UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

One- and two-dimensional seismic modeling

by

Myung Woong Lee

Open-File Report 78-270

1978

This report is preliminary and has not been edited or reviewed for conformity with U.S. Geological Survey standards or nomenclature.

This report was written during the time the author worked under consulting contract No. 14-08-0001-14677, for the Branch of Oil and Gas Resources, U.S. Geological Survey, Denver, Colorado, from October 1974 to September 1975.

TABLE OF CONTENTS

ABSTRACT

Numerical seismic modeling techniques were developed by using finite difference solutions to one- or two-dimensional inhomogeneous elastic wave equations. Analytic solutions for the modeling of plane wave propagation in horizontally layered media were also obtained.

The first part of this report presents solutions to a one-dimensional elastic wave propagation equation by an analytic method and a finite difference method. The second part presents some solutions to a two-dimensional elastic wave equation by a finite difference method in an orthogonal cartesian coordinate system. The third part presents some calculations of wide angle reflection coefficients and interference patterns associated with a thin bed with varying angles of longitudinal wave incidence using Haskell's matrix method.

This study showed that a finite difference approach for numerical seismic modeling may be a good method, particularly in two-dimensional cases when there is emphasis on the amplitude and shape of seismic signals returning to the surface.

The following three computer programs and user's manuals are included in the appendices:

(1) Synthetic seismogram computer program for a plane wave in perfectly elastic media.

(2) One-dimensional finite difference computer program.

(3) Two-dimensional finite difference computer program.

FIGURES

Figure | Page

Figure | Page

TABLES

APPENDICES

INTRODUCTION

In seismic exploration, it is very important to compute the shape and amplitude of reflected and transmitted seismic signals for a complex subsurface geologic model. Particularly, in stratigraphic trap oil exploration, the examination of amplitude anomalies of seismic signals play an important role in the interpretation stage in contrast to the travel time anomaly used in the structural oil trap exploration. Also the advent of vertical seismic profiles has increased the applicability of the seismic modeling still more.

The purpose of this report is to study some of the basic theories of elastic wave propagation and to make a computer program to calculate reflected and transmitted seismic signals very efficiently.

Numerous authors (Peterson and others, 1955; Wuenschel, 1960; Trorey, 1962) have studied plane wave propagation in a horizontally layered media using an analytic solutions to a one-dimensional wave equation. For a perfectly elastic medium, this analytic solution approach is the best, in this author s opinion. However, for the realistic earth material which always has some degree of attenuation, this analytic solution is difficult to program. Therefore, we studied an inhomogeneous, attenuating one-dimensional wave equation by a finite difference scheme along with the analytic solution approach.

One of the advantage of this finite difference approach over the analytic solution approach can be found in making synthethic vertical seismic profiles, or VSP s. In finite difference schemes, we must calculate seismic signals at all grid points to solve the wave equation. Therefore, the execution time for one output trace is exactly the same as for the outputs at all grid points in a model. On the other hand, one of the disadvantages of finite difference approaches is its inaccuracey due to the accumulation of local truncation errors, propagation errors, and grid dispersion errors, which almost surely will increase as the length of a seismogram increases.

We studied very simple models using finite difference approaches and compared these with corresponding analytic solutions. We found good agreement.

For an irregular boundary and/or non-normal incidence, we conventionally use ray tracing techniques. For the computation of arrival times of seismic signals from the different geologic boundaries, this approach provides reliable information. But for true amplitude calculations, and particularly in the study of converted waves, this ray tracing technique fails. Thus, as in the one-dimensional case, we studied two-dimensional wave propagation by a finite difference approach.

Aboudi (1971) computed elastic wave fields by a finite difference scheme with a body force as a forcing function. Alterman and Aboudi (1970) studied a one-layered half-space in a cylindrical coordinate system by implementing the analytic solution around the source region with a difference scheme. Also Alford and others (1974) investigated diffraction problems and the accuracy of finite difference schemes in an acoustic material by solving for a displacement potential function with an analytic solution around the source.

