NORSAR ROYAL NORWEGIAN COUNCIL FOR SCIENTIFIC AND INDUSTRIAL RESEARCH

Norsar Technical Report No. 4/80

A GENERAL COMPUTER ALGORITHM FOR CALCULATING ZERO-VALUE CONTOURS OF A BIVARIATE SCALAR FUNCTION, WITH SPECIAL APPLICATION TO SEISMIC RAY-TRACING IN COMPLEX 3-DIMENSIONAL MODELS

by
Håvar Gjøystdal

June 1978

NORSAR Contribution No. 279

A GENERAL COMPUTER ALGORITHM FOR CALCULATING ZERO-VALUE CONTOURS OF A BIVARIATE SCALAR FUNCTION, WITH SPECIAL APPLICATION TO SEISMIC RAY-TRACING IN COMPLEX 3-DIMENSIONAL MODELS

by

Håvar Gjøystdal

June 1978

GECO
GEOPHYSICAL COMPANY
OF NORWAY A·S

Rektor Natvig-Pedersens vei 29
N-4000 Stavanger
Norge - Norway

Main office and
geophysical services:
Teleph: (045) 55 000
Telex: 33 268 GECOS N

Petroleumgeological and
- technological services:
Teleph: (045) 89 902 — 89 903
Telex: 33 325 GECOL N

TABLE OF CONTENTS

1. Introduction

In this paper we will describe a mathematical procedure which was developed in order to find zero-value contours of a scalar function of two variables. Although the method is a rather general one, it originated from a particular problem which had to be solved in connection with the tracing of seismic rays in complex 3-dimensional geological models: When a ray is to be traced from a given source point in a given initial direction, it is generally impossible to know in advance where the ray happens to arrive at the surface. A very common problem lies in the fact that twe want to trace exactly those rays which are returning very close to a predefined receiver line at the model surface, say within a distance of a few meters from the line. As the arrival point of the ray is uniquely determined by two parameters specifying the initial ray direction, the distance to the line may be considered as a scalar function of two variables. Thus the problem reduces to finding those pairs of initial ray direction parameters corresponding to points on the zero-value contours of this scalar function. Another slightly different situation occurs when tracing rays normally from a given reflecting horizon to the predefined line on the surface (the so-called normal incidence paths (NIP)). In this case the ray direction is given, and the two independent variables of the 'distance function' are made up by the horizontal coordinates of the start point of the ray.

In both examples just mentioned the problem reduces to calculating sample points (x_1,x_2) satisfying the equation

$$f(x_1,x_2) = 0$$

where f denotes the 'distance function' and (x_1,x_2) are the independent variables which may be alternatively defined in each case. It should be noted that the procedure which will be described is especially designed for problems involving very time-consuming operations in order to calculate the functional values, implying that the number of such operations has to be kept to a minimum. (In the ray-tracing examples, each functional value calculation includes a complete tracing of a ray through the model.) Consequently, the common method of mapping the function over large regions of the x_1x_2-plane and then obtaining the wanted contours by interpolation, is rather unsatisfactory in such cases. For further details concerning ray-tracing problems, see Gjøystdal (1978), Hubral (1976), Julian and Gubbins (1977), Sha (1973), Sorrels et al (1971).

2. Mathematical Description of the Problem

We now turn to the more general mathematical description of the problem.

Assume that a scalar function of two variables is defined on a region S of the x_1x_2-plane

$$y = f(\bar{x}), \qquad \bar{x} \in S \tag{1}$$

where $\bar{x} = (x_1,x_2)$ is a vector notation for the independent variables.

Assume further that S may be divided into regions S_i, i=1,...,n in such a way that f is continuous on each S_i (that is, f is piecewise continuous on S).

The method to be described in this paper was designed in order to calculate solutions of the equation

$$f(\bar{x}) = 0 \tag{2}$$

provided that such solutions exist somewhere in S. The procedure is based on the assumption that two operations are available:

Given an arbitrary point $\bar{x}$

- Determine if $\bar{x} \in S$
- If $\bar{x} \in S$, find $f(\bar{x})$.

Geometrically, the calculation of solutions of (2) corresponds to determination of intersection curves (or tangential curves) between the surface $y = f(\bar{x})$ and the $\bar{x}$-plane. In special cases one may think of having f = 0 in whole regions of the $\bar{x}$-plane, however, our method will be restricted to cases in which the solutions are made up by one or more continuous curves in S. Our aim will be to sample these curves (in the following denoted by C) using a given sampling interval ΔS, i.e., to calculate solutions of the form

$$\bar{x}_{ij} , \quad i=1,N_j, \quad j = 1,M \tag{3}$$

Here $\bar{x}_{ij}$ denotes sample point no. i of the j-th curve C_j, N_j is the number of samples for C_j and M the total number of separate curves.

The distance between each sample shall be approximately given by $|\bar{x}_{i+1,j} - \bar{x}_{ij}| \approx \Delta S$, except for the samples close to the end points of each C_j which in general shall have a smaller sampling interval in order to determine the discontinuities of the function more accurately. In case C_j is a closed curve, it should be sampled in such a way that the first and last sample points are identical, i.e., $\bar{x}_{1j} = \bar{x}_{N_j j}$. See Fig. 1. It should also be noted that in a numerical procedure, the equation (2) must be replaced by the requirement

$$|f(\bar{x})| < \varepsilon \tag{4}$$

i.e., a point $\bar{x}$ is a satisfactory solution when giving functional values closer to zero than a certain predefined limit.

