T-3038

LONG WAVELENGTH STATIC ANALYSIS USING FIRST BREAKS AND GENERALIZED LINEAR INVERSION

by

Shih-Yeng Kuo

T-3038

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in Partial fulfillment of the requirements for the degree of Doctor of Philosophy, Geophysics.

Golden, Colorado

Date 20 March , 19 85

Signed: Shih-Yeng Kuo

Shih-Yeng Kuo

Signed: William A. Schneider

Dr. William A. Schneider
Thesis Advisor.

Golden, Colorado

Date 21 March , 19 85

P R Romig

Dr. Phillip R. Romig
Head
Department of Geophysics.

ABSTRACT

The goal of this research is to study the long wavelength static problem of the seismic reflection method. An iterative inversion procedure for refraction data has been developed. A forward refraction modeling program, designed to handle complicated structures, is used in conjunction with a generalized linear inversion (GLI) technique and observed first arrival times. The output is a numerical description of the long wavelength near surface structure. Once the near surface model is created, static corrections are directly computed.

This method can resolve both the velocity and the depth of the refractor. Both synthetic and real examples prove this method useful in the detection of long wavelength structures. Solutions are stable and unique for reasonable initial guesses. For a long wavelength structure and limited number of inverse parameters, the speed of convergence is fast. Since the inversion is a GLI method, it is not very sensitive to noise. This method may be used as a weathering layer velocity detector.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vii
NOMENCLATURE AND SYMBOLS	x
ACKNOWLEDGEMENTS	xii
CHAPTER 1: INTRODUCTION	1
1.1: Previous Methods	1
1.1.1: Shot Hole Method	3
1.1.2: Reflection Method	3
1.1.3: Refraction Methods	4
1.2: Iterative Refraction Data Inversion Method	6
CHAPTER 2: REFRACTION FORWARD MODELING	7
2.1: Refraction Ray Tracing Theory	8
2.1.1: Constant Velocity Layer	10
2.1.2: Linear Vertical and/or Lateral Velocity Layer	10
2.1.3: Input Velocity Function	14
2.2: Refraction Modeling Program	14
2.3: Synthetic Modeling Example	16
2.3.1: Constant Velocity Layer	16

TABLE OF CONTENTS (Continued) Page

2.3.2: Linear Vertical and Lateral Velocity Changes Model........... 18

2.3.3: Input Velocity Function.......... 18

CHAPTER 3: GENERALIZED LINEAR INVERSION......... 24

3.1: Theory of Generalized Linear Inversion... 24

3.2: The GLI Algorithm........................ 29

3.2.1: Parameter Determination.......... 29

3.2.2: Initial Guess..................... 29

3.2.3: Sensitivity Matrix and Difference Matrix Determination............. 30

3.2.4: Singular Value Decomposition..... 30

3.2.5: Correction Determination......... 31

3.2.6: Result Examination............... 32

CHAPTER 6: REFRACTION MODELING AND INVERSION SYSTEM................................ 33

4.1: Simple Synthetic Model Tests............. 35

4.1.1: Model with Straight Line Boundaries....................... 35

4.1.2: Model with Curved Boundary....... 36

4.1.3: Noisy Case....................... 45

4.2: Practical Application.................... 45

4.2.1: Calculation of the Sensitivity Matrix.......................... 51

TABLE OF CONTENTS (Continued) Page

4.2.2: Synthetic Model Test............. 55

4.3: Weathering Layer Velocity Study.......... 62

4.3.1: Undulated Refraction Model Test.. 62

4.3.2: V_0 Inversion Test................ 67

CHAPTER 5: REAL DATA TEST....................... 68

CHAPTER 6: CONCLUSIONS AND RECOMMENDATIONS...... 86

REFERENCES....................................... 88

APPENDIX: LISTING OF THE REFRACTION MODELING AND INVERSION PROGRAMS................ 90

LIST OF FIGURES

Figure			Page
1	:	Elevation and weathering static corrections	2
2	:	Single refractor model with straight line boundaries and constant layer velocities	11
3	:	Main algorithm of the modeling program	15
4	:	Synthetic first break picks diagram of the model in Figure 2	17
5	:	Two-refractor model with linear vertical and linear lateral velocity change	19
6	:	Model response for Figure 5 with velocity gradient	20
7	:	Two refractor model with curved boundaries and input velocity function	21
8	:	Model response for Figure 7	22
9	:	The algorithm of the refraction Modeling-Inversion System	34
10	:	Initial guess and original model for straight line boundary case	37
11	:	Travel time curves (lower part) and residual errors (upper part) for straight line boundary example	39
12	:	Output and original models for straight boundary case	40
13	:	Initial guess and original model for curved boundary case	41
14	:	Travel time curves (lower part) and residual errors (upper part) for curved boundary example	43
15	:	Output and original models for curved boundary case	44

Figure LIST OF FIGURES (Continued) Page

Figure			Page
16	:	Curved boundary model response with and without noise	46
17	:	Travel time curves (lower part) and residual erros (upper part) for noisy case example	48
18	:	Output and original models for noisy data inversion	49
19	:	First breaks of the final output model and the original noisy first breaks	50
20	:	Using linear approximation to obtain the partial derivative in the sensitivity matrix related to the depth change Δz	53
21	:	Using linear approximation to obtain the partial derivative in the sensitivity matrix related to the velocity change ΔV	54
22	:	Synthetic model with undulated surface	57
23	:	Initial guess for syncline model	58
24	:	QC plot for the initial guess model	59
25	:	Output model by GLI	60
26	:	QC plot for the output model	61
27	:	Initial guess model with wrong V_0	63
28	:	QC plot for the guess model with wrong V_0	64
29	:	Output model with wrong V_0	65
30	:	QC plot for the output model with wrong V_0	66
31	:	Surface elevation profile of the real data example	69
32	:	Stack section (with elevation correction)	70

Figure		LIST OF FIGURES (Continued)	Page

Figure		Title	Page
33	:	First break picks of the real data set	71
34	:	Stack section (with conventional refraction static correction)	72
35	:	Near surface model for 2500 ft/sec V_0	74
36	:	Near surface model for 3000 ft/sec V_0	75
37	:	Near surface structure of good initial guess (from conventional method)	76
38	:	Constant thickness initial guess model (150 ft at all the geometry control points)	77
39	:	QC plot of good initial guess (from the output of conventional method)	78
40	:	QC plot of good initial guess (first iteration)	79
41	:	QC plot of good initial guess (second iteration)	80
42	:	QC plot of bad initial guess (constant thickness)	81
43	:	QC plot of bad initial guess (first iteration)	82
44	:	QC plot of bad initial guess (second iteration)	83
45	:	Stack section (with GLI refraction static corrections)	85

NOMENCLATURE AND SYMBOLS

weathering layer: The low velocity layer, a zone of low velocity material near the earth's surface at the base of which the velocity abruptly increases.

long wavelength structure: In this thesis, the wavelength is compared with the geophone spreading; a structure with the wavelength longer than 10-station interval can be treated as a long wavelength structue.

GLI: Generalized linear inversion.

CDP: Common depth point.

QC : Quality control.

RMS: Root mean square.

CPU: Central processing unit.

V_0 : Weathering layer velocity.

V_1 : The velocity of the refractor.

p : Ray parameter.

x,z: The horizontal and vertical coordinates.

dV_x , dV_z: The lateral and vertical velocity changes in unit distance.

A : Sensitivity matrix.

A^T : Transpose of A.

C : Observed model response.

P : Model parameter.

ΔC : The difference between the observations and the input model response.

NOMENCLATURE AND SYMBOLS (continued)

ΔP: The correction of the parameter P.

u : The eigenvector associated with the column of A.

v : The eigenvector associated with the row of A.

λ : The eigenvalue of A.

U : The matrix whose column vectors are eigenvectors U .

V : The matrix whose xolumn vectors are eigenvectors V .

Λ : The diagonal matrix whose elements are non-zero eigenvalues λ .

Λ^{-1} : Inverse of Λ.

θ_{cr}: Critical angle.

S : The dip angle of the refractor.

T_i : Intercept time.

ACKNOWLEDGEMENT

Initially, I would like to personally thank my advisor, Dr. William A. Schneider, for his support and technical advice. Also, I would thank Dr. Robert N. Shurtleff for his valuable advice in my research.

I am greateful to Golden Geophysical Corporation for offering my both the financial support and the facilities necessary for the completion of this thesis. I also wish to thank the members of the staff at Golden who assisted me in my research.

Finally, I appreciate Dr. Frank Hadsell, Dr. Samuel B. Romberger, Dr. Phillip R. Romig and Dr. Richard Yeatts who served as the members of my thesis committee. I would like to thank them for their valuable oppinions.

CHAPTER 1

INTRODUCTION

The static problem of the seismic reflection method is caused by near surface effects such as surface elevation variations and near sub-surface velocity and thickness variations. It is a major problem in seismic processing, especially for land data. Incorrect static corrections cause deteroration of the stack due to misalignment within the common depth points (CDP), distortion in the structural times of deep reflectors, attenuation of high frequency response and incorrect velocity analysis results. Correct static corrections remove the near-surface inhomogeneous variations and allow referencing the seismic data to a uniform datum plane (Figure 1). Recent advances in other reflection seismology techniques make higher quality seismic data available, which in turn makes solving the static problem more and more important.

1.1 Previous Methods

People have been trying to solve the reflection seismic static problem for a long time. There are several different ways to determine the static corrections, using either

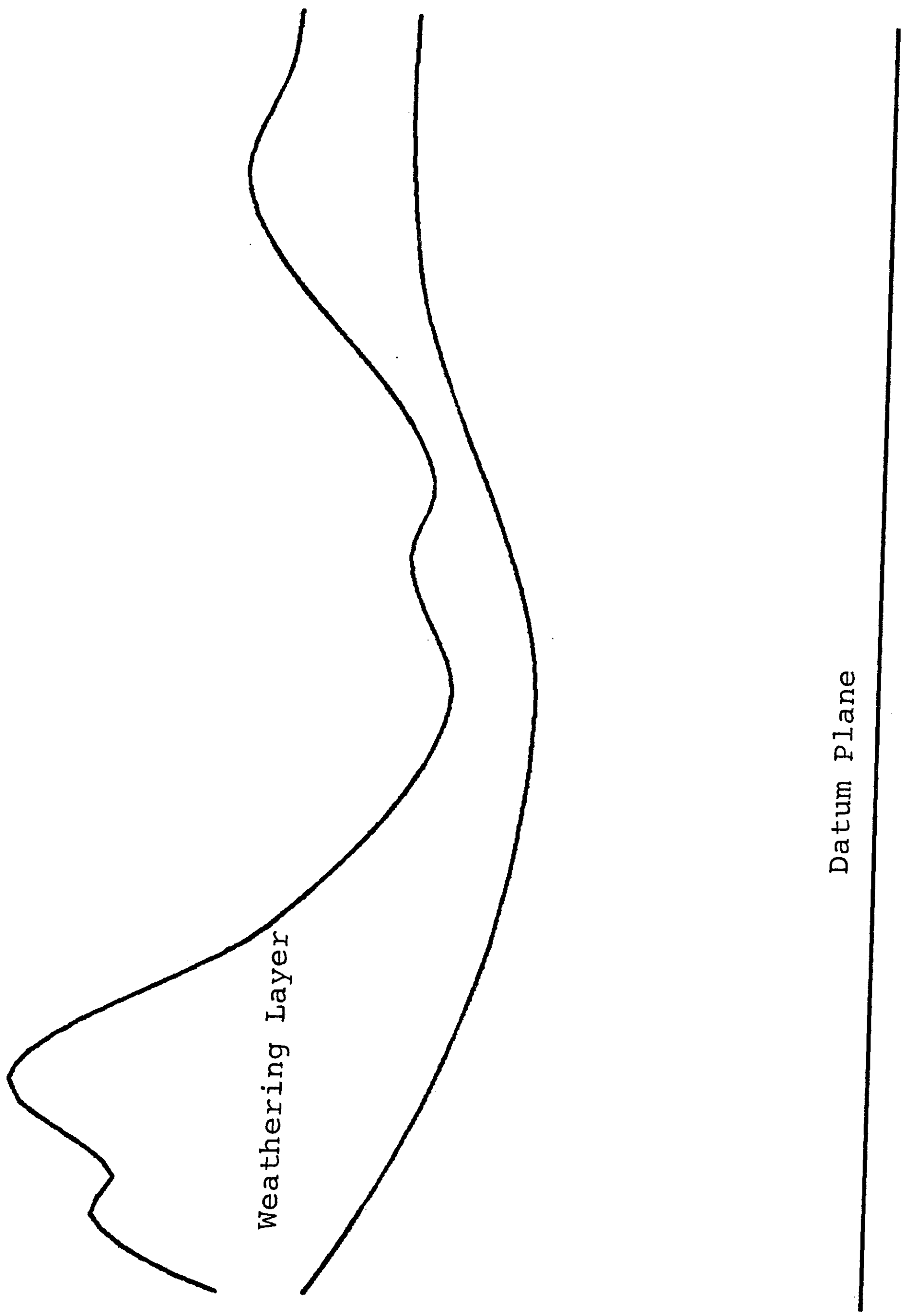


Figure 1   Elevation and weathering static corrections

reflected or refracted data. In this section we will give a brief description of each, with the reference listed in brackets.

1.1.1 Shot Hole Method (Dobrin, 1976)

This method determines the thickness of the weathering layer by a series of shots at different depths in a deep hole. It is good for the station just above the hole but not accurate for other stations. It is expensive and sometimes impossible to use for very thick weathering zones.

1.1.2 Reflection Method (Waters, 1978 ; Hileman, Embree, Pflueger, 1976 ; Wiggins, Larner, Wisecup, 1975)

It is possible to cross-correlate selected windows of the traces concerned to find out at which time difference the best match of the significant signals is obtained. The correlation window is taken over prominent reflections after the normal moveout (the variation of reflection arrival time because of variation in the shotpoint-to-geophone distance) has been removed.

The basic formula for the total travel time difference between two traces can be represented as the following sum:

$$T = T_d + T_{ws} + T_{wr} + T_{nmo} + T_n$$

T_d : layer dip effect

T_{ws} : weathering layer static related to the source

T_{wr} : weathering layer static related to the receiver

T_{nmo} : normal moveout error

T_n : noise effect

The problem can be handled in the CDP, common shot, common receiver or common offset domains. None of these can give the unique solution because this is an under-determined problem (simultaneous equations with more unknowns than the number of equations).

It is still a very good way to determine the high frequency statics under the assumption of a smooth reflector in the scale of the station interval. It is obvious that the reflection method is not perfect because there is no way to distinguish whether long wavelength anomalies (greater than one spread length) are caused by the reflector or by the weathering layer.

1.1.3 Refraction Methods

Instead of studying reflection signals, these methods use the first break picks (head waves through the weathering layer) to find the static corrections. In this way, the problem in approached directly. However, the near surface velocity inversion case will degrade these methods.

Time Delay Method (Gardner, 1967 ; Wyrobek, 1956)

The intercept time can be separated into a source component and a receiver component revealing the depth of the refractor can be found. This method is however very sensitive to the dip of the refractor. Dip angles as low as 5 degrees can seriously degrade the results.

Reciprocal Method and GRM (generalized reciprocal method) (Dobrin, 1976 ; Palmer, 1980)

The reciprocal method averages the forward and reverse travel times so that the results are not as sensitive to the dip angle. The conventional reciprocal method smooths refractor irregularities, but GRM can better handle refractor irregularities. It is possible to fit a travel time curve with a straight line to find the intercept time as the low frequency component and the deviations from the fitted line as the high frequency component of the static corrections. These methods are still dependent on dip angle and they have other shortcomings as well.

1.2 Iterative Refraction Data Inversion Method

Since the high frequency static problem can be well solved by the reflection method, the goal of this research is to study the low frequency static problem. In this method, we are attempting to recover a numerical description of the subsurface so that the static corrections might be applied. As in many real earth problems, it is impossible to find an analytical solution, and an approximate linear technique must be used. For this inversion scheme, the end result is the optimun solution of a system of linearly independent equations. The method developed in this thesis combines refraction forward modeling and a linear inversion scheme to find the long wavelength subsurface structure. It only requires a reasonably accurate set of first break picks and a simple initial guess such as flat boundaries. This method has an advantage over similar methods because it is robust with respect to steep dips of the refractors. Also, because this method relies on the generalized linear inversion (GLI) process, the noise inherent in real data sets can be improved. Another advantage to this technique is that a weathering layer velocity (V_0) estimate is produced from the output of the inverison.

CHAPTER 2

REFRACTION FORWARD MODELING

All refraction analysis methods assume some models of the near surface structure. This is normally taken to be a series of layers whose thickness and velocities may vary. The arrival times of the first breaks depend on the layer thickness and velocities, and the problem in to determine those unknowns from the measured first breaks.

There are two main parts in this thesis, one is refraction forward modeling, and the other is linear inversion for unknown parameters (e.g. depth and velocity). The whole process uses the first arrival times and then combines the forward modeling program and the generalized linear inversion technique to get the numerical description of the near subsurface structures through iteration.

There are several ways to perform the refraction modeling. For complicated models, the computing time needed for the calculation restricts the choice to the most efficient one. Ray tracing is a fast and accurate way to model refractions if the arrival time is the only response to be searched but not the amplitude or waveform.

2.1 Refraction Ray Tracing Theory

The first arrivals are separated into three kinds of signals: direct arrivals, refracted arrivals and diffracted arrivals if any diffraction point exists. The diffraction points occur where there are sudden changes (discontinuity) of the slope or the velocity along the refractor. The diffracted arrivals are from the waves travelling through the diffraction process. The refraction refers only to head waves.

The direct wave is the wave travelling along the surface. The travel time of the direct wave can be easily calculated by integration along the surface boundary.

The refracted wave is the most important signal which provides information about the refractor. By Snell's Law, when the incident angle equals to the critical angle, the refracted ray will travel along the boundary and the ray will leave the boundary by the critical angle. The refracted arrival is not necessarily the first arrival and should be compared with the other kinds of arrivals. The ray path in a homogeneous, isotropic media is a straight line but is a curved ray for other cases. For example, if the ray passes through a region where the velocity change is linear, the ray path is a circle. When a ray goes through a boundary, it obeys Snell's Law.

If a diffraction point exists, the first arrival may come from a combination of refracted and diffracted waves. The diffraction point could be treated as another source and the travel time can be calculated by a similar method.

For a given trace, there is a corresponding shot and a receiver position. Direct arrival is found by integrating between these points along the surface. To find the refracted arrivals, the rays are shot at the critical angle from several points along the refractor. If a singular point exists, rays will be shot from either side of it. The program analyzes these rays and tries to hit the shot position to within a pre-determined accuracy. An accuracy of 0.5 ft can usually be achieved in two to three trials for a curved refractor. For all the cases have been tried, the program has found all rays that theoretically exist.

For diffracted arrivals, the program starts from a diffraction point and finds the rays that connect it to the shot and/or the receiver points. These are combined with the refracted ray paths found above to give refracted-diffracted, diffracted-refracted and diffracted-diffracted arrivals.

All these arrival times provide the information needed to construct a synthetic first break seismogram and the two or three earliest arrivals will be used as the input to the GLI method.

To use the forward modeling program as part of the inversion process, it needs to be fast and accurate. It needs to have the ability of finding all or most of the rays in short computer time. Much effort has gone into the program to make it efficient. For a typical model such as that shown in Figure 2, which has one refractor with four diffraction points, to find all the rays for a system of 55 shots using 48 receivers takes 23 seconds CPU time on a VAX 11/780.

2.1.1 Constant Velocity Layer

If the velocity of a layer is constant, then the ray paths are straight lines and the critical angles are the same.

2.1.2 Linear Vertical and/or Lateral Velocity Layer

When the velocity changes linearly in the vertical and/or the lateral direction, the ray paths are curved and

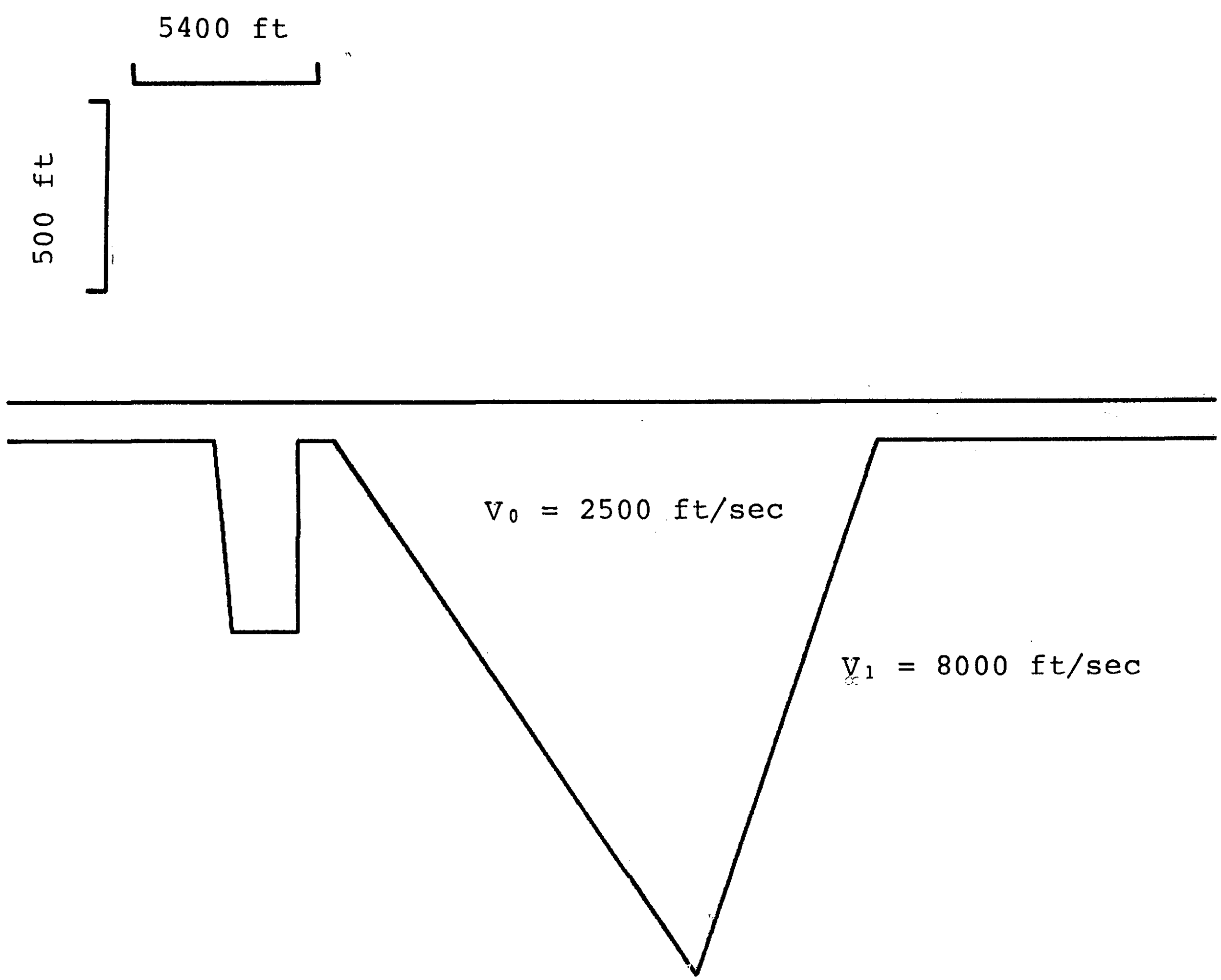


Figure 2 Single refractor model with straight line boundaries and constant layer velocities

the critical angle is a function of the local velocity contrast. From Dobrin (1976), if the model has the condition:

$$V = V_0 + kz$$

where V = velocity at depth z

V_0 = velocity at zero depth

k = linear velocity change constant

The ray parameter p can be difined as a constant for any particular ray with a given initial angle of penetration into the earth. It describes each ray in terms of the surface velocity V_0 and the emergence angle i_0 in the x-z plane.

$$p = \frac{\sin i_0}{V_0}$$

The relationships of the horizontal travel distance x, the vertical travel distance z and the travel time t are:

$$x = \frac{1}{kp} \{\sqrt{1 - p^2 V_0^2} - \sqrt{1 - p^2 (V_0+kz)^2}\}$$

$$t = \frac{1}{k} \ln \frac{(V_0+kz)\ (1 + \sqrt{1-p^2 V_0^2})}{V_0 (1 + \sqrt{1-p^2 (V_0+kz)^2})}$$

$$\left(x - \frac{\sqrt{1-p^2V_0{}^2}}{kp}\right)^2 + \left(z + \frac{V_0}{k}\right)^2 = \frac{1}{k^2p^2}$$

These equations define the ray path as a circle.

The above derivation can be rotated 90 degrees to give the ray path for the case of linear lateral velocity changes. In this situation the centers of all the circles will lie on a vertical line.

For the case where the velocity changes linearly in both the vertical and lateral direction, the ray path is still a circle. All these circles will have their centers on a straight line. This straight line will be neither vertical nor horizontal, but will intersect the horizontal direction at an angle θo depending on the gradient of the wavespeed.

$$\theta o = \tan^{-1} (dV_x/dV_z)$$

where dV_x = lateral velocity change in unit distance

dV_z = vertical velocity change in unit distance

2.1.3 Input Velocity Function

The velocities may be changed at any point in the model. If a velocity function has been specified, the critical angle and the ray path can be calculated analytically and approximately.

2.2 Refraction Modeling Program

The model allows flexible refractor geometry to be specified. The refractors can be piecewise linear or curved. The curved refractors are specified as a set of coordinates at arbitrary spacing. Diffraction points along the refractor are specified by the user.

The program can handle up to three layers (two refractors). Since the program needs picked first breaks as input, it doesn't make much sense to compute more layer. In practice, most data can be treated as a one refractor problem. The algorithm of the modeling program is shown in Figure 3 and described as follow:

a. Set subsurface geometry which includes source-receiver relationship, geophone interval and surface elevation.
b. Set equal spaced sample points as well as some special points such as diffraction points and velocity discontinuous points.

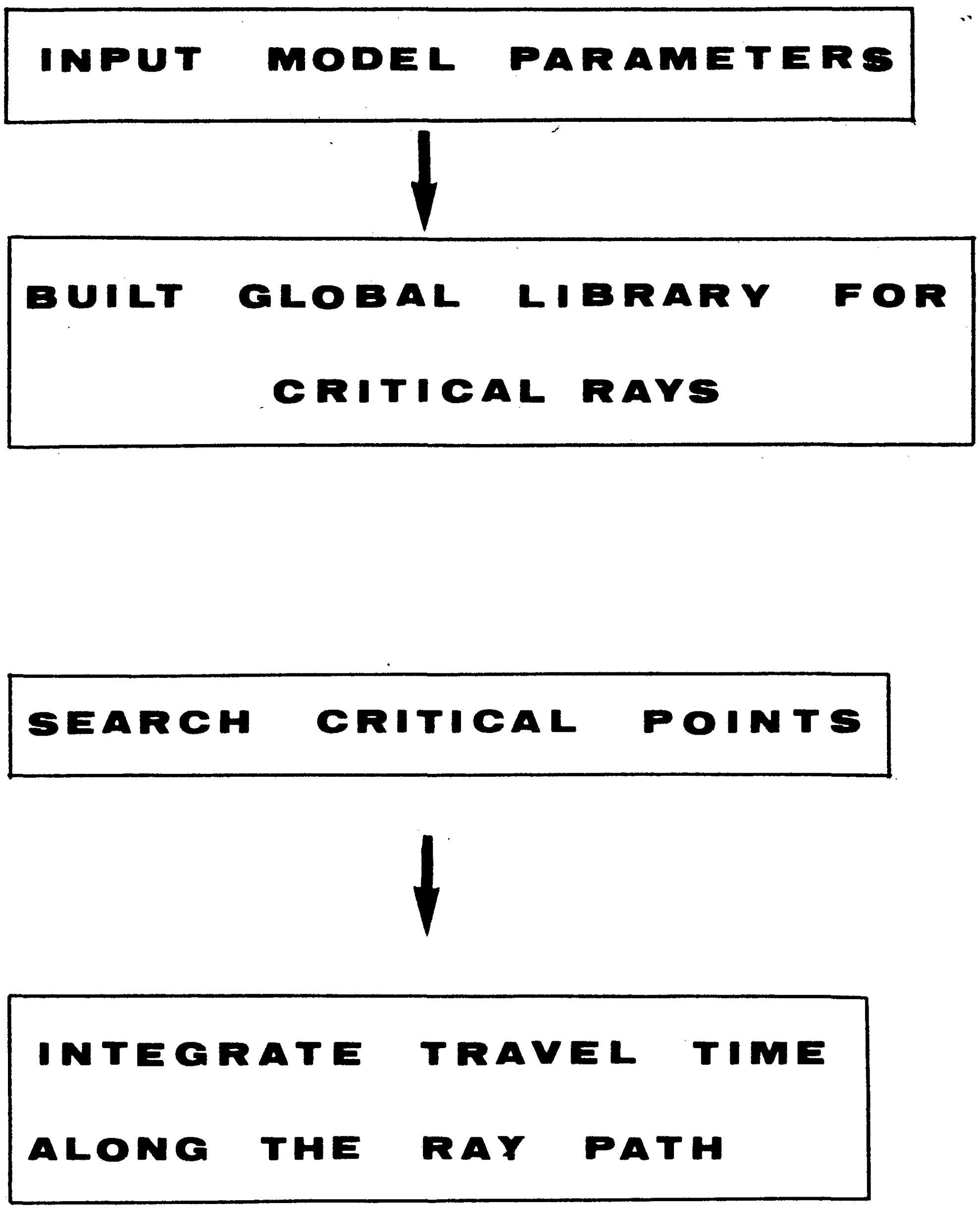


Figure 3 Main algorithm of the modeling program

c. Built a library which contains the information about the critical rays from the sample points. (such as travel times and outgoing positions on the surface)
d. Find the critical point corresponding to a given shot point to a prescribed accuracy by the input.
e. Find the critical points corresponding to the given receivers.
f. Integrate the travel time along the subsurface from shot critical points to receiver points.
g. Add all the travel times in d, e and f together to get the total travel time.

2.3 Synthetic Modeling Example

2.3.1 Constant Velocity Layer

Figure 2 is a single refractor model with constant velocities separated by straight line boundaries. In this model, there also exists a few diffraction points. There are 55 source shots in all, and the shot points occur at every other station of a 48 channel split spread receiver configuration. The synthetic first break pick diagram for this model is shown in Figure 4. It is interesting to note that this simple model generates a very complicated pick

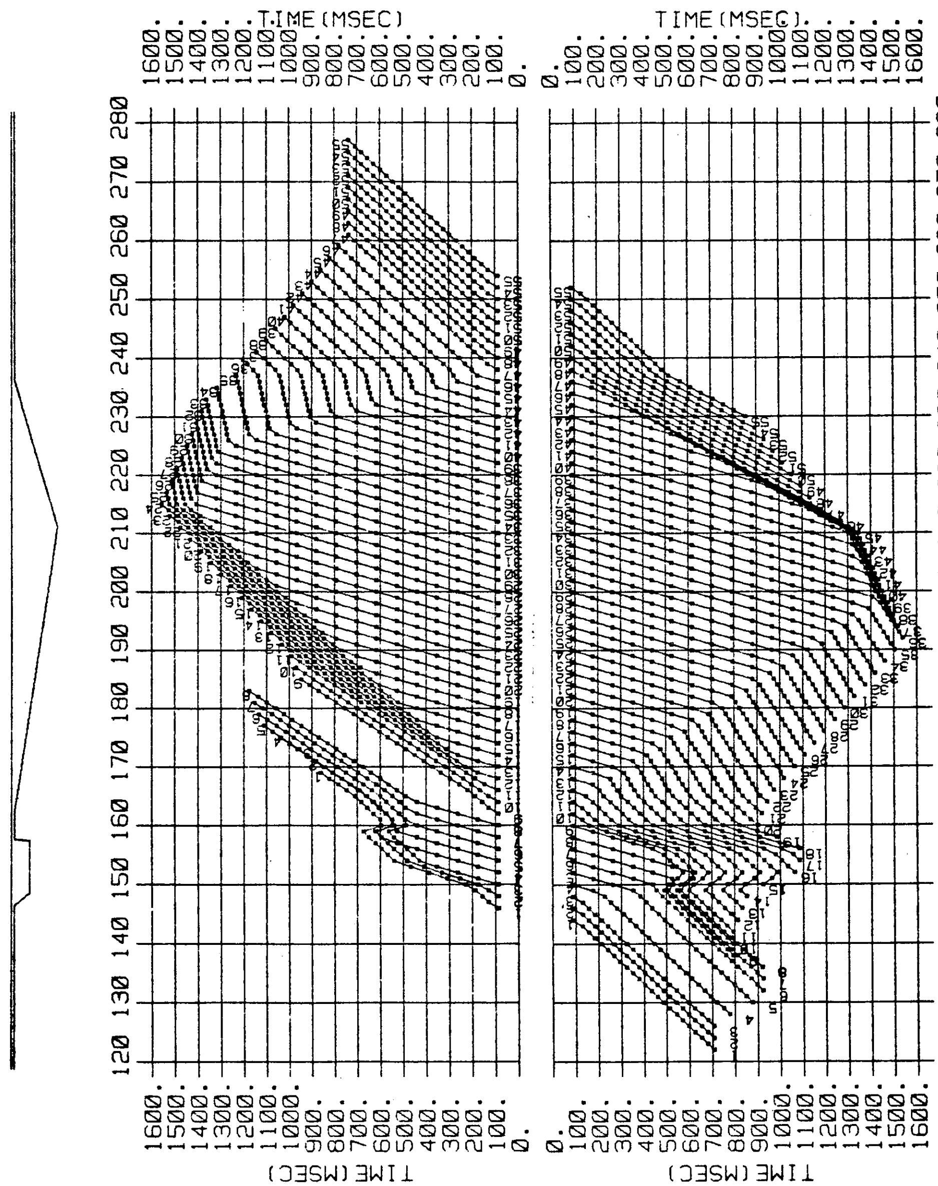


Figure 4 Synthetic first break picks diagram of the model in Figure 2

diagram. Even though the input consists of only a single interface separating two constant velocity media, the slopes of the travel time curves suggest that there exist many layers. The geometry of the interface is responsible for these changes in the slope of the travel time curves.