Those authors, in common, used a homogeneous wave equation and fitted the boundary conditions at many boundaries. If these boundaries are simple (vertical or horizontal interface) and there are not many of them, this approach may be appropriate. But for complex geological models, this homogeneous formulation may not be adequate. Therefore, we studied an inhomogeneous wave equation using a finite difference approach.

Finally we studied wide angle reflection coefficients and interference patterns due to a thin bed by Haskell's matrix method.

This report has three parts. The first part of this report presents the solution of a one-dimensional elastic wave equation by analytic and finite difference approaches. The second part presents a finite difference scheme for solving a two-dimension inhomogeneous elastic wave equation. The third part presents the reflection coefficients of a thin bed.

Three computer programs and users's manuals are included in the appendices:

(1) Synthethic seismogram computer program for a plane wave in perfectly elastic media.

(2) One-dimensional finite difference computer program.

(3) Two-dimensional finite difference computer program.

ONE-DIMENSIONAL WAVE PROPAGATION

In geophysical exploration, it is very useful to compute reflected and transmitted seismic signals for a horizontally layered half-space, assuming plane wave propagation. Numerous authors have studied this one-dimensional wave propagation problem, either without multiples (Peterson and others, 1955) or with multiples (Wuenschel, 1960; Trorey, 1962), either without attenuation (Peterson and others, 1955; Wuenschel, 1960) or with attenuation (Trorey, 1962).

The purpose of this study is to make a computer program to solve this horizontally-layered problem. We followed Wuenschel's approach to make a synthetic seismogram for a perfectly elastic medium, with all multiples. We also studied the phase and amplitude distortion of a reflected plane wave with attenuation but without multiples. Finally, we studied the latter problem using a finite difference scheme to solve an inhomogeneous, attenuating one-dimensional wave equation.

Analytic Solution

(A) Perfectly Elastic Medium

To calculate the reflected and transmitted seismic signals, consider Figure 1, composed of N layers of homogeneous, isotropic and perfectly elastic material.

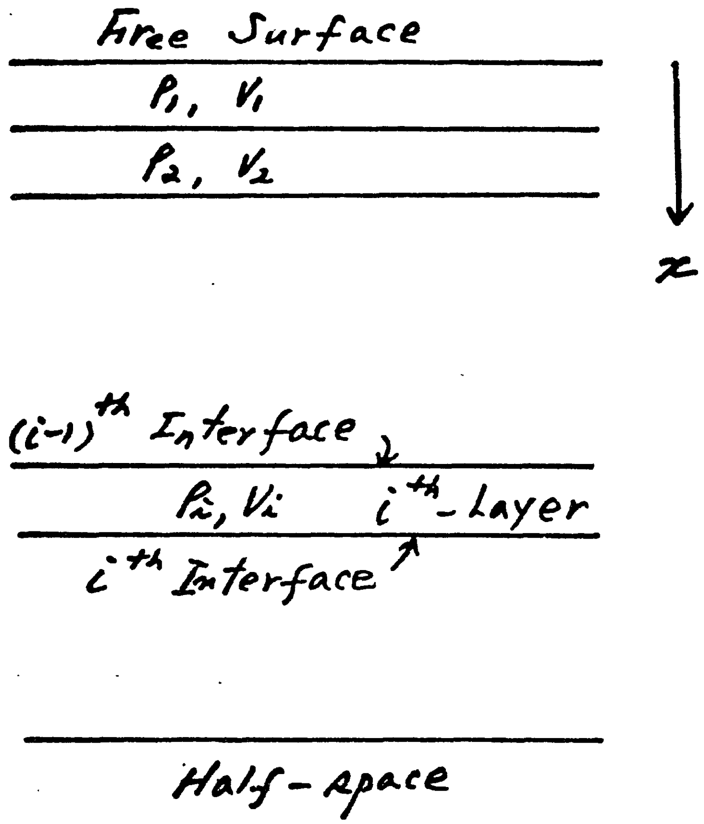


Figure 1. Multi-layered half-space.