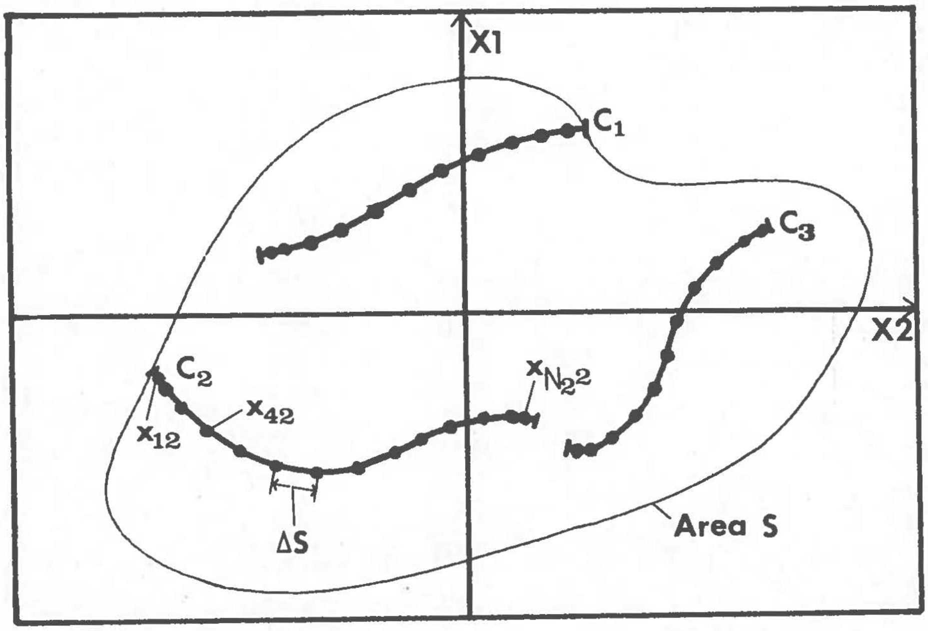


Fig. 1. The sampling of 'zero-value curves' C_j in the x_1x_2-plane.

3. The Gradient Search Method

We shall now describe a method which we have called 'the gradient search method' (or GS-method), and which can be regarded as forming a sub-process of the sampling procedure described in section 2.

We assume that an initial point $\bar{x}_1$ is given in such a way that

$$|f(\bar{x}_1)| > \varepsilon \qquad \bar{x}_1 \in S \tag{5}$$

which means that $\bar{x}_1$ does not fulfill the requirement (4), or that f is too far from zero for $\bar{x}_1$ to be considered as a solution. The aim of the GS-method is to displace $\bar{x}_1$ into a new point $\bar{x}_c$ for which

$$|f(\bar{x}_c)| < \varepsilon \qquad \bar{x}_c \in S \tag{6}$$

We shall follow an iterative scheme and we denote the iteration index by k, k=1 referring to the start point $\bar{x}_1$. Having found a point $\bar{x}_k$, we compute the gradient

$$\nabla f(\bar{x}_k) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}\right)_{\bar{x}_k} \tag{7}$$

by use of the common approximation

$$\frac{\partial f}{\partial x_1} = \frac{f(x_1+\Delta x, x_2) - f(\bar{x})}{\Delta x}$$

$$\frac{\partial f}{\partial x_2} = \frac{f(x_1, x_2+\Delta x) - f(\bar{x})}{\Delta x} \tag{8}$$

where Δx is a small increment chosen to give a satisfactory estimate of ∇f.

We define a displacement direction vector as

$$\bar{d}_k = -\mathrm{sgn}[f(\bar{x}_k)] \cdot \frac{\nabla f_k}{|\nabla f_k|} \quad (9)$$

where

$$\mathrm{sgn}(\theta) = \begin{cases} -1 & \theta < 0 \\ 0 & \theta = 0 \\ +1 & \theta > 0 \end{cases} \quad (10)$$

$\bar{d}_k$ is a unit vector parallel to ∇f_k pointing towards the $\bar{x}$-plane intersection curve C. The displacement of the point $\bar{x}_k$ towards C is performed by an iterative process with constant step size D:

$$\bar{x}_{k+1} = \bar{x}_k + D \cdot \bar{d}_k \quad (11)$$

The process is repeated until one of the following cases occurs:

- The functional value becomes less than a given ε_N $(|f(\bar{x}_k)| < \varepsilon_N)$
- The functional value changes sign from one iteration to the next $(f(\bar{x}_k) \cdot f(\bar{x}_{k-1}) < 0)$
- The total number of steps exceeds a predefined limit.

If the last case occurs, the process is terminated and declared as unsuccessful, otherwise a more effective process is initiated to search for the final solution. The method used here is very similar to the well-known Newton iteration method frequently used in the one-dimensional case, which makes use of the curve tangent in the point x_k, and locates the next point x_{k+1} in the intersection point between the tangent and the x-axis. (f.ex., see Acton, 1970). In a similar way we shall compute the intersection point between the $\bar{x}$-plane and a line ℓ which

- lies in the tangential plane to the surface $z = f(\bar{x})$ and goes through the tangential point $\bar{x}_k$
- has a horizontal projection in the $\bar{x}$-plane parallel to $\nabla f(\bar{x}_k)$.

Representing ℓ by a unit vector $\bar{v} = (v_1, v_2, v_3)$ starting in $\bar{x}_k$ and pointing along ℓ towards the $\bar{x}$-plane, $\bar{v}$ may be determined from the following equations

$$\bar{v} \cdot \bar{n} = 0 \Rightarrow v_1 \frac{\partial f}{\partial x_1} + v_2 \frac{\partial f}{\partial x_2} - v_3 = 0 \qquad (12)$$

$$(v_1, v_2) || \nabla f \Rightarrow v_1 \frac{\partial f}{\partial x_2} = v_2 \frac{\partial f}{\partial x_1} \qquad (13)$$

$$|\bar{v}| = 1 \Rightarrow v_1^2 + v_2^2 + v_3^2 = 1 \qquad (14)$$

where $\bar{n} = (\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, -1)$ is the $f(\bar{x})$ surface normal vector in $\bar{x}_k$.