2.3.2 Linear Vertical and Lateral Velocity Changes Model

Figure 5 represents a two refractor model with linear vertical and linear lateral velocity changes. The response for this model is shown in Figure 6. From this time section, we can see that the velocity model generates curved first breaks. For this model there was only one source shot.

2.3.3 Input Velocity Function

The program can accept an input velocity function at arbitrary points along the boundary so that it can handle more complicated models such as fault structure (velocity discontinuity). Figure 7 is a two-refractor model with curved boundaries and input velocity function. The model response is shown in Figure 8. The ray paths in this model are approximately estimated by straight lines. The velocity between two given velocities is linearly interpolated. For

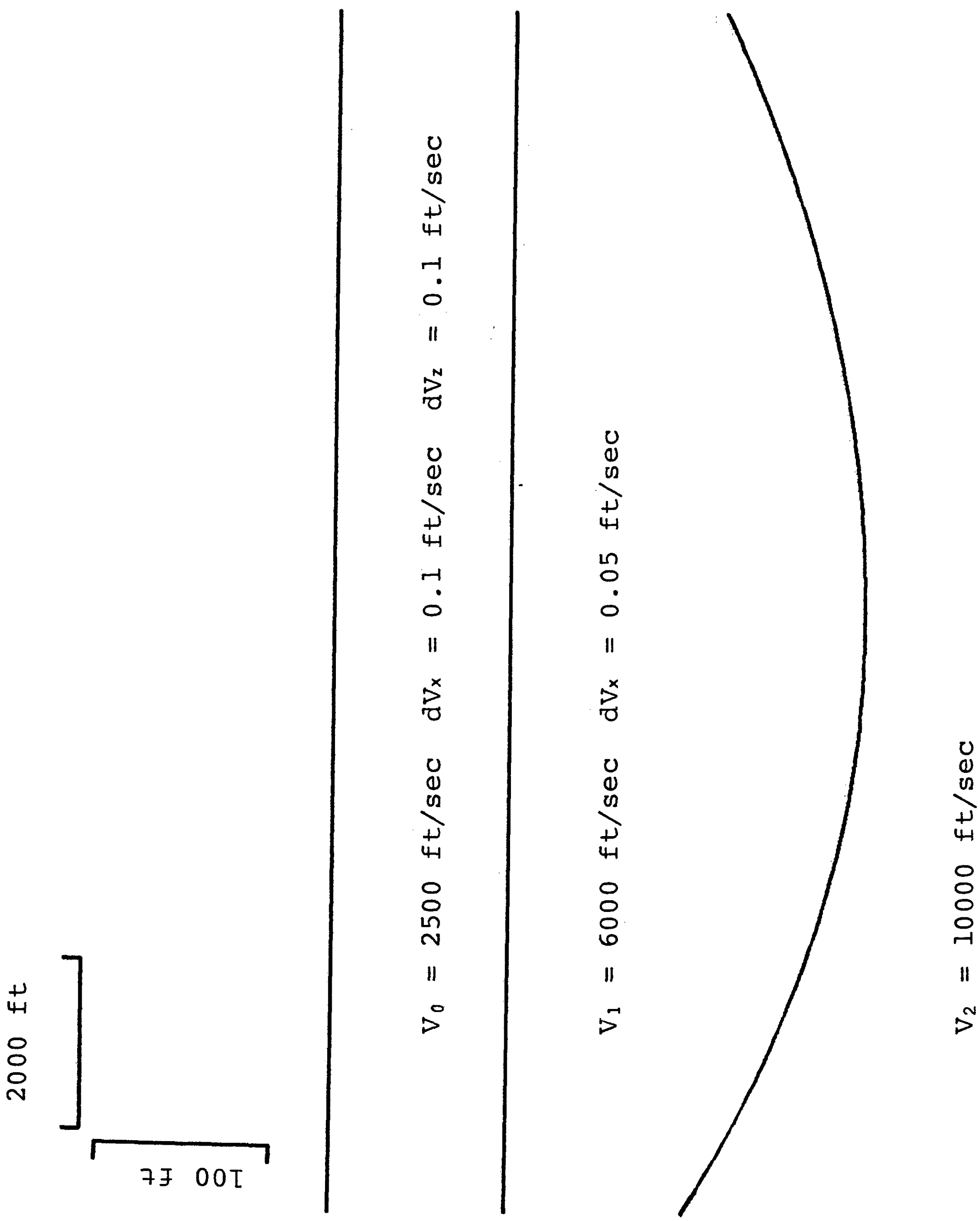


Figure 5 Two-refractor model with linear vertical and linear lateral velocity change

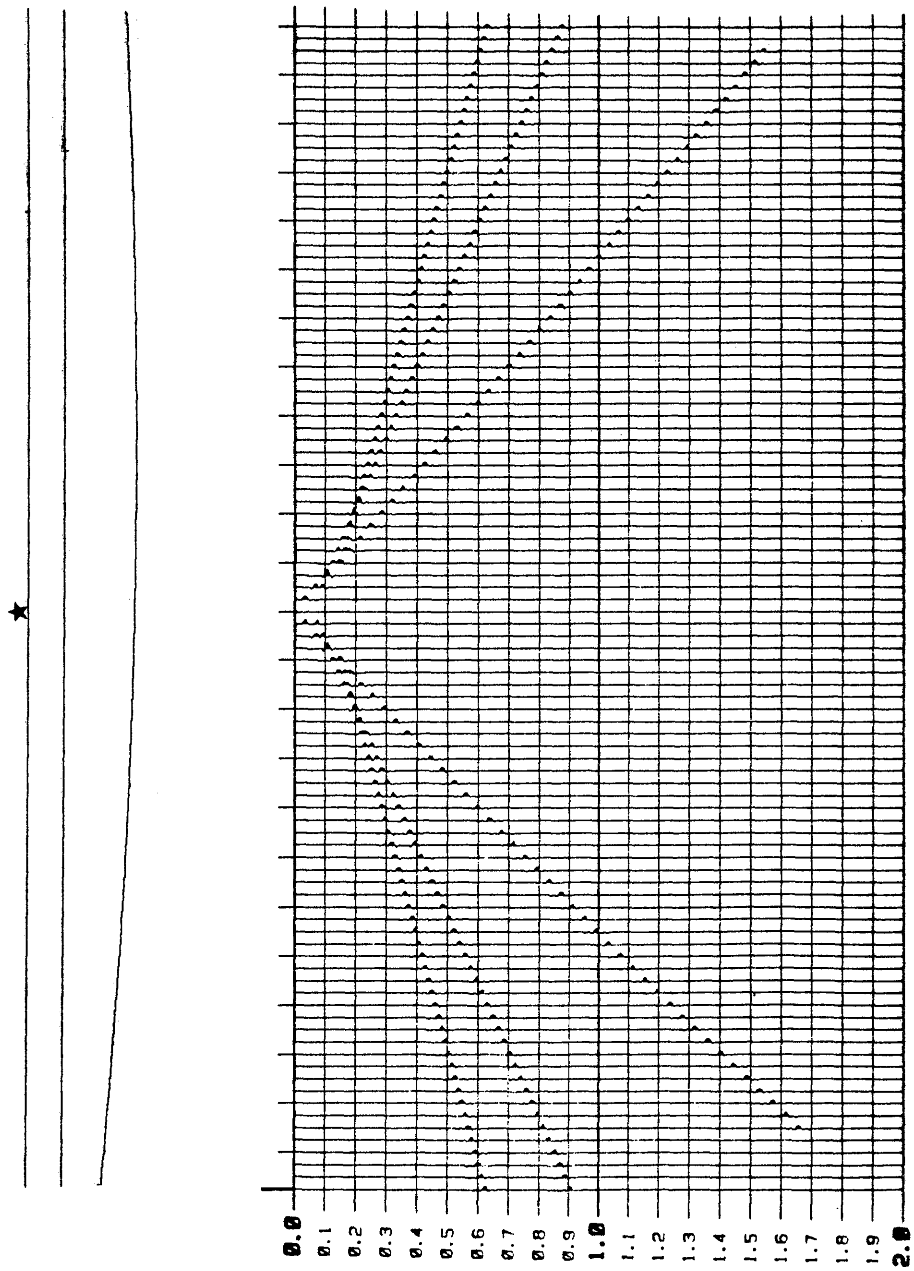


Figure 6 Model response for Figure 5 with velocity gradient

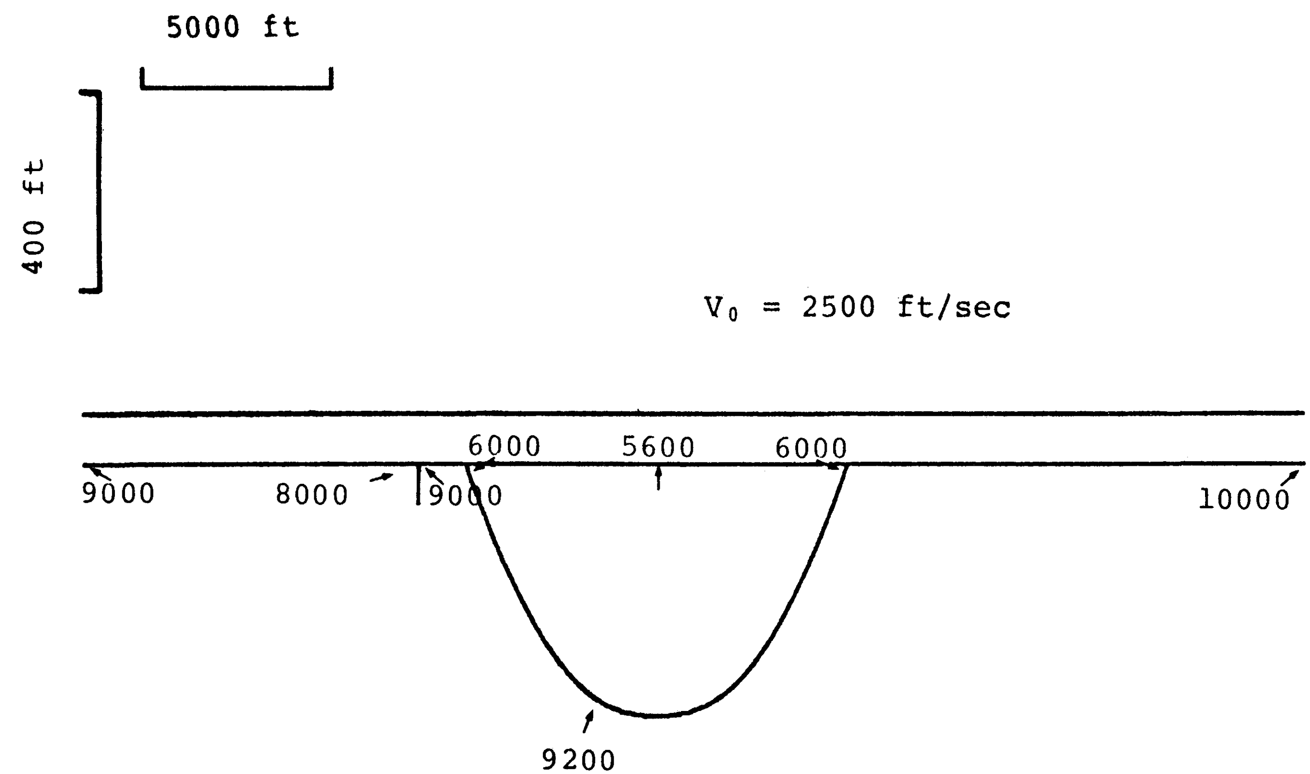


Figure 7 Two refractor model with curved boundaries and input velocity function

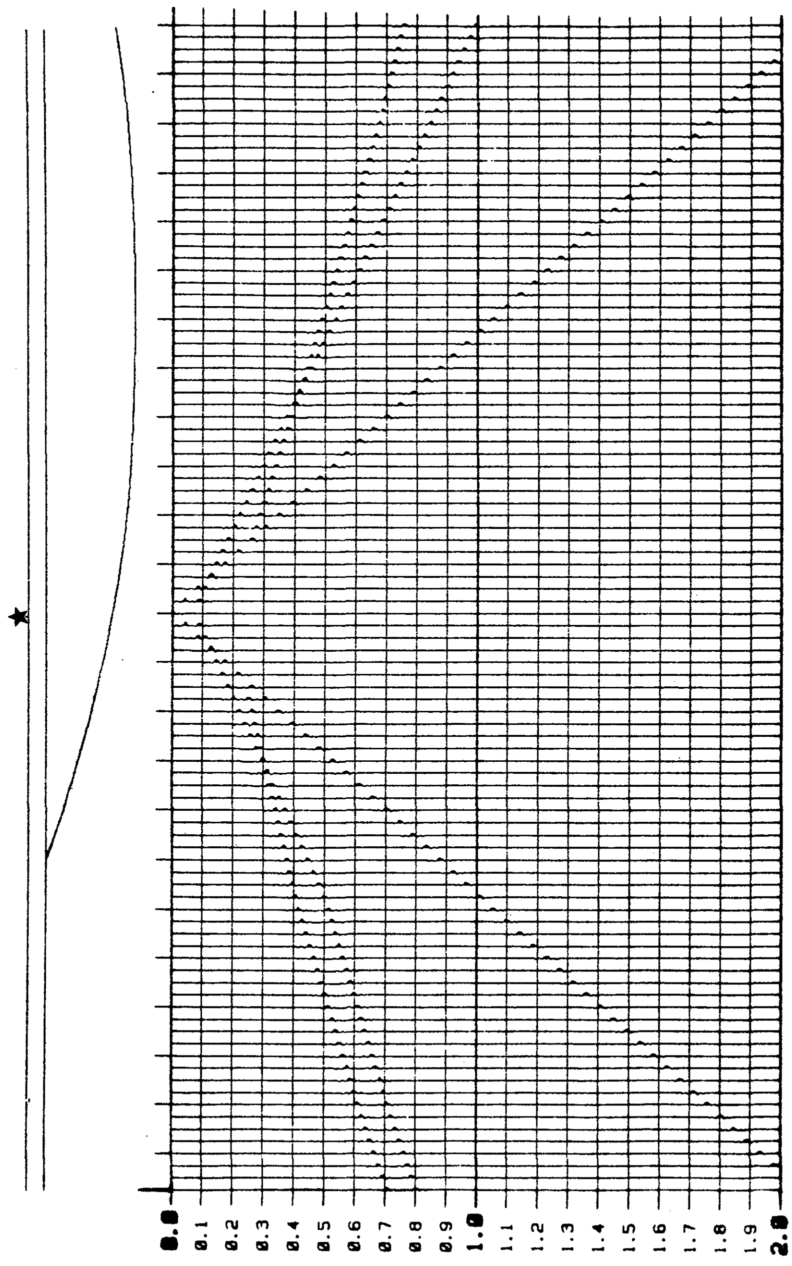


Figure 8   Model response for Figure 7

the bottom layer, velocity discontinuity can be handled. The linear vertical velocity change can also be applied. The travel times are calculated by integration along the ray path.

$$t = \frac{1}{k} \ln \left(\frac{V_0 + kx}{V_0} \right)$$

V_0 = velocity at starting point

x = distance between starting and ending points

k = linear velocity change rate between starting and ending points

CHAPTER 3

GENERALIZED LINEAR INVERSION

Geophysical inversion may be viewed as an attempt to automatically fit the response of an idealized subsurface earth model to a finite set of field observations. The models are mathematical in nature. These equations in turn depend on n parameters which are estimated from the actual data. The model response consists of the synthetic data produced by a particular realization of the model. The purpose of the inversion is to extract the model parameters estimated from a series of attempted fits of the model response to the observed data.

The theory and application of GLI have been well developed in past twenty years; (Backus and Gilbert, 1967; 1968; Wiggins, 1972; Aki and Richards, 1980).

3.1 Theory of Generalized Linear Inversion

For a measurable physical quantity C which is a function of P_i $(i = 1,2,....,n)$: i.e. $C = f(P_1, P_2,....,P_n)$.

$$\Delta C = C_{observation} - C_{calculation}$$

Applying the Taylor's expansion, ΔC_i can be express as

$$\Delta C_i = \frac{\partial C_i}{\partial P_1}\Delta P_1 + \frac{\partial C_i}{\partial P_2}\Delta P_2 + \ldots + \frac{\partial C_i}{\partial P_n}\Delta P_n + \text{higher order terms}$$

In tensor form, this relationship is given by

$$\Delta C_i = \frac{\partial C_i}{\partial P_j}\Delta P_j + \text{higher order terms}$$

or in matrix form

$$\begin{bmatrix} \frac{\partial C_1}{\partial P_1} & \frac{\partial C_1}{\partial P_2} & \cdots & \frac{\partial C_1}{\partial P_n} \\ \frac{\partial C_2}{\partial P_1} & & & \vdots \\ \vdots & & & \vdots \\ \frac{\partial C_m}{\partial P_1} & \cdots & \cdots & \frac{\partial C_m}{\partial P_n} \end{bmatrix} \begin{bmatrix} \Delta P_1 \\ \vdots \\ \vdots \\ \Delta P_n \end{bmatrix} = \begin{bmatrix} \Delta C_1 \\ \vdots \\ \vdots \\ \Delta C_m \end{bmatrix}$$

It is convenient to define the sensitivity matrix A with the element

$$A_{ij} = \frac{\partial C_i}{\partial P_j}$$

and simplify the above equation to the form

$$A\Delta P = \Delta C$$

where $\Delta P = P_{model} - P_{input}$

The objective is to determine the correction vector ΔP so as to find the parameters (P's) of the model. The GLI method is used to solve this over-determined problem and the singular value decomposition method is used to find the inverse matrix of the non-square matrix A (A is an m x n matrix).

A symmetric (m + n) x (m + n) matrix A' is defined as:

$$A' = \begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} .$$

This assures the existence of an orthogonal set of eigen-vectors w_i $(i = 1,2,....,m + n)$ with eigenvalues λ_i: i.e.

$$A' \, w_i = \lambda_i \, w_i \, .$$

w_i can be devided into two parts, u_i and v_i, and the above equation can be written by:

$$\begin{pmatrix} 0 & A \\ A^T & 0 \end{pmatrix} \begin{pmatrix} u_i \\ v_i \end{pmatrix} = \lambda_i \begin{pmatrix} u_i \\ v_i \end{pmatrix}$$

or $A \, v_i = \lambda_i \, u_i$

$$A^T \, u_i = \lambda_i \, v_i \, .$$

It is easy to get the following relationships:

$$A^T A v_i = \lambda_i^2 v_i$$

$$A A^T u_i = \lambda_i^2 u_i .$$

Each of u_i and v_i forms an orthogonal set of eigenvectors with real eigenvalues.

It is convinient to define matrix U which contains all the eigenvectors u_i associated with the column and V which contains all the eigenvectors v_i associated with the row. The matrix Λ which is a diagonal matrix of all the non-zero eigenvalues λ_i. If there is no zero eigenvalue in the system, the above equations can be concluded to:

$$A V = U \Lambda .$$

Since the eigenvectors are orthogonal,

$$A = U \Lambda V^T .$$

If there are zero eigenvalues, the rank of matrix A is reduced to k (k < n). The matrix A can be still decomposed to the following form: (Aki and Richards, 1980; Wiggins, 1972; Nobel, 1977)

$$A_{m \times n} = U_{m \times k} \Lambda_{k \times k} V^T_{k \times n}$$

where $U_{m \times k}$ contains k eigenvectors u_i of length m related to the non-zero eigenvalues and $V_{n \times k}$ contains k eigenvectors

v_i of length n related to the non-zero eigenvalues. Thus, ΔP can be obtained by:

$$\Delta P = V\Lambda^{-1}U^{T}\Delta C.$$

For the GLI method to work well, the number of the nonvanishing eigenvalues of the sensitivity matrix shall not be less than n. This implies that all the parameters are linearly independent and that a unique solution can be obtained.

Actually, the GLI method seeks to minimize the norm of the vector

$$A \quad \Delta P = \Delta C.$$

The least square problem always has a solution; the solution is unique if and only if the rank of the m x n matrix A equals n. In other words, if k equals n.

For $k < n$, there are zero eigenvalues in the system and the solution is not unique. The least square solution takes the average over some parameters. The resolution matrix VV^T will be examined to find how the parameters can be solved (Aki, 1980; Wiggins, 1972). The row vectors of VV^T form weighting coefficients and the generalized inversion corrections are expressed as a weighted average of the true model corrections.

Another very important requirement for GLI method is the constraint that the problem be approximately linear near the solution point. Namely, when the higher order terms in above Taylor's expansion are ignored, and there are no singularities or local minima near the solution.

3.2 The GLI Algorithm

3.2.1 Parameter Determination

The first step in GLI algorithm is to determine both the number of independent unknowns which exist in the problem and the number of unknowns which need to be solved for as free parameters. All the other variables in the problem are assumed to be fixed. It is very important to provide a judicious choice of input parameters to make the inversion process both efficient and unique.

3.2.2 Initial Guess

The initial guess model is created by the user. Sometimes a bad initial guess can give a divergent inversion result, or give an unwanted result at some other local root mean square (RMS) error minimum in the model parameter space. The initial guess generally depends on the character of the problem and should be carefully chosen.

3.2.3 Sensitivity Matrix and Difference Matrix Determination

The model response is calculated from the input model parameters, and the difference matrix is obtained by subtracting the calculated values from the observed values. The sensitivity matrix can be obtained analytically by derivative; but, in many cases, the numerical perturbation methods must be used. A small perturbation of input parameter will change the model response. The element A_{ij} in the sensitivity matrix A can be calculated numerically by the ratio of ΔC_i (the difference between the original response and perturbed response at ith observation for the change of jth parameter) and ΔP_j (perturbed amount of jth parameter).

3.2.4 Singular Value Decomposition

The sensitivity matrix A is normalized before decomposition to improve the resolution of each parameter. Next A is decomposed into the product of three matrices as mentioned above.

The rank of the sensitivity matrix is the number of non-zero eigenvalues, and it is initially equal to n. Actually the singular decomposition method will fail when zero eigenvalues exist, but this generally does not happen in real case if the parameters are chosen properly. For many problems where m is much greater than n, the eigenvalues approach zero exponentially which makes it difficult to distinguish between very small eigenvalues and zero eigenvalues (Wiggins, 1972). Thus, it is very common to face with the task of selecting some cutoff eigenvalue λ_{cutoff} and ignoring all eigenvalues below that value. If small eigenvalues exist, the number of the free parameters should decrease to make the inversion process stable. From the resolution matrix, which is the product of VV^T, to see the resolution for each parameter. The resolution matrix is no longer an identity matrix if k is less than n. The data can determine only a weighted average over certain related parameters. The matrix with the best resolving power is nearly always the vector with the maximum diagonal element.

3.2.5 Correction Determination

Understanding the matrix manipulations in section 3.1, it is not difficult to find the corrections. From the physical laws and some knowledge about the problem, it is easy to set a few constraints for the corrections to make sure these corrections are reasonable. (e.g. the thickness of the layer should be greater than zero, the layer velocity should be positive, etc.)

3.2.6 Result Examination

The error of the corrected model will be examined to determine the direction of the inversion process. To generate a better solution, the loop from 3.2.2 to 3.2.6 is repeated until the errror is small. When the error is small or when it reaches a minimum, the process stops, and the final model is obtained.

CHAPTER 4

REFRACTION MODELING AND INVERSION SYSTEM

The algorithm for the refraction modeling and inversion is shown in Figure 9 and this discussion will deal with the lower part (below the dash line) of the flow chart. The first break times are assumed to be well-picked, although the method can also handle noisy data as well (an example is given later). A second assumption supposes an initial guess, created via outputs of other refraction interpretation methods or starting from a constant thickness weathering layer.

The goal of the inversion is to find the subsurface structure and obtain the static corrections from the elevation and the thickness of the weathering layer. The model parameters to be perturbed are chosen to be the thickness of the weathering layer and the velocity of the refractor. An equal distance interval or arbitrary chosen x coordinates can be used to sample the geometry control points. The thickness of the weathering layer at each specified geometry control point and the velocity of the refractor are input as initial guess parameters.

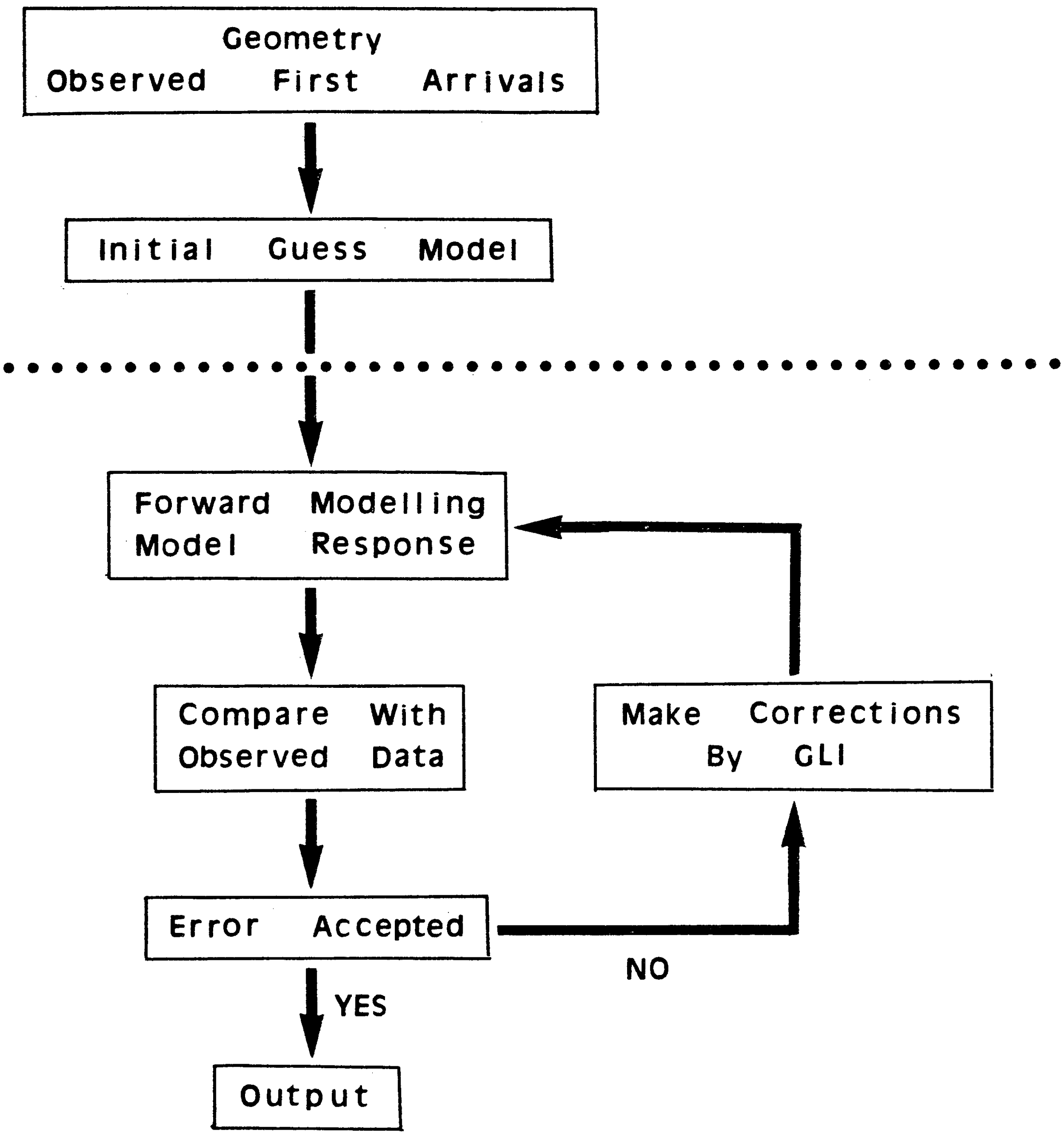


Figure 9 The algorithm of the refraction Modelling-Inversion System

As shown in chapter 3, the GLI corrections are made by the product of the difference matrix (the difference between the field observations and the input model responses) and the inverse sensitivity matrix. There are a few constraints on the model parameters: 1) the thickness of the layer should be greater than zero; 2) the velocity of the refractor should be greater than the velocity of the weathering layer in order to get the refracted response. These constraints for the model parameters are applied in every iteration to make sure the corrections are reasonable. If the RMS error is small enough or reaches a stable minimum, the GLI loop is finished and the best-fit earth model is obtained.

4.1 Simple Synthetic Model Tests

In this section, two models with constant layer velocities will be tested. The weathering layer velocity and the first break picks of two reciprocal shots which cover the region of interest are assumed to be known. Initial guesses were made far from the true model.

4.1.1 Model with Straight Line Boundaries

The original model and the flat initial guess are shown in Figure 10. There are three depth parameters D_1, D_2, D_3 and one velocity parameter V_1 to be solved for. The iteration results for all steps are listed in Table 1. The original model can be recovered in four iterations. The eigenvalues in this system are 3.6, 1.6, 0.84 and 0.57, all of them are significant and well behaved ($\lambda_{max}/\lambda_{min}=6.3$). There is no cutoff eigenvalue and a unique solution for this problem can be obtained. Figure 11 shows the convergence of the errors between observed and calculated first breaks for this model. The output model and the original model are shown in Figure 12.

4.1.2 Model with Curved Boundary

The original model and flat initial guess are shown in Figure 13. There are five parameters this time, four for depth and one for velocity. The output for each iteration is shown in Table 2. In this example, the original model can be recovered in four iterations and five significant eigenvalues within the range 3.1 to 0.35 are achieved. Figure 14 shows the convergence of the errors for this example, and Figure 15 shown the output and the original models.

