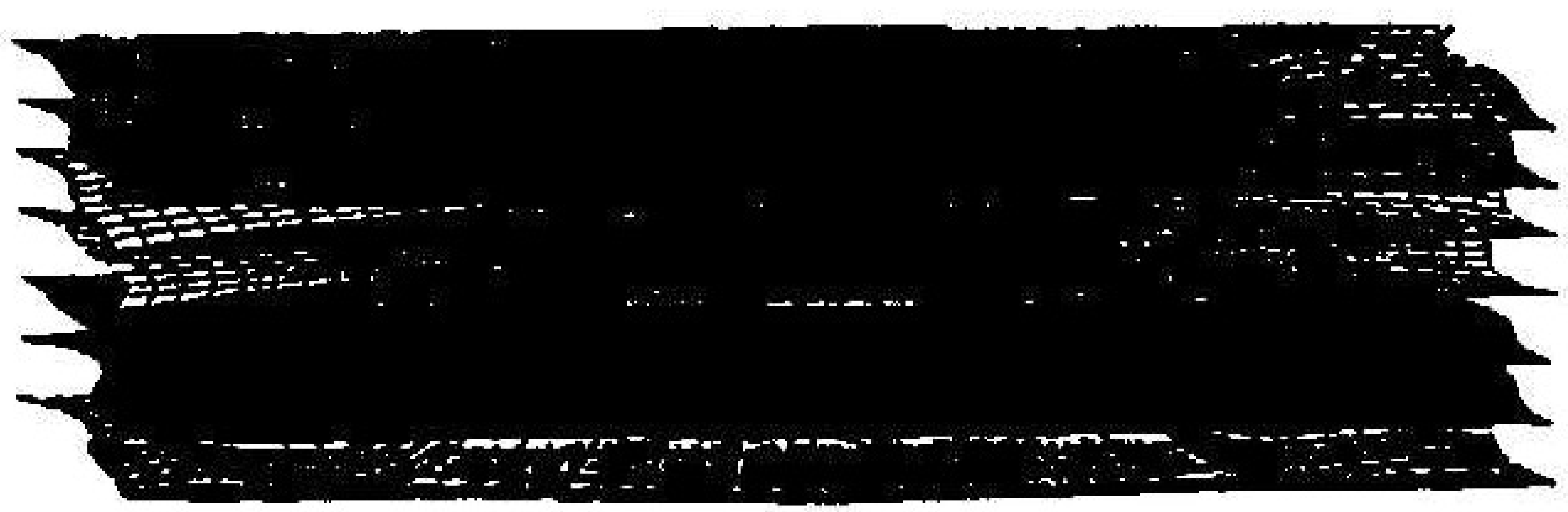

Figure 3. Multi-layer structure pierced by a salt dome (hidden).

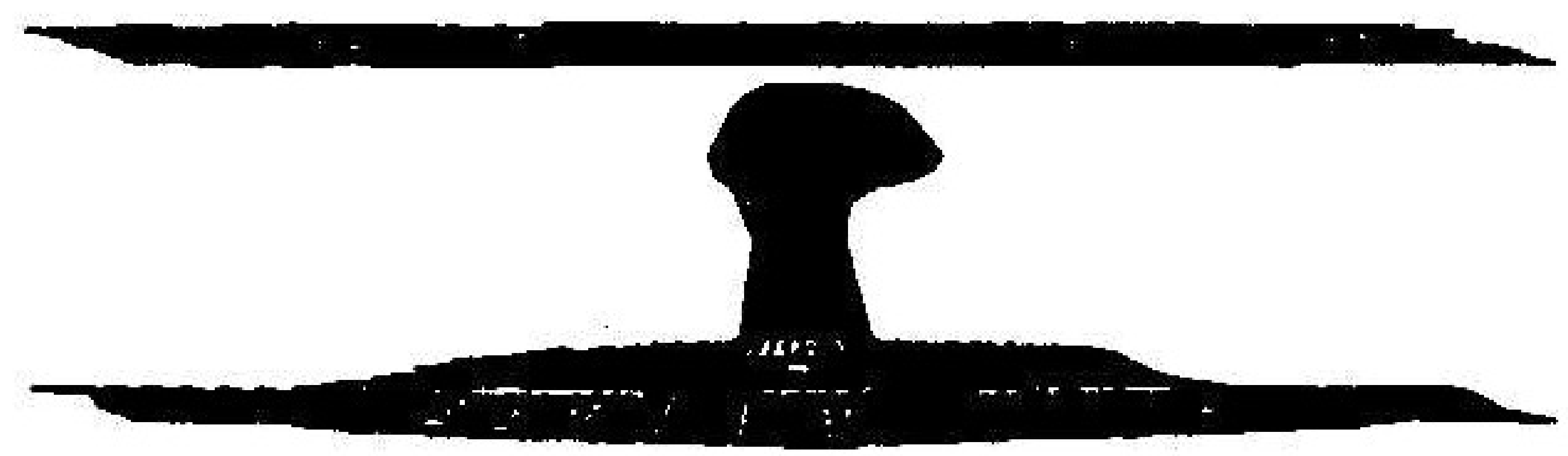

Figure 4. Detail of salt dome, including overhang, and almost vertical walls.