Now, equations (12), (13), and (14) give the components

$$v_1 = -\text{sgn}(f(\bar{x})) \cdot \frac{\frac{\partial f}{\partial x_1}}{|\nabla f| \sqrt{1+|\nabla f|^2}}$$

$$v_2 = -\text{sgn}(f(\bar{x})) \cdot \frac{\frac{\partial f}{\partial x_2}}{|\nabla f| \sqrt{1+|\nabla f|^2}} \qquad (15)$$

$$v_3 = -\text{sgn}(f(\bar{x})) \cdot \frac{f}{\sqrt{1+|\nabla f|^2}}$$

where all variables refer to the point $\bar{x}_k$. Given a point $\bar{x}_k$ in the iterative process, the next point $\bar{x}_{k+1}$ is simply found by computing $\bar{v}$ in $\bar{x}_k$ and then the point of intersection between ℓ (given by $\bar{v}$) and the $\bar{x}$-plane. The process, in the following called the 'Newton process', is repeated until one (or some) of the following cases occur:

- $|f(\bar{x}_k)| < \varepsilon$, successful process, stop
- $|f(\bar{x}_{k+1})| > |f(\bar{x}_k)|$, diverging process, stop
- iteration index k exceeds a given maximum value, stop.

The GS-method is thus completely described, $\bar{x}_c$ (se eq. (6)) being the last value of $\bar{x}_k$. As we shall see, also the last value of the gradient ∇f_k will be of great importance when the GS-method is to be implemented in the general process which will be described in the following section. Fig. 2 gives a schematical illustration of the two-step process forming the basis of the GS-method.

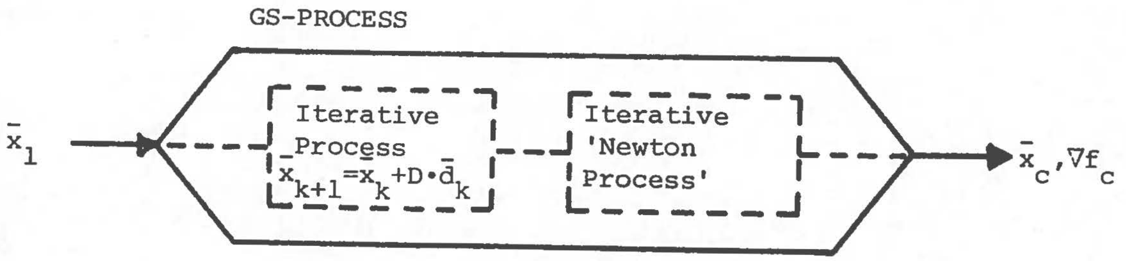

GS-PROCESS
$\bar{x}_1$
Iterative Process $\bar{x}_{k+1}=\bar{x}_k+D\cdot\bar{d}_k$
Iterative 'Newton Process'
$\bar{x}_c, \nabla f_c$

Fig. 2. Schematical illustration of the GS-method.

4. Procedure for Calculating the C_j-Curves

The procedure may be divided into two major steps:

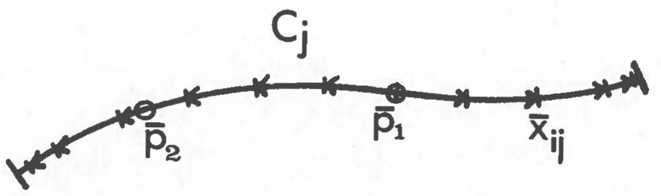
A. Determine at least one point belonging to each curve C_j, i.e., determine a set of points

$$\bar{p}_i \ , \ i=1,N_p \qquad\qquad N_p \geq M \tag{16}$$

such that

$$|f(\bar{p}_i)| < \varepsilon, \qquad\qquad \bar{p}_i \in S \tag{17}$$

B. Starting in $\bar{p}_i$, search in both directions along the corresponding curve C_j to find the sampled version of C_j defined in (3). When C_j has been determined, eliminate from (16) all points $\bar{p}_i$ which happen to belong to the C_j just found, to avoid that C_j is sampled more than once (see Fig. 3). Repeat the procedure for all remaining $\bar{p}_i$.



Step A
Determination of curve points $\bar{p}_1$ and $\bar{p}_2$

Step B
Search along C_j to find samples $\bar{x}_{ij}$, starting in $\bar{p}_1$. Eliminate $\bar{p}_2$.

Fig. 3. Sampling of a curve C_j by the two-step procedure described. (In this example M=1 and N_p=2).

We now turn to a more detailed description of the procedures involved in steps A and B.

A. Determination of Start Points

- Specify a set of points $\bar{g}_i'$, $i=1,N_g'$ distributed in the $\bar{x}$-plane according to a preselected rule. F.ex., $\bar{g}_i'$ may be uniformly distributed on a rectangular grid in the $\bar{x}$-plane.
- Select those $\bar{g}_i'$ fulfilling the requirement

$$|f(\bar{g}_i')| < \varepsilon_g , \qquad \bar{g}_i' \in S, \tag{18}$$

ε_g being a predefined positive value. The resulting set of points is denoted by

$$\bar{g}_i , \qquad i=1,N_g \tag{19}$$

By choosing the original grid points $\bar{g}_i'$ dense enough, we may in principle obtain as good start values $\bar{g}_i$ as we wish. The choice of initial grid and value of ε_g should be made on the basis of preliminary knowledge of the function $f(\bar{x})$ (if any). Too many points will reduce effectiveness of the process; too few points will increase the probability of loosing solutions.