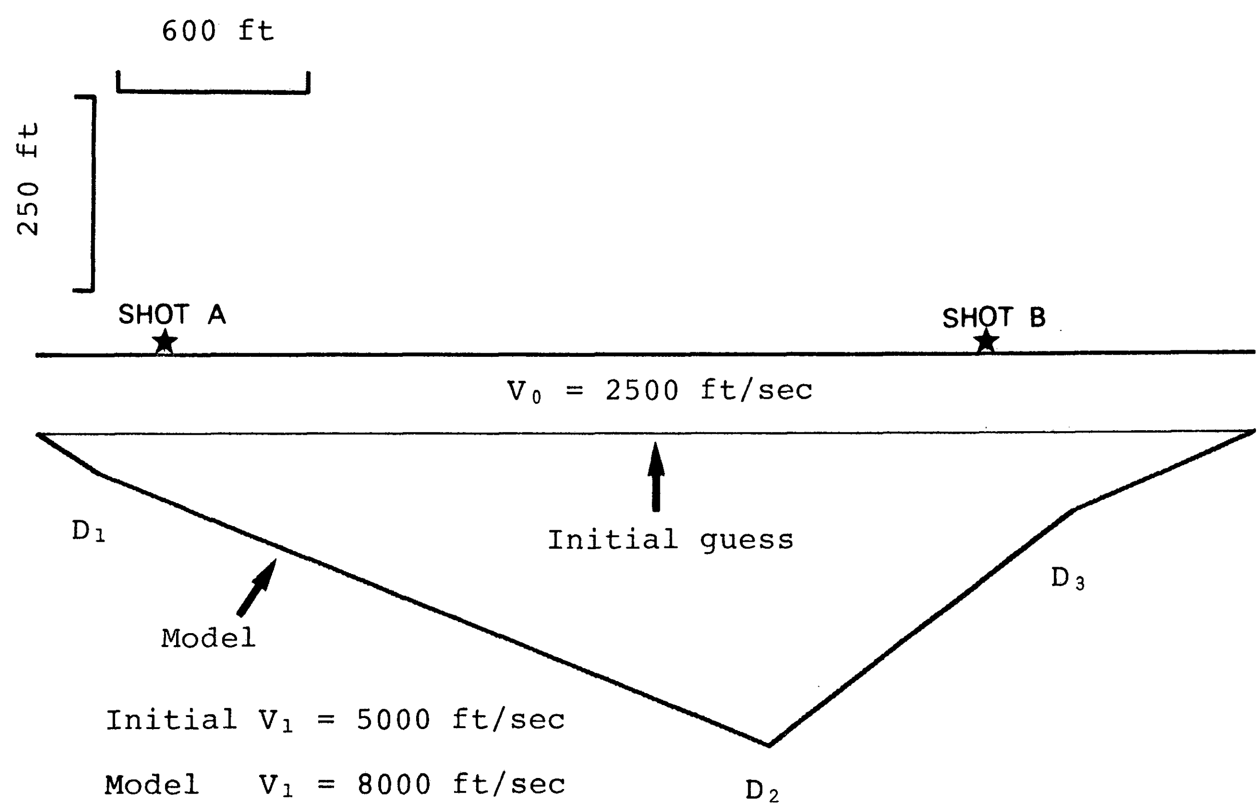


Figure 10 Initial guess and original model for straight line boundary case

	V_1	D_1	D_2	D_3
INITIAL GUESS	5000	100	100	100
ITERATION 1	6079	97	466	38
ITERATION 2	7360	137	498	180
ITERATION 3	7954	150	500	201
ITERATION 4	8000	150	500	200
MODEL	8000	150	500	200

Table 1 Straight line boundary inversion

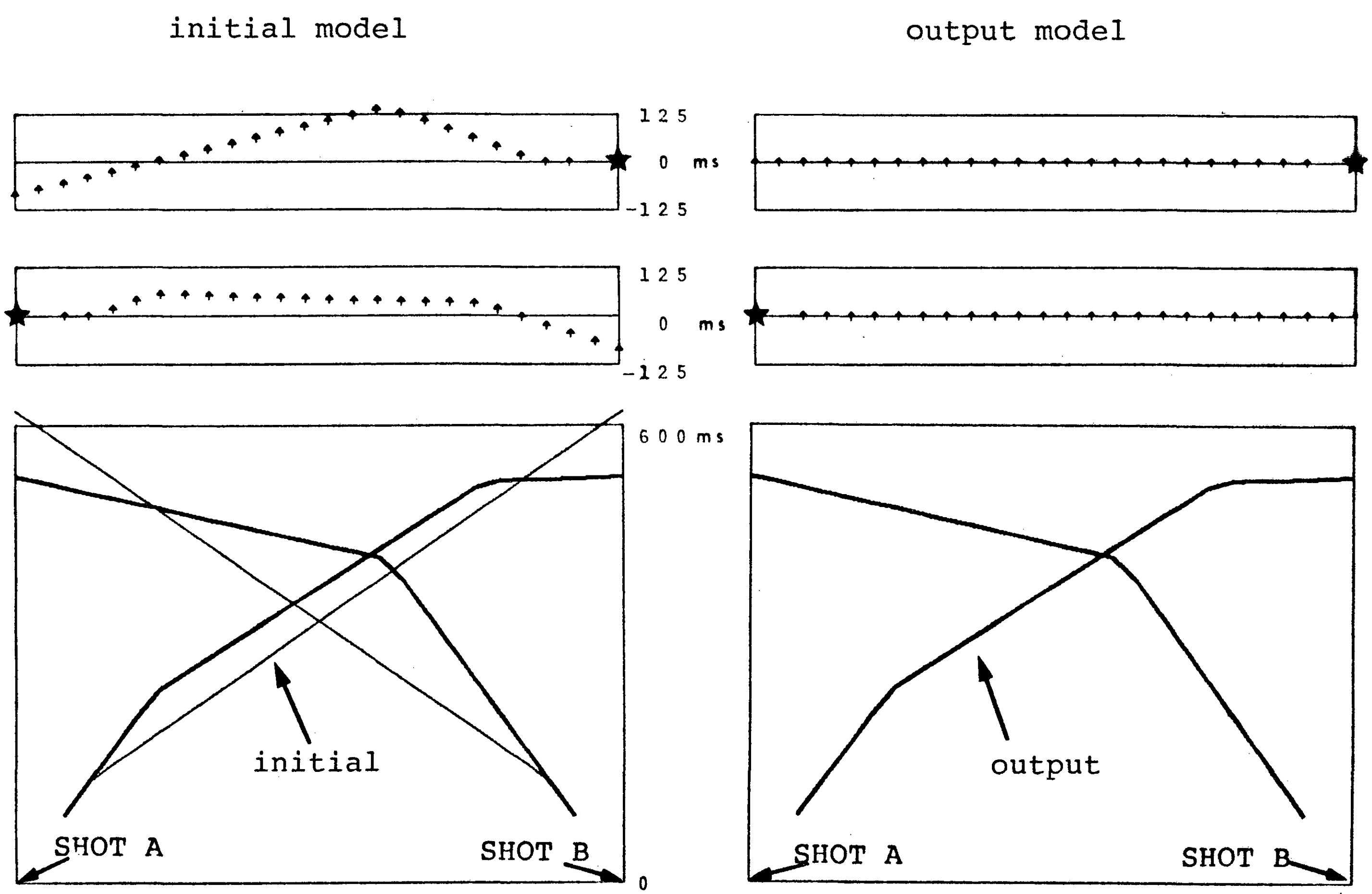


Figure 11 Travel time curves (lower part) and residual errors (upper part) for straight line boundary example

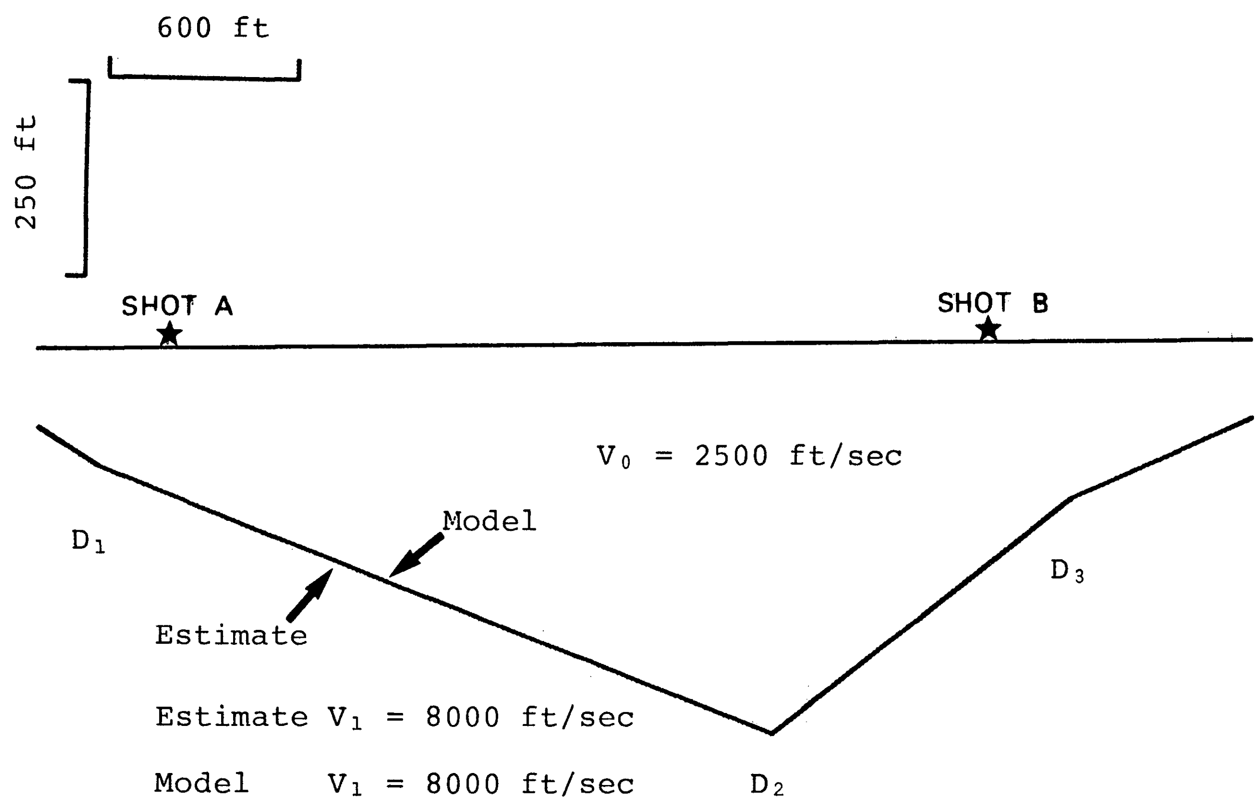


Figure 12. Output and original models for straight boundary case

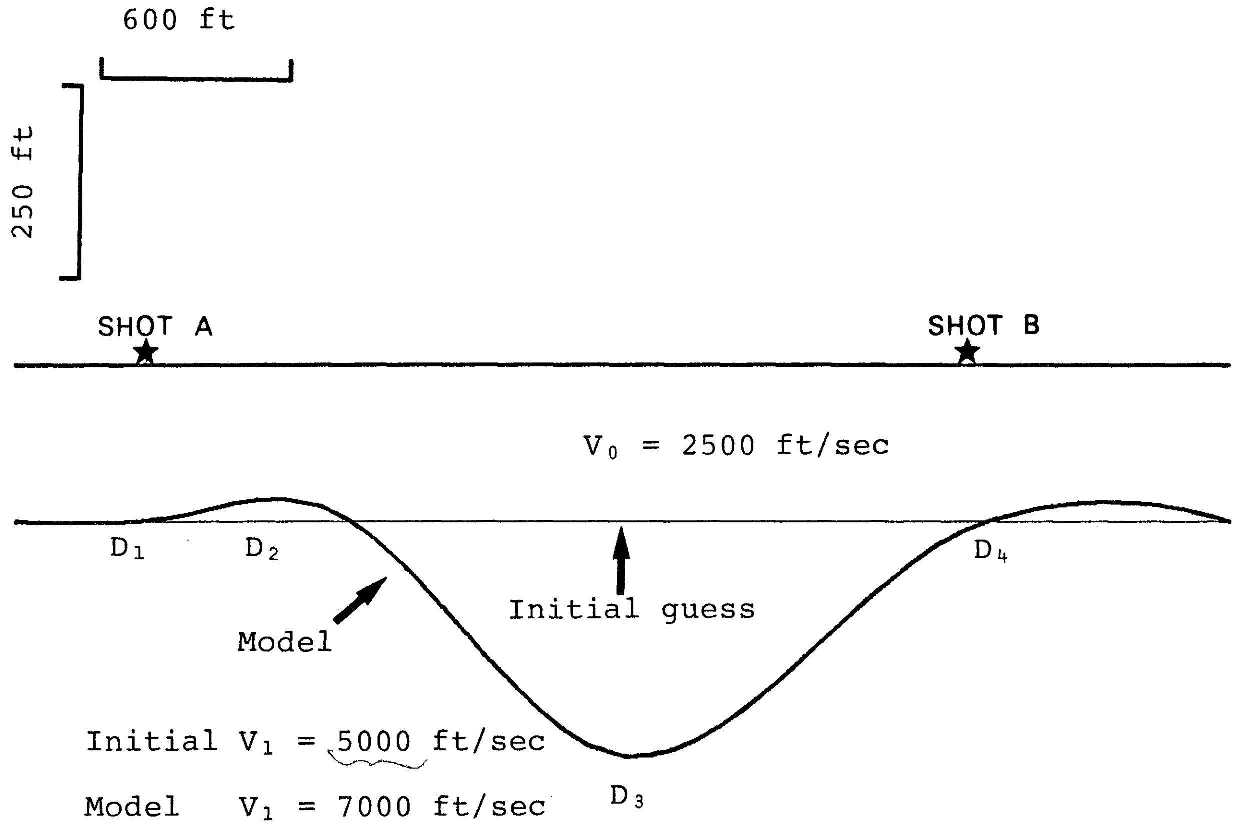


Figure 13 Initial guess and original model for curved boundary case

	V_1	D_1	D_2	D_3	D_4
INITIAL GUESS	5000	200	200	200	200
ITERATION 1	6091	171	155	496	155
ITERATION 2	6833	197	180	498	194
ITERATION 3	6996	200	180	500	200
ITERATION 4	7000	200	180	500	200
MODEL	7000	200	180	500	200

Table 2 Curved boundary inversion

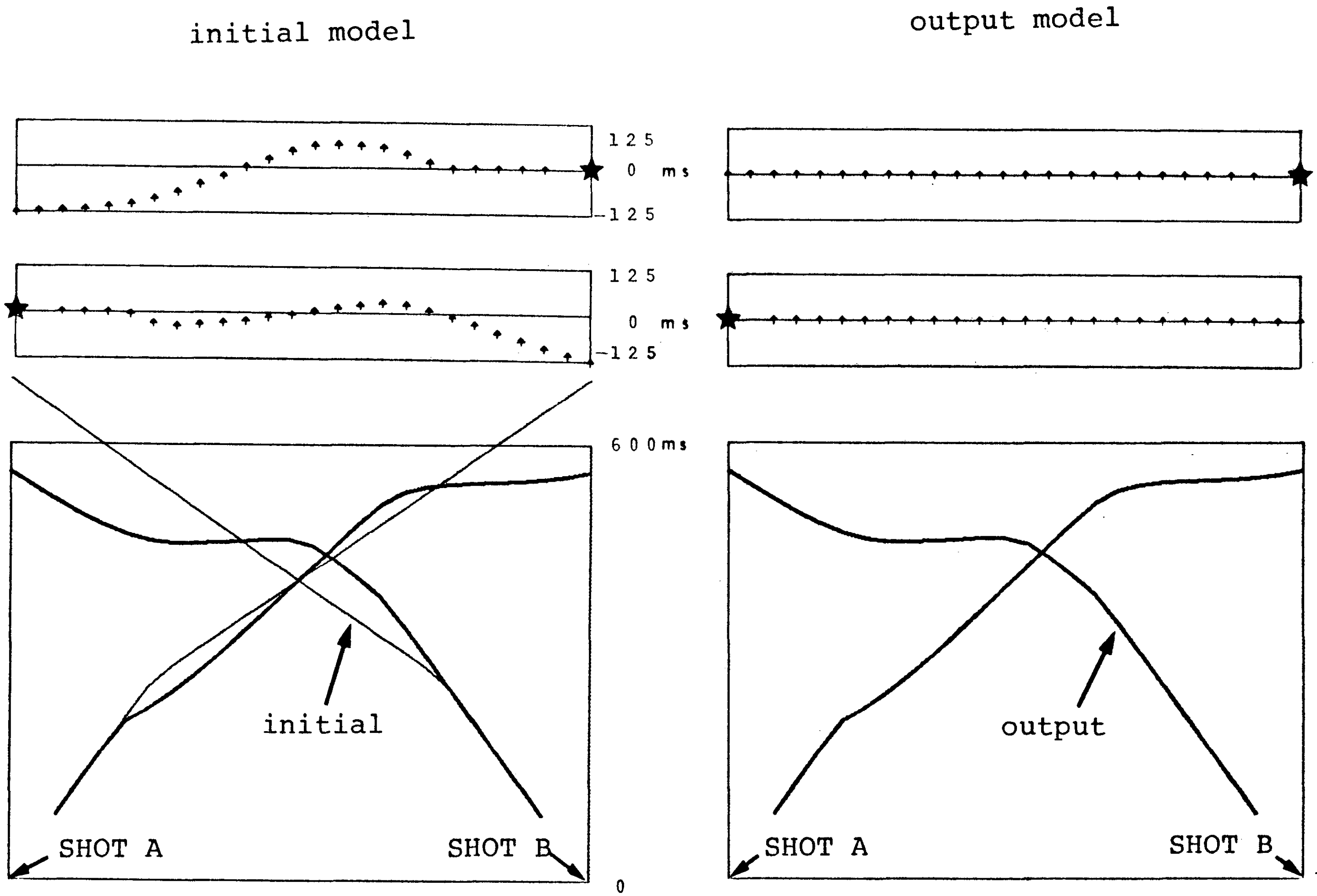


Figure 14 Travel time curves (lower part) and residual errors (upper part) for curved boundary example

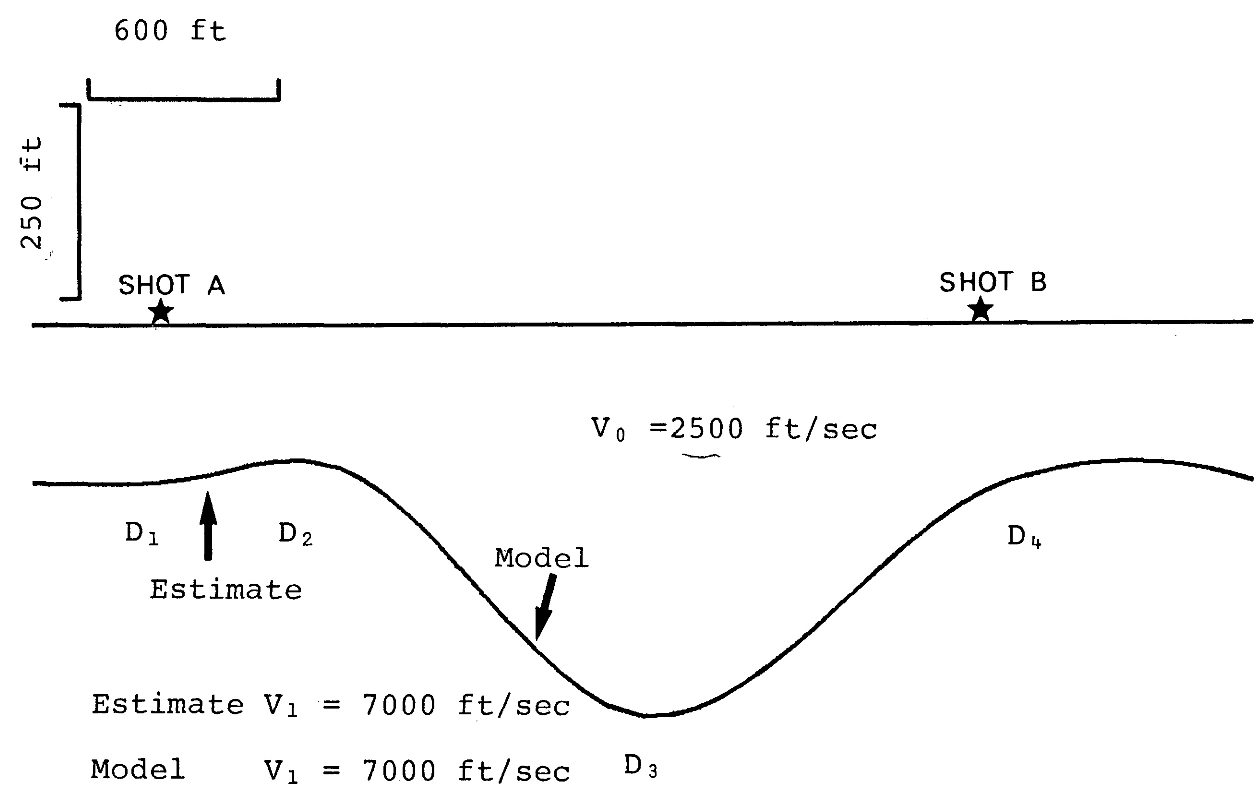


Figure 15 Output and original models for curved boundary case

4.1.3 Noisy Case

The previous two examples are noise free. However, there is some noise in real geophysical problems, and it is important to test the inversion process for stability in the noisy case. An addition of 20 ms. uniform random noise was applied to the second model response and both the clean and noisy travel time curves are shown in Figure 16. Using the same initial guess as in 4.1.2, the initial model is not exactly recovered after four iterations but the process is stable and the answer is acceptable. Table 3 shows the results for each iteration. Figure 17 shows the convergence of the errors for this example. The output model and original model are shown in Figure 18. The first breaks of the final output model and the original noisy first breaks are shown in Figure 19.

4.2 Practical Application

The above synthetic tests give confidence that the inversion algorithm should work well for real data sets. A VAX 11/780 module was developed to allow the implementation of the algorithm as part of the processing flow on actual reflection seismic data.

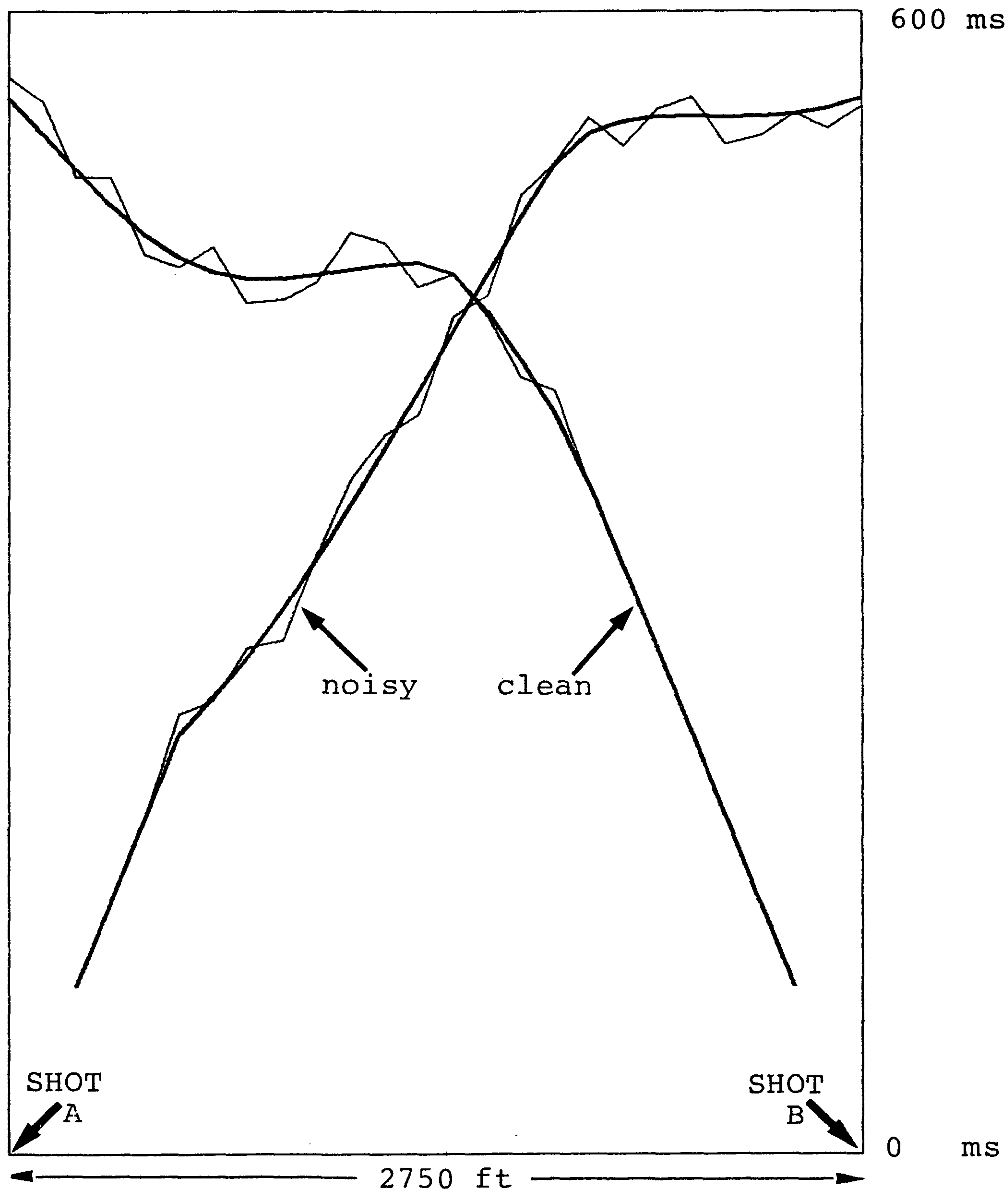


Figure 16 Curved boundary model response with and without noise

	V_1	D_1	D_2	D_3	D_4
INITIAL GUESS	5000	200	200	200	200
ITERATION 1	6216	184	164	508	154
ITERATION 2	7028	210	191	509	190
ITERATION 3	7203	211	189	512	196
ITERATION 4	7209	211	189	512	196
MODEL	7000	200	180	500	200

Table 3 Noisy case inversion

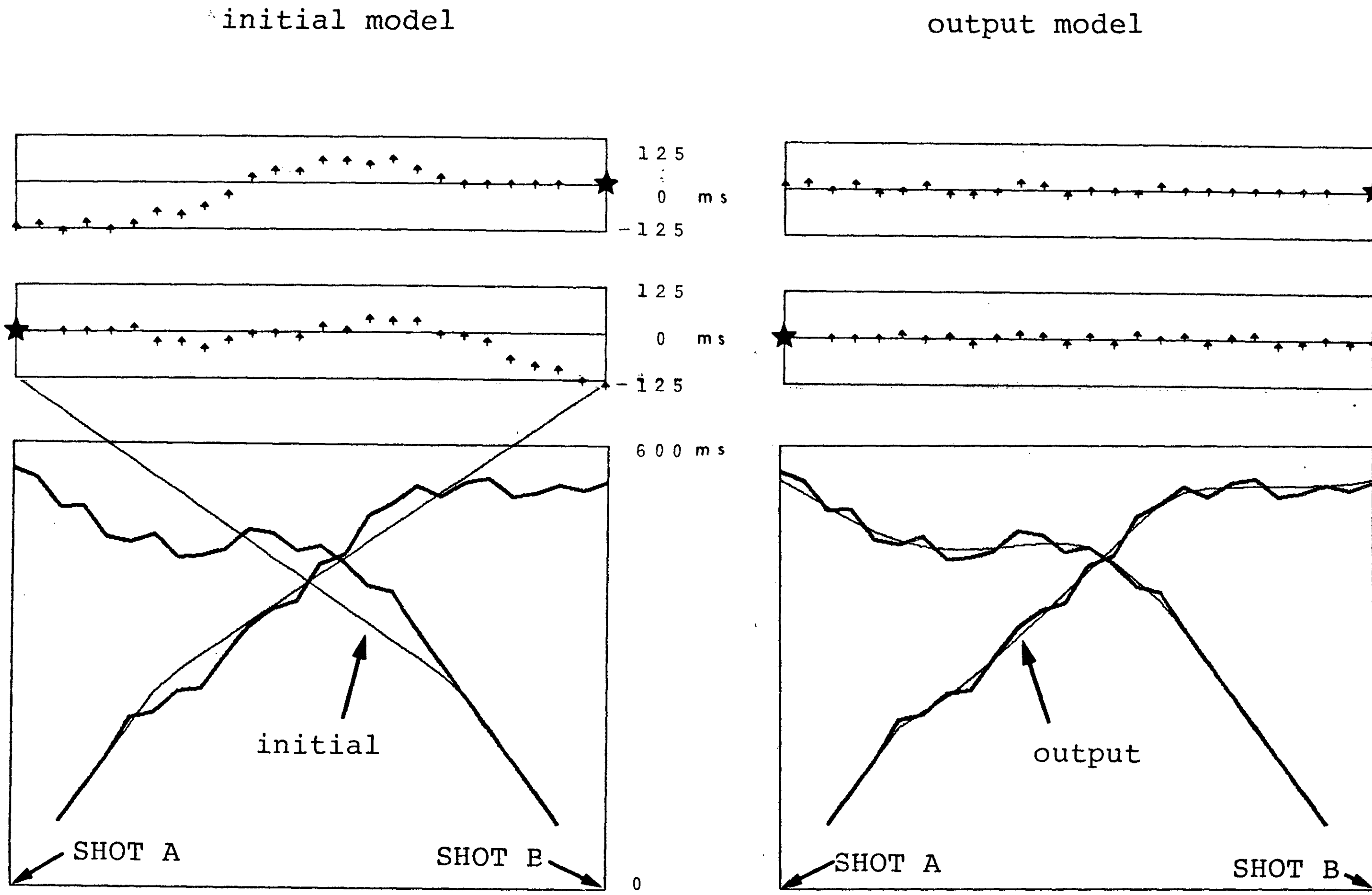


Figure 17 Travel time curves (lower part) and residual errors (upper part) for noisy case example

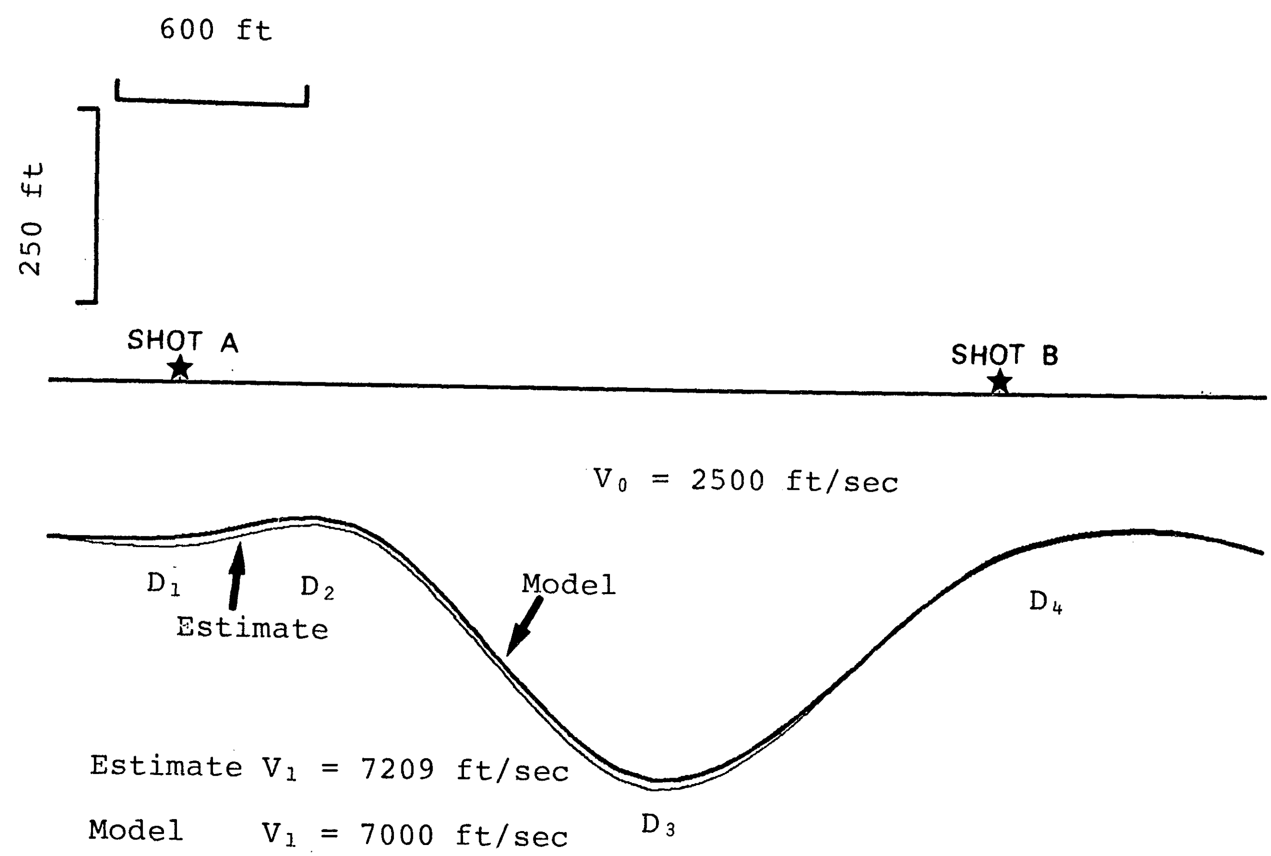


Figure 18 Output and original models for noisy data inversion.