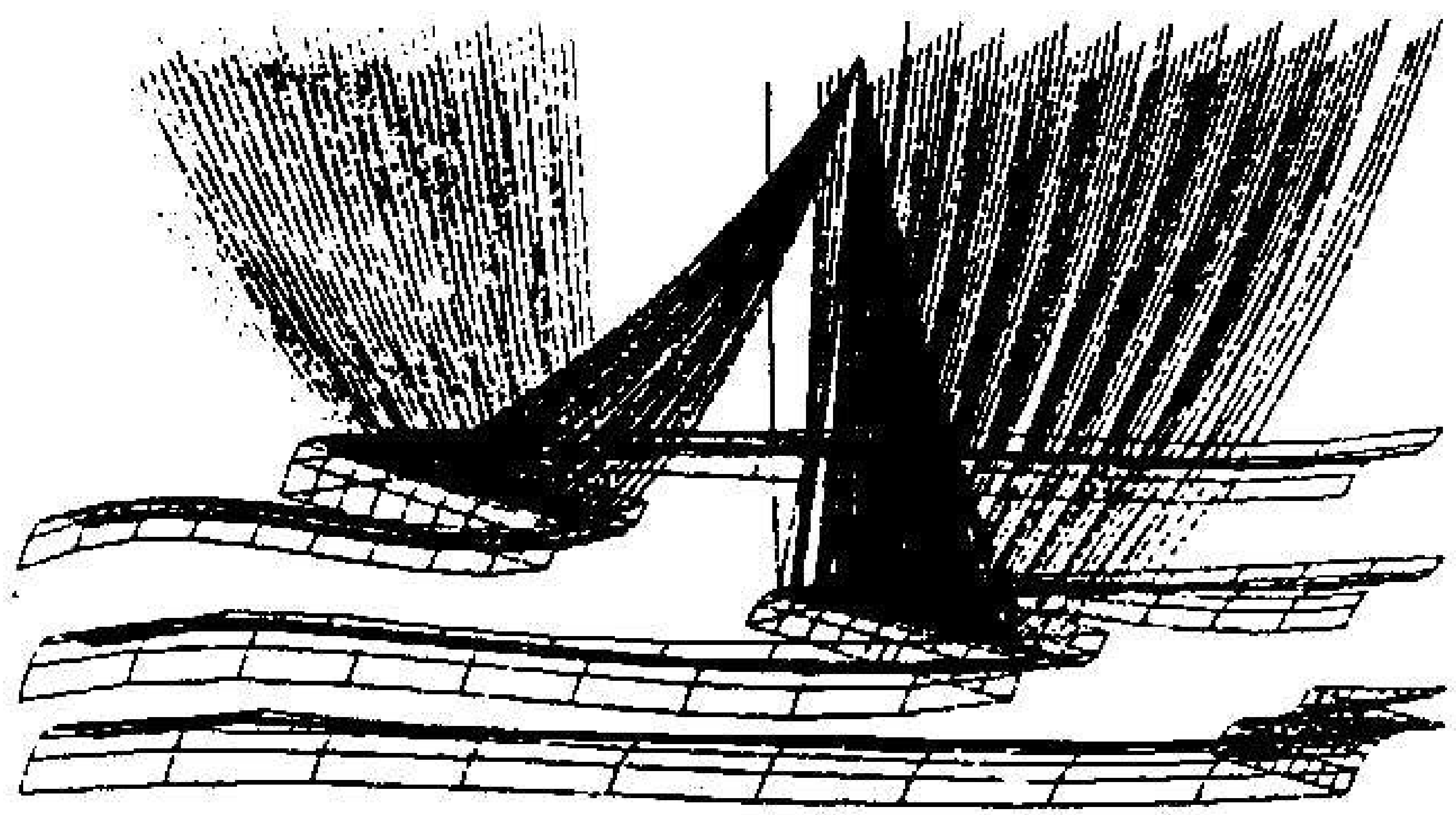

Figure 5. Two-point ray tracing through reverse fault model. Principal reflection from fault. 25 x 10 receiver array. Velocity in region 1, $v_1 = 2 + 0.4z$ kf/sec.