The one-dimensional wave equation to be solved in each layers is

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{V^2}\frac{\partial^2 u}{\partial t^2}$$

where u is the particle displacement, V is the compressional velocity of the medium, and t is time.

Let

$$\bar{U}(s) = \int_0^\infty e^{-st} u(x,t)\,dt.$$

Then the Laplace transformed solution of the wave equation for the n-th layer is

$$\begin{aligned} \bar{U}_n &= I_n e^{-xs/V_n} + R_n e^{xs/V_n} \\ &= I_n e^{-sn\Delta z} + R_n e^{sn\Delta z} \end{aligned} \tag{1}$$

where $n\Delta z$ is one way travel time from the free surface to n-th layer.

The transformed stress in the n-th layer can be written as

$$\bar{\sigma}_n = -s I_n Z_n e^{-sn\Delta z} + s R_n Z_n e^{sn\Delta z} \tag{2}$$

where $Z_n = \rho_n V_n$

ρ_n : density in the n-th layer
$\bar{\sigma}_n$: normal stress.

The boundary condition to be satisfied in each layer is:

(1) When there is no source at n-th interface,

$$\begin{aligned} \bar{U}_n &= \bar{U}_{n+1} \\ \bar{\sigma}_n &= \bar{\sigma}_{n+1} \end{aligned} \tag{3}$$

(2) When there is a unit impulsive velocity source at n-th interface,

$$\frac{\partial u_{n+1}}{\partial t} - \frac{\partial u_n}{\partial t} = \delta(t), \quad \sigma_{n+1} = \sigma_n$$

or

$$\begin{aligned} \bar{U}_{n+1} - \bar{U}_n &= \frac{1}{s} \\ \bar{\sigma}_{n+1} &= \bar{\sigma}_n \end{aligned} \tag{4}$$

When there is a source on the free surface, we used a pressure source.

Assume that there is a unit impulsive velocity source at k-th interface. Using Equation (1), (2), and (4), we can show that, in matrix notation,

$$\begin{bmatrix} I_k \\ R_k \end{bmatrix} = \frac{1}{t_{k(k+1)}} \begin{bmatrix} 1 & r_{k(k+1)} z^{-k} & Q_1 \\ r_{k(k+1)} z^{k} & 1 & Q_2 \end{bmatrix} \begin{bmatrix} I_{k+1} \\ R_{k+1} \\ 1 \end{bmatrix} \qquad (5)$$

where

$$r_{k(k+1)} = \frac{Z_k - Z_{k+1}}{Z_k + Z_{k+1}}$$

$$t_{k(k+1)} = \frac{2 Z_k}{Z_k + Z_{k+1}}$$

$$Q_1 = \frac{-Z_k z^{-k/2}}{s\,(Z_k + Z_{k+1})}$$

$$Q_2 = \frac{-Z_k z^{k/2}}{s\,(Z_k + Z_{k+1})}$$

$$z^{\frac{1}{2}} = e^{-s \Delta \xi}$$

When there is no source at an interface, we can show the following:

$$\begin{bmatrix} I_n \\ R_n \end{bmatrix} = \frac{1}{t_{n(n+1)}} \begin{bmatrix} 1 & r_{n(n+1)} z^{-n} \\ r_{n(n+1)} z^{n} & 1 \end{bmatrix} \begin{bmatrix} I_{n+1} \\ R_{n+1} \end{bmatrix} \tag{6}$$

$$\triangleq [C_n] \begin{bmatrix} I_{n+1} \\ R_{n+1} \end{bmatrix}$$

Therefore, if there is no source, by iteration,

$$\begin{bmatrix} I_n \\ R_n \end{bmatrix} = [C_n][C_{n+1}][C_{n+2}] \begin{bmatrix} I_{n+3} \\ R_{n+3} \end{bmatrix} \tag{7}$$