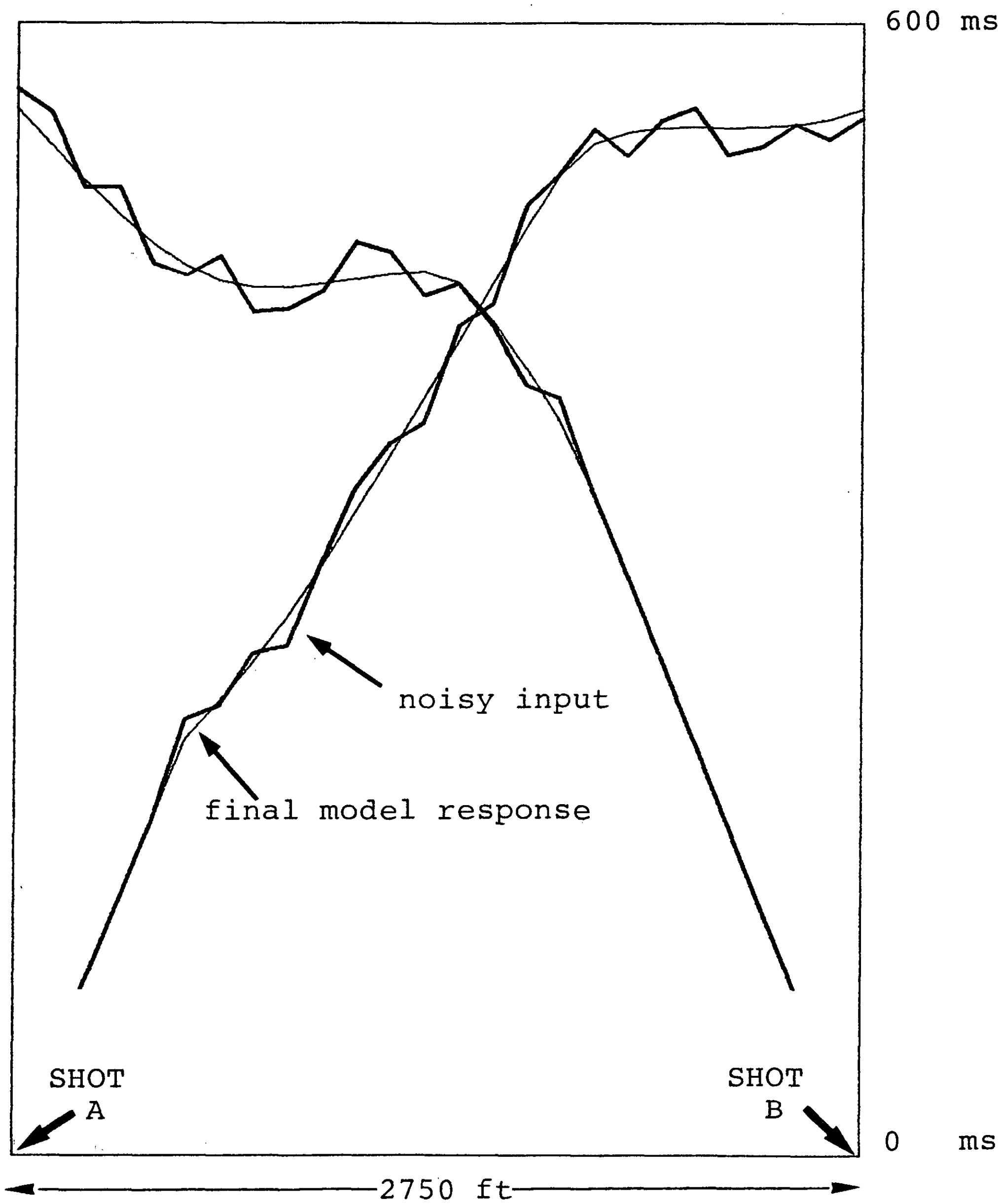


Figure 19 First breaks of the final ouput model and the original noisy first breaks

All the examples shown above were obtained by using a single pair of reciprocal shot records. However, for standard reflection shooting, there are many pairs of reciprocal shots. All reciprocal shots, within the whole section supply the information about the solution of the structure under the entire line. A subset of them is chosen to feed the inversion program and this over-determined problem can be solved by minimizing the least square error.

4.2.1 Calculation of the Sensitivity Matrix

The method shown in section 3.2.2 may be used to calculated the sensitivity matrix numerically. For standard seismic line, a large number of shots and iterations are required to solve the inversion problem. For a curved boundary model, the modeling process uses a searching type algorithm rather than an analytical solution to obtain the critical ray. This greatly increases the computer time required to get the sensitivity matrix. The computer time necessary for such calculations may not be practical on VAX class machines. If instead the curved boundary segment between the geometry control points is replaced by the straight line, the travel time differences from perturbations of the model parameters can be calculated analytically and the elements in the sensitivity matrix can be obtained in an efficient manner.

For a single refractor case, a depth change scheme is shown in Figure 20 and the velocity change scheme is shown in Figure 21. In Figure 20, (gx_1, gz_1) and (gx_2, gz_2) are two consecutive geometry control points and (x_0, z_0) are the coordinates of the surface station. When the control point (gx_1, gz_1) is moved up a small distance Δz, the emergence angle of the critical ray will change from θ_1 to θ_2, and the critical point will move from (x_1, z_1) to (x_2, z_2). The critical angle is expressed as θ_{cr}. The partial derivative at ith observation corresponding to the jth geometry control depth change can be calculated analytically through the following procedure:

$$\theta_{cr} = \sin^{-1} \left(\frac{V_0}{V_1} \right)$$

$$S_1 = \tan^{-1} ((gz_2-gz_1)/(gx_2-gx_1))$$

$$S_2 = \tan^{-1} ((gz_2-gz_1-1)/(gx_2-gx_1))$$

$$\theta_1 = \pm\theta_{cr} - S_1$$

$$\theta_2 = \pm\theta_{cr} - S_2$$

$$x_1 = (z_0-gz_2+\tan S_1 \, gx_2-\cot \theta_1 \, x_0) \, / \, (\tan S_1 - \cot \theta_1)$$

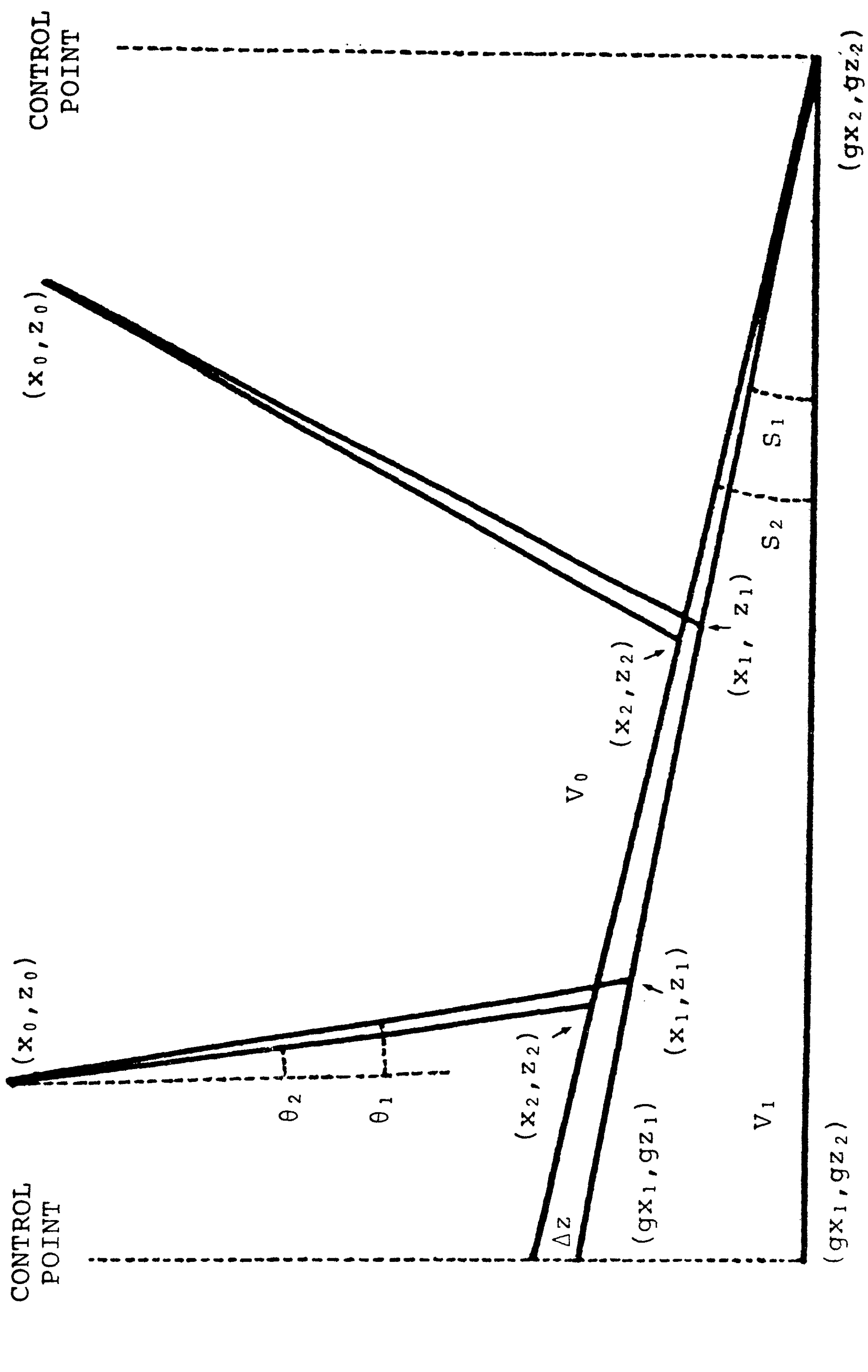


Figure 20 Using linear approximation to obtain the patial derivative in the sensitivity matrix related to the depth change Δz

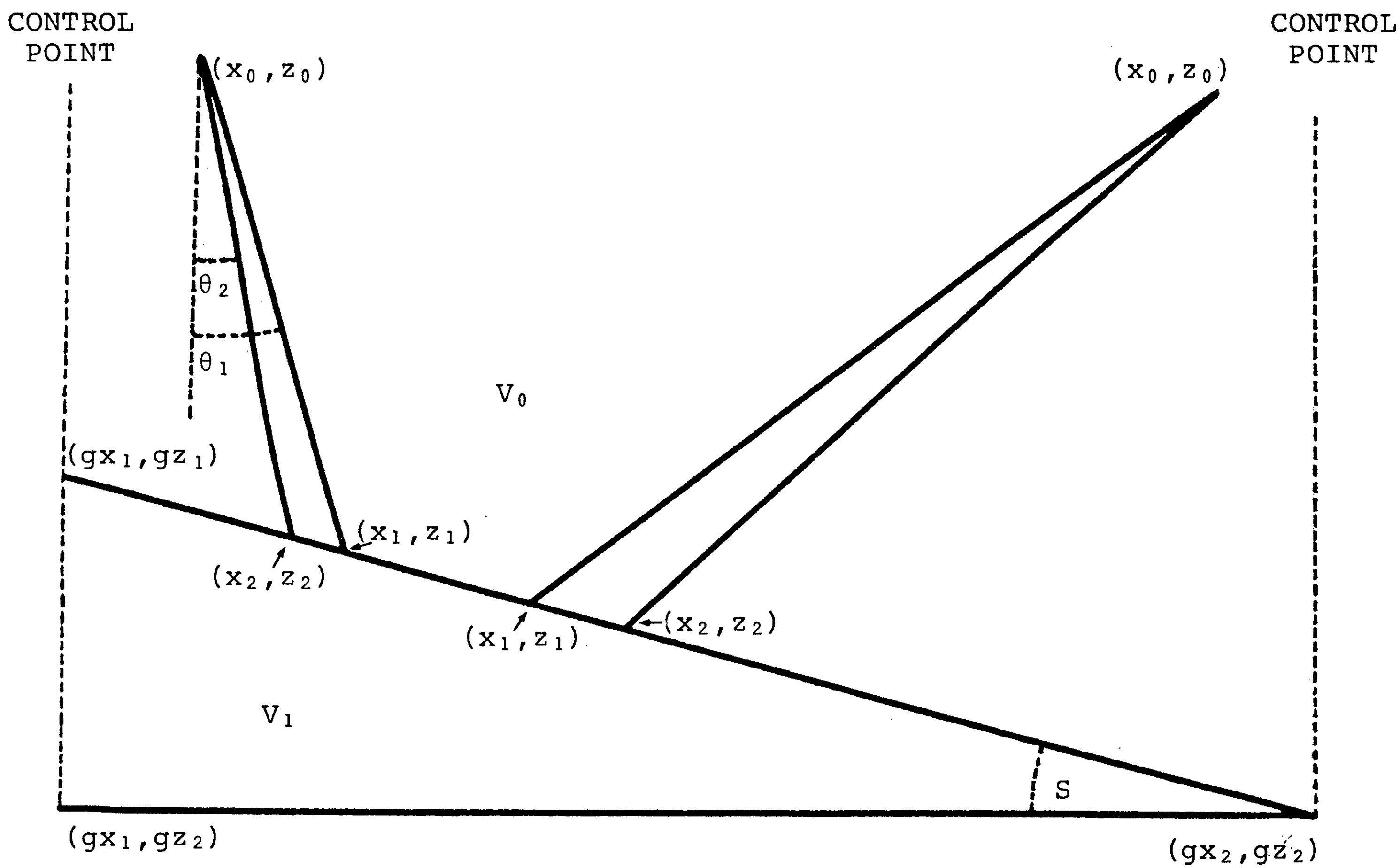


Figure 21 Using linear approximation to obtain the partial derivative in the sensitivity matrix related to the velocity change ΔV

$$\Delta T_d = \{\sqrt{(x_1-gx_2)^2+(z_1-gz_2)^2} - \sqrt{(x_2-gx_2)^2+(z_2-gz_2)^2}\} /V_1.$$

the total time difference

$$\Delta T_t = \Delta T_u + \Delta T_d$$

$$\frac{\partial C_i}{\partial V_1} = \Delta T_{ti} / \Delta V_1.$$

The corresponding elements in the sensitivity matrices were compared after forward modeling process and linear approximation. The differences between them were very small, typically less than 0.1 percent for the models examined.

4.2.2 Synthetic Model Test

For real data sets, it is common to define constant distance intervals for sampling of the geometry control points. The number of control points is chosen by the frequency of the structure. For a high frequency strucutre, more sample points are needed to describe the structure well, typically two or more per wavelength. In other words, shots and receivers may be skipped to reduce the computation effort, but at the expense of short wavelength statics.

A single refractor model with undulated surface and long wavelength syncline subsurface structure is shown in Figure 22. There are twenty-three geometry control points across the surface but only three geometry control points across the refractor. An 18-station interval is chosen for geometry control points, while a 6-station interval for shot points and a 6-receiver interval for first break picks being used. The initial guess from a delay-time refraction method is shown in Figure 23 and a quality control (QC) plot which represents the error of the input model response from the theoretical response is shown in Figure 24. The bounds for the QC plot are limited to ±50 ms, in case the error is too large. There are nineteen depth parameters and one velocity parameter. The weathering layer thickness at the model edge is fixed to avoid instability because sometimes depth parameters at the edges are not used in the forward modeling process. After three interations, the central part is correctly recovered and only small errors are evident around the edge. The output model and final QC plot are shown in Figure 25 and Figure 26. This scheme has proved to be an accurate and efficient approach for detecting low frequency weathering structure.

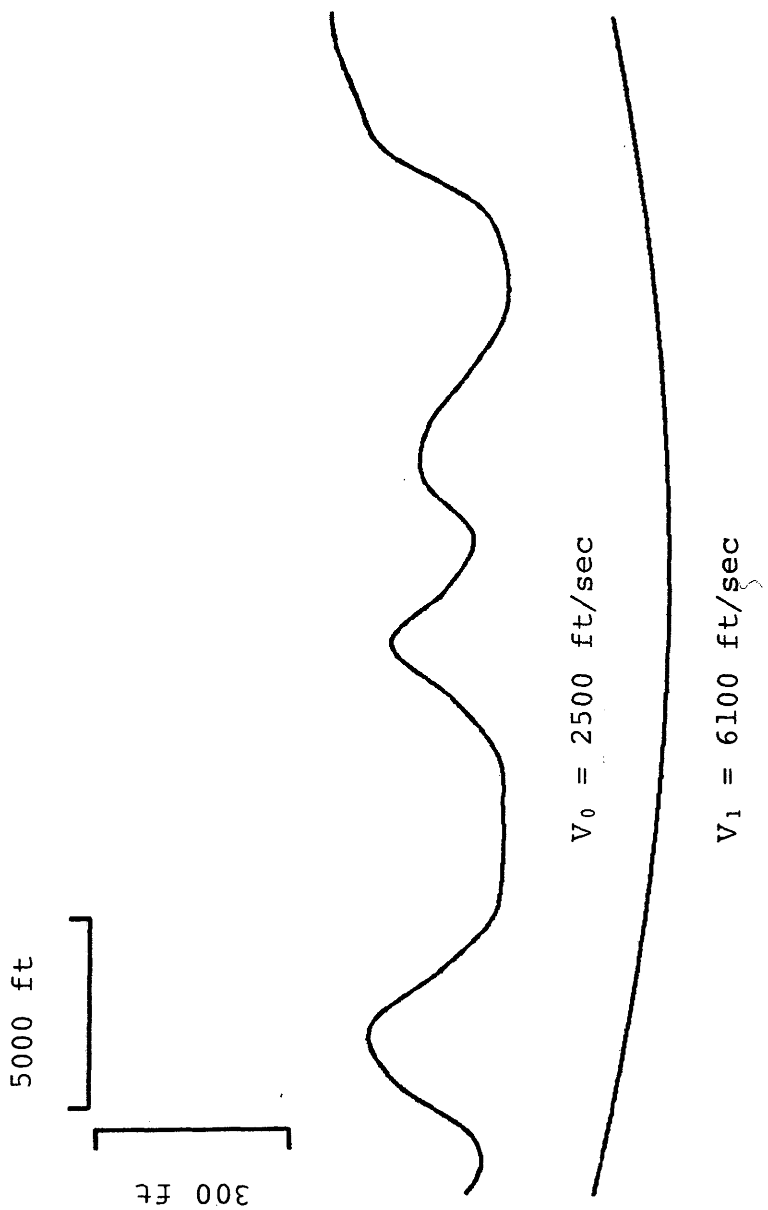


Figure 22 Synthetic model with undulated surface

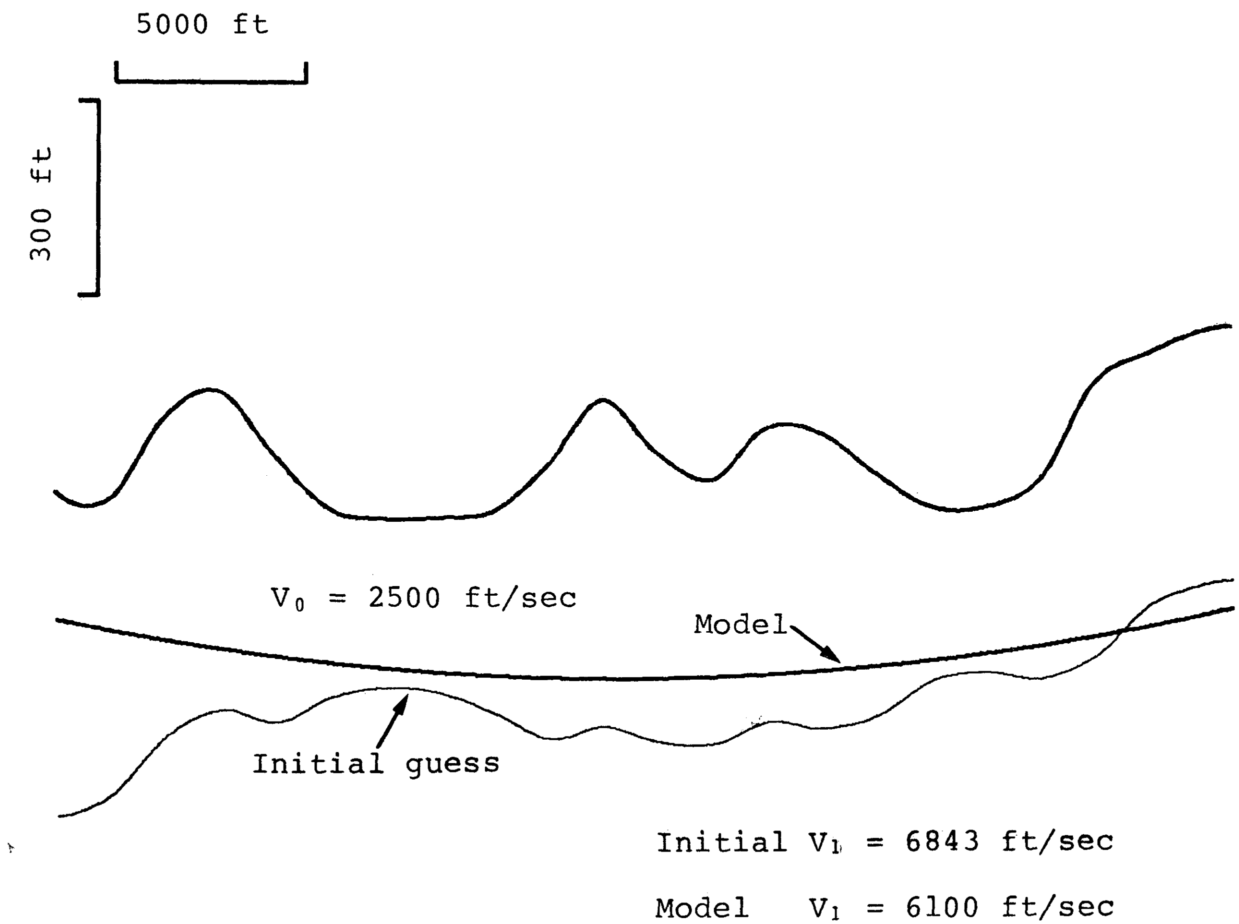


Figure 23 Initial guess for syncline model

RMS ERROR = 41.96 ms

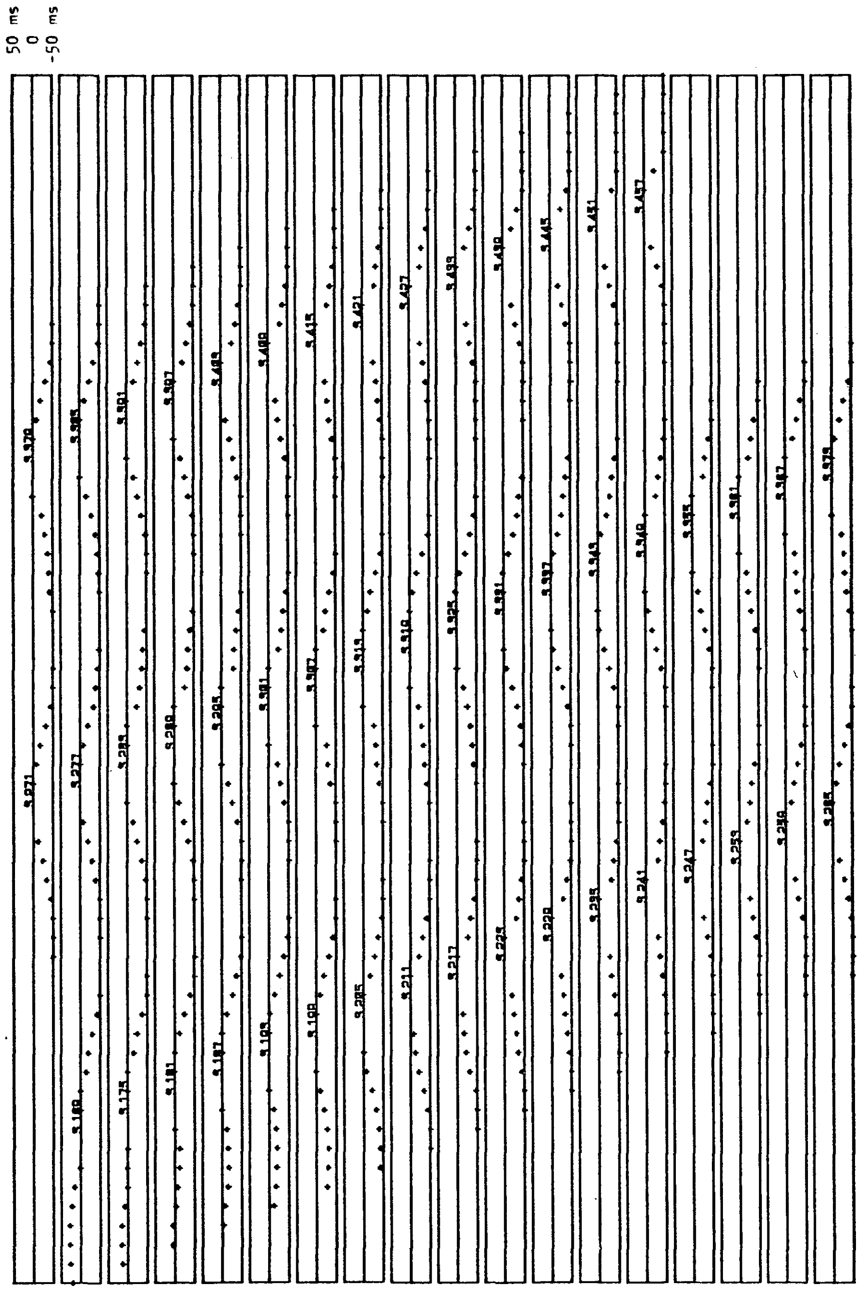


Figure 24 QC plot for the initial guess model

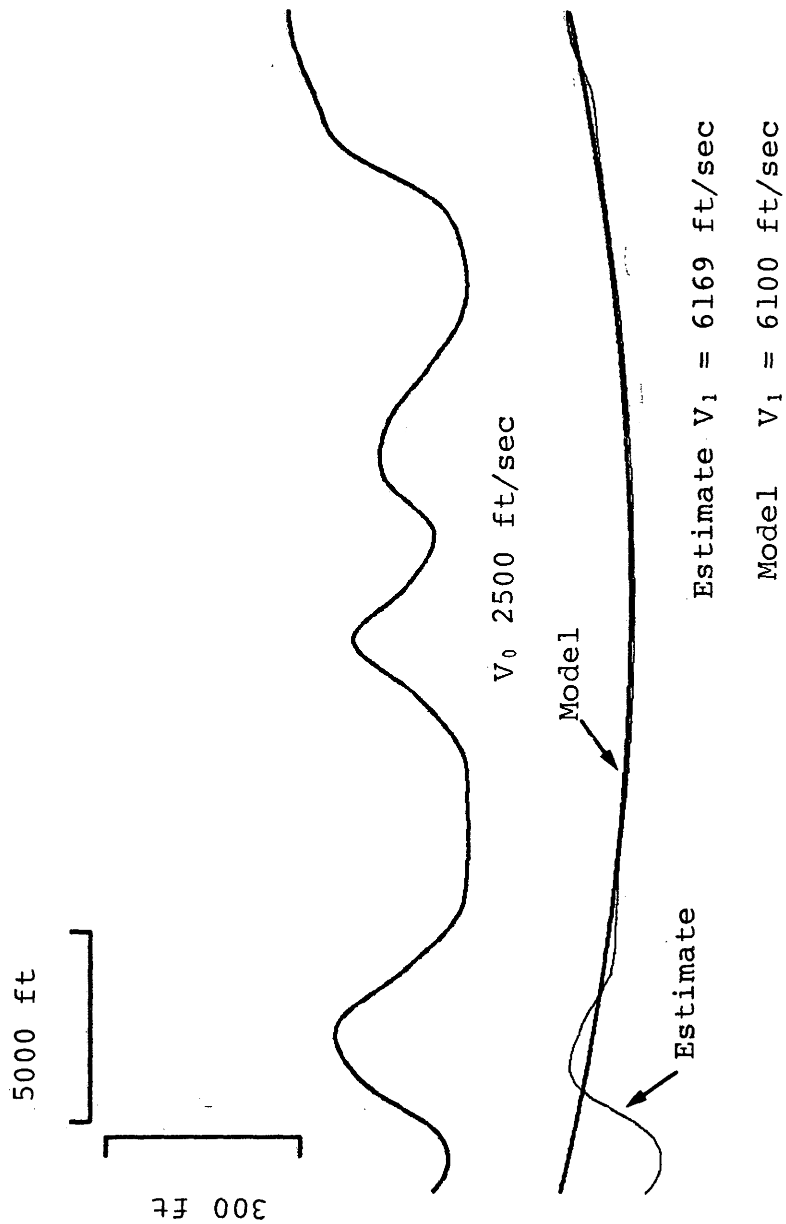


Figure 25 Output model by GLI

RMS ERROR = 5.36 ms

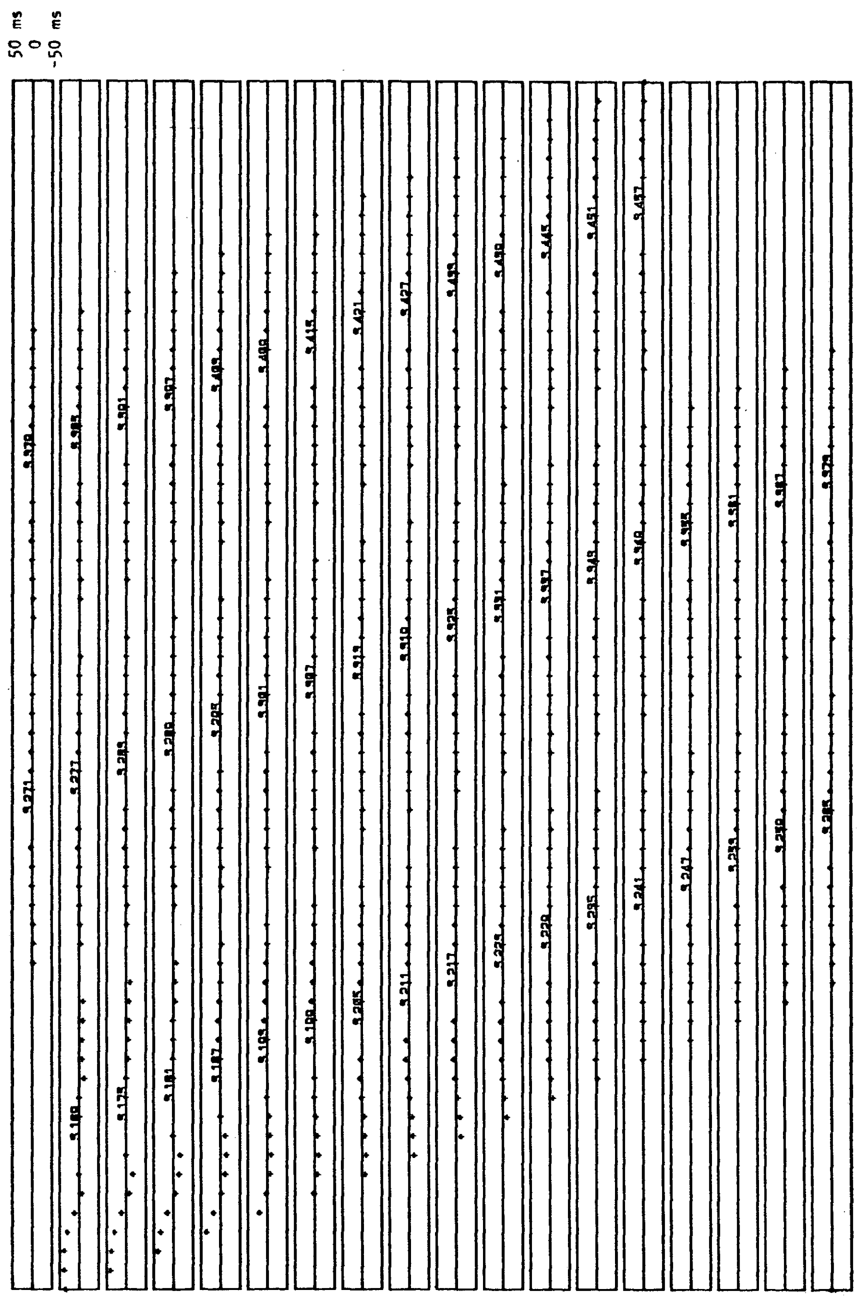


Figure 26 QC plot for the output model

4.3 Weathering Layer Velocity Study

V_0 (weathering layer velocity) is a very critical parameter in the computation of refraction statics. Theoretically, for a flat surface and a flat refractor, V_0 is not solvable from refraction data according to the equation:

$$T_i = 2z\sqrt{\frac{1}{V_0^2} - \frac{1}{V_1^2}}$$

$$T_i = 2\,z\,\frac{\sqrt{V_1^2 - V_0^2}}{V_1 V_0}$$

where $1/V_1$ is the slope of the travel time curve and T_i is the intercept time. It is obvious that the depth z and the velocity V_0 are dependent and there is no unique solution for this problem.