The next step is to apply the GS-method (described in section 2) to each of the N_g points $\bar{g}_i$, which according to Fig. 2 should result in the wanted points p_i in (16) together with the corresponding gradients.

$$\bar{g}_i , \quad i=1,N_g \xrightarrow{\text{GS-method}} \bar{p}_i, \nabla f_i, \ i=1,N_p \tag{20}$$

where $N_p \leq N_g$

B. Given Start Points on the Zero-value Curve C_j, Search for Samples along C_j

The sampling procedure starts by finding the tangent unit vector $\bar{t}$ of C_j in the start point $\bar{p}$ by use of the equations

$$\bar{t} \cdot \nabla f = 0 \tag{21}$$

$$|\bar{t}| = 1 \tag{22}$$

Equations (21) and (22) give two solutions $\bar{t}_1$ and $\bar{t}_2$ pointing in opposite directions along the curve C_j. We shall sample C_j in both directions until the end points are reached. Denoting the i-th sample point by $\bar{x}_i$ and the gradient in $\bar{x}_i$ by ∇f_i we have

$$\bar{x}'_{i+1} = \bar{x}_i + \Delta S \cdot \bar{t}_i \ , \qquad t_i \perp \nabla f_i \tag{23}$$

ΔS being the predefined sampling distance. Using the GS-method we get

$$\bar{x}'_{i+1} \xrightarrow{\text{GS-method}} \bar{x}_{i+1} \ , \ \nabla f_{i+1} \tag{24}$$

where the new sample point $\bar{x}_{i+1}$ satisfies the requirements $|f(\bar{x}_{i+1})| < \varepsilon$, $\bar{x}_{i+1} \in S$.

This procedure is repeated until an end point is reached, that is, until the region of definition S is exceeded or a discontinuity occurs. The procedure is to be performed in both directions from the start point $\bar{p}$, i.e., for $\bar{t} = \bar{t}_1$ and $\bar{t} = \bar{t}_2$.

Special attention should be paid to the sample points close to the ends of the curve, in order to define the end samples more accurately. The first time a sample point is not found due to exceeding of the curve C_j, we use half the sampling distance ΔS and try once more. This 'halving operation' may be repeated any given number of times, and the determination of end points is thus continuously improved.

Each time a new curve C_j has been sampled and the sample points $\bar{x}_{ij}$, $i=1,N_j$ have been determined, one should go through the remaining start points $\bar{p}_k$ to avoid that another sampling process is started for the same C_j. Every $\bar{p}_k$ for which

$$|\bar{p}_k - \bar{x}_{ij}| < \Delta S \tag{25}$$

will be considered as belonging to the curve C_j and will thus be omitted.