By combining Equation (5) and Equation (7), assuming there is only one source at k-th interface, the following equation can be derived:

$$\begin{bmatrix} I_k \\ R_k \end{bmatrix} = \frac{1}{t_{k(k+1)}} \begin{bmatrix} 1 & r_{k(k+1)} z^{-k} & Q_1 \\ r_{k(k+1)} z^{k} & 1 & Q_2 \end{bmatrix} \begin{bmatrix} \prod_{j=k+1}^{N} [C_j] \begin{bmatrix} I_{N+1} \\ R_{N+1} \end{bmatrix} \\ 1 \end{bmatrix} \tag{8}$$

and

$$\begin{bmatrix} I_1 \\ R_1 \end{bmatrix} = \prod_{j=1}^{k-1} [C_j] \begin{bmatrix} I_k \\ R_k \end{bmatrix} \tag{9}$$

Equations (8) and (9) are the essence of the synthetic seismogram computation.

From Equations (8) and (9), it can also be shown

$$\begin{bmatrix} I_1 \\ R_1 \end{bmatrix} = \prod_{j=1}^{N} \left[C_j \right] \begin{bmatrix} I_{N+1} \\ R_{N+1} \end{bmatrix} + \prod_{j=1}^{k-1} \left[C_j \right] \begin{bmatrix} Q_1' \\ Q_2' \end{bmatrix} \qquad (10)$$

where

$$Q_1' = \frac{Q_1}{t_{(k+1)k}} \quad , \quad Q_2' = \frac{Q_2}{t_{k(k+1)}} .$$

In the following subsection we calculate the geophone velocity response for the specific source and detector location.

<u>(1) Source at free surface</u>...As mentioned before, in this case, the source is a pressure impulse.

From Equation (2), at $x=0$,

$$\bar{\sigma}_0 = -s \rho_1 V_1 (I_1 - R_1) = -\bar{p}(s)$$

where $\bar{p}(s)$ is the Laplace transformed source function.

Define

$$\left[C \right] = \prod_{j=1}^{N} \left[C_j \right] = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} . \qquad (11)$$

Since there is no source from the first to the N-th interface, using Equation (9),

$$I_1 = I_{N+1} C_{11} + R_{N+1} C_{12}$$

$$R_1 = I_{N+1} C_{21} + R_{N+1} C_{22} .$$

Notice that there is no incoming plane wave from the half-space, so we can set $R_{N+1} = 0$.
Therefore,

$$I_1 = \frac{\bar{p}(s)\, C_{11}}{s Z_1 (C_{11} - C_{21})} \tag{12}$$

$$R_1 = \frac{\bar{p}(s)\, C_{21}}{s Z_1 (C_{11} - C_{21})} .$$

Define

$$w = \frac{\partial u}{\partial t} , \quad \bar{w} = \int_0^\infty e^{-st} w(t)\, dt .$$

When a detector is on the free surface, using Equations (1) and (12),

$$\bar{W}_0 = \frac{\bar{p}(s)(C_{11} + C_{21})}{\rho_1 V_1 (C_{11} - C_{21})} . \tag{13}$$

When a detector is on the ℓ-th interface,

$$\bar{W}_\ell = \bar{p}(s)\left(s I_\ell \, \zeta^{\frac{\ell}{2}} + s R_\ell \, \zeta^{-\frac{\ell}{2}}\right).$$

Let

$$\prod_{j=\ell}^{N} \left[C_j\right] = \begin{bmatrix} D_{11}^{\ell} & D_{12}^{\ell} \\ D_{21}^{\ell} & D_{22}^{\ell} \end{bmatrix} \tag{14}$$