4.3.1 Undulated Refraction Model Test

All the above examples assumed a known V_0 (weathering layer velocity). For standard reflection shooting, it is very difficult to measure V_0 because the direct wave is hardly detected. A wrong V_0 will cause the inversion process to find an incorrect model. The same model as above is used but the V_0 is assumed to be 3000 ft/sec in stead of 2500 ft/sec. The initial guess and final output are shown in Figure 27, Figure 28, Figure 29 and Figure 30. After three

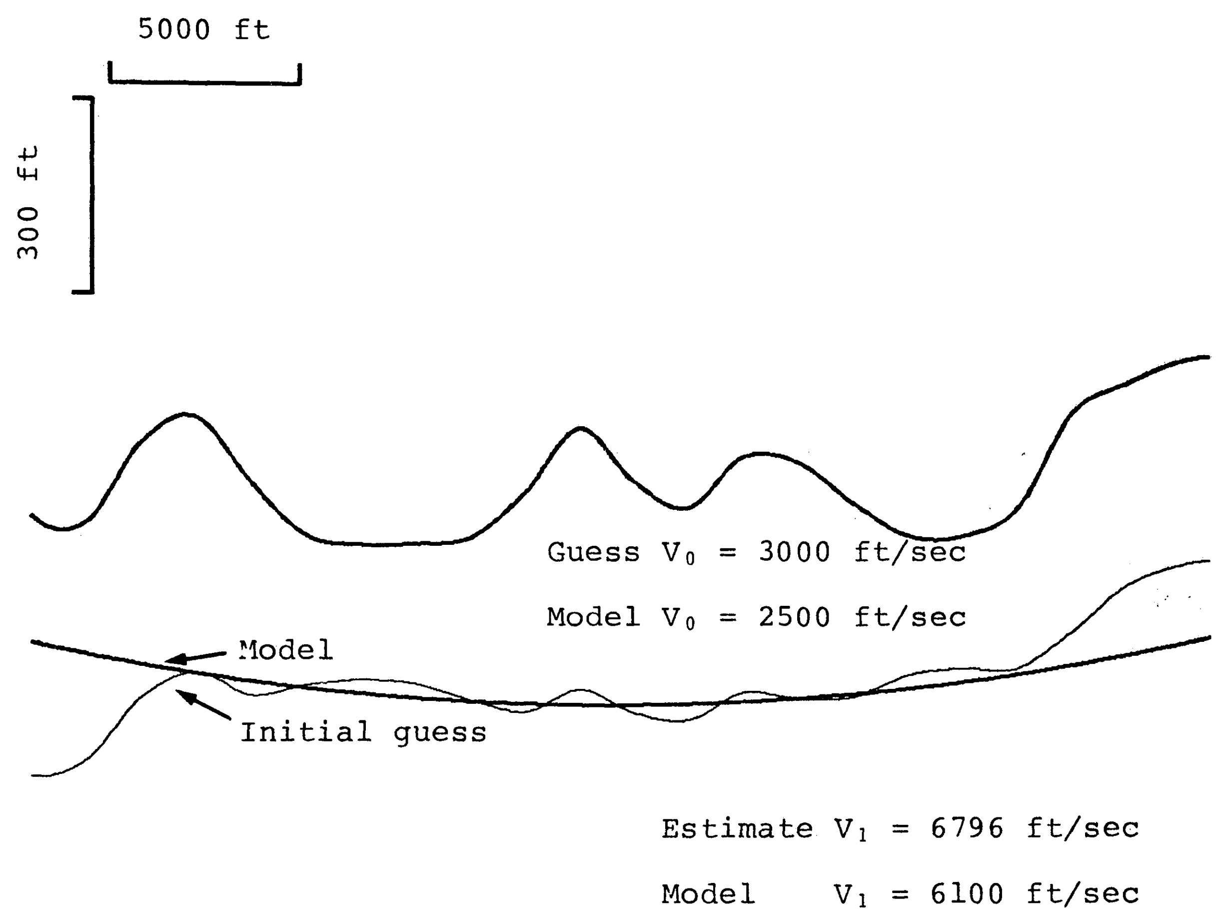


Figure 27 Initial guess model with wrong V_0

RMS ERROR = 46.87 ms

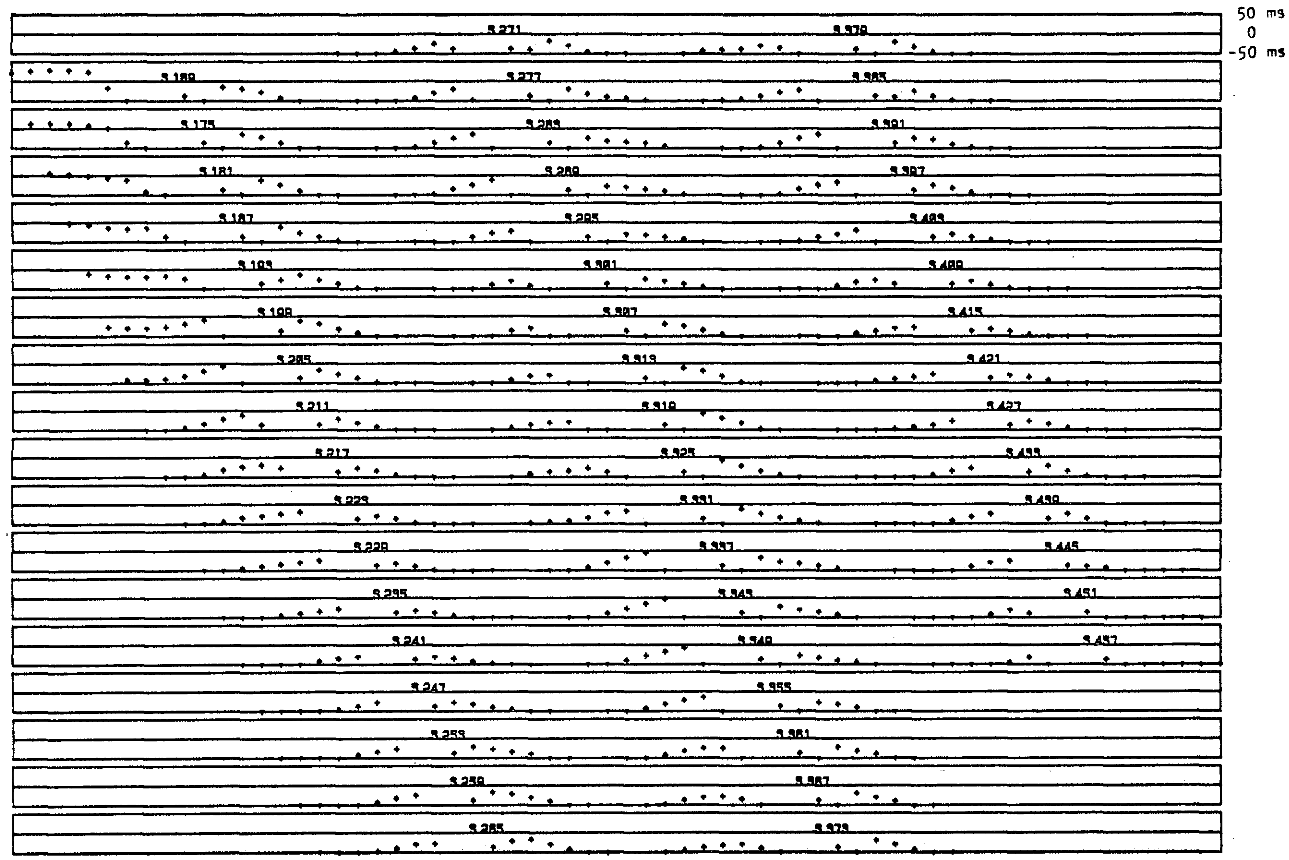


Figure 28 QC plot for the guess model with wrong V_0

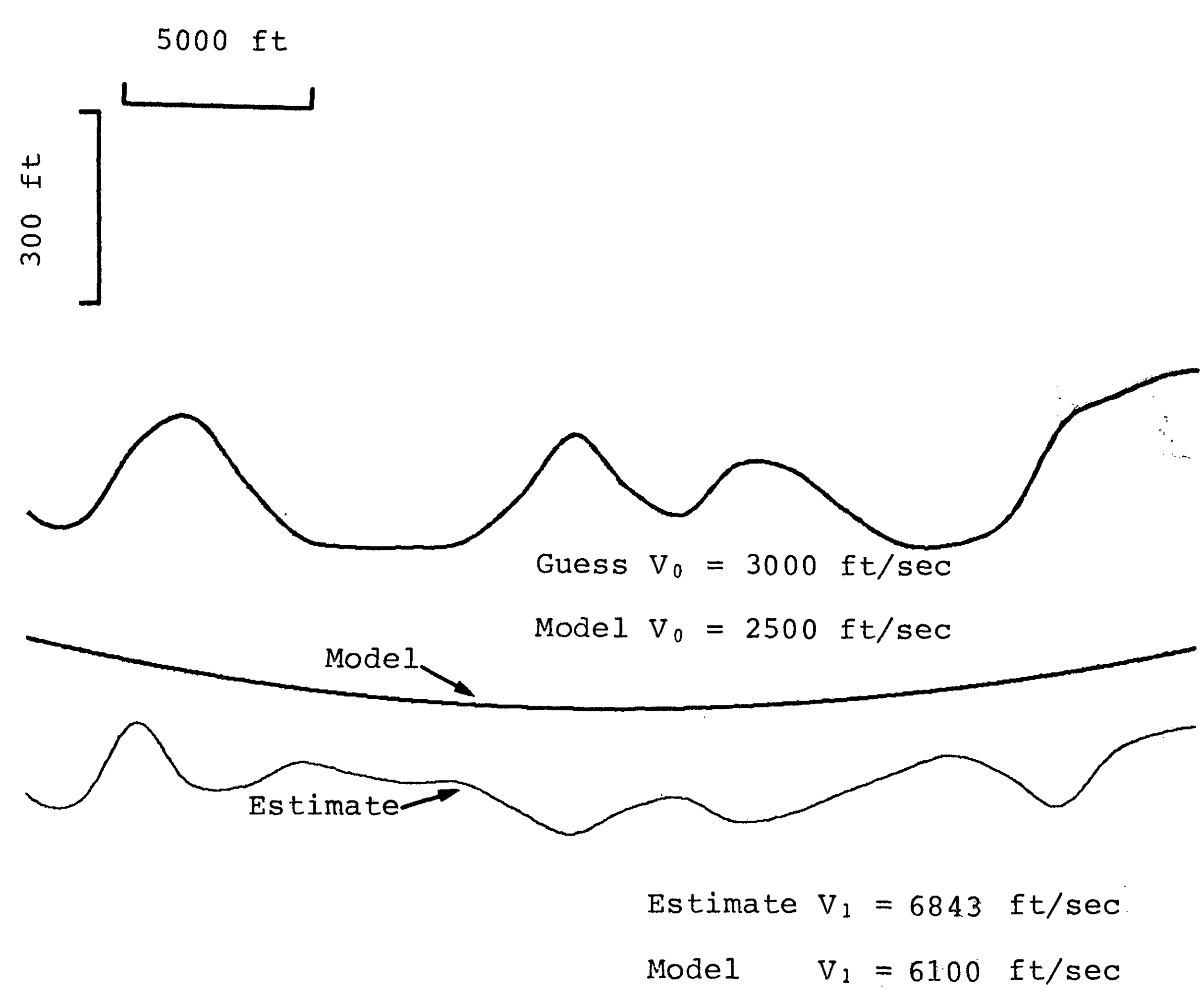


Figure 29 Output model with wrong V_0

RMS ERROR = 24.24 ms

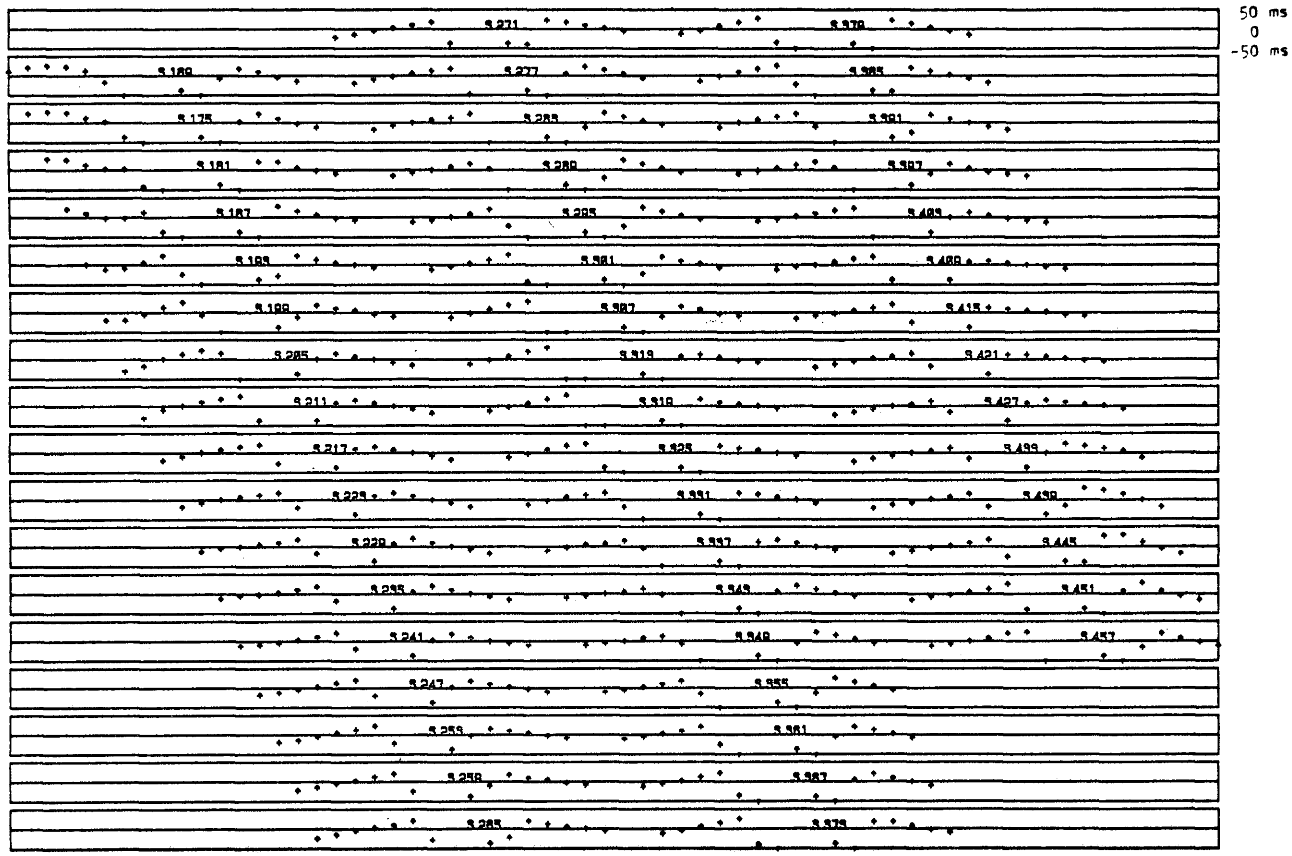


Figure 30 QC plot for the output model with wrong V_0

iterations, the original structure can't be recovered but the best fit model is obtained under the assumption of wrong V_0. Interestingly, the result shows a reverse image of the surface on the subsurface structure. In practice, this could be due to the input V_0 being incorrect.

4.3.2 V_0 Inversion Test

When V_0 is chosen as an added parameter in the inversion process, there is a relatively small eigenvalue in the system ($\lambda_{max}/\lambda_{min} = 30$) related to the parameter V_0. That means it is more difficult to solve for V_0 than for any of the other parameters. Actually, there may be some local minima between the wrong initial guess and the true model. Thus, the final solution may be incorrect.

For real data sets, inversion for slightly different V_0's is suggested to determine the proper weathering layer velocity. The V_0 corresponding to the smallest RMS error yields the best solution and should be the most reasonable geologic model.

CHAPTER 5

REAL DATA TEST

A seismic line from an undisturbed area with an undulated surface was used to test the GLI technique. This line consists of 96 traces of 24 fold Vibroseis data which is sample at 4 msec, filtered between 18 and 90 hertz, and recorded at an 80 foot group interval. The surface elevation profile is shown in Figure 31 and the brute stack section is shown in Figure 32. It is obvious that there is a low frequency static problem in this example. The first break picks of this line are shown in Figure 33. Refraction static correction method which combines the time delay and reciprocal methods (now is referred to as the conventional method) could improve the stack section. The improved stack section is shown in Figure 34 and there are still some low frequency components believed to be caused by statics. With the conventional method, the best stack section is achieved using 3000 ft/sec as the weathering layer velocity, while 2500 ft/sec is a more reasonable V_0 for the area, which has sand dunes present.

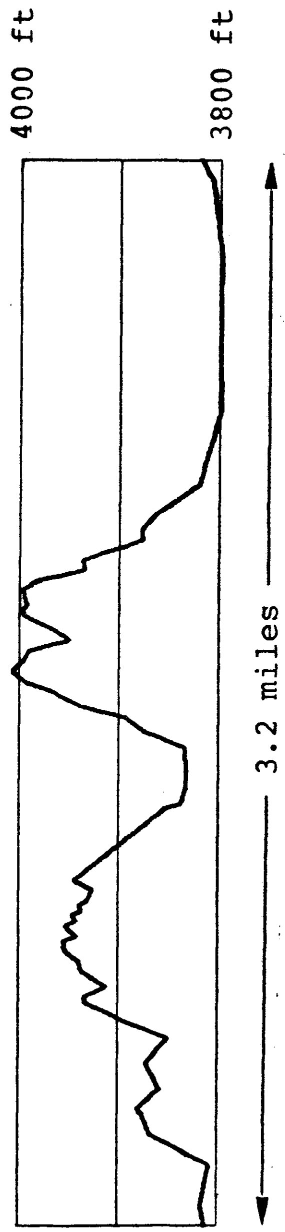


Figure 31 Surface elevation profile of the real data example

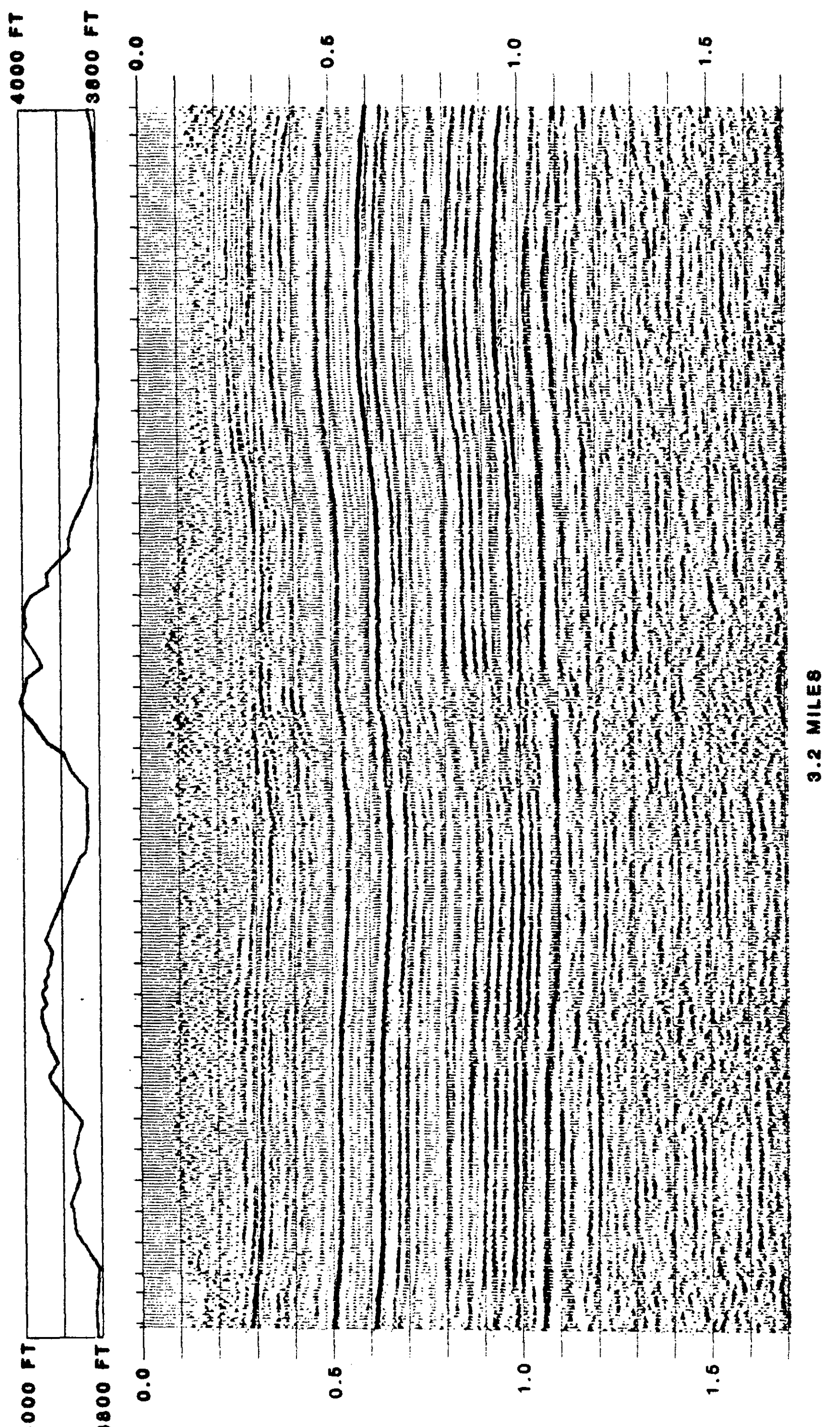


Figure 32 Stack section (with elevation correction)

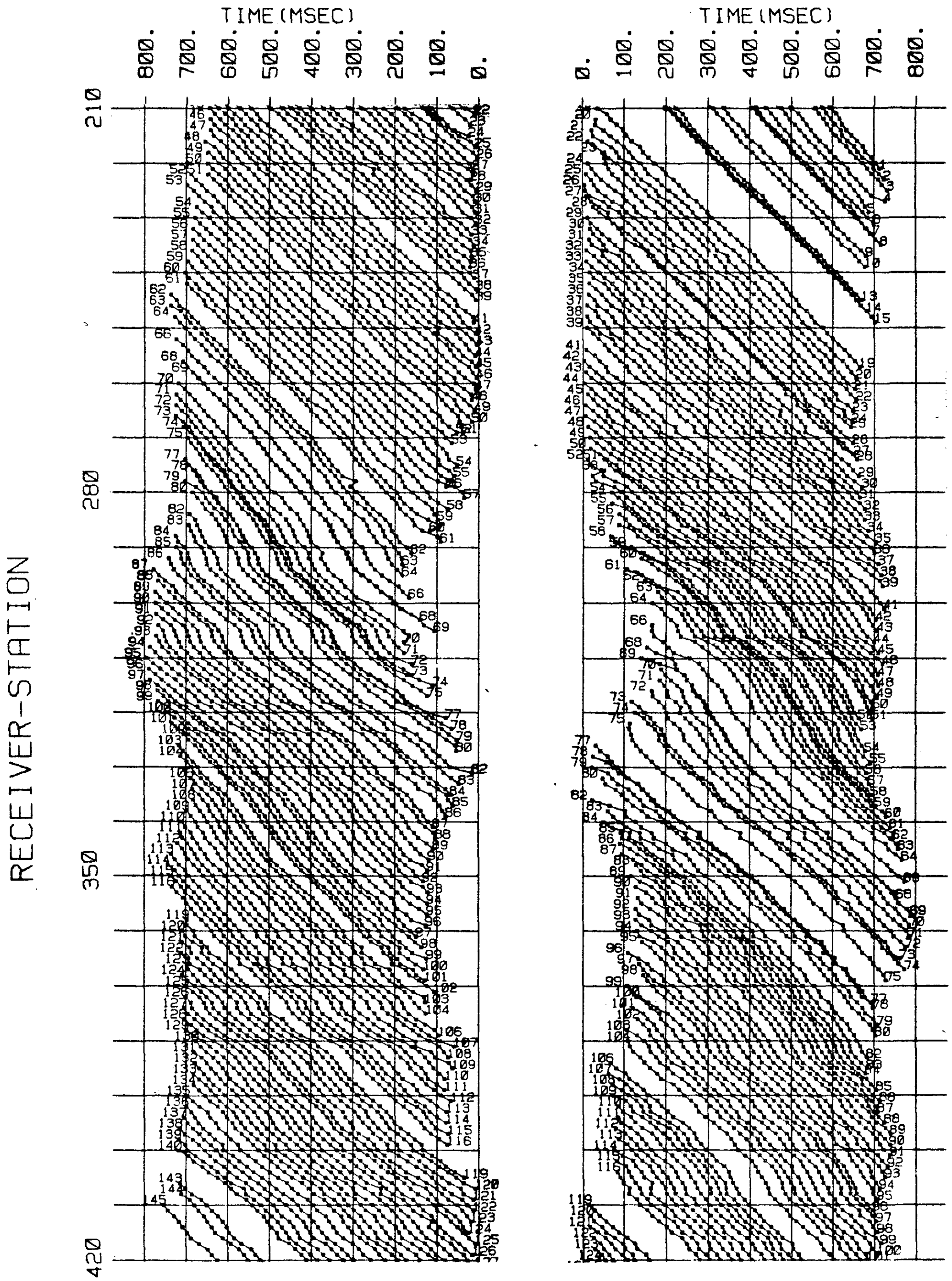
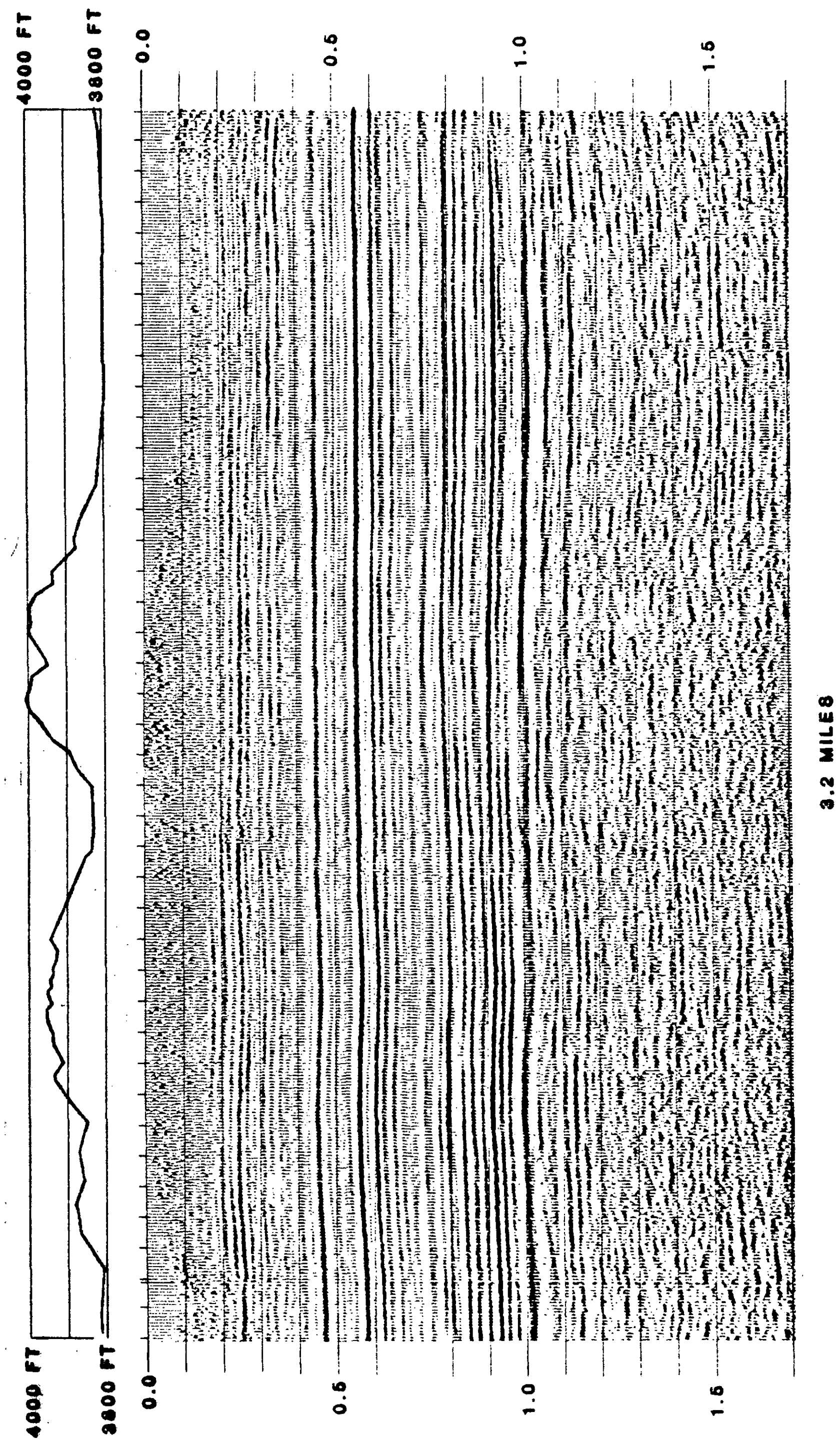


Figure 33 First break picks of the real data set



Fifure 34 Stack section (with conventional refraction static correction)

The V_0 test is done by running the inversion process for varying V_0's. These values include: 3000 ft/sec, 2500 ft/sec and 2000 ft/sec. A 6-station geometry control point interval, a 6-station shot interval and a 2-station receiver interval are used as input to the program. When the V_0 equals 2500 ft/sec, the output model has the smallest RMS error of 10.3, compared with the RMS error of 11.0 when the V_0 equals 3000 ft/sec and the RMS error of 10.5 when the V_0 equals 2000 ft/sec. When the V_0 is chosen to be 2500 ft/sec, the output model has a more reasonable geologic structure, less related to the surface undulation and without the wobble phenomena as in the output model when the V_0 is chosen to be 3000 ft/sec (Figure 35 and Figure 36). Thus, a value of 2500 ft/sec for V_0 will be used in this line.