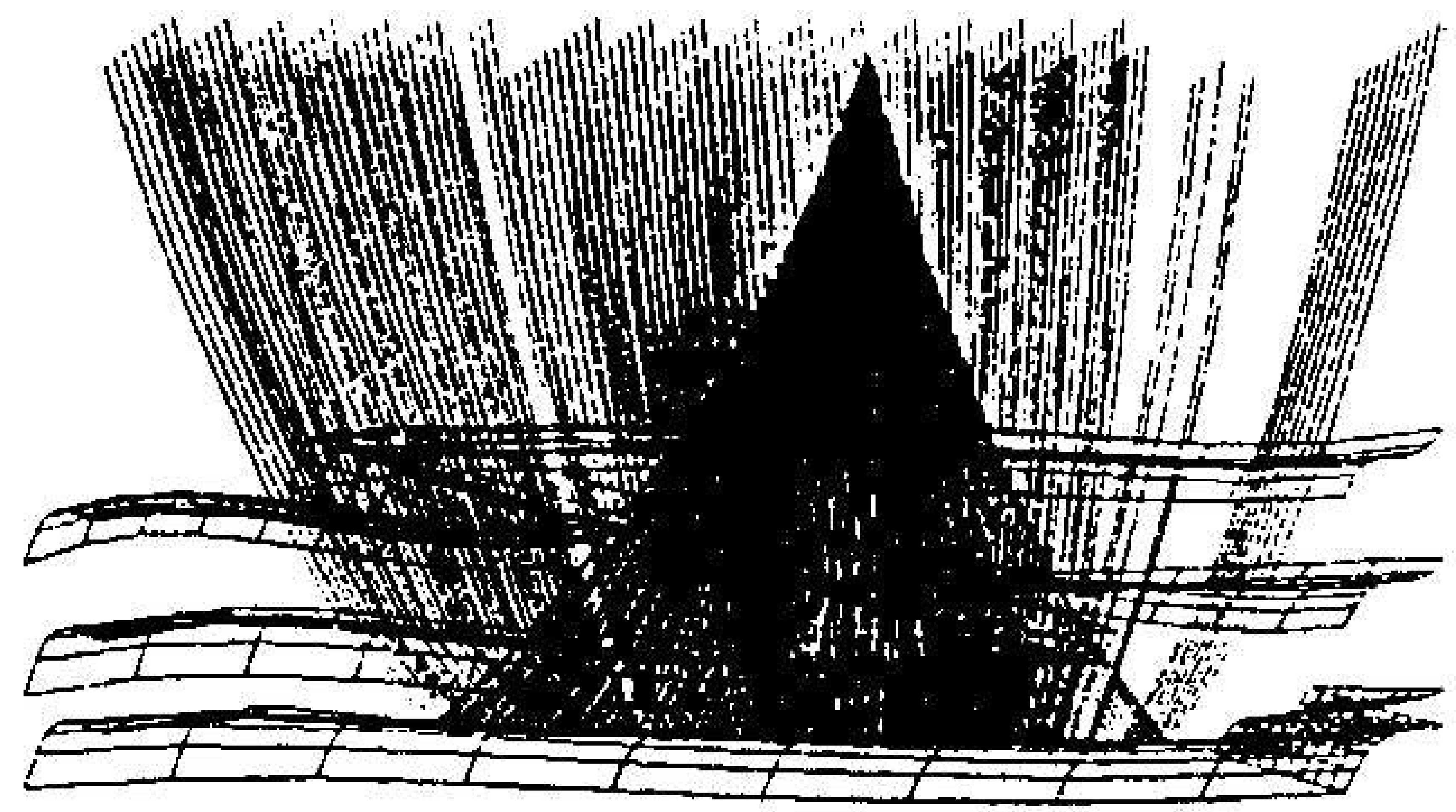

Figure 6. Same as Figure 5. Bounce from bottom interface.

Figure 7. Two-point ray tracing through salt dome model. Reflection from third interface. 50 x 20 array.