Then,

$$I_\ell = D_{11}^{\ell} I_{N+1}$$

$$R_\ell = D_{21}^{\ell} I_{N+1}$$

Therefore,

$$\bar{W}_\ell = \frac{\bar{p}\, \zeta^{\frac{\ell}{2}}}{\rho_1 V_1 (C_{11} - C_{21})} \left(D_{11}^{\ell} + \zeta^{-\ell} D_{21}^{\ell}\right) \tag{15}$$

(2) Source at the m-th Interface...From Equation (10),

$$Q_1' = \frac{-\zeta^{-m/2}}{2\lambda} \quad , \quad Q_2' = \frac{-\zeta^{m/2}}{2\lambda}$$

Since there is a source on the m-th interface, a free boundary condition must be satisfied at $x=0$.
Therefore, from Equation (4), $I_1 = R_1$.

Define

$$\prod_{j=1}^{m-1} [C_j] = \begin{bmatrix} S_{11}^m & S_{12}^m \\ S_{21}^m & S_{22}^m \end{bmatrix} . \tag{16}$$

From Equations (9) and (10), using Equation (16),

$$I_{N+1} = \frac{Q_1'(S_{21}^m - S_{11}^m) + Q_2'(S_{22}^m - S_{12}^m)}{C_{11} - C_{21}}$$

$$I_1 = G_{11}(S_{22}^m Q_2' + S_{21}^m Q_1') - G_{21}(S_{11}^m Q_1' + S_{12}^m Q_2'),$$

where

$$G_{11} = \frac{C_{11}}{C_{11} - C_{21}} \quad , \quad G_{21} = \frac{C_{21}}{C_{11} - C_{21}} .$$

When a detector is on the free surface,

$$\overline{W}_0 = \zeta^{-m/2} \bar{P}(\lambda)(G_{21}\bar{S} - G_{11}\overset{+}{S}) , \tag{17}$$

where $\bar{S} = S_{11}^m + \zeta^m S_{12}^m$

$\overset{+}{S} = S_{21}^m + \zeta^m S_{22}^m$.

When a detector is on the ℓ-th interface and $\ell > m$, we can show that

$$\overline{W}_\ell = \frac{\zeta^{\left(\frac{\ell-m}{2}\right)}\bar{P}}{2}\left(\frac{E}{C_{11} - C_{21}}\right)(D_{11}^\ell + \zeta^{-\ell} D_{21}^\ell) , \tag{18}$$

where

$$E = S_{11}^m - S_{21}^m + \zeta^n S_{12}^m - \zeta^m S_{22}^m .$$

When a detector is on the ℓ-th interface and $\ell < m$, we can show that

$$\overline{W}_\ell = \frac{Z_1}{2Z_\ell} \zeta^{-\frac{(\ell+m)}{2}} (D_{11}^\ell - D_{21}^\ell + \zeta^\ell D_{22}^\ell - \zeta^\ell D_{12}^\ell)(G_{21}\bar{S} - G_{11}\overset{+}{S}). \quad (19)$$

The main computation for a synthetic seismogram is the multiplication of the layer matrix. An iterative scheme to calculate matrix multiplication is as follows:

From Equation (6)

$$[C_n] = \frac{1}{t_{n(n+1)}} \begin{bmatrix} 1 & r_{n(n+1)}\zeta^{-n} \\ r_{n(n+1)}\zeta^n & 1 \end{bmatrix}.$$

Let

$$[G_n] = \begin{bmatrix} {}_ng_{11} & {}_ng_{12} \\ {}_ng_{21} & {}_ng_{22} \end{bmatrix} = t_{n(n+1)} [C_n] \quad (20)$$

$$= \begin{bmatrix} 1 & R_n\zeta^{-n} \\ R_n\zeta^n & 1 \end{bmatrix}$$

where $R_n = r_{n(n+1)}$.

Define

$$\prod_{n=i}^{j} [G_n] = [D_m^i] = [G_i][G_{i+1}] \cdots\cdots [G_{i+m-1}] \quad (21)$$

where $m = j - i + 1$.