In order to test the stability of the algorithm, the good initial guess (from the output of the conventional method, Figure 37) as well as the bad initial guess (constant thickness weathering layer, Figure 38) are used. Figure 39, 40 and 41 show the QC plots of the initial guess from conventional method and two iteration outputs. Figure 42, 43 and 44 show the QC plots of the constant thickness

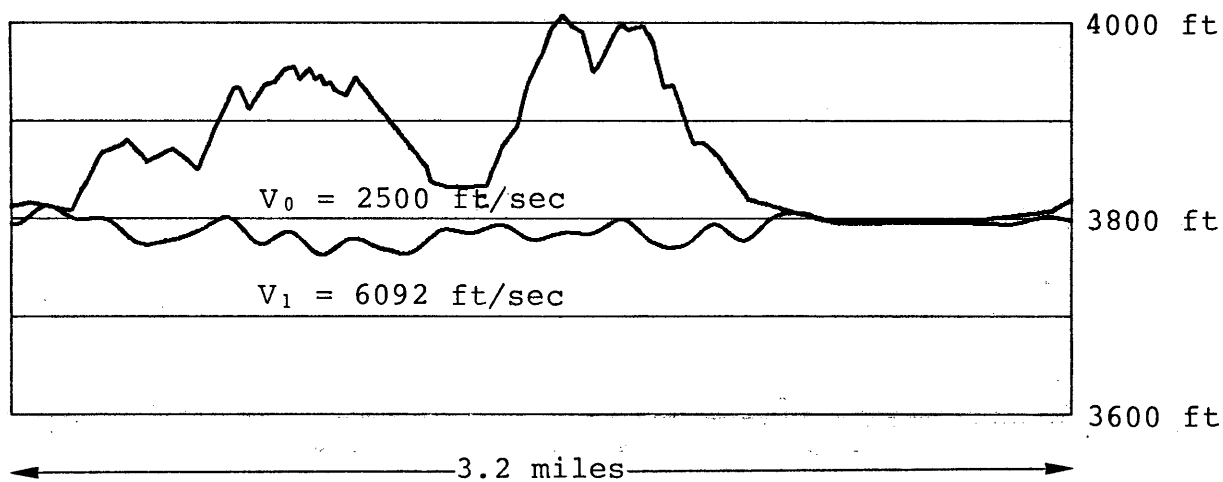


Figure 35 Near surface model for 2500 ft/sec V_0

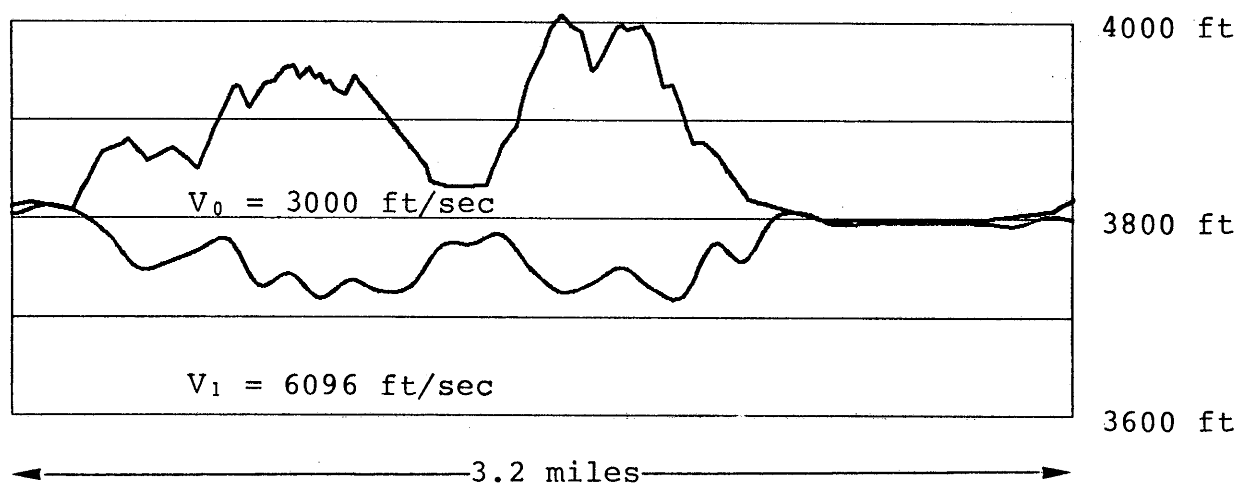


Figure 36 Near surface model for 3000 ft/sec V_0

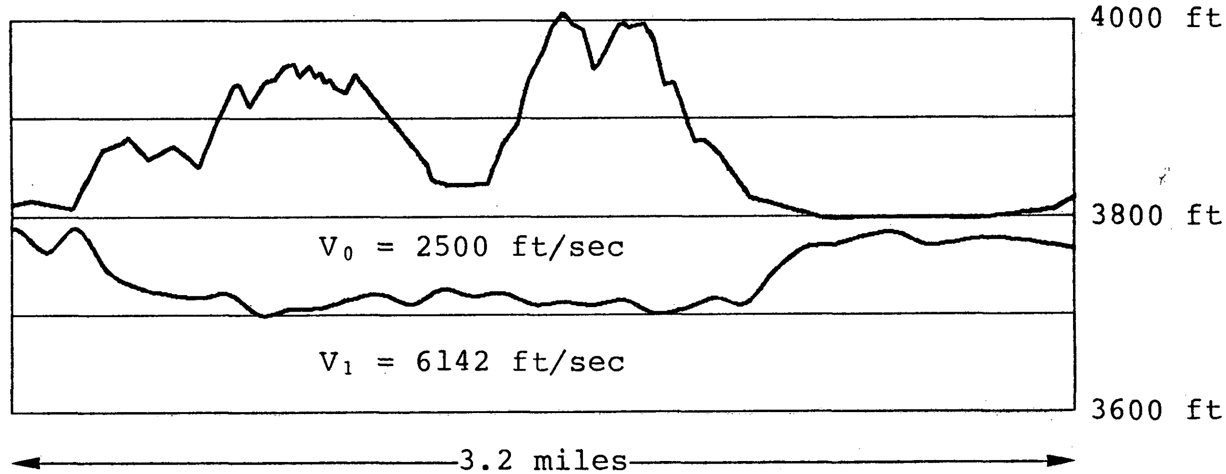


Figure 37 Near surface structure of good initial guess (from conventional method)

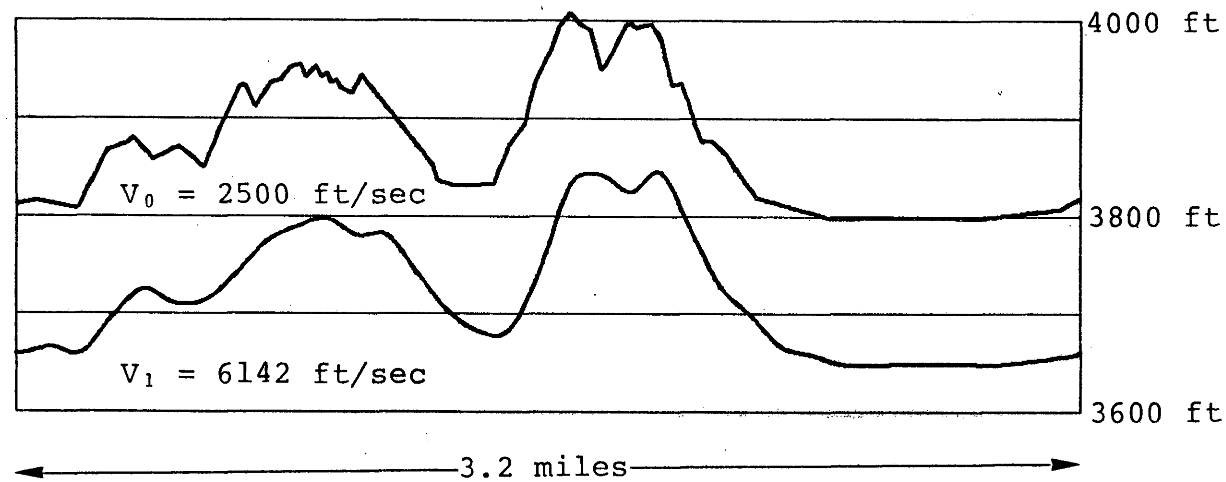


Figure 38 Constant thickness initial guess model (150 ft at all the geometry control points)

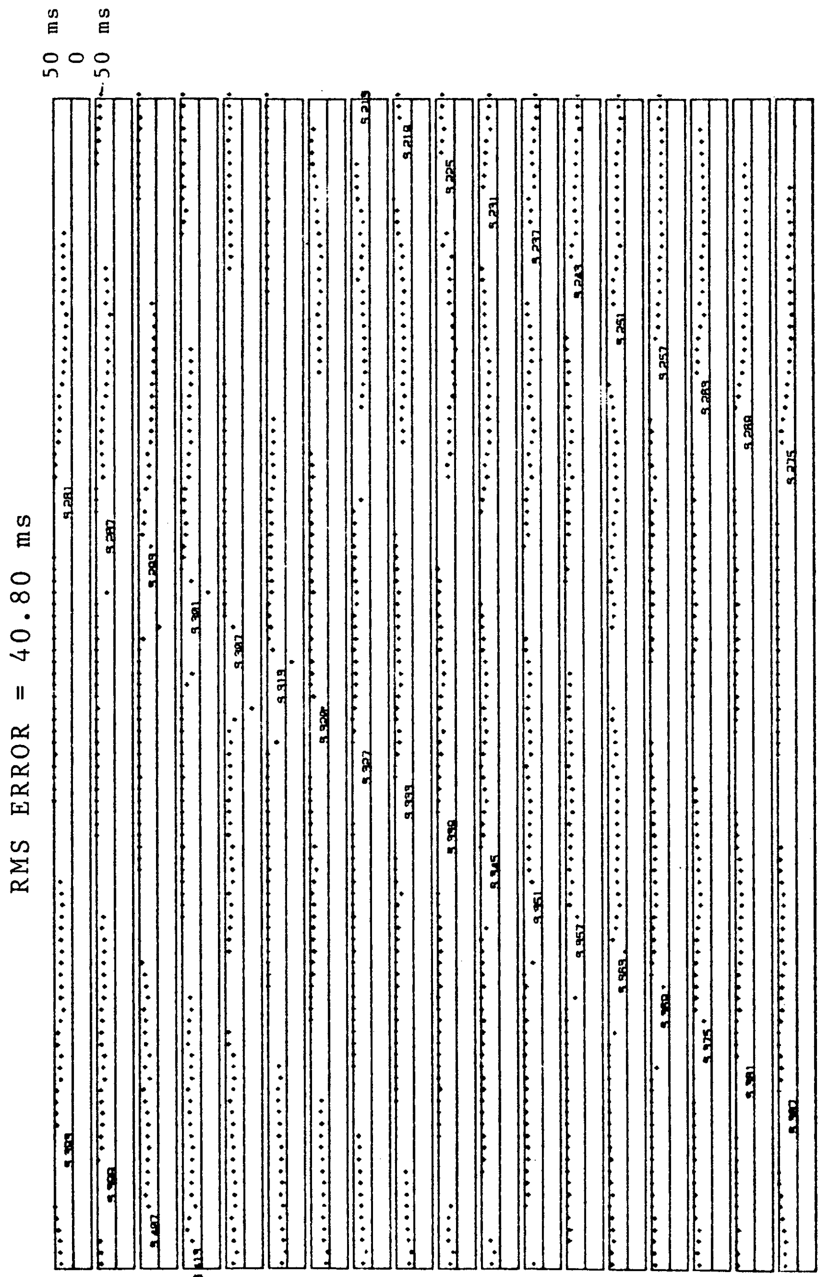


Figure 39 QC plot of good initial guess (from the output of conventional method)

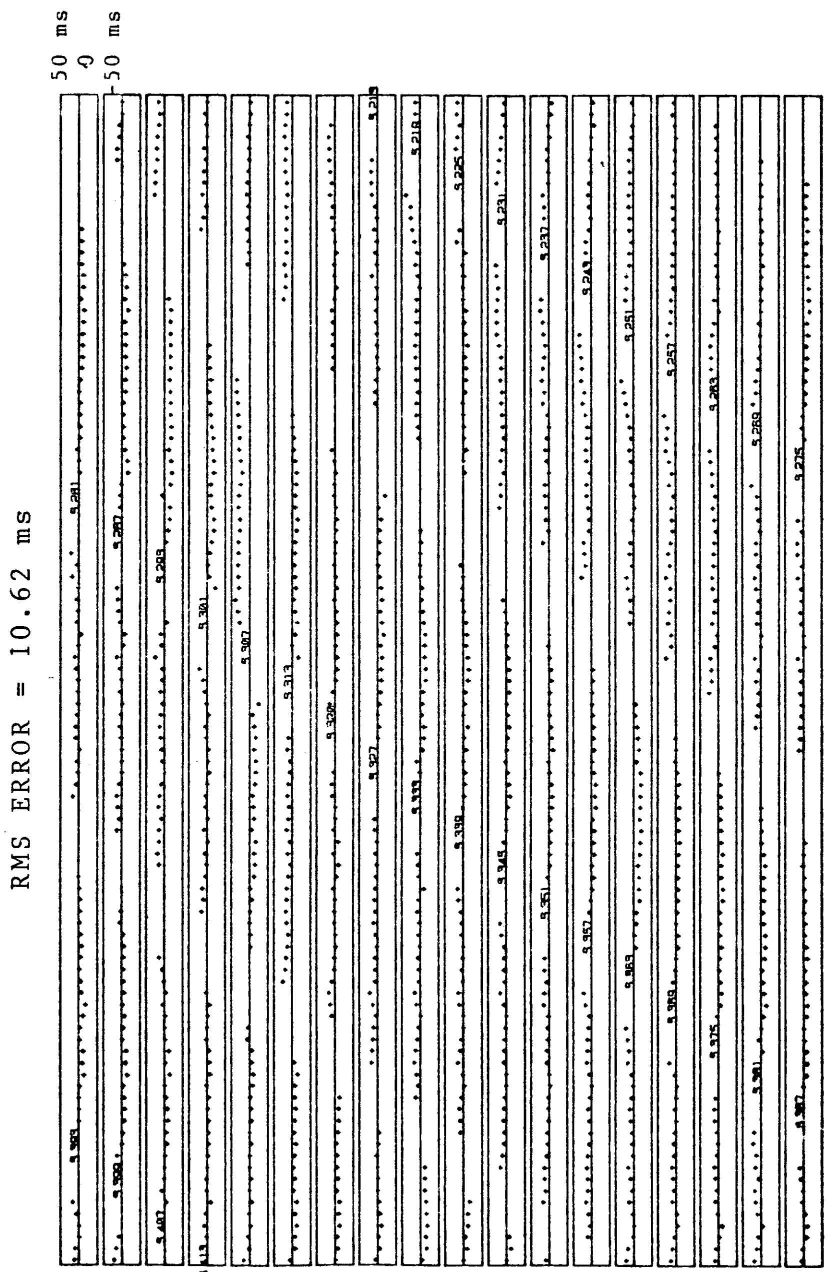


Figure 40 QC plot of good initial guess (first iteration)

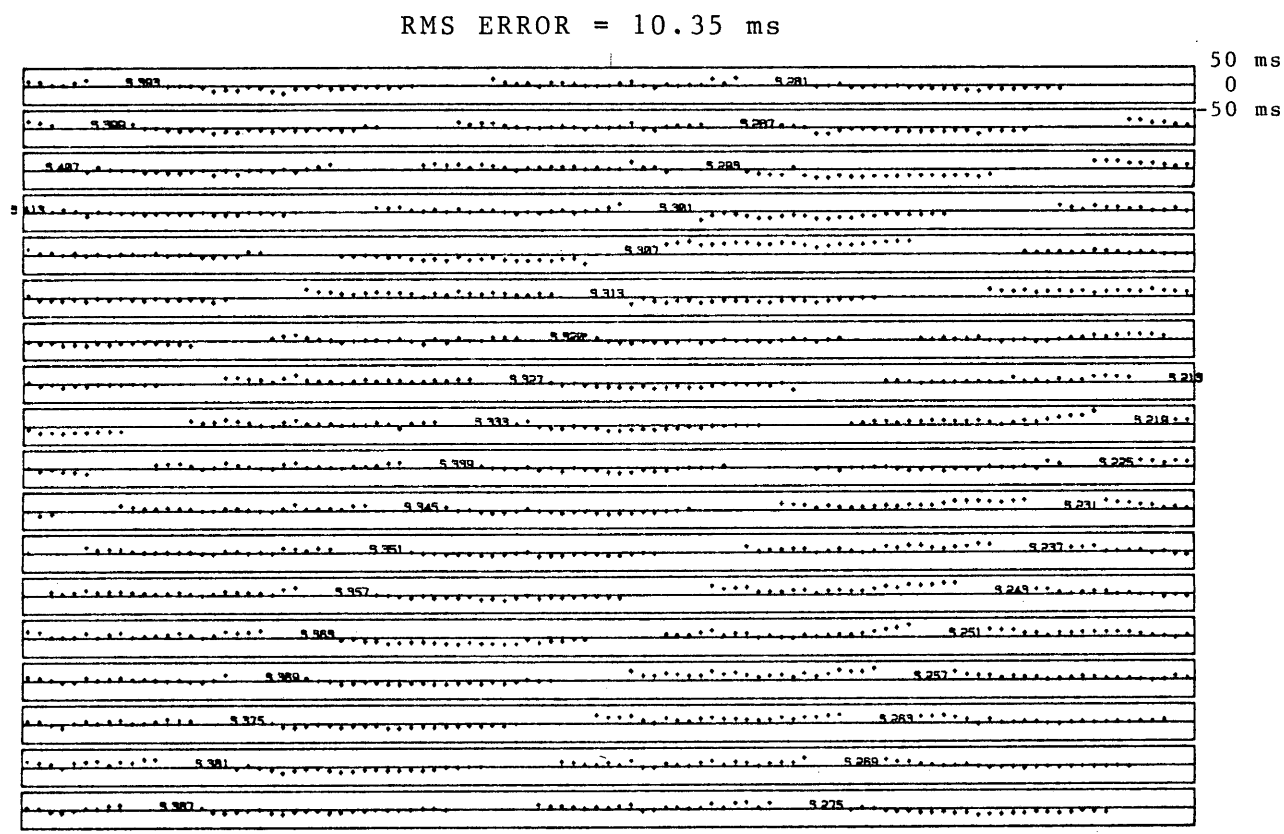


Figure 41 QC plot of good initial guess (second iteration)

RMS ERROR = 64.21 ms

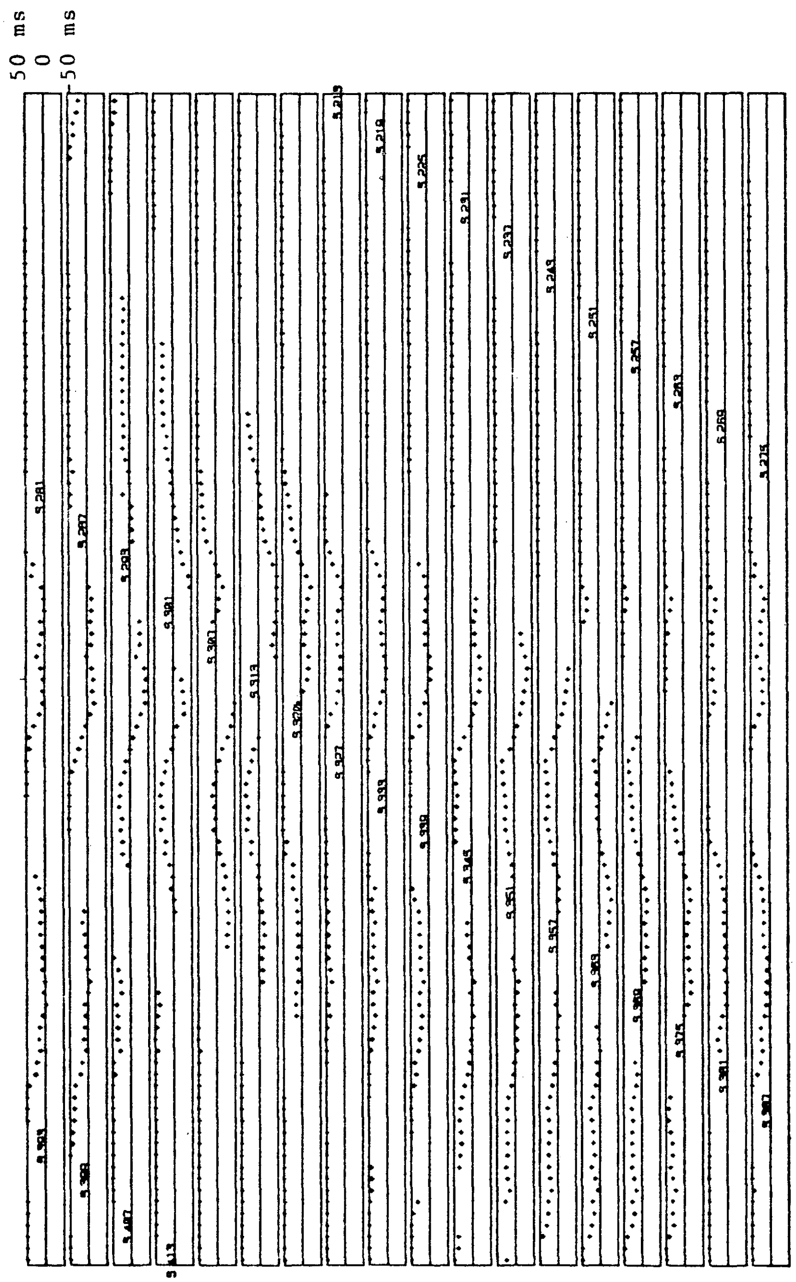


Figure 42 QC plot of bad initial guess (constant thickness)

RMS ERROR = 10.75 ms

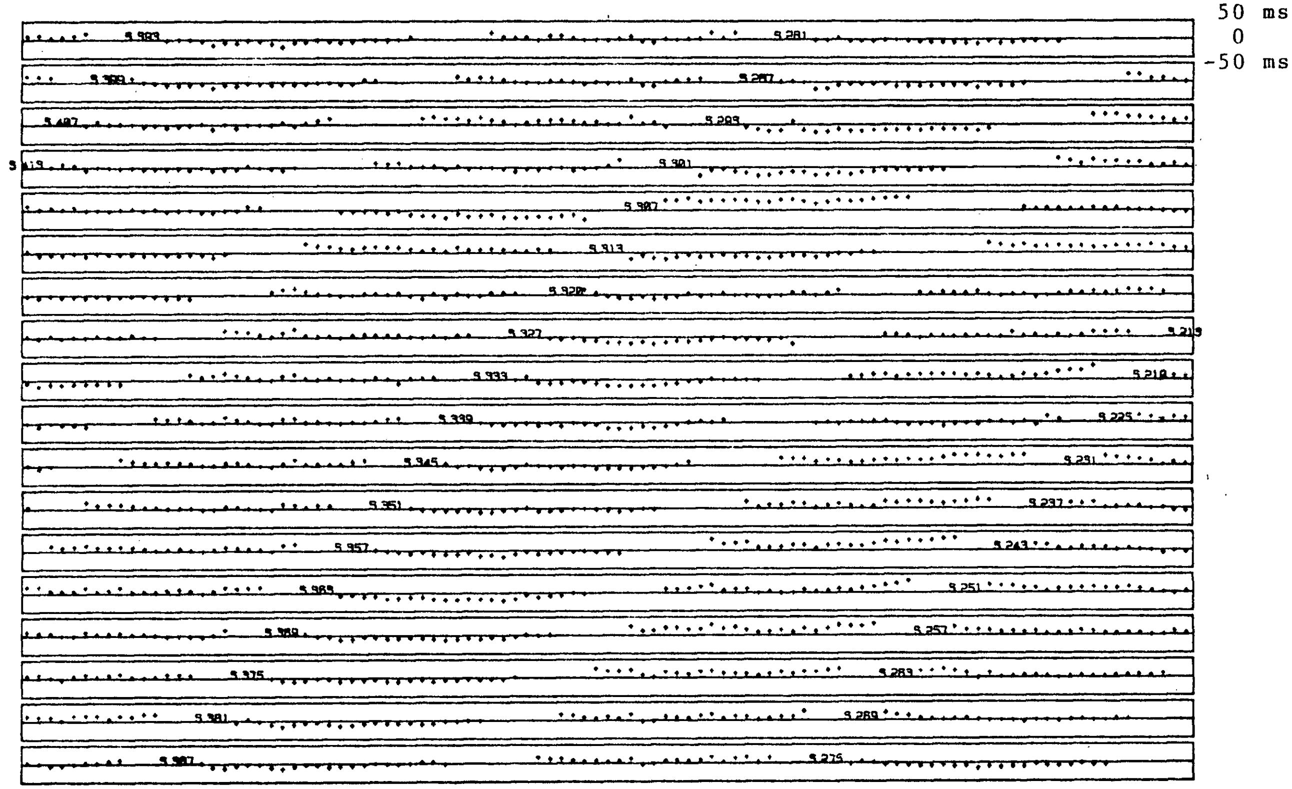


Figure 43 QC plot of bad initial guess (first iteration)

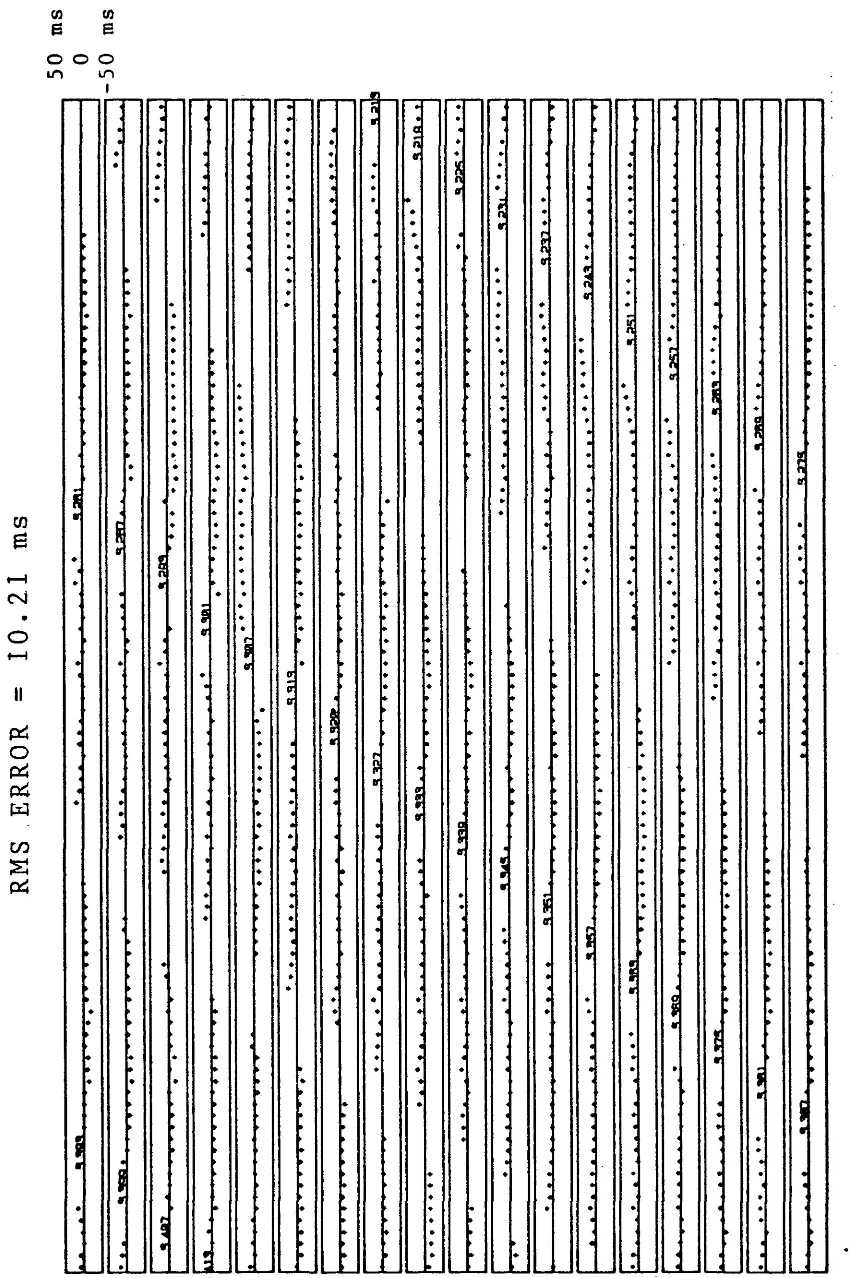


Figure 44 QC plot of bad initial guess (second iteration)

initial model (150 ft thick under all the geometry control points at the surface) and two iteration outputs. After two iterations, the RMS errors of these two initial guesses reach the same minimum about 10.3 and appear stable. The very similar output models provide confidence about the uniqueness of the solution.

Dividing the thickness of the weathering layer at each station by V_0, it is easy to determine the static corrections. These static corrections are applied to the raw data (after deconvolution) and the stack section is shown in Figure 45. The low frequency structure has improved which leads to a more reliable stacked section.

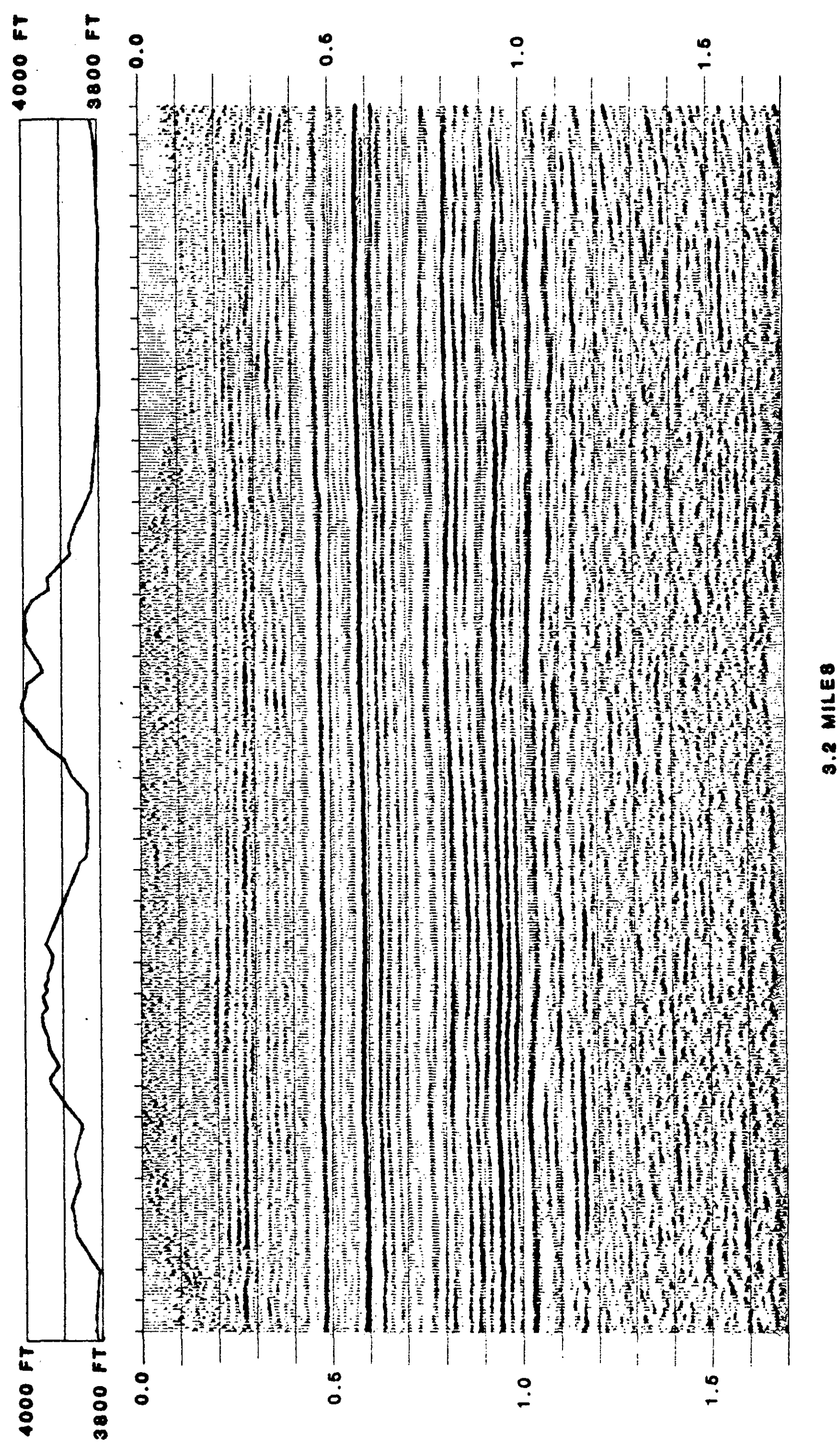


Figure 45 Stack section (with GLI refraction static corrections)

CHAPTER 6

CONCLUSIONS AND RECOMMENDATIONS

In conclusion, the GLI method proves to be a powerful tool for solving low frequency static problems. The advantages are:

a. This method is dip angle (both surface and refractor) independent, unlike all current other refraction methods.

b. This method can be used as the main part of a V_0 detection scheme, keying on the RMS error and the output model.

c. This method performs the elevation correction and the static correction simultaneously.

d. The QC plots of this method are very useful because they check the GLI correction at each step to make sure the whole process is converging. Also, bad shots may be easily located and properly edited. Surface consistent high frequency errors may be treated as high frequency statics, applied to the data after the final iteration.

The disadvantage of this method is that there is a rapid increase in computer run time when the number of parameters increase.

Further research is still needed for the GLI statics computation technique. The inversion program may be connected to more flexible modeling programs in chapter 2 to handle the spatial dependent velocity case, when a slowly varying velocity earth model may not be used. It is possible to create a high frequency static detection system to extract the high frequency statics from the QC plots. Lastly, inverstigations into better methods of determining the velocity V_0 would be most worthwhile.

REFERENCES

Aki, Keiiti, and Richards, Paul G., (1980), "Qualitative Seismology Theory and Methods": W. H. Freeman and Company.

Backus, G. E., and Gilbert, J. F., (1976), "Numerical Application of a Formalism for Geophysical Inverse Problems", Geophys. J. 13, 247-276.

Dobrin, Milton B., (1976), "Introduction to Geophysical Prospecting", Third Edition, McGraw-Hill, Inc.

Gardner, L. W., (1967), "Refraction Seismograph Profile Interpretation", pp. 338-347, in "Seismic Refraction Prospecting", Society of Exploration Geophysicists.

Hampson, Dan, and Russell, Brian, (1984), "First-Break Interpretation Using Generalized Linear Inversion", 1984 SEG annual Meeting.

Hileman, J. A., Embree, P., and Pflueger, J.C. (1968), "Automated Static Corrections", Geophysical Prospecting, Vol. 16, No. 3, pp.326-358.

Lines, L. R., and Treitel, S., (1984), "Tutorial A Review of Least-Squares Inversion and Its Application to Geophysical Problems", Geophysical Prospecting 32, 159-186, 1984.

Nobel, Ben, and Daniel, James W., (1977), "Applied Linear Algera": Prentice-Hall, Inc.

Palmer, Derecke, (1986), "The Generalized Reciprocal Method of Seismic Refraction Interpretation": Society of Exploration Geophysicists.

Slotnick, Morris M., (1950), "A Graphical Method for the Interpretation of Refraction Profile Data", Geophysics, v. 15, p. 163-180.

Tarrant, L. H., (1956), "A Rapid Method of Determining the Form of a Seismic Refractor From Line Profile Results", Geophys. Prosp., v. 4, p. 131-139.

Waters, Kenneth H., (1978), "Reflection Seismology": John Wiley & Sons, Inc.

REFERENCE (Continued)

Wiggins, R. A., Larner, K. L. and Wisecup, R. P. (1976), Residual statics analysis as a general linear inverse problem, Geophysics 41, p. 927-938.

Wiggins, Ralph A., (1972), "The General Linear Inverse Problem: Implication of Surface Waves and Free Oscillations for Earth Structure", Reviews of Geophysics and Space Physics, Vol. 10, No. 1, p. 251-285, February 1972.