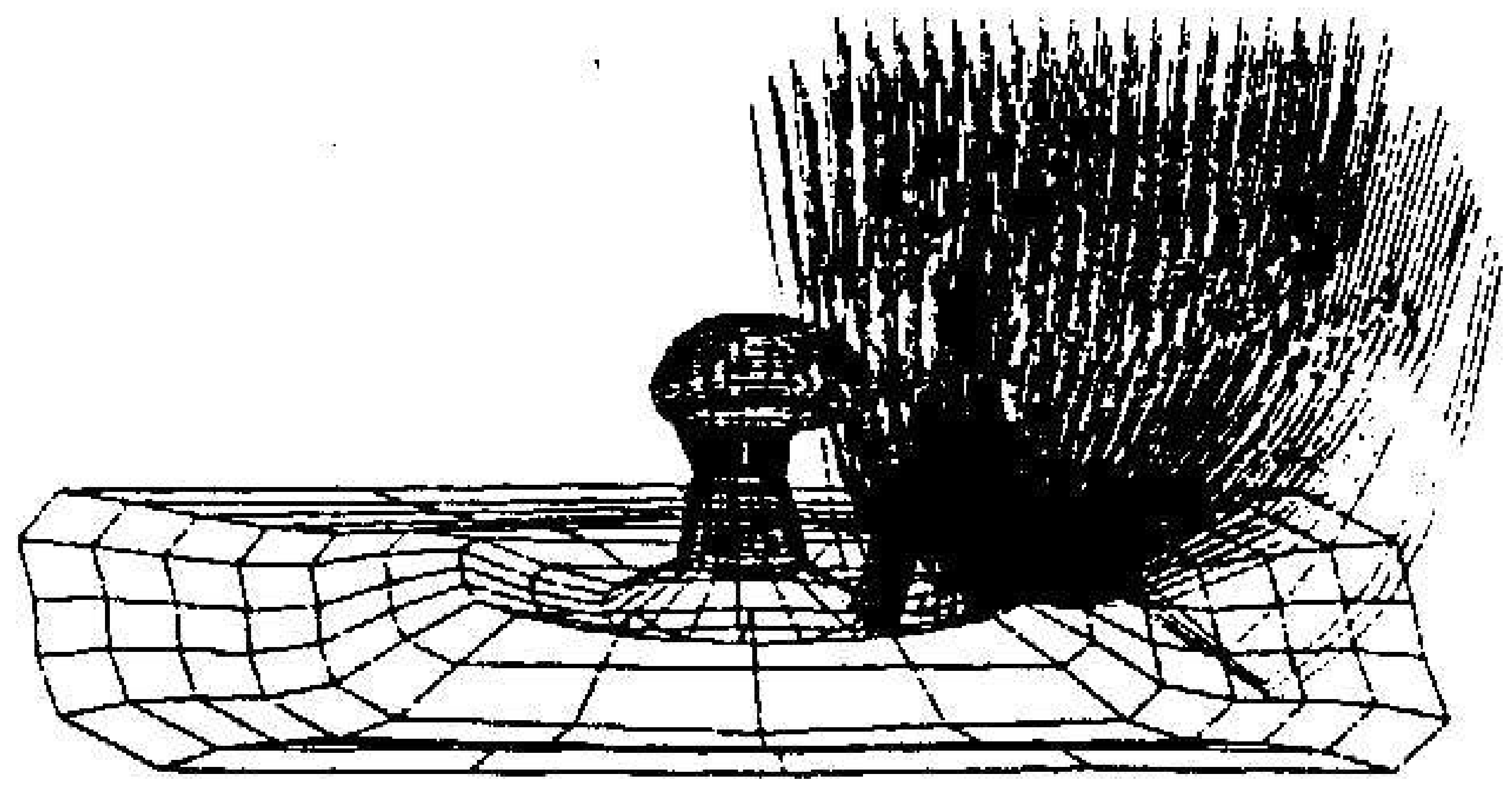

Figure 8. Same as Figure 7. Reflection from dome and bottom interface. Other interfaces present but not drawn.