From Equation (21)

$$[D_m^i] = [D_{m-1}^i][G_{i+m-1}].$$

Let

$$[D_m^i] = \begin{bmatrix} {}_md_{11}^i & {}_md_{12}^i \\ {}_md_{21}^i & {}_md_{22}^i \end{bmatrix}.$$

Then, by the matrix multiplication,

$$
\begin{aligned}
{}_m d_{11}^{\,i} &= {}_{m-1}d_{11}^{\,i} + ({}_{m-1}d_{12}^{\,i})({}_{i+m-1}g_{21}) \\
{}_m d_{12}^{\,i} &= ({}_{m-1}d_{11}^{\,i})({}_{i+m-1}g_{12}) + ({}_{m-1}d_{11}^{\,i})({}_{i+m-1}g_{22}).
\end{aligned} \tag{22}
$$

By the properties of matrix ,

$$ {}_m d_{12}^{\,i}(z) = {}_m d_{21}\left(\frac{1}{z}\right) $$

From Equation 20 and Equation 22,

$$
\begin{aligned}
{}_1 d_{11}^{\,i} &= 1 \triangleq \hat{E}_{11}^{\,i}(0) \\
{}_1 d_{12}^{\,i} &= k_i z^{-i} \triangleq \hat{E}_{12}^{\,i}(i) z^{-i} \\
{}_2 d_{11}^{\,i} &= ({}_1 d_{11}^{\,i})({}_{i+1}g_{11}) + ({}_1 d_{12}^{\,i})({}_{i+1}g_{21}) \\
&= 1 + k_i k_{i+1} z^{1} \\
&\triangleq \hat{E}_{11}^{\,i}(0) + \hat{E}_{11}^{\,i}(1) z \\
{}_2 d_{12}^{\,i} &= k_{i+1} z^{-(i+1)} + k_i z^{-i} \\
&\triangleq \hat{E}_{12}^{\,i}(i) z^{-i} + \hat{E}_{12}^{\,i}(i+1) z^{-(i+1)}.
\end{aligned}
$$

By induction, we can show that

$$
\begin{aligned}
{}_m d_{11}^{\,i} &= 1 + \hat{E}_{11}^{\,i}(1) z + \hat{E}_{11}^{\,i}(2) z^2 + \cdots + \hat{E}_{11}^{\,i}(m-1) z^{m-1} \\
{}_m d_{12}^{\,i} &= \hat{E}_{12}^{\,i}(i) z^{-i} + \hat{E}_{12}^{\,i}(i+1) z^{-(i+1)} + \cdots + \hat{E}_{12}^{\,i}(i+m-1) z^{-(i+m-1)}.
\end{aligned} \tag{23}
$$

Combining Equation 22 and Equation 23,

$$
\begin{aligned}
{}_m d_{11} &= [1 + \hat{E}_{11}^{\,i}(1) z + \cdots + \hat{E}_{11}^{\,i}(m-2) z^{m-2}]\,{}_{i+m-1}g_{11} \\
&\quad + [\hat{E}_{12}^{\,i}(i) z^{-i} + \hat{E}_{12}^{\,i}(i+1) z^{-i-1} + \cdots + \hat{E}_{12}^{\,i}(i+m-2) z^{-(i+m-2)}]\,{}_{i+m-1}g_{21} \\
&= 1 + z[\hat{E}_{11}^{\,i}(1) + \hat{E}_{12}^{\,i}(i+m-2) k_{i+m-1}] \\
&\quad \vdots \\
&\quad + z^{k}[\hat{E}_{11}^{\,i}(k) + \hat{E}_{12}^{\,i}(i+m-k-1) k_{i+m-1}] \\
&\quad \vdots \\
&\quad + z^{m-2}[\hat{E}_{11}^{\,i}(m-2) + \hat{E}_{12}^{\,i}(i+1) k_{i \ldots}] + z^{m-1}\hat{E}_{12}^{\,i}(i) k_{i+m-1}.
\end{aligned} \tag{24}
$$

This is the iterative formula for the matrix multiplication.