APPENDIX

PROGRAM LISTING

```
C*********************************************************************
C
C       COMMON BLOCKS
C
C
C*********************************************************************
         DIMENSION V(3),NSR(3),XFH(400),YPH(400),A1(80),A2(80),A3(80)
         DIMENSION  P1(60),P2(60),P3(60),IA1(20),IA2(20),IA3(20),SR1(20)
         DIMENSION  SR2(20), SR3(20), SM1(20), SM2(20), SM3(20)
         DIMENSION AT1(20),AT2(20),AT3(20),DENT(450),DINT(450),RSHX2(450)
         DIMENSION ESHX2(450),RTIM2(450),ESHY2(450),RSHY2(450)
         DIMENSION ETIM2(450),RSHX3(450),RSHY3(450),RTIM3(450),ETIM3(450)
         DIMENSION ESHX3(450),ESHY3(450)
C
       COMMON /MODIN/ XFH,YPH,A1,A2,A3,P1,P2,P3,IA1,IA2,IA3,SR1,SR2,SR3,
     &              SM1,SM2,SM3,INA,INB,CRA,CRB,NSR,IHK,ICA,ICB,ICC,
     &              LN,V,DI1,ER,AT1,AT2,AT3,DENT,DINT,RSHX2,RSHY2,RTIM2,
     &              ESHX2,ESHY2,ETIM2,RSHX3,RSHY3,RTIM3,ESHX3,ESHY3,ETIM3



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       THE MAIN ROUTINE FOR THE MODELING-INVERSION PROCESS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      DIMENSION UXA(30,3),LYA(30,3),NSP(2),VE(3),RAF(200)
      DIMENSION IXAR(20),IXBR(20),MV(2),GES(200),DRI(2000)
      DIMENSION WEI(20),OGF(200),EG(20),VEG(500),CAL(20),DEP(20)
      DIMENSION VTR(2000),CES(200),ERF(200)
C
C
C     READ THE PARAMETERS FOR THE INVERSION AND MODELING PROGRAM
C     LN   : NUMBER OF THE LAYERS OF THE MODEL
C     DI1  : GEOPHONE INTERVAL
C     XST  : THE DISTANCE OF THE FIRST STATION FROM THE ORIGIN
C     NC   : THE NUMBER OF THE RECEIVERS
C     ILS  : THE STATION INTERVAL FROM THE SHOT TO THE FIRST GEOPHONE
C     ER   : THE TOLERANT DISTANCE FOR THE RAY TO THE SPECIFIED GEOPHONE
C     DIR  : THE DIRECTION OF THE GEOPHONE SPREAD
C            0 ------ RECIPROCAL
C            1 ------ LEFT TO RIGHT
C            -1 ----- RIGHT TO LEFT
C     ERIN : THE ACCEPT ERROR FOR THE INVERSION PROCESS
C     GSTA : THE CONSTRAIN FOR THE THICKNESS OF THE WEATHERING LAYER
C     NCON : THE MAXIMUM ITERATION FOR THE INVERSION PROCESS
```

```
C
C
      READ(7,*)LN,DI1,NC,ILS,ER,XST,DIR,ERIN,GSTA,NCON
      NS=NC+ILS
C
C
C     SET THE BASIC PARAMETER FOR THE MODEL
C
C
      CALL MOCE(LN,DI1,XST,NC,ILS,ER,DIR,ERIN,NSP)
      IF(DIR.EQ.0.)NR=NC*2
      IF(DIR.EQ.1.OR.DIR.EQ.-1.)NR=NC
      LS=NR/NC
C
C
C     READ IN THE OBSERVATIONS
C
C
      READ(8,*)(OBS(I),I=1,NR)
      I=1
C
C
C     READ IN THE SURFACE GEOMETRY CONTROL POINTS
C
C
  220 READ(9,*)UXA(I,1),UYA(I,1)
      IF(UYA(I,1).EQ.10000.)GO TO 230
      I=I+1
      GO TO 220
C
C
C     READ IN THE WEATHERING LAYER VELOCITY
C
C
  230 READ(9,*)VE(1)
C
C
C     SUBROUTINE MOGE SETS THE NUMERICAL EXPRESSION FOR THE MODEL
C
C
      CALL MOGE(XST,NS,1,VE,VEX,VEZ,1,IDCF,DIR,UXA,UYA,KNDF,IPX)
      IF(IPX.NE.0)GO TO 807
      DO 232 J=2,LN
      I=1
C
C
C    READ IN THE REFRACTOR GEOMETRY CONTROL POINTS
C
C
  233 READ(9,*)UXA(I,J),UYA(I,J)
      IF(UYA(I,J).EQ.10000.)GO TO 234
      I=I+1
      GO TO 233
C
C
C     READ IN THE VELOCITY OF THE REFRACTOR
C
C
```

```
  234 READ(9,*)VE(J)
        IF(J.EQ.2)NA=I-1
        IF(J.EQ.3)NB=I-1
  232 CONTINUE
C
C
C     READ IN THE LOCATIONS OF THE ADJUSTABLE POINT OF THE MODEL
C     MX1,MX2 ----- INDICES OF THE FIRST AND SECOND REFRACTORS
C     MN ----- NUMBER OF SAMPLE LOCATIONS OF THE FIRST REFRACTOR
C     MM ----- NUMBER OF SAMPLE LOCATIONS OF THE SECOND REFRACTOR
C     MV ----- VELOCITY INDICES
C     IXAR --- LOCATION OF SAMPLE POINT OF THE FIRST REFRACTOR
C     IXBR --- LOCATION OF SAMPLE POINT OF THE SECOND REFRACTOR
C
C
        READ(10,*)MX1,MN
        IF(MX1.NE.0)READ(10,*)(IXAR(I),I=1,MN)
        READ(10,*)MX2,MM
        IF(MX2.NE.0)READ(10,*)(IXBR(I),I=1,MM)
        LN1=LN-1
        READ(10,*)(MV(I),I=1,LN1)
 1200   INCO=0
 1400   INCO=INCO+1
        DO 388 J=2,LN
        CALL MODE(XST,NS,J,VE,VEX,VE2,100,IDOF,DIR,UXA,UYA,KNDF,IPX)
        IF(IPX.NE.0)GO TO 807
  388   CONTINUE
C
C
C     SUBROUTINE REFRA CALCULATES THE REFRACTED ARRIVALS FOR THE INPUT
C     MODEL AND SOURCE-RECEIVER GEOMETRY
C
C
        CALL REFRA(LS,NC,XST,NS,ILS,NSF,DIR,GES)
C
C
C     SUBROUTINE EROR CALCULATES THE ROOT MEAN SQURE ERROR BETWEEN THE
C     OBSERVATIONS AND THE GUESS MODEL
C
C
        CALL EROR(GES,OBS,NR,ERE)
        WRITE(15,1004)ERE
        WRITE(15,1002)(VE(I),I=2,LN)
        WRITE(15,1002)(UYA(I,2),I=1,NA)
        IF(LN.EQ.3)WRITE(15,1002)(UYA(I,3),I=1,NB)
        WRITE(15,*)(GES(I),I=1,NR)
C
C
C     IF THE ERROR IS SMALL ENOUGH OR THE MAXIMUM ITERATION NUMBER
C     IS REACHED, STOP THE INVERSION LOOP
C
C
        IF(ERE.LT.ERIN.OR.INCO.GT.NCON)GO TO 806
        IST=0
        DO 401 I=1,LN1
        II=I+1
        IF(MV(I).EQ.0)GO TO 401
```

```
C
C
C      PERTURBATE THE REFRACTOR VELOCITY OF THE INPUT MODEL
C
C
          VDE=VE(II)*0.01
          VE(II)=VE(II)+VDE
          CALL MOGE(XST,NS,II,VE,VEX,VEZ,0,IDDF,DIR,UXA,UYA,KNDF,IPX)
          IF(IPX.NE.0)GO TO 807
          VE(II)=VE(II)-VDE
C
C
C      CALCULATE FIRST BREAKS OF THE VELOCITY PERTURBED MODEL
C
C
          CALL REFRA(LS,NC,XST,NS,ILS,NSP,DIR,RAF)
          IST=IST+1
C
C
C      CREATE THE ELEMENTS IN SENSITIVITY MATRIX RELATED TO VELOCITY
C      CHANGE
C
C
          CALL SENS(IST,NR,VDE,RAF,GES,DRI,WEI)
  401     CONTINUE
          IF(MX1.EQ.0)GO TO 410
C
C
C      PERTURBATE THE DEPTH OF THE FIRST REFRACTOR
C
C
          DO 411 J=1,MN
          I=IXAR(J)
          XID=UYA(I,2)*0.01
          IF(ABS(XID).LT.1.)XID=1.
          UYA(I,2)=UYA(I,2)+XID
          II=I+1
          IF(UXA(II,2).EQ.UXA(I,2))UYA(II,2)=UYA(I,2)
          CALL MOGE(XST,NS,2,VE,VEX,VEZ,2,IDDF,DIR,UXA,UYA,KNDF,IPX)
          IF(IPX.NE.0)GO TO 807
          UYA(I,2)=UYA(I,2)-XID
          IF(UXA(II,2).EQ.UXA(I,2))UYA(II,2)=UYA(I,2)
C
C
C      CALCULATE THE FIRST BREAKS OF THE PERTURBED MODEL WHOSE FIRST
C      REFRACTOR DEPTH CHANGED
C
C
          CALL REFRA(LS,NC,XST,NS,ILS,NSP,DIR,RAF)
          IST=IST+1
C
C
C      SENSITIVITY MATRIX SET
C
C
          CALL SENS(IST,NR,XID,RAF,GES,DRI,WEI)
  411     CONTINUE
  410     IF(MX2.EQ.0)GO TO 420
```

```
C
C
C      SAME PERTURBED METHOD APPLIED TO THE SECOND REFRACTOR IF REQUIRED
C
C
        DO 421 J=1,MM
        I=IXBR(J)
        XTD=UYA(I,3)*0.01
        IF(ABS(XTD).LT.1.0)XTD=1.0
        UYA(I,3)=UYA(I,3)+XTD
        II=I+1
        IF(UXA(II,3).EQ.UXA(I,3))UYA(II,3)=UYA(I,3)
        CALL MOGE(XST,NS,3,VE,VEX,VEZ,3,IDDF,CIF,UXA,UYA,KNDF,IPX)
        IF(IPX.NE.0)GO TO 807

        UYA(I,3)=UYA(I,3)-XTD
        IF(UXA(II,3).EQ.UXA(I,3))UYA(II,3)=UYA(I,3)
        CALL REFRA(LS,NC,XST,NS,ILS,NSP,DIR,RAF)
        IST=IST+1
        CALL SENS(IST,NR,XTD,RAF,GES,DRI,WEI)
  421   CONTINUE
C
C
C      FIND THE DIFFERENCE BETWEEN THE OBSERVATION AND UPDATE MODEL
C
C
  420   CALL DEFR(OES,GES,OGF,NR)
        NPRAM=MM+MN+MV(1)+MV(2)
        AMX=0.
        DO 281 I=1,NPRAM
        IF(WEI(I).GT.AMX)AMX=WEI(I)
  281   CONTINUE
C
C
C      NORMALIZE THE SENSITIVITY MATRIX
C
C
        CALL RNORM(DRI,WEI,NR,NPRAM,AMX)
C
C
C      SUBROUTINE SVD PERFORMS THE SINGULAR VALUE DECOMPOSITION
C      AND RETURN THE DECOMPOSED MATRICES
C      DRI ----- SENSITIVITY MATRIX AND RETURNED AS THE EIGENVECTORS
C                ASSOCIATED WITH THE COLUMN OF THE SENSITIVITY MATRIX
C      EG ------ EIGENVALUE ARRAY
C      VEG ----- EIGENVECTORS ASSOCIATED WITH THE ROW OF THE SENSITIVITY
C                MATRIX
      CALL SVD(DRI,EG,VEG,NR,NPRAM,NR,NPRAM,0,0,0)
        WRITE(17,*)(EG(L),L=1,NPRAM)
C
C
C      STOP WHEN THE EIGENVALUE ARE TOO SMALL
C
C
        IF(EG(NPRAM).LT.1E-4)GO TO 809
        DO 335 I=1,NR
        OGF(I)=OGF(I)*1000.
  335   CONTINUE
```

```
C
C
C     SUBROUTINE VTRAN FINDS THE TRANSPOSE OF THE INPUT MATRIX,
C     SUBROUTINE VMUL FINDS THE PRODUCT OF TWO MATRICES
C
C
      CALL VTRAN(DRI,VTR,NR,NPRAM)
      CALL VMUL(VTR,OCF,CAL,NPRAM,NR,1)
      DO 280 I=1,NPRAM
      CAL(I)=CAL(I)/EG(I)
 280  CONTINUE
      CALL VMUL(VEG,CAL,DEP,NPRAM,NPRAM,1)
C
C
C     DEP CONTAIN ALL THE CORRECTIONS OF THE MODEL
C
C
      CALL RNORM(DEP,WEI,1,NPRAM,AMX)
      LC=0
      DO 251 I=1,LN1
      II=I+1
      IF(MV(I).EQ.0)GO TO 251
      LC=LC+1
      VE(II)=VE(II)+DEP(LC)
 251  CONTINUE
      IF(MX1.EQ.0)GO TO 253
      IDEX=0
      DO 252 I=1,MN
      LK=IXAR(I)
      LC=LC+1
      UYA(LK,2)=UYA(LK,2)+DEP(LC)
C
C
C     PUT THE CONSTRAINTS TO THE CORRECTIONS
C
C
      CALL GETR(UXA(LK,2),UYA(LK,2),2,IDEX,GSTA)
      LK1=LK+1
      IF(UXA(LK1,2).EQ.UXA(LK,2))UYA(LK1,2)=UYA(LK,2)
 252  CONTINUE
      IF(IDEX.NE.0)GO TO 1400
 253  IF(MX2.EQ.0)GO TO 1400
      DO 254 I=1,MM
      LK=IXER(I)
      LC=LC+1
      UYA(LK,3)=UYA(LK,3)+DEP(LC)
      LK1=LK+1
      IF(UXA(LK1,3).EQ.UXA(LK,3))UYA(LK1,3)=UYA(LK,3)
 254  CONTINUE
      GO TO 1400
 808  WRITE(6,*)ERE
      WRITE(6,*)'INPUT NCON'
      READ(6,1001)NCON
      IF(NCON.EQ.0)GO TO 132
      GO TO 1200
 809  PRINT*,'EIGENVALUE TOO SMALL'
      GO TO 132
```

```
 807    IF(IPX.EQ.1)PRINT*,'IMPROPER GEOMETRY'
        IF(IPX.EQ.2)PRINT*,'STATION OUT OF BOUNDARY'
        IF(IPX.EQ.3)PRINT*,'NO REFRACTION DATA'
        IF(IPX.EQ.4)PRINT*,'BAD SUBSURFACE'
        IF(IPX.EQ.5)PRINT*,'BAD LOWER SUBSURFACE'
 132 PRINT*,' PROGRAM FINISHED'
1001    FORMAT(I2)
1002    FORMAT(/,6(1X,F8.1))
1004    FORMAT(8(1X,F8.6))
        END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      CREATE THE SENSISIVITY MATRIX
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
        SUBROUTINE SENS(IST,NR,DIF,A,B,C,W)
        DIMENSION A(1),B(1),C(1),D(200),W(1)
        NWE=IST*NR
        NW=NWE-NR+1
        J=0
        ACUM=0.
        CALL DEFR(A,B,D,NR)
        DO 123 I=NW,NWE
        J=J+1
        C(I)=D(J)*1000./DIF
        ACUM=ACUM+C(I)*C(I)
 123    CONTINUE
C
C      W IS THE WEIGHTING FACTOR OF ONE SPECIFIED PARAMETER
C
        W(IST)=SQRT(ACUM)
        RETURN
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      MATRIX NORMALIZATION
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
        SUBROUTINE RNORM(A,W,NR,NN,XN)
        DIMENSION A(1),W(1)
        DO 120 I=1,NN
        II=(I-1)*NR+1
        IJ=I*NR
        WT=1./W(I)
        DO 110 L=II,IJ
        A(L)=A(L)*XN*WT
  110   CONTINUE
  120   CONTINUE
        RETURN
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      MATRIX TRANSPOSE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
        SUBROUTINE VTRAN(A,B,M,N)
        DIMENSION A(M,N),B(N,M)
        DO 100 I=1,M
        DO 100 J=1,N
        B(J,I)=A(I,J)
  100   CONTINUE
        RETURN
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      SUBROUTINE DEFR CALCULATES THE DIFFERENCE ARRAY FROM TWO GIVEN
C      ARRAYS (D = A - B)
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE DEFR(A,B,D,NC)
      DIMENSION A(1),B(1),D(1)
      DO 101 I=1,NC
      D(I)=A(I)-B(I)
  101 CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      SUBROUTINE GETR CHECKS AND LIMITS THE CORRECTIONS OF THE DEPTH
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE GETR(X,Y,IND,IDEX,GSTA)
      INCLUDE 'MODIN/NOLIST'
      IPX=0
C
C
C      CHECK THE CORRECTIONS IF THEY ARE REASONABLE
C
C
      IF(IND.EQ.2)CALL CHECK(1,A1,IA1,X,Y,IPX,ST)
      IF(IND.EQ.3)CALL CHECK(2,A2,IA2,X,Y,IPX,ST)
      IF(IPX.EQ.0)RETURN
C
C
C      IF THE CORRECTIONS ARE NOT REASONABLE, PUT THE CONSTRAINTS
C
C
      IDEX=IDEX+1
      Y=ST-GSTA
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      MATRIX MULTIPLICATION
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
        SUBROUTINE VMUL(A,U,T,M,N,L)
        DIMENSION A(M,N),U(N,L),T(M,L)
        DO 100 I=1,M
        DO 100 K=1,L
  100   T(I,K)=0.
        DO 203 I=1,M
        DO 202 J=1,L
        DO 201 K=1,N
        T(I,J)=T(I,J)+A(I,K)*U(K,J)
  201   CONTINUE
  202   CONTINUE
  203   CONTINUE
        RETURN
        END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      CHECK THE THICKNESS OF THE WEATHERING LAYER IF IT IS NEGATIVE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE CHECK(LA,A,IA,SSR,SSM,IFX,ST)
      DIMENSION A(1),IA(1)
      LPH=LOCA(SSR,LA,0)
      IF(LPH.EQ.0)GO TO 33
      LM=4*LPH
      CALL FVAL(ST,A,LM,IA(LPH),SSR)
      DE=ST-SSM
      IF(DE.LT.0.)IFX=4
      RETURN
   33 IFX=2
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      SET THE INITIAL PARAMETERS FOR THE PROGRAMS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE MOCE(LDN,DDI1,XST,NDC,ILDS,EDR,DDIR,ERDIN,NSP)
      DIMENSION NSP(2)
      INCLUDE 'MODIN/LIST'
      LN=LDN
      DI1=DDI1
      ER=EDR
      ILS=ILDS
      NC=NDC
      IDIR=IDDIR
      ERIN=ERDIN
      NRT=ILS+NC
      IF(DDIR.EQ.0.)GO TO 111
      IF(DDIR.EQ.1.)GO TO 112
      NSP(1)=NRT
      RETURN
  112 NSP(1)=1
      RETURN
  111 NSP(1)=1
      NSP(2)=NRT
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      COMPARE THE ARRIVAL TIMES TO FIND THE FIRST ARRIVAL
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE REIN(G1,G2,NC,LEN,LEE)
      DIMENSION G1(1),G2(1)
      DO 20 I=LEN,LEE
      IF(LEN.EQ.1)GO TO 22
      IJ=I-NC
      GO TO 24
   22 IJ=I
   24 IF(G2(IJ).LT.0.)GO TO 20
      IF(G2(IJ).GT.G1(I))GO TO 20
      G1(I)=G2(IJ)
   20 CONTINUE
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      SET THE BASIC MODEL PARAMETERS FROM THE INPUT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE MOGE(XST,NS,M,VE,IDF,DIR,UXA,UYA,IPX)
        DIMENSION UXA(30,3),UYA(30,3),UX(20),UY(20),VE(3)
        INCLUDE 'MODIN/LIST'
      IPX=0
      V(M)=VE(M)
      IF(IDF.EQ.0)GO TO 145
  193 ISR=1
      ICL=0
      I=1
      IT=1
        UX(I) = UXA( IT,M )
        UY(I) = UYA( IT,M )
      IG=1
      CALL EQV(M,IG,UX(I),ICR)
      IF(ICR.EQ.0)GO TO 11
      IPX=1
      RETURN
   11 I=I+1
      IT=IT+1
      IG=IG+1
        UX(I) = UXA( IT,M )
        UY(I) = UYA( IT,M )
      IF(UY(I).EQ.10000.)GO TO 13
      IF(UX(I-1).EQ.UX(I).AND.UY(I-1).EQ.UY(I))GO TO 12
      CALL EQV(M,IG,UX(I),ICR)
      IF(ICR.EQ.0)GO TO 11
      IPX=1
      RETURN
   12 NL=I-1
      CALL SUBS(M,NL,ICL,UX,UY)
      UX(1)=UX(I)
      UY(1)=UY(I)
      I=1
      IG=IG-1
      GO TO 11
   13 NL=I-1
      CALL SUBS(M,NL,ICL,UX,UY)
      IF(M.EQ.1)ICA=IG-2
      IF(M.EQ.2)ICB=IG-2
      IF(M.EQ.3)ICC=IG-2
  145 IF(M.EQ.1)GO TO 508
      IF(M.EQ.3)GO TO 778
      IF(V(1).GT.V(2))GO TO 888
      INA=1
C
C     CRA,CRB ----CRITICAL ANGLES
C
      CRA=ASIN(V(1)/V(2))
      GO TO 502
  888 INA=2
```

```
 778 IF(V(2).GT.V(3))GO TO 889
     INE=1
     CRB=ASIN(V(2)/V(3))
     GO TO 502
 889 INE=2
 502 IF(IDF.EQ.0)GO TO 505
 508 CALL SEGM(M)
     IF(M.NE.1)GO TO 505
     XPH(1)=XST
     LPH=LOCA(XST,1,0)
     IF(LPH.NE.0)GO TO 498
     IPX=2
     RETURN
 498 LM=4*LPH
     CALL FVAL(YPH(1),A1,LM,IA1(LPH),XPH(1))
     DO 503 L=2,NS
     XPH(L)=XPH(L-1)+DI1
     LPH=LOCA(XPH(L),1,0)
     IF(LPH.NE.0)GO TO 499
     IPX=2
     RETURN
 499 LM=4*LPH
     CALL FVAL(YPH(L),A1,LM,IA1(LPH),XPH(L))
 503 CONTINUE
     RETURN
 505 IF(LN.EQ.2.AND.INA.EQ.2)IPX=3
     IF(LN.EQ.3.AND.INA.EQ.2.AND.INB.EQ.2)IPX=3
     IF(IPX.EQ.3)RETURN
     IF(IDF.EQ.0)RETURN
     IF(M.NE.2)GO TO 904
     NIB=ICB+1
     DO 804 I=1,NIB
     CALL CHECK(1,A1,IA1,SR2(I),SM2(I),IPX,SUT)
     IF(IPX.EQ.4)RETURN
 804 CONTINUE
 904 IF(M.NE.3)GO TO 905
     NIB=ICC+1
     DO 805 I=1,NIB
     CALL CHECK(2,A2,IA2,SR3(I),SM3(I),IPX,SUT)
     IF(IPX.NE.0)RETURN
 805 CONTINUE
 905 IF(LN.EQ.3.AND.IDF.EQ.2)GO TO 906
     GO TO 907
 906 NIB=ICC+1
     DO 806 I=1,NIB
     CALL CHECK(2,A2,IA2,SR3(I),SM3(I),IPX,SUT)
     IF(IPX.EQ.0)GO TO 806
     IPX=5
     RETURN
 806 CONTINUE
 507 CALL NITE
     RETURN
3005 FORMAT(3X,I5,3X,I5,3X,I5)
3001 FORMAT(10(1X,F11.5))
     END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      REFRA IS A MAIN SUBROUTINE TO HANDLE THE WHOLE MODELING PROCESS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE REFRA(LS,NC,XST,NS,ILS,NSP,DIR,RAF)
      DIMENSION TST(100),TS1T(100),TS2T(100),TS(100),TS1(100),TS2(100)
      DIMENSION RAF(1),NSP(2)
        INCLUDE 'MODIN/LIST'
      CAL SESET(XST,NS)
      DO 123 I=2,LN
      LT=I-1
  123 CONTINUE
      ILL=NC/2
      DO 335 LH=1,LS
      NOS=NSP(LH)
      OSX=XPH(NOS)
      OSY=YPH(NOS)
      IF(DIR.EQ.0..AND.LH.EQ.2)GO TO 567
      IF(DIR.EQ.-1.)GO TO 567
      IDOR=1
      IFS=NOS+ILS
      IFE=IFS+NC-1
      IF(IFS.GT.NS)GO TO 335
      IHT=IFE
      IF(IFE.GT.NS)IHT=NS
      LB=IHT-IFS+1
      CALL RAYM(NOS,OSX,OSY,1.,IDOR,LB,IFS,IHT,TS,TS1,TS2)
      GO TO 334
  567 IFS=NOS-ILS
      IFE=IFS-NC+1
      IDOR=-1
      IF(IFS.LT.1)GO TO 335
      IHT=IFE
      IF(IFE.LT.1)IHT=1
      LB=IFS-IHT+1
      CALL RAYM(NOS,OSX,OSY,-1.,IDIR,LB,IFS,IHT,TST,TS1T,TS2T)
      DO 331 I=1,LB
      IP=LB-I+1
      TS(I)=TST(IP)
      TS1(I)=TS1T(IP)
      IF(LN.EQ.3)TS2(I)=TS2T(IP)
  331 CONTINUE
      GO TO 334
  568 IFSL=NOS
      IFEL=NOS-ILL+1
      IDOR=0
      IF(IFSL.LT.1)GO TO 570
      IHT=IFEL
      IF(IFEL.LT.1)IHT=1
      LBB=IFSL-IHT+1
      CALL RAYM(NOS,OSX,OSY,-1.,IDOR,LBB,IFSL,IHT,TST,TS1T,TS2T)
      DO 332 I=1,LBB
      IP=LBB-I+1
      TS(I)=TST(IP)
```

```
      TS1(I)=TS1T(IP)
      IF(LN.EQ.3)TS2(I)=TS2T(IP)
  332 CONTINUE
  570 IFSR=NOS+1
      IFER=NOS+ILL
      IHT=IFER
      IDCR=0
      IF(IFSR.GT.NS)GO TO 335
      IF(IFER.GT.NS)IHT=NS
      LB=LEB+IHT-IFSR+1
      LB1=LBB+1
      CALL RAYM(NOS,OSX,OSY,1.,IDCR,LBB,IFSR,IHT,TST,TS1T,TS2T)
      DO 333 I=LB1,LB
      IP=I-LBB
      TS(I)=TST(IP)
      TS1(I)=TS1T(IP)
      IF(LN.EQ.3)TS2(I)=TS2T(IP)
  333 CONTINUE
  334 IF(LH.EQ.1)LEN=1
      IF(LH.EQ.2)LEN=NC+1
      LEE=LB+LEN-1
      IM=0
      DO 203 I=LEN,LEE
      IM=IM+1
      RAF(I)=TS(IM)
  203 CONTINUE
      CALL REIN(RAF,TS1,NC,LEN,LEE)
      IF(LN.EQ.3)CALL REIN(RAF,TS2,NC,LEN,LEE)
  335 CONTINUE
      RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     RAYM FINDS ALL KINDS OF DIRECT AND REFRACTED ARRIVALS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE RAYM(NOS,OSX,OSY,DIR,IDOR,LB,IFS,IHT,TST,TS1T,TS2T)
      DIMENSION TST(1),TS1T(1),TS2T(1)
      INCLUDE '*MODIN/NOLIST'
      I=1
      IOFR=0
      IDOP=0
      IDIR=INT(DIR)
      DER=-DIR
      DO 501 I=1,100
      TS1T(I)=-0.1
      TS2T(I)=-0.1
  501 CONTINUE
      CALL SURW(OSX,OSY,DIR,IFS,IHT,TST)
```

```
      DO 671 IND=2,LN
      IF(INA.EQ.2.AND.IND.EQ.2)GO TO 671
      IF(INB.EQ.2.AND.IND.EQ.3)GO TO 671
      CALL RAYC(IND,NOS,IFS,IHT,OSX,OSY,DIR,DER,0.,TDC,XDC,YDC,ICU)
      IF(ICU.NE.0)GO TO 671
      CALL RAYU(IND,DIR,XDC,YDC,XCR,YCR,TIN,INX)
      IF(INX.EQ.0)GO TO 534
      XCR=XDC
      TIN=0.
  534 TCR=TDC+TIN
      TRP=TDC
      CRP=XDC
      CALL DETE(DIR,IFS,IHT,XCR,IFT)
      IF(IFT.EQ.0)GO TO 671
      GXN=XDC
      II=IDIR*(IFT-IFS)
      IF(II.EQ.0)GO TO 890
      DO 888 J=1,II
      IF(IND.EQ.2)TS1T(J)=-0.1
      IF(IND.EQ.3)TS2T(J)=-0.1
  888 CONTINUE
  890 JE=IDIR*(IHT-IFT)+1
      DO 672 J=1,JE
      IJ=IFT+IDIR*(J-1)
      JJ=II+J
      FX=XPH(IJ)
      FY=YPH(IJ)
      CALL RAYC(IND,NOS,IJ,IHT,FX,FY,DIR,DIR,XDC,TD,XD,YD,ICU)
      IF(ICU.NE.0)GO TO 601
      GU=(XD-GXN)*DIR
      IF(GU.LT.0.)GO TO 302
      AX=GXN
      BX=XD
      IF(IND.EQ.2)CALL GRAT(P2,A2,2,SR2,IA2,SM2,AX,BX,V(2),TK,AT2)
      IF(IND.EQ.3)CALL GRAT(P3,A3,3,SR3,IA3,SM3,AX,BX,V(3),TK,AT3)
      TRP=TRP+TK
      IF(IND.EQ.2)TS1T(JJ)=TRP+TD
      IF(IND.EQ.3)TS2T(JJ)=TRP+TD
      GXN=XD
      GO TO 672
  302 AX=XDC
      BX=XD
      IF(IND.EQ.2)CALL GRAT(P2,A2,2,SR2,IA2,SM2,AX,BX,V(2),TK,AT2)
      IF(IND.EQ.3)CALL GRAT(P3,A3,3,SR3,IA3,SM3,AX,BX,V(3),TK,AT3)
      IF(IND.EQ.2)TS1T(JJ)=TDC+TK+TD
      IF(IND.EQ.3)TS2T(JJ)=TDC+TK+TD
      GO TO 672
  601 IF(IND.EQ.2)TS1T(JJ)=-0.1
      IF(IND.EQ.3)TS2T(JJ)=-0.1
  672 CONTINUE
  671 CONTINUE
 2900 FORMAT(5X,I4,3X,I4,3X,I4,3X,I4)
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       SEARCH THE CRITICAL RAY ASSOCIATED TO THE GIVEN SURFACE POINT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE RAYC(IND,NOS,IFS,IHT,OSX,OSY,DIR,DER,XDD,TDC,XDC,YDC,
     &ICU)
      DIMENSION IGN(10),TUP(10),XUP(10),YUP(10)
      INCLUDE 'MOCIN/NCLIST'
      ICU=0
      TCC=0.
      I=0
      J0=1
      IDX=-INT(DER)
      IF(IND.EQ.3)GO TO 723
      NPK=NSR(2)
      CALL CHFG(NOS,2,DIR,DER,IFS,IHT,NPK,IS,IE)
      IF(DER.EQ.1.)CALL RAGE(RSHX2,OSX,IDX,IGN,IS,IE)
      IF(DER.EQ.-1.)CALL RAGE(ESHX2,OSX,IDX,IQN,IS,IE)
      GO TO 640
  723 NPK=NSR(3)
      CALL CHFG(NOS,3,DIR,DER,IFS,IHT,NPK,IS,IE)
      IF(DER.EQ.1.)CALL RAGE(RSHX3,OSX,IDX,IQN,IS,IE)
      IF(DER.EQ.-1.)CALL RAGE(ESHX3,OSX,IDX,IGN,IS,IE)
  640 IF(IQN(1).EQ.0)GO TO 593
  641 I=I+1
      TOR=1E5
      IF(I.GT.10.OR.IGN(I).EQ.0)GO TO 630
      CALL FIND(IND,DER,OSX,OSY,IGN(I),XUP(I),YUP(I),TUP(I),ISD)
      GO TO 641
  630 IF(DIR.NE.DER)GO TO 650
      I=I-1
      DO 643 J=1,I
      IF(TUP(J).GT.1E4)GO TO 643
      SHX=DIR*(XUP(J)-XDD)
      IF(SHX.LT.0.)GO TO 643
      AX=XDD
      BX=XUP(J)
      IF(IND.EQ.2)CALL GRAT(P2,A2,2,SR2,IA2,SM2,AX,BX,V(2),TI,AT2)
      IF(IND.EQ.3)CALL GRAT(P3,A3,3,SR3,IA3,SM3,AX,BX,V(3),TI,AT3)
      TTS=TI+TUP(J)
      IF(TOR.LE.TTS)GO TO 643
      TOR=TTS
      JC=J
  643 CONTINUE
      IF(TOR.GT.1E4)GO TO 593
      TDC=TUP(JC)
      XDC=XUP(JC)
      YDC=YUP(JC)
      RETURN
  650 I=I-1
      TOR=TUP(1)
      IF(I.EQ.1)GO TO 705
      DO 653 J=2,I