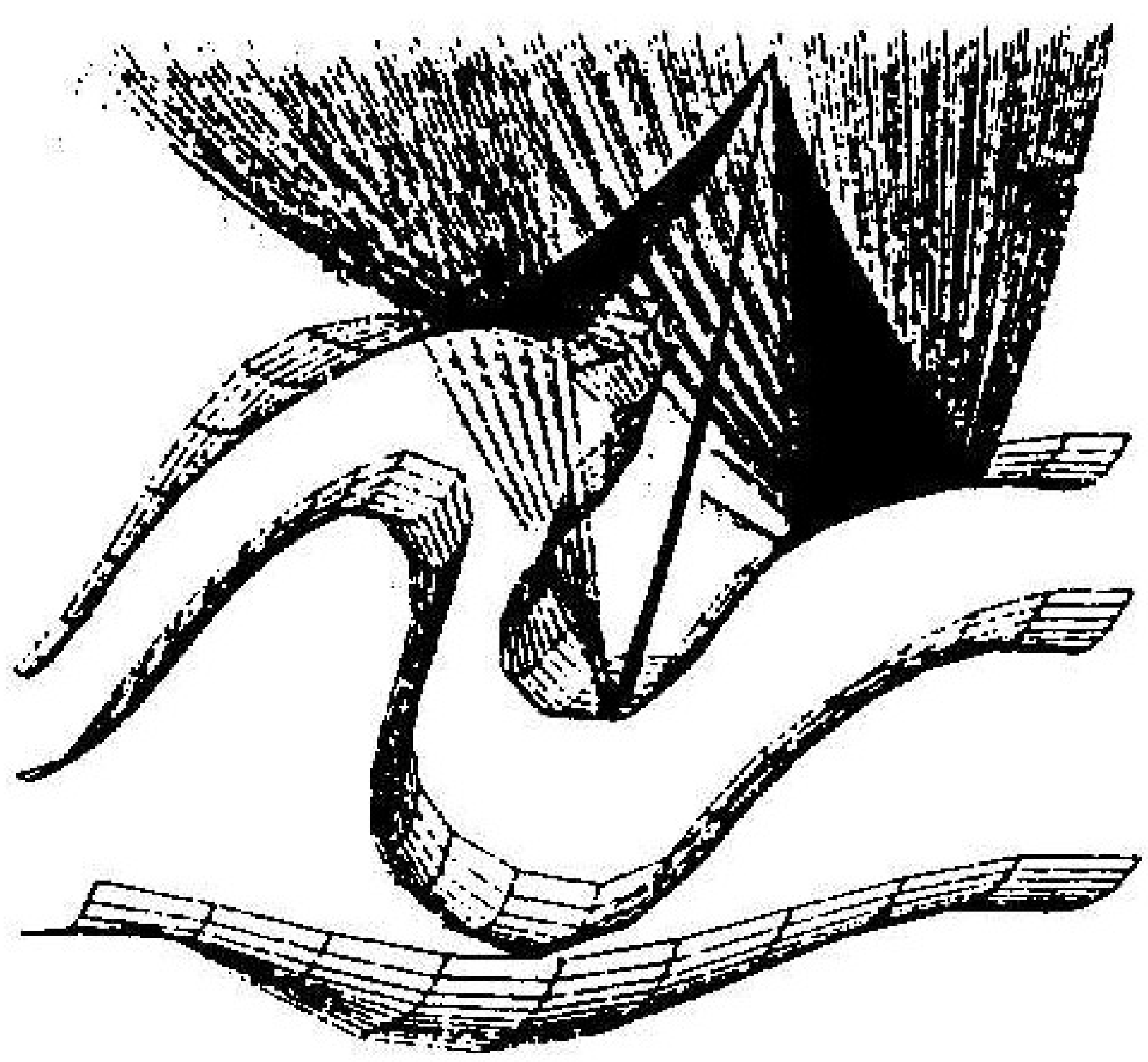

Figure 9. Two-point ray tracing through fold model. 25 x 10 receiver array. Featuring multi-path arrivals.

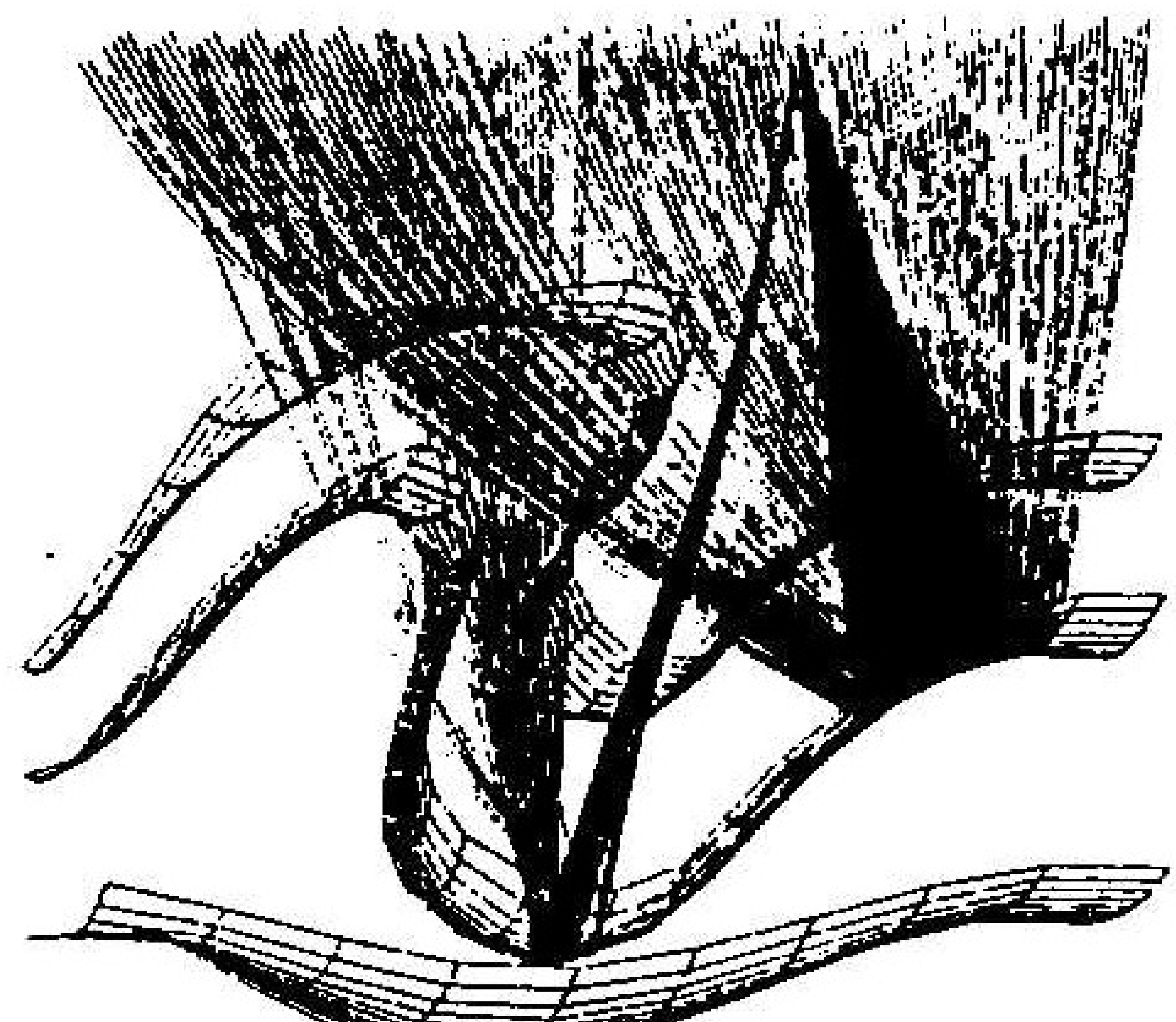

Figure 10. Same as in Figure 9. Reflection from second interface.