Likewise, we can show that

$$
\begin{aligned}
{}_m d_{12}^{i} = {} & z^{-i} E_{12}^{i}(i) + z^{-(i+1)}\{ E_{12}^{i}(i+1) + E_{11}^{i}(m-2) R_{i+m-1} \} \\
& + z^{-(i+2)}\{ E_{12}^{i}(i+2) + E_{11}^{i}(m-3) R_{i+m-1} \} \\
& \vdots \\
& + z^{-(i+k)}\{ E_{12}^{i}(i+k) + E_{11}^{i}(m-k-1) R_{i+m-1} \} \\
& \vdots \\
& + z^{-(i+m-2)}\{ E_{12}^{i}(i+m-2) + E_{11}^{i}(1) R_{i+m-1} \} \\
& + z^{-(i+m-1)} R_{i+m-1} .
\end{aligned}
\tag{25}
$$

(B) Attenuating medium

In the case where the medium is not perfectly elastic, the reflection and transmission coefficients are complex qualities. So we studied the effect of these complex reflection and transmission coefficients for a plane wave in a horizontally layered half-space without multiples, and showed how to compute a synthetic seismogram for this situation.

A solution of the one-dimensional wave equation in an attenuating medium can be written as

$$\bar{U}(\omega) = I(\omega) e^{-\alpha x} e^{-i\frac{\omega x}{V}} + R(\omega) e^{\alpha x} e^{i\frac{\omega x}{V}} \tag{26}$$

where $\bar{U}(\omega)$ is the Fourier transform of the displacement and α is an attenuation constant.

To calculate reflection and transmission coefficient, consider the following situation.

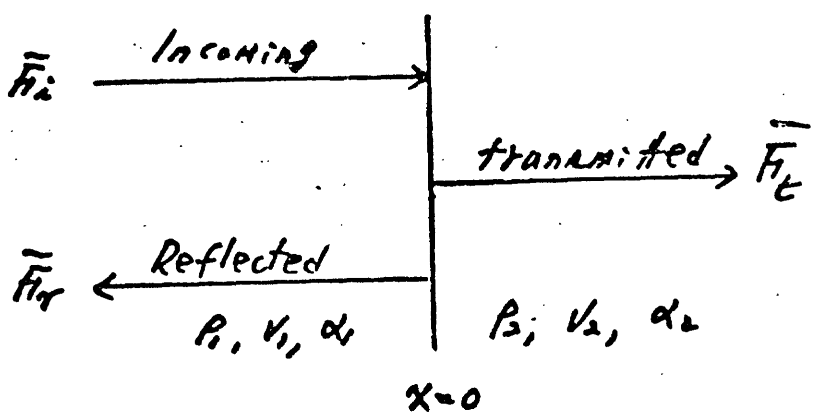


Figure 2. Normal incidence of a plane wave at a boundary.

From Equation 26, the stress can be written

$$\bar{\sigma}(\omega) = -\rho V^2 I(\alpha + \frac{i\omega}{V}) e^{-\alpha x} e^{-i\omega \frac{x}{V}} + \rho V^2 R(\alpha + \frac{i\omega}{V}) e^{\alpha x} e^{i\omega \frac{x}{V}}.$$

The boundary conditions at $x=0$ are

$$\bar{U}_1 = \bar{U}_2$$

$$\bar{\sigma}_1 = \bar{\sigma}_2 .$$

Let

$$\bar{h}_i = A_i e^{-\alpha_1 x} e^{-i\omega \frac{x}{V_1}}$$

$$\bar{h}_r = A_r e^{\alpha_1 x} e^{i\omega \frac{x}{V_1}}$$

$$\bar{h}_t = A_t e^{-\alpha_2 x} e^{-i\omega \frac{x}{V_2}} .$$

Therefore, the boundary conditions become,

$$A_t = A_i + A_r$$

$$-\rho_1 V_1 A_i (V_1 \alpha_1 + i\omega) + \rho_1 V_1 A_r (\alpha_1 V_1 + i\omega) = -\rho_2 V_2 A_t (\alpha_2 V_2 + i\omega)$$