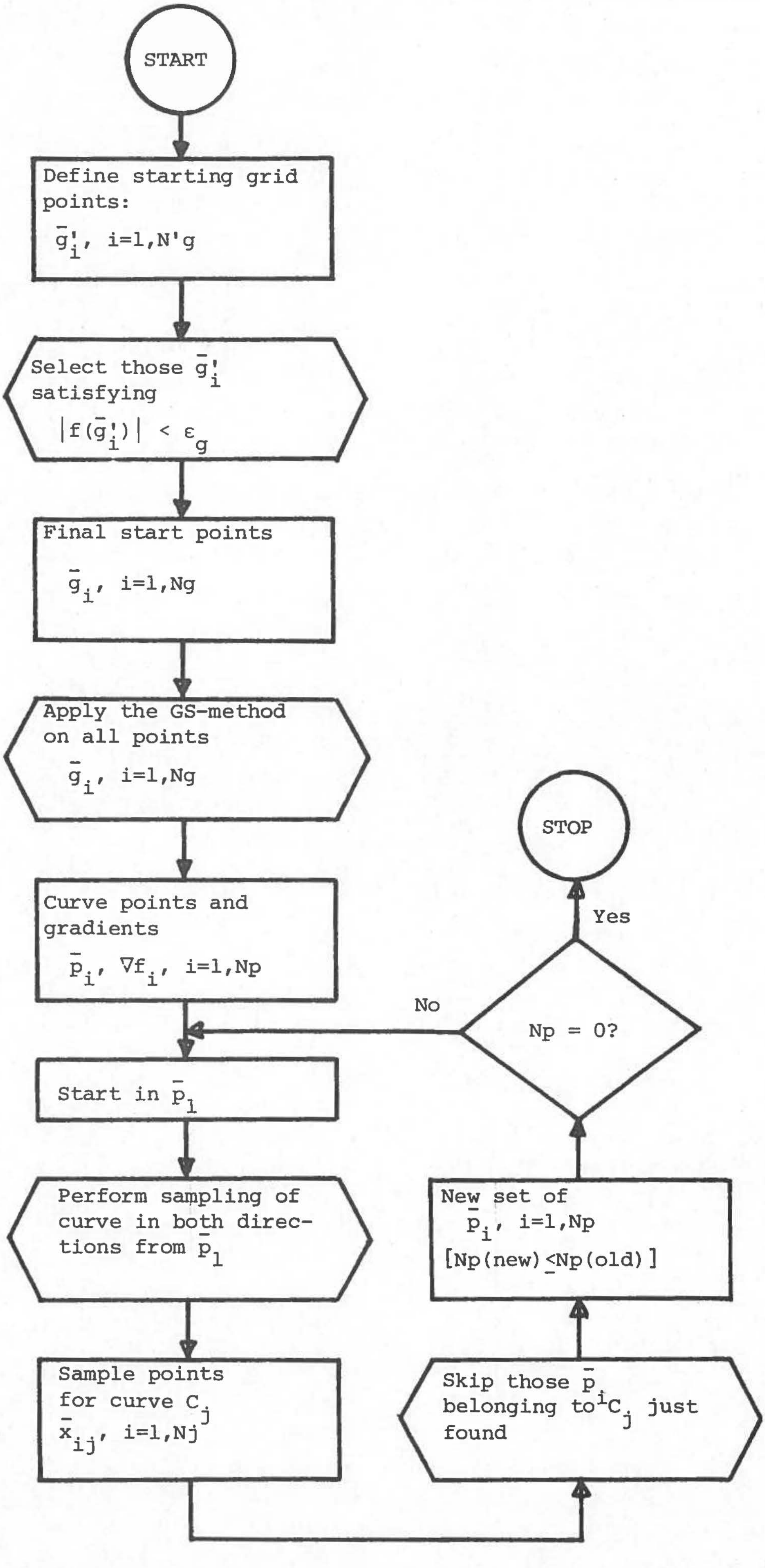

START
Define starting grid points:
$\bar{g}'_i$, i=1,N'g
Select those $\bar{g}'_i$ satisfying
$|f(\bar{g}'_i)| < \varepsilon_g$
Final start points
$\bar{g}_i$, i=1,Ng
Apply the GS-method on all points
$\bar{g}_i$, i=1,Ng
Curve points and gradients
$\bar{p}_i$, ∇f_i, i=1,Np
Start in $\bar{p}_1$
Perform sampling of curve in both directions from $\bar{p}_1$
Sample points for curve C_j
$\bar{x}_{ij}$, i=1,Nj
Skip those $\bar{p}_i$ belonging to C_j just found
New set of
$\bar{p}_i$, i=1,Np
[Np(new) $\leq$ Np(old)]
Np = 0?
No
Yes
STOP

Fig. 4. Flow-chart showing the principles of the zero-value curve search process.

The last thing to do will be to reorganize the sampling sequence of x_{ij}, $i=1,N_j$ for all curves C_j. Instead of starting in an arbitrary point $\bar{p}$ on the curve and counting the samples in both directions from $\bar{p}$, the sample sequence is changed so that first and last samples are counted from one end point to the other.

Fig. 4 gives a flowchart of the process described in this paper.

5. Program Documentation

In this section we shall give a description of the input/output parameters for a number of FORTRAN IV subroutines which were written in order to perform the procedures described in sections 2, 3 and 4. We shall refer to these sections when necessary.

```
I. SUBROUTINE PROCED (X,Y,NY, IPROC,IC)
     DIMENSION X(2), Y(NY)
```

Purpose:

This routine performs the two basic operations described on page 2, that is, it determines if a point $\bar{x}$ belongs to the region of definition S of the function f, and computes the functional value $f(\bar{x})$.

Parameter description:

Input

X(I), I=1,2	-	Vector of independent variables $\bar{x}$.
NY	-	Dimension of the array Y (see description of Y in the output parameter list).
IPROC	-	Function or process index no. The routine may contain more than one function, IPROC points to the one wanted.

Output

Y(1)	-	Functional value $f(\bar{x})$ if $\bar{x} \in S$.
Y(I), I=2,NY	-	Array containing additional values to be saved if wanted.

IC - Code index describing the output state of the process. F.ex., IC=0 means $\bar{x} \in S$, IC $\neq$ 0 means $\bar{x} \notin S$. For IC $\neq$ 0, the value of IC may contain information about why the process fails, i.e., why $f(\bar{x})$ cannot be computed.

Remarks:

The subroutine PROCED has to be written by the user in order to perform functional value computations. (For the seismic ray-tracing problem referred to in Section 1, this routine performs the basic operation of tracing a ray through a 3-D model, the functional value Y(1) being the distance between the ray arrival point and the recording line. Additional ray-tracing results, such as the travel time, may be stored in Y(I), I=2,NY). Other input parameters needed to perform the functional calculations should be given through a COMMON statement.

Subroutines called:

Must be specified by the user.

```
II. SUBROUTINE IGRASP(IPROC,NY,XSTART,DX,STEPL,NSTEP,NITMAX,EPSNWT,EPS,
         XEND,YEND,GRAD,ICODE)
         DIMENSION XSTART(2), XEND(2), YEND(NY), GRAD(2)
```

Purpose:

This routine performs the 'gradient search process' described in Section 3, i.e., moves a point from start position $\bar{x}_1$, where $|f(\bar{x}_1)|>\varepsilon$ to a final position $\bar{x}_c$ satisfying $|f(x_c)|<\varepsilon$ (see eqs. (5) and (6)).

Parameter description:

Input

IPROC - Function or process index number pointing to the right process in subroutine PROCED.

NY - Dimension of the output array YEND, YEND(1) containing the final functional value, if any.

XSTART(I), I=1,2 - Start point of process ($\bar{x}_1$ in eq. (5)).

DX - Small x-increment to be used in gradient computations (Δx in eq. (8)).

STEPL - Step length in iterative process (D in eq. (11)).

NSTEP - Total number of steps to be allowed (i.e., total number of iterations, using eq. (11)).

NITMAX - Total number of iterations by the tangent/$\bar{x}$-plane intersection process (or Newton process) described on pp. 5 and 6.

EPSNWT - The Newton process is started when the functional value becomes less than this parameter ($|f(\bar{x}_k)|<\varepsilon_N$, page 5).

EPS - The functional value is said to be zero if it is less than this small positive number (ε in eqs. (5) and (6)).

GRAD(I), I=1,2 - Gradient (∇f) in the initial point XSTART(I). This parameter may or may not be given. It will be changed in the routine.

ICODE - Input/output option parameter. ICODE=0 by input means that input gradient will not be given. ICODE=1 by input means that input gradient will be given. (Note that this parameter is also used as an output parameter and is thus changed within the routine.)

Output

XEND(I), I=1,2 - Final position of point $\bar{x}_c$ such that $|f(\bar{x}_c)|<\varepsilon$ (see eq. (6)).

YEND(1) - Final functional value $f(\bar{x}_c)$.

YEND(I), I=1,2 - Additional parameters wanted, corresponding to the array Y in subroutine PROCED.