      IF(TUP(J).GT.1E4)GO TO 653
```

```
      AX=XUP(J)
      BX=XUP(JO)
      IF(IND.EQ.2)CALL GRAT(P2,A2,2,SR2,IA2,SM2,AX,BX,V(2),TI,AT2)
      IF(IND.EQ.3)CALL GRAT(P3,A3,3,SR3,IA3,SM3,AX,BX,V(3),TI,AT3)
      TTS=TI+TUP(JO)
      IF(TTS.LE.TUP(J))GO TO 653
      TOR=TUP(J)
      JO=J
  653 CONTINUE
  705 IF(TOR.GT.1E4)GO TO 993
      TOC=TUP(JO)
      XOC=XUP(JO)
      YOC=YUP(JO)
      RETURN
  993 ICU=8
      WRITE(6,1300)IND
      RETURN
 1300 FORMAT(2X,I1,3X,27HHARD TO FIND CRITICAL POINT)
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      CALCULATE THE TRAVEL TIME AND OUTGOING POINT FROM A DEEPER
C      REFRACTION POINT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE RAYU(IND,DIR,GX,GY,XCR,YCR,TCR,INX)
      INCLUDE 'MODIN/NOLIST'
      INX=0
      IUL=0
      IF(IND.EQ.3)GO TO 921
  146 LP=LOCA(GX,2,1)
      LM=LP*4
      CALL FVAL(GY,A2,LM,IA2(LP),GX)
      LZ=LP*3
      CALL SLOPE(P2,LZ,GX,SS,IA2(LP))
      THE=ATAN(SS)+DIR*(1.5707964-CRA)
  164 DCHE=DIR*THE
      IF(DCHE.LT.0.01.AND.DCHE.GT.-1.0)GO TO 238
      CALL BIGC(1,GX,GY,THE,XCR,YCR,A1,IA1,ICA,EF,LP)
      CALL SELFC(THE,2,GX,GY,XCR,YCR,SR2,SM2,IYES)
      IF(IYES.EQ.0)GO TO 487
      TCR=1E5
      RETURN
  487 TCR=SQRT((XCR-GX)**2+(YCR-GY)**2)/V(1)
      RETURN
  921 LP=LOCA(GX,3,1)
      LM=LP*4
      CALL FVAL(GY,A3,LM,IA3(LP),GX)
      LZ=LP*3
      CALL SLOPE(P3,LZ,GX,SS,IA3(LP))
      THE=ATAN(SS)+DIR*(1.5707964-CRB)
```

```
523 DCHE=DIR*THE
    IF(DCHE.LT.0.01.AND.DCHE.GT.-1.0)GO TO 238
    CALL BIGC(2,GX,GY,THE,XC,YC,A2,IA2,ICB,ER,LPP)
    IF(LPP.EQ.101.OR.LPP.EQ.-101)GO TO 238
    CALL SELFC(THE,3,GX,GY,XC,YC,SR3,SM3,IYES)
    IF(IYES.EQ.0)GC TO 356
    IUL=3
356 TCR1=SQRT((XC-GX)**2+(YC-GY)**2)/V(2)
    LZ=LPP*3
    CALL SLOPE(P2,LZ,XC,SR,IA2(LPP))
    TH1=ATAN(SR)
    CALL SNEL(THE,TH1,V(2),V(1),REG)
    DCHE=DIR*REG
    IF(DCHE.LT.0.01.AND.DCHE.GT.-1.0)GC TC 238
    CALL BIGC(1,XC,YC,REG,XCR,YCR,A1,IA1,ICA,ER,LPP)
    IF(IUL.NE.0)GO TO 808
    CALL SELFC(REG,2,XC,YC,XCR,YCR,SR2,SM2,IYES)
    IF(IYES.EQ.0)GO TO 489
808 TCR=1E5
    RETURN
489 TCR=TCR1+SQRT((XCR-XC)**2+(YCR-YC)**2)/V(1)
    RETURN
238 INX=9
    IF(DIR.EQ.1.)XCR=1E8
    IF(DIR.EQ.-1.)XCR=-1E8
    RETURN
    END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      FIND THE FUNCTION VALUE OF A POLYNOMIAL
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
       SUBROUTINE FVAL(Y,A,LM,N,X)
       DIMENSION A(1)
       Y=0.
       IF(N.LE.1)GO TO 202
       LMM=LM-N+1
       LM1=LM-1
       DO 201 J=LMM,LM1
       J1=LM-J
       Y=Y+A(J)*(X**J1)
  201  CONTINUE
  202  Y=Y+A(LM)
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      SET THE SAMPLE POINTS FOR THE SECOND REFRACTOR
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE BOSET(XST,NS)
      INCLUDE 'MODIN/NOLIST'
      L=2
      SLI=XST+FLOAT(NS-1)*DI1
      IP=LOCA(XST,3,0)
      IP=IP+1
      IH=INT(XST/DI1)
      IF(IH.EQ.0)GO TO 512
      IF(IH.GT.15)IH=15
      IHK=IH
      DINT(1)=XST-FLOAT(IH)*DI1
      IF(IH.EQ.1)GO TO 513
      DO 514 I=2,IH
      DINT(I)=DINT(I-1)+DI1
  514 CONTINUE
      GO TO 513
  512 DINT(1)=XST
      IH=1
      IHK=1
  513 IH1=IH+1
      IH2=IH+NS
      K=IH1
  505 ARB=DINT(K-1)+DI1
      IF(ARB.GE.SLI)GO TO 506
      IF(ARB.GT.SR3(L))GO TO 403
      DINT(K)=ARB
      K=K+1
      GO TO 505
  403 XTE=SR3(L)-DINT(K-1)
      IF(XTE.LT.0.2)GO TO 404
      DINT(K)=SR3(L)-0.2
      K=K+1
  404 DINT(K)=SR3(L)
      L=L+1
      K=K+1
      GO TO 505
  506 DINT(K)=SLI
      K=K+1
      IF(IH.EQ.0)GO TO 516
      IH1=K
      IH2=IH1+IH-1
      DO 517 I=IH1,IH2
      DINT(I)=DINT(I-1)+DI1
  517 CONTINUE
      NSR(3)=IH2
      RETURN
  516 DINT(K)=DINT(K-1)+XST
      NSR(3)=K
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      SET THE SAMPLE POINTS FOR THE FIRST REFRACTOR
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE SESET(XST,NS)
      INCLUDE 'MODIN/NCLIST'
      L=2
      SLI=XST+FLOAT(NS-1)*DI1
      IF(LN.EQ.3)CALL ECSET(XST,NS)
      IP=LOCA(XST,2,C)
      IP=IP+1
      IH=INT(XST/DI1)
      IF(IH.EQ.0)GO TO 512
      IF(IH.GT.15)IH=15
      IHK=IH
      DENT(1)=XST-FLOAT(IH)*DI1
      IF(IH.EQ.1)GO TO 513
      DO 514 I=2,IH
      DENT(I)=DENT(I-1)+DI1
  514 CONTINUE
      GO TO 513
  512 DENT(1)=XST
      IH=1
      IHK=1
  513 IH1=IH+1
      IH2=IH+NS
      K=IH1
  505 ARB=DENT(K-1)+DI1
      IF(ARB.GE.SLI)GO TO 506
      IF(ARB.GT.SR2(L))GO TO 403
      DENT(K)=ARB
      K=K+1
      GO TO 505
  403 XTE=SR2(L)-DENT(K-1)
      IF(XTE.LT.0.2)GO TO 404
      DENT(K)=SR2(L)-0.2
      K=K+1
  404 DENT(K)=SR2(L)
      L=L+1
      K=K+1
      GO TO 505
  506 DENT(K)=SLI
      K=K+1
      IF(IH.EQ.0)GO TO 516
      IH1=K
      IH2=IH1+IH-1
      DO 517 I=IH1,IH2
      DENT(I)=DENT(I-1)+DI1
  517 CONTINUE
      NSR(2)=IH2
      RETURN
  516 DENT(K)=DENT(K-1)+XST
      NSR(2)=K
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      LOCATION DETERMINATION BETWEEN TWO SAMPLE POINTS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE RAGE(G,X1,IDX,IQN,IS,IE)
      DIMENSION G(1),IQN(1)
      DO 370 I=1,10
      IQN(I)=0
  370 CONTINUE
      J=1
      IF(IDX.EQ.0)GO TO 350
      IF(IDX.EQ.-1)GO TO 343
      I=IS
      A=G(I)-X1
  543 B=G(I+1)-X1
      C=A*B
      IF(C.GT.0)GO TO 544
      IQN(J)=I
      J=J+1
      I=I+1
      IF(I.GE.IE)RETURN
      A=B
      GO TO 543
  544 A=B
      I=I+1
      IF(I.GE.IE)RETURN
      GO TO 543
  343 I=IE
      A=G(I+1)-X1
  344 B=G(I)-X1
      C=A*B
      IF(C.GT.0)GO TO 345
      IQN(J)=I
      J=J+1
      I=I-1
      IF(I.LE.IS)RETURN
      A=B
      GO TO 344
  345 A=B
      I=I-1
      IF(I.LE.IS)RETURN
      GO TO 344
  350 I=IS
      A=G(I)-X1
  352 B=G(I+1)-X1
      C=A*B
      IF(C.GT.0)GO TO 351
      IQN(J)=I
      RETURN
  351 A=B
      I=I+1
      IF(I.GE.IE)RETURN
      GO TO 352
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      USING INTERPOLATION METHOD TO FIND THE ACCURATE POINT WHERE THE
C      CRITICAL RAY HITS THE SPECIFIED GEOPHONE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE RAYJ(OX,OY,X1,X2,C1,C2,D1,D2,T1,T2,XDC,YDC,TDC,
     $  IND,DIR,ISD)
      INCLUDE 'MODIN/NOLIST'
      NCON=1
      ISD=0
      ERP1=SQRT((C1-OX)**2+(D1-OY)**2)
      IF(ERP1.GE.ER)GO TO 611
      IF(T1.EQ.1E5)GO TO 683
      XDC=X1
      LP=LOCA(XDC,IND,1)
      LM=LP*4
      IF(IND.EQ.2)CALL FVAL(YDC,A2,LM,IA2(LP),XDC)
      IF(IND.EQ.3)CALL FVAL(YDC,A3,LM,IA3(LP),XDC)
      TDC=T1
      RETURN
  611 ERP2=SQRT((C2-OX)**2+(D2-OY)**2)
      IF(ERP2.GT.ER)GO TO 612
      IF(T2.EQ.1E5)GO TO 683
      XDC=X2
      LP=LOCA(XDC,IND,1)
      LM=LP*4
      IF(IND.EQ.2)CALL FVAL(YDC,A2,LM,IA2(LP),XDC)
      IF(IND.EQ.3)CALL FVAL(YDC,A3,LM,IA3(LP),XDC)
      TDC=T2
      RETURN
  612 DD=ADJ(X1,X2,C1,C2,OX,DI1)
      CALL RAYU(IND,DIR,DD,DF,CX,CY,TIM,INX,0,0.)
      IF(INX.NE.0)GO TO 677
      ERR=SQRT((CX-OX)**2+(CY-OY)**2)
      IF(ERR.LT.ER)GO TO 682
  677 NCON=NCON+1
      IF(NCON.GT.15)GO TO 683
      IF(INX.EQ.0)GO TO 678
      IF(CX.EQ.C1)GO TO 621
      IF(CX.EQ.C2)GO TO 620
      GO TO 683
  678 ATE=(CX-OX)*(C1-OX)
      IF(ATE.GT.0.)GO TO 621
  620 C2=CX
      D2=CY
      X2=DD
      GO TO 612
  621 C1=CX
      D1=CY
      X1=DD
      GO TO 612
  682 XDC=DD
      YDC=DF
      TDC=TIM
      RETURN
```

```
  683 ISO=2
      TOC=1E5
      RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       LOCATION DETERMINATION CALLING PROGRAM
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE CHRG(NOS,INO,DIR,DER,IFS,IHT,NSS,IS,IE)
      DIMENSION IQN(10)
      INCLUDE 'MODIN/NOLIST'
      CTE=XPH(NOS)
      DTE=XPH(IFS)
      ETE=XPH(IHT)
      IF(IND.EQ.3)GO TO 109
      CALL RAGE(DENT,CTE,0,IQN,NOS,NSS)
      ILP=IQN(1)
      CALL RAGE(DENT,DTE,0,IQN,IFS,NSS)
      ILQ=IQN(1)
      CALL RAGE(DENT,ETE,0,IQN,IHT,NSS)
      ILR=IQN(1)
      GO TO 105
  109 CALL RAGE(DINT,CTE,0,IQN,NOS,NSS)
      ILP=IQN(1)
      CALL RAGE(DINT,CTE,0,IQN,IFS,NSS)
      ILQ=IQN(1)
      CALL RAGE(DINT,ETE,0,IQN,IHT,NSS)
      ILR=IQN(1)
  105 IF(DIR.EQ.1.)GO TO 121
      IF(DIR.EQ.DER)IS=ILR-20
      IF(DIR.NE.DER)IS=ILQ-20
      IF(IS.LT.1)IS=1
      IE=ILP+20
      IF(IE.GT.NSS)IE=NSS
      RETURN
  121 IS=ILP-20
      IF(IS.LE.0)IS=1
      IF(DIR.EQ.DER)IE=ILQ+20
      IF(DIR.NE.DER)IE=ILR+20
      IF(IE.GT.NSS)IE=NSS
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      FIND THE CRITICAL RAY WHICH HITS THE SPECIFIED RECEIVER
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE FIND(IND,DIR,OX,OY,IGN,XDC,YDC,TDC,ISD)
      INCLUDE 'MODIN/NOLIST'
      IF(IND.EQ.3)GO TO 220
      X1=DENT(IGN)
      X2=DENT(IGN+1)
      XTE=ABS(X1-X2)
      IF(XTE.LT.0.5)GO TO 333
      IF(DIR.EQ.-1.)GO TO 440
      C1=RSHX2(IGN)
      C2=RSHX2(IGN+1)
      D1=RSHY2(IGN)
      D2=RSHY2(IGN+1)
      T1=RTIM2(IGN)
      T2=RTIM2(IGN+1)
      GO TO 222
  440 C1=ESHX2(IGN)
      C2=ESHX2(IGN+1)
      D1=ESHY2(IGN)
      D2=ESHY2(IGN+1)
      T1=ETIM2(IGN)
      T2=ETIM2(IGN+1)
      GO TO 222
  220 X1=DINT(IGN)
      X2=DINT(IGN+1)
      XTE=ABS(X1-X2)
      IF(XTE.LT.0.5)GO TO 333
      IF(DIR.EQ.-1.)GO TO 221
      C1=RSHX3(IGN)
      C2=RSHX3(IGN+1)
      D1=RSHY3(IGN)
      D2=RSHY3(IGN+1)
      T1=RTIM3(IGN)
      T2=RTIM3(IGN+1)
      GO TO 222
  221 C1=ESHX3(IGN)
      C2=ESHX3(IGN+1)
      D1=ESHY3(IGN)
      D2=ESHY3(IGN+1)
      T1=ETIM3(IGN)
      T2=ETIM3(IGN+1)
  222 CALL RAYJ(OX,OY,X1,X2,C1,C2,D1,D2,T1,T2,XDC,YDC,TDC,IND,DIR,ISD)
      RETURN
  333 ISD=2
      TDC=1E5
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      FIND THE CRITICAL RAYS FROM THE GIVEN SAMPLE POINTS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
       SUBROUTINE RAYTR(IND)
       INCLUDE 'MODIN/NOLIST'
       IF(IND.EQ.3)GO TO 520
       DO 523 I=1,NSR(2)
       DXX=DENT(I)
       CALL RAYU(2,1.,DXX,DYY,RX,RY,RTT,INX)
       RTIM2(I)=RTT
       RSHX2(I)=RX
       RSHY2(I)=RY
       CALL RAYU(2,-1.,DXX,DYY,RX,RY,RTT,INX)
       ETIM2(I)=RTT
       ESHX2(I)=RX
       ESHY2(I)=RY
   523 CONTINUE
       RETURN
   520 DO 562 I=1,NSR(3)
       DXX=DINT(I)
       CALL RAYU(3,1.,DXX,DYY,RX,RY,RTT,INX)
       RTIM3(I)=RTT
       RSHX3(I)=RX
       RSHY3(I)=RY
       CALL RAYU(3,-1.,DXX,DYY,RX,RY,RTT,INX)
       ETIM3(I)=RTT
       ESHX3(I)=RX
       ESHY3(I)=RY
   562 CONTINUE
       RETURN
       END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      INTERPOLATION OF TWO VALUES
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
       FUNCTION ADJ(XA,XB,XY,XZ,XT,DI)
       ARE=ABS(XY-XZ)-DI
       IF(ARE.GT.0)GO TO 944
       AR=(XY-XT)/(XZ-XT)
       ADJ=(AR*XB-XA)/(AR-1)
       RETURN
   944 ADJ=(XA+XB)*0.5
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      CALCULATE THE CUBIC SPLINE POLYNOMIAL FOR EACH SEGMENT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
       SUBROUTINE SUBS(M,NL,ICL,UX,UY)
       DIMENSION UX(1),UY(1)
       IF(NL.EQ.2)GO TO 80
       IF(NL.EQ.3)GO TO 81
       CALL CUBS(M,3,ICL,UX,UY)
   79  CALL CUBS(M,4,ICL,UX,UY)
       NL=NL-1
       DO 90 I=1,NL
       UX(I)=UX(I+1)
       UY(I)=UY(I+1)
   90  CONTINUE
       IF(NL.EQ.3)GO TO 82
       GO TO 79
   80  CALL CUBS(M,NL,ICL,UX,UY)
       RETURN
   81  CALL CUBS(M,NL,ICL,UX,UY)
   82  CALL CUBS(M,-1,ICL,UX,UY)
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      FIND THE LINE INTEGRATION FOR ALL THE SEGMENTS ON THE CURVE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
       SUBROUTINE NITE
       INCLUDE 'MODIN/NOLIST'
       DO 203 I=1,ICA
       CALL GEFAN(P1,IA1(I),I,SR1(I),SM1(I),SR1(I+1),SM1(I+1),V(1)
      &,AT1(I))
  203  CONTINUE
       DO 201 I=1,ICB
       CALL GEFAN(P2,IA2(I),I,SR2(I),SM2(I),SR2(I+1),SM2(I+1),V(2)
      &,AT2(I))
  201  CONTINUE
       IF(LN.EQ.2)RETURN
       DO 202 I=1,ICC
       CALL GEFAN(P3,IA3(I),I,SR3(I),SM3(I),SR3(I+1),SM3(I+1),V(3)
      &,AT3(I))
  202  CONTINUE
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      USING NEWTON'S METHOD TO FIND THE ZERO OF A POLYNOMIAL
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE SOLV(XG,YG,B,N2,A,LM,N,ER,IER)
      DIMENSION B(1),Q(4),A(1)
      LV=N-1
      Q(1)=0.
      Q(2)=3.*B(1)
      Q(3)=2.*B(2)
      Q(4)=B(3)
      XF=XG
      CALL FVAL(YF,B,N,N,XG)
      IF(ABS(YF).LT.ER)GO TO 303
      CALL SLOPE(Q,LV,XG,S,N)
      IF(ABS(S).LT.1E-5)GO TO 305
      XG=XG-YF/S
      XS=XG
      CALL FVAL(YS,B,N,N,XG)
      IF(ABS(YS).LT.ER)GO TO 303
      CALL SLOPE(Q,LV,XG,S,N)
      IF(ABS(S).LT.1E-5)GO TO 305
      XG=XG-YS/S
      XT=XG
      CALL FVAL(YT,B,N,N,XG)
      IF(ABS(YT).GT.ABS(YF).OR.ABS(YT).GT.ABS(YS))GO TO 304
  444 CALL FVAL(Y,B,N,N,XG)
      IF(ABS(Y).LT.ER)GO TO 303
      CALL SLOPE(Q,LV,XG,S,N)
      IF(ABS(S).LT.1E-5)GO TO 305
      XG=XG-Y/S
      GO TO 444
  303 CALL FVAL(YG,A,LM,N,XG)
      RETURN
  304 W1=YF*YS
      W2=YF*YT
      IF(W1.GT.0..AND.W2.GT.0.)GO TO 305
      IF(W1.LT.0.)CALL RESO(XF,XS,YF,YS,XG,B,N,N,ER)
      IF(W2.LT.0.)CALL RESO(XF,XS,YF,YS,XG,B,N,N,ER)
      CALL FVAL(YG,A,LM,N,XG)
      RETURN
  305 IER=2
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      DETERMINE THE LOCATION OF A GIVEN POINT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      FUNCTION LOCA(XC,M,LCD)
      INCLUDE 'MODIN/NOLIST'
      I=1
      IF(M.EQ.2)GO TO 65
      IF(M.EQ.3)GO TO 66
      IF(XC.LT.SR1(1))GO TO 64
      IF(XC.NE.SR1(ICA+1))GO TO 54
      LOCA=ICA
      RETURN
   54 IF(XC.LT.SR1(I+1))GO TO 67
      I=I+1
      IF(I.GT.ICA)GO TO 68
      GO TO 54
   65 IF(XC.LT.SR2(1))GO TO 64
      IF(XC.NE.SR2(ICB+1))GO TO 55
      LOCA=ICB
      RETURN
   55 IF(XC.LT.SR2(I+1))GO TO 67
      I=I+1
      IF(I.GT.ICB)GO TO 68
      GO TO 55
   66 IF(XC.LT.SR3(1))GO TO 64
      IF(XC.NE.SR3(ICC+1))GO TO 56
      LOCA=ICC
      RETURN
   56 IF(XC.LT.SR3(I+1))GO TO 67
      I=I+1
      IF(I.GT.ICC)GO TO 68
      GO TO 56
   67 LOCA=I
      RETURN
   68 IF(LCD.EQ.1)GO TO 69
      LOCA=0
      RETURN
   69 LOCA=101
      RETURN
   64 IF(LCD.EQ.1)GO TO 63
      LOCA=0
      RETURN
   63 LOCA=-101
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      FIND ZERO BY INTERPOLATION WHEN NEWTON'S METHOD FAILED
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE RESO(XR,XL,YR,YL,XG,B,LM,N,ER)
      DIMENSION B(N)
  172 XM=(XR+XL)/2.
      CALL FVAL(YA,B,LM,N,XM)
      IF(YA.GT.ER)GO TO 175
      XG=XM
      RETURN
  175 W1=YA*XR
      IF(W1.LT.0.)GO TO 170
      XR=XM
      GO TO 172
  170 XL=XM
      GO TO 172
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      SOLVE FOR THE INTERSECTION OF A LINE AND A CURVE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE SSC(THE,X,Y,YH,XG,YG,LM,N,A,ER,IER)
      DIMENSION A(1),B(4)
      N1=N-1
      N2=N-2
      TES=ABS(THE)-1.5707964
      IF(ABS(TES).LT.0.001)GO TO 411
      S=TAN(THE)
      BI=Y-S*X
      B(4)=A(LM)-BI
      B(3)=A(LM-1)-SU
      B(2)=A(LM-2)
      B(1)=A(LM-3)
      XG=X-(Y-YH)/S
  230 CALL SOLV(XG,YG,B,N2,A,LM,N,ER,IER)
      RETURN
  411 XG=X
      CALL FVAL(YG,A,LM,N,XG)
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      FIND THE LINE INTEGRAL ALONG A CURVED BOUNDARY BETWEEN TWO
C      GIVEN POINTS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE GRAT(P,A,NG,SR,IA,SM,SX,EX,V,TL,AT)
      DIMENSION P(1),SR(1),IA(1),SM(1),A(1),AT(1)
      I1=LOCA(SX,NG,1)
      LM=4*I1
      CALL FVAL(SY,A,LM,IA(I1),SX)
      I2=LOCA(EX,NG,1)
      LM=4*I2
      CALL FVAL(EY,A,LM,IA(I2),EX)
      IF(EX.GE.SX)GO TO 40
      IT=I1
      I1=I2
      I2=IT
      TSX=SX
      TSY=SY
      SX=EX
      SY=EY
      EX=TSX
      EY=TSY
   40 SXX=SX
      SYY=SY
      EXX=EX
      EYY=EY
      IY=I2-I1
      IF(IY.EQ.0)GO TO 81
      I=I1
      CALL GEFAN(P,IA(I),I,SXX,SYY,SR(I+1),SM(I+1),V,TL)
   84 I=I+1
      IY=IY-1
      IF(IY.EQ.0)GO TO 83
      TL=TL+AT(I)
      GO TO 84
   83 CALL GEFAN(P,IA(I),I,SR(I),SM(I),EXX,EYY,V,TLE)
      TL=TL+TLE
      RETURN
   81 CALL GEFAN(P,IA(I1),I1,SXX,SYY,EXX,EYY,V,TL)
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     INTEGRATION BETWEEN TWO POINTS AT ONE CURVE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE GEFAN(P,LEN,I,SXX,SYY,EXX,EYY,V,TL)
      DIMENSION P(1),SP(11),SI(11)
      IU=8
      IU1=9
      TL=0.
      N2=LEN-1
      IF(N2.GT.1.OR.DV.NE.0.)GO TO 71
      TL=TL+SQRT((EXX-SXX)**2+(EYY-SYY)**2)/V
      RETURN
   71 LZ=3*I
      H=ABS(EXX-SXX)/IU
      DO 72 J=1,IU1
      SI(J)=SXX+FLOAT(J-1)*H
      CALL FVAL(TP,P,LZ,N2,SI(J))
      SP(J)=SQRT(1.+TP**2)
   72 CONTINUE
      S1=SP(1)+SP(IU1)
      S2=0.
      DO 73 J=2,IU,2
      S2=S2+4.*SP(J)
   73 CONTINUE
      IU2=IU-1
      DO 74 J=3,IU2,2
      S1=S1+2.*SP(J)
   74 CONTINUE
      TL=ABS((S1+S2)*H/(3.*V))+TL
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     DETERMINE RAY DIRECTION BY SNELL'S LAW
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE SNEL(TH,THP,VI,VO,REG)
      IF(TH.GE.0.)DNR=1.
      IF(TH.LT.0.)DNR=-1.
      SNA=1.570796-ABS(TH)+DNR*THP
      RAV=VO*SIN(SNA)/VI
      RAG=ASIN(RAV)
      REG=DNR*(1.570796-RAG+DNR*THP)
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       FIND THE SLOPE OF A CURVE AT SPECIFIED POINT
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE SLOPE(P,LZ,X,S,N)
      DIMENSION P(1)
      S=0.
      IF(N.EQ.1)GO TO 809
      IF(N.EQ.2)GO TO 810
      NN=N-1
      LZZ=LZ-NN+1
      LZ1=LZ-1
      DO 800 I=LZZ,LZ1
      S=S+P(I)*(X**(LZ-I))
  800 CONTINUE
  810 S=S+P(LZ)
  809 RETURN
      END


CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C       FIND THE INTERSECTION OF A STRAIGHT LINE AND A CURVE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE BIGC(IND,OX,OY,SU,XG,YG,A,IA,IBO,ER,LPP)
      DIMENSION A(1),IA(1)
      LP=LOCA(OX,IND,1)
      LPP=LP
  799 LM=LPP*4
      IF(LPP.LE.IBO.AND.LPP.GE.1)GO TO 780
      IF(LPP.GT.IBO)GO TO 781
      LPP=-101
      RETURN
  781 LPP=101
      RETURN
  780 CALL FVAL(YO,A,LM,IA(LPP),OX)
      ERS=0.01*ER
      CALL SSC(SU,OX,OY,YO,XG,YG,LM,IA(LPP),A,ERS,IER)
      IF(IER.EQ.2)GO TO 792
      LPP=LOCA(XG,IND,1)
      IF(LPP.EQ.LP)RETURN
  792 IF(SU.GT.0.)GO TO 793
      LPP=LP-1
      LP=LPP
      GO TO 799
  793 LPP=LP+1
      LP=LPP
      GO TO 799
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      CALCULATE THE ARRIVAL TIME OF THE DIRECT WAVE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE SURW(OSX,OSY,DIR,IFS,IFE,TST)
      DIMENSION TST(1)
      INCLUDE 'MODIN/NOLIST'
      JE=INT(DIR)*(IFE-IFS)+1
      X1=XPH(IFS)
      X2=OSX
      CALL GRAT(P1,A1,1,SR1,IA1,SM1,X2,X1,V(1),DT,AT1)
      TST(1)=DT
      IF(JE.EQ.1)RETURN
      DO 111 J=2,JE
      IJ=IFS+INT(DIR)*(J-2)
      IJ1=IJ+INT(DIR)
      X1=XPH(IJ)
      X2=XPH(IJ1)
      CALL GRAT(P1,A1,1,SR1,IA1,SM1,X1,X2,V(1),DT,AT1)
      TST(J)=TST(J-1)+DT
  111 CONTINUE
      RETURN
      END



CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      DETERMINE THE RANGE OF THE GEOPHONES WHICH CAN RECEIVE THE
C      SPECIFIND WAVE
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE DETE(DIR,IS,IHT,XCR,IFT)
      INCLUDE 'MODIN/NOLIST'
      IFT=IS
      IF(DIR.EQ.-1.)GO TO 222
      IF (XCR.LE.XPH(IFT))RETURN
  226 IFT=IFT+1
      IF(IFT.GT.IHT)GO TO 229
      IF(XCR.GE.XPH(IFT))GO TO 226
      RETURN
  222 IF(XCR.GE.XPH(IFT))RETURN
  224 IFT=IFT-1
      IF(IFT.LT.IE)GO TO 229
      IF(XCR.LE.XPH(IFT))GO TO 224
      RETURN
  229 IFT=0
      RETURN
      END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      CHECK THE RAY IF IT CROSSES THE BOUNDARY OF ITSELF
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
       SUBROUTINE SELFC(AG,IND,XI,YI,XO,YO,SR,SM,IYES)
       DIMENSION SR(1),SM(1)
       IYES=0
       GG=ABS(ABS(AG)-1.5707964)
       IF(GG.LT.0.001)RETURN
       CALL FOUN(IND,XI,XO,IU,IV)
       IF(IU.EQ.0)RETURN
       DO 702 I=IU,IV
       XT=SR(I)
       YT=SM(I)
       SMU=ABS(TAN(AG))
       XD=ABS(XT-XI)
       YH=YI+SMU*XD
       DF=YH-YT
       IF(DF.GE.0.)GO TO 702
       IYES=1
       RETURN
  702  CONTINUE
       RETURN
       END




CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C      FIND THE INTERVALS BETWEEN TWO END POINTS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
       SUBROUTINE FOUN(IND,X1,X2,IU,IV)
       INCLUDE 'MODIN/NOLIST'
       IU=0
       IV=0
       I1=LOCA(X1,IND,1)
       I2=LOCA(X2,IND,1)
       IF(I1.EQ.I2)RETURN
       IF(I1.LT.I2)GO TO 333
       II=I1
       I1=I2
       I2=II
  333  IU=I1+1
       IV=I2
       RETURN
       END
```

```
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
C     SET THE VERTICAL COORDINATES OF THE GEOMETRY CONTROL POINTS
C
C
CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
C
C
      SUBROUTINE SEGM(I)
      INCLUDE 'MODIN/NOLIST'
      IF(I.EQ.2)GO TO 50
      IF(I.EQ.3)GO TO 60
      DO 776 J=1,ICA
      LM=4*J
      CALL FVAL(SM1(J),A1,LM,IA1(J),SR1(J))
  776 CONTINUE
      JJ=ICA+1
      CALL FVAL(SM1(JJ),A1,LM,IA1(J-1),SR1(JJ))
      RETURN
   50 DO 775 J=1,ICB
      LM=4*J
      CALL FVAL(SM2(J),A2,LM,IA2(J),SR2(J))
  775 CONTINUE
      JJ=ICB+1
      CALL FVAL(SM2(JJ),A2,LM,IA2(J-1),SR2(JJ))
      RETURN
   60 DO 774 J=1,ICC
      LM=4*J
      CALL FVAL(SM3(J),A3,LM,IA3(J),SR3(J))
  774 CONTINUE
      JJ=ICC+1
      CALL FVAL(SM3(JJ),A3,LM,IA3(J-1),SR3(JJ))
      RETURN
      END
```