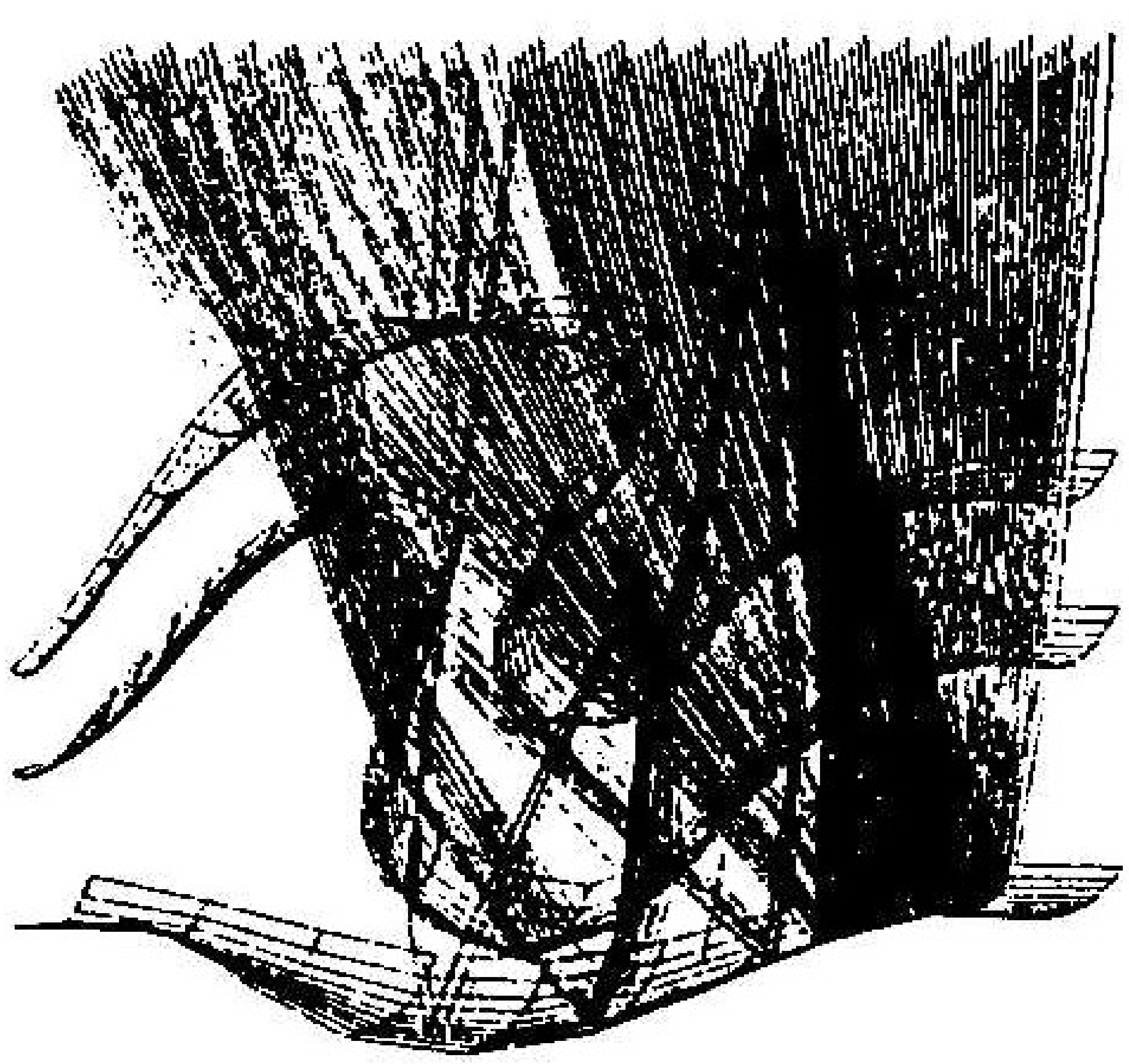

Figure 11. Same as in Figure 10. Reflection from third interface.