GRAD(I), I=1,2 - Gradient ∇f in final point XEND.

ICODE - Output code index.

ICODE=0: Final point found, i.e., normal return

ICODE=1: Max. no. of steps (NSTEP) has been exceeded

ICODE=2: No intersection point found between tangent and $\bar{x}$-plane in Newton process

ICODE=3: Max. no. of iterations exceeded in Newton process

ICODE=3+IC: The functional value $f(\bar{x})$ could not be found for the reason indicated by IC. (IC is the output code index described in subroutine PROCED.)

Remarks:

If ICODE=0 by input the gradient GRAD(I) is computed in the initial point and need not be given. This will be the case when subroutine IGRASP is used to refine grid points as described in A) on page 8 ($\bar{g}_i' \to \bar{g}_i$). If ICODE=1 by input, the initial gradient must be given. This will, f.ex., be the case when operating very close to the end of a C-curve, where the function $f(\bar{x})$ may have a discontinuity making gradient computations difficult. In this case, the <u>last</u> successfully computed gradient value should be used.

Subroutines called:

PROCED

LINPLA (for description, see computer listing).

```
III. SUBROUTINE SZEROV(IPROC,NY,XSTART,NDIR,GRAD,DX,STEPL,NSTEP,NITMAX,
     EPSNWT,EPS,NSAMAX,DSAMP,NSAEND,XSAMP,YSAMP,NSAMP,ICODE)
     DIMENSION XSTART(2), GRAD(2), XSAMP(NSAMP,2), YSAMP(NSAMP,NY)
```

Purpose:

This routine performs a search for zero-values of the function $f(\bar{x})$ (that is, the procedure B) in section 4) when start values $\bar{p}_i$, ∇f_i (eq. (20)) is given on a curve C_j. The result of a successful run is a sampling of the C_j-curve from the start point in a given direction until an end point is reached.

Parameter description:

<u>Input</u>

IPROC
NY
DX
STEPL
NSTEP
NITMAX
EPSNWT
EPS

See description of IGRASP

XSTART(I), I=1,2 – Start point in sampling process (on zero-value curve).

NDIR – Sampling direction: NDIR = +1 or -1.
The sampling direction unit vector is given by $\bar{d}$ = NDIR $\cdot$ (∇f_2, ∇f_1), ∇f_1 and ∇f_2 being the components of ∇f. Obviously, $\bar{d}$ is normal to ∇f.

GRAD(I), I=1,2 – Gradient ∇f in start point.

NSAMP – Number of samples (in the direction given by NDIR).

NSAMAX – Max. limit for NSAMP.

DSAMP – Sampling distance along curve (ΔS in eq. (23)).

NSAEND – Number of additional samples at the end of the curve. The sampling distance is halved NSAEND times before sampling process is terminated.

Output

XSAMP(I,J), I=1,NSAMP, J=1,2 – Array of sample points along zero-value curve.

YSAMP(I,1) I=1,NSAMP – Array of functional values corresponding to XSAMP(I,J).

YSAMP(I,J), I=1,NSAMP J=2,NY – Additional parameters wanted corresponding to Y in subroutine PROCED.

ICODE – Output code index:
ICODE contains the last code index from subroutine IGRASP indicating the reason for an eventual stop in the sampling process.

Remarks:

Each call to the routine performs sampling in one direction along the curve. To obtain a complete sampling of the curve, one has to make two successive calls with NDIR=1 and -1, respectively.

Subroutines called:

IGRASP

6. Program Example - Application to a 3-dimensional Ray-tracing Problem

In this section we shall describe an example of applying the procedures to a seismic ray-tracing problem in a very simple 3-dimensional model. An intersection through the model is given in Fig. 5a). We shall assume that all sections normal to the x_2-axis are equal, i.e., the model is independent of the x_2-coordinate. The model consists of one single reflecting interface ABC having a discontinuous change in curvature only in B. Fig. 5b) shows a predefined receiver line ℓ on the model surface, striking 45° relative to the x_1-axis. The aim of this example will be to find the foot points of a set of normal incidence rays (NIP-rays) starting in the reflecting interface ABC in a direction normal to the interface, and arriving inside the rectangular area DEFG in Fig. 5b) and closer to the receiver line than a given ε. Denoting the horizontal coordinates of the foot-point (start point) of an NIP-ray by $\bar{s} = (s_1,s_2)$, and the corresponding arrival point on the surface by $\bar{r} = (r_1,r_2)$, we know that $\bar{r}$ is a function of $\bar{s}$:

$$\begin{aligned} r_1 &= f_1(\bar{s}) \\ r_2 &= f_2(\bar{s}) \end{aligned} \qquad (26)$$

Assuming that d is the normal distance from $\bar{r}$ to the receiver line ℓ, defined to be positive at one side of ℓ and negative at the other, we immediately have that d is a function of start points $\bar{s}$ defined on a region R of the x_1x_2-plane corresponding to arrival points $\bar{r}$ within the rectangle DEFG:

$$d = f(\bar{s}) \qquad \bar{s} \in R \qquad (27)$$

It should be stressed that the region R is not itself the area covered by the rectangle DEFG. In fact, R is the set of all foot points $\bar{s}$ which give a NIP-ray arriving within this rectangle:

$$R = \{\bar{s}|\bar{r}=[f_1(\bar{s}),\ f_2(\bar{s})] \in \text{DEFG}\} \qquad (28)$$

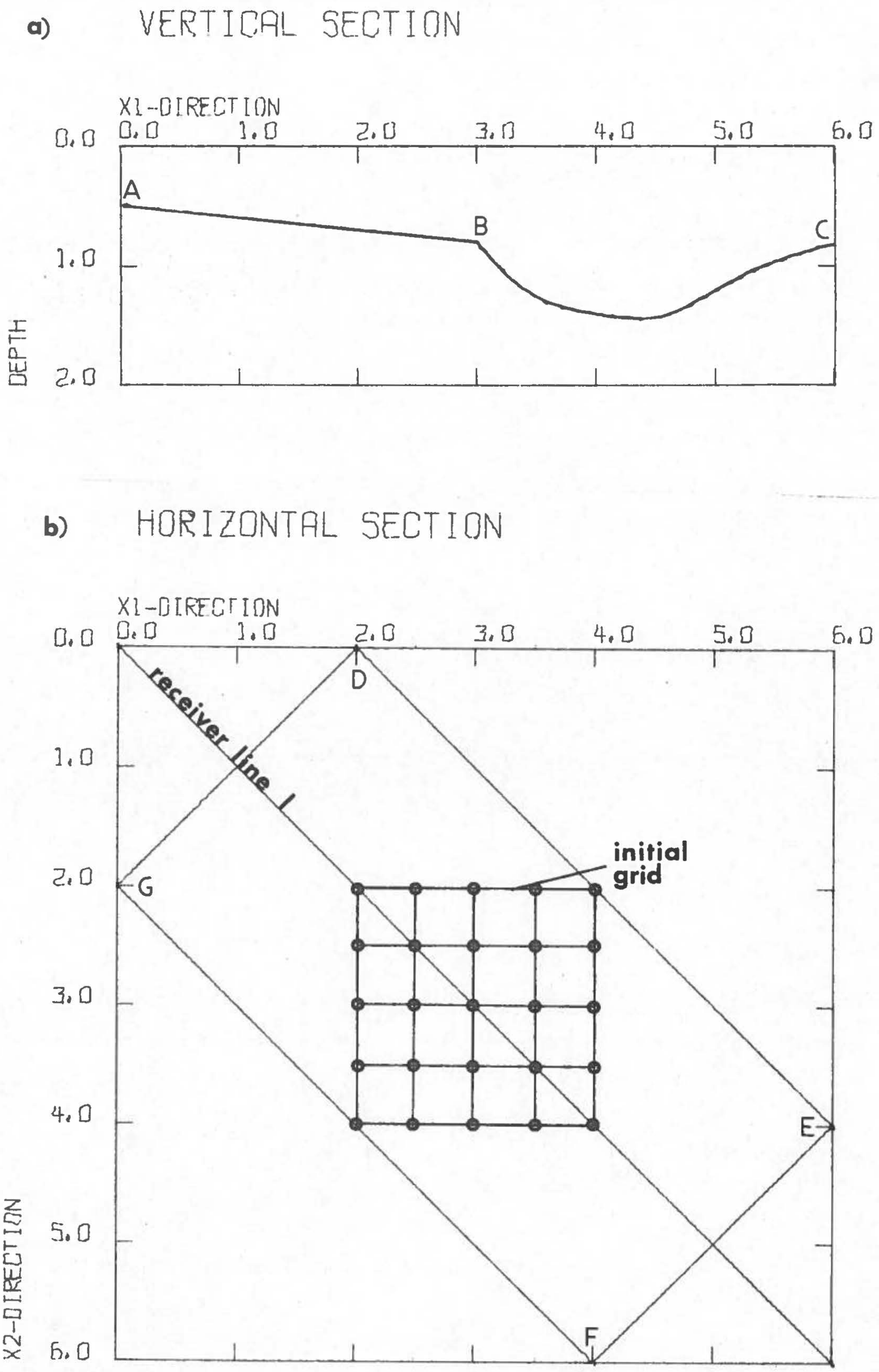


Fig. 5. Vertical (a) and horizontal (b) model section.

The problem of finding the foot points $\bar{x}_{ij}$ of NIP-rays arriving closer to ℓ than ε now reduces to the problem of finding the 'zero-value contours' or the 'C_j-curves' of the function $d = f(\bar{s})$ within the region R which corresponds to the region S of definition of f. In order to apply the routines described in Section 5, we proceed as follows:

- Write a subroutine PROCED (description on page 11) containing the ray-tracing algorithm needed to trace a ray from $\bar{s}$ to $\bar{r}$. X(I) then corresponds to $\bar{s}$ and the functional value Y(1) corresponds to the distance d. Y(I), I=2,NY may contain other interesting ray-tracing results such as travel time, travel distance, etc. The output code index IC is defined to be zero if the ray arrives at the surface within the rectangular area DEFG in Fig. 5b). If the ray arrives outside this area, IC is set equal to 1.
- Choose a rectangular grid in the x_1x_2-plane (ex. shown on Fig. 5b)) and trace rays from grid points to the surface, selecting the ones giving a distance to the receiver line less than a predefined ε_g (i.e., select points corresponding to the set $\bar{g}_i$, i = 1,Ng in eq. (19)).
- By subsequent use of the subroutine IGRASP, displace the points $\bar{g}_i$ in order to obtain receiver points closer to the line ℓ than ε, that is, find start points $\bar{p}_i$, ∇f_i, i=1,Np given in (20).
- Then, by applying the subroutine SZEROV, search in both directions along the 'ray foot point curve', starting in the points $\bar{p}_i$ in a direction normal to ∇f_i. In this way, foot points for the NIP-rays are successively found until the surface arrival points fall outside the predefined rectangle or a discontinuity in the foot point curve is reached. The latter phenomenon is expected to take place when passing the edge B of the model, since the arrival point in this case will jump away from the line due to the sudden change in the initial ray direction.