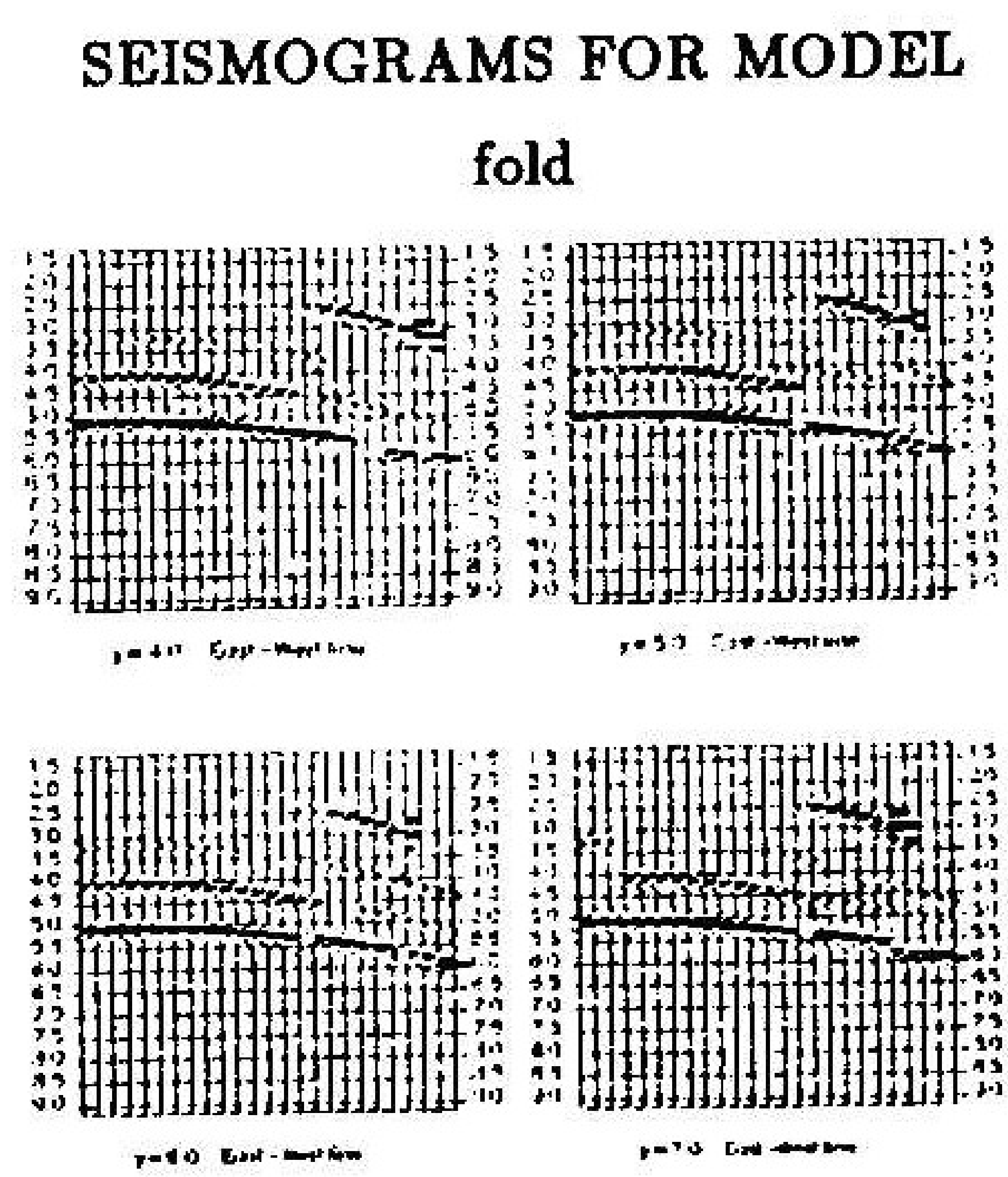

Figure 12. Selected synthetic seismograms along four geophone lines, corresponding to the 2P arrivals of Figures 9 through 11.