Table 1 shows the values of the input parameters used in this example, and the results of a complete sampling process are shown in Fig. 6. We see that the 'foot point curve' is split into two branches, due to the

PROGRAM VARIABLE	VALUE
Initial grid values (Fig. 5b):	
Coord. of grid lower lefth. corner	X1=2., X2=4.
Grid spacing	X1-direction: 0.5 X2-direction: 0.5
Number of grid points	X1-direction: 5 X2-direction: 5
Limit for grid point selection (ε_g in eq. (18))	1.0
DX	0.1
STEPL	0.25
NSTEP	20
NITMAX	10
EPSNWT	0.5
EPS	0.01
DSAMP	0.1
NSAEND	4

Table 1. Values of input parameters used in the ray-tracing example.

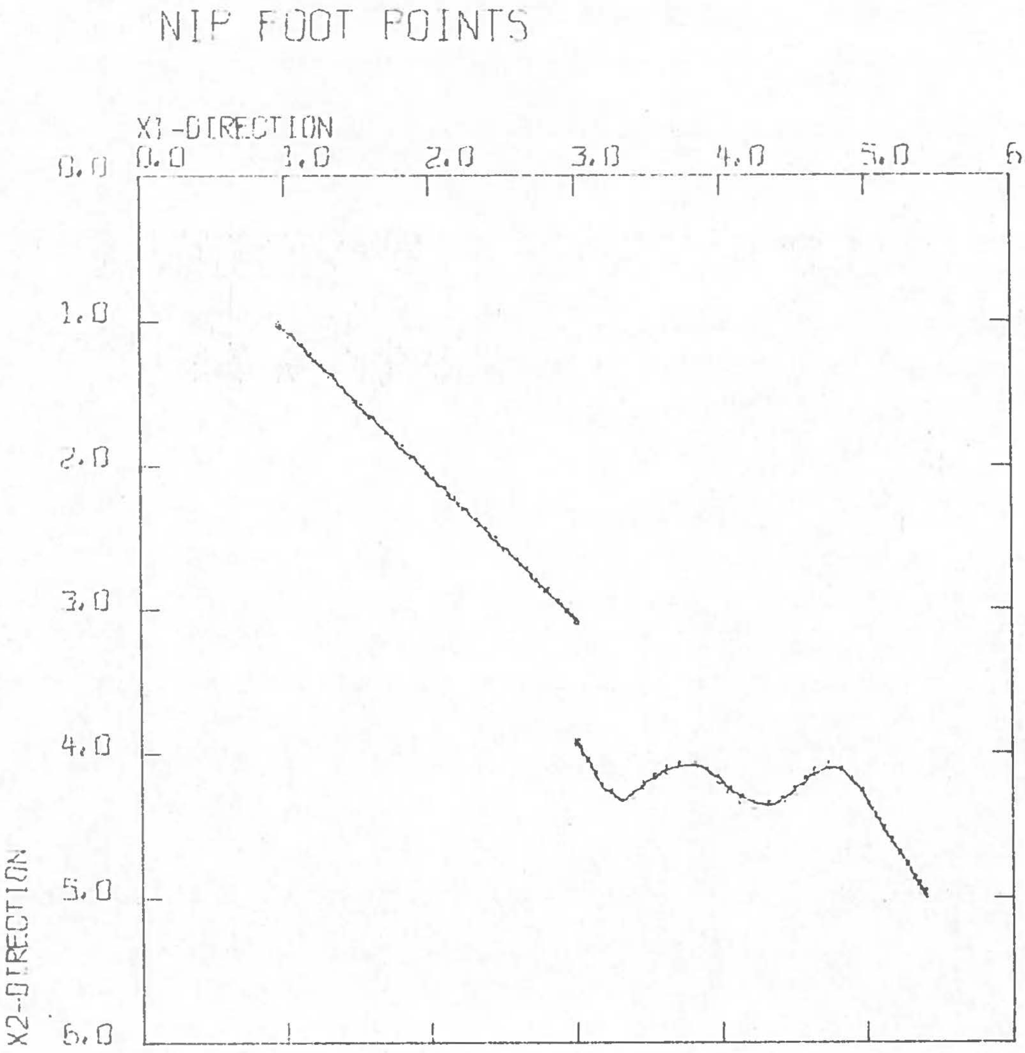


Fig. 6. Plot of the NIP-foot point curves for the model in Fig. 5.

discontinuity edge B just mentioned. By use of these curves, we may trace any number of NIP-rays arriving at the receiver line, each time computing the total travel time along the ray. This will make possible the construction of a synthetic seismic time section which should contain all possible branches of the time/distance (TX)-curves, provided that the initial grid was chosen dense enough to 'catch' all the 'foot point curves'. Obviously, the grid density should be selected in accordance with the model complexity, that is, a model with rapid variations and/or numerous discontinuities in the interface curvatures will generally give rise to several branches of the foot point curves.

For more details concerning the application of the method to different kinds of ray-tracing problems, we refer to Gjøystdal (1978).

ACKNOWLEDGEMENTS

This work has mainly been carried out at GECO, Høvik, and completed at NTNF/NORSAR, Kjeller, on behalf of GECO. It has received financial support from the Royal Norwegian Council for Scientific and Industrial Research and from Norsk Hydro.

Especially, I would like to thank Dr. B. Ursin for many valuable discussions.

REFERENCES

Acton, F.S., 1970. Numerical methods that work, Harper International Edition, Harper and Row Publishers, London.

Gjøystdal, H., 1978. Computation of seismic ray paths between given source and receiver line in a complex 3-D model, Technical Report, GECO, Høvik.

Hubral, P., 1976. Some remarks on computing travel times in plane layered media, Geophysical Prospecting 24, 719-724.

Julian, B.R., and Gubbins, D., 1977. Three-dimensional seismic ray tracing J. Geophys. 43, 95-113.

Sha, P.M., 1973. Ray tracing in three dimensions, Geophysics 38, 600-604.

Sorrels, G.G., Crowley, J.R., and Veith, K.F., 1971. Methods for computing ray paths in complex geological structures, Bull. Seism. Soc. Amer., 61, 27-53.