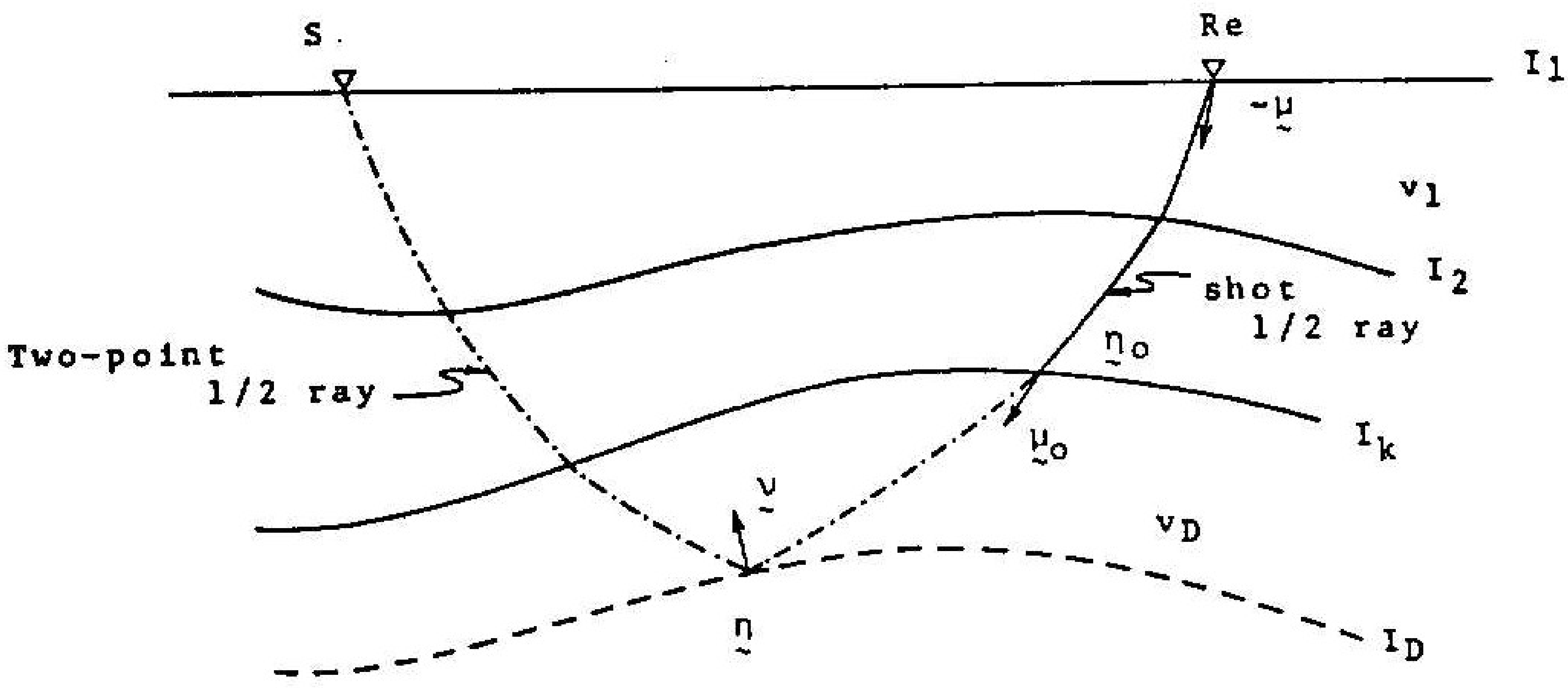

Figure 13. Schematic representation of two-point-shooting migration.

References

[1] T. D. Garvey, J . D. Lowrance , M. A. Fischler, An Inference Technique for Integrating Knowledge from Disparate Sources, Proc. 7th Joint Conference Artificial Intelligence, 319–325. AAAI, Menlo Park, CA, 1981.

[2] P. E. Gill, W. Murray, M. A. Saunders , M. H. Wright, User's Guide for NPSOL (Version 4.0): A FORTRAN Package for Nonlinear Programming, Technical Report 86–2, Systems Opt. Lab., Stanford University, CA, 1986.

[3] M. Lentini, V. Pereyra , PASVA4: An Ordinary Boundary Solver for Problems with Discontinuous Interfaces and Algebraic Parameters, *Mat. Appl. Comp.*, 2 (1983), 103–118.

[4] L. R. Lines, A. K. Schultz , S. Treitel, Cooperative Inversion of Geophysical Data, *Geophysics* 53 (1988), 8–20.

[5] J. D. Lowrance, Evidential Reasoning with GISTER: A Manual, Artificial Intelligence Center, SRI Int., Menlo Park, CA, 1987.

[6] W. Millavec, V. Pereyra, GEOBLD: An Interactive Three Dimensional Geometrical Modeling System-User Manual, Weidlinger Associates Inversion Project (WAIP), Report 87–04, 1987.

[7] V. Pereyra, Two-point Ray Tracing in Heterogeneous Media and the Inversion of Travel Time Data, in Computational method in Applied science and Engineering (Eds. R. Glowinski , J. L. Lions), North-Holland Pub. Co., Amsterdam, 553–570, 1980.

[8] V. Pereyra, W. H. K. Lee , H. B. Keller, Solving Two-Point Seismic Ray Tracing Problems in a Heterogeneous Medium, *Bull. American Geophys. Soc.*, 70 (1980), 79–99.

[9] V. Peretra, Deferred Corrections Software and Its Applications to Seismic Ray Tracing, *Comp. Appl.*, 5 (1984), 211–226.

[10] V. Pereyra, Modeling with Ray Tracing in Two-Dimensional Curved Homogeneous Layered Media, in New Computing Environments: Microcomputers in Large Scale Computing (Ed. A. Wouk), 39–67 SIAM Pub., philadelphia, PA, 1987.

[11] V. Pereyra, Improved Automatic Two-Point Ray Tracing in Inhomogeneous Three-Dimensional Media, WAIP, Report 87–01, 1987.

[12] V. Pereyra, Time-To-Depth, Pre-Stack Migration of Curved Horizons on Piecewise Heterogeneous Three-Dimensional Media, WAIP, Report 87–02, 1987.

[13] V. Pereyra, Numerical Methods for Inverse Problems in Three-Dimensional Geophysical Modeling, in special volume on Computational Geophysics (Ed. V. Pereyra), *Appl. Num. Math.*, 1988.

[14] V. Pereyra, Numerical Methods for Inverse Problems in Petroleum Exploration, invited paper, *Acta Cientifica Venzolana,* 1988.

[15] G. L. Wojcik, D. K. Vaughan, M. Barenberg, M. B. Hulit , J. Mould, Large-Scale, Explicit Wave Simulation on the CRAY 2, *App. Num. Math.,* 1988.

[16] S. Wright, V. Pereyra, Adaptation of a Two-Point Boundary Value Solver to a Vector-Multiprocessor Environment, to appear in *SIAM J. Matrix Anal. and Appl.*, 1988.