Let

$$Z_i = \rho_i V_i$$

$$\beta_i = \alpha_i V_i$$

Then,

$$\tilde{R} \triangleq \frac{A_r}{A_i} = \frac{Z_1(\alpha_1 V_1 + i\omega) - Z_2(\alpha_2 V_2 + i\omega)}{Z_1(\alpha_1 V_1 + i\omega) + Z_2(\alpha_2 V_2 + i\omega)}$$

$$\tilde{T} \triangleq \frac{A_t}{A_i} = \frac{2 Z_1 (V_1 \alpha_1 + i\omega)}{Z_1(\alpha_1 V_1 + i\omega) + Z_2(\alpha_2 V_2 + i\omega)}$$

Let's assume α_i is proportional to the seismic wave frequency, so that

$$\alpha_i \to \alpha_i \, \omega \, \mathrm{sgn}(\omega).$$

Define

$$r_1 = \sqrt{(z_1\beta_1 + z_2\beta_2)^2 + (z_1 + z_2)^2}$$

$$r_2 = \sqrt{(z_1\beta_1 - z_2\beta_2)^2 + (z_1 - z_2)^2}$$

$$\alpha_1 = \tan^{-1} \frac{z_1 + z_2}{z_1\beta_1 + z_2\beta_2}$$

$$\alpha_2 = \tan^{-1} \frac{z_1 - z_2}{z_1\beta_1 - z_2\beta_2}$$

$$r_3 = 2z_1 \sqrt{\beta_1^2 + 1}$$

$$\alpha_3 = \tan^{-1} \frac{1}{\beta_1}$$

Then,

$$\tilde{R} = \frac{r_2}{r_1} e^{i(\alpha_2 - \alpha_1)\,\mathrm{sgn}(\omega)} \triangleq R e^{i\theta_r}$$

$$\tilde{T} = \frac{r_3}{r_1} e^{i(\alpha_3 - \alpha_1)\,\mathrm{sgn}(\omega)} \triangleq T e^{i\theta_t}.$$

The general form of $\tilde{R}_n$, $\tilde{T}_n$ at n-th interface can be,

$$\tilde{R}_n = R_n e^{i\theta_n \,\mathrm{sgn}\,\omega}$$

$$\tilde{T}_n = T_n e^{i\theta_n' \,\mathrm{sgn}\,\omega}$$

$$\tilde{T}_n' = T_n' e^{i\theta_n'' \,\mathrm{sgn}\,\omega}.$$

where $\tilde{T}_n'$ is the transmission coefficient from $(n+1)$-th to n-th layer.

The layer thickness is adjusted such that $2\frac{d_n}{V_n} = \tau$ is constant in all layers.

This means that the two-way travel time in each layer is the same.

Therefore, the reflected displacement on the free surface due to the n-th reflector can be written in the following form in the frequency domain with unit delta function input.

$$\overline{U}_n(\omega) = 2 D_{n-1}(\omega) e^{-H_n|\omega|} R_n e^{i(\theta_n + E_{n-1})\,sgn(\omega)} e^{-i\omega n\tau}, \qquad (27)$$

where

$$H_n = 2\sum_{i=1}^{n} \alpha_i d_i$$

$$E_n = \sum_{i=1}^{n} (\theta_i' + \theta_i'')$$

$$D_n = \prod_{i=1}^{n} T_i T_i' .$$

The synthetic seismogram is the sum of the series of reflections. Therefore, for a N-layered half-space, the total impulse response in the frequency domain is

$$\overline{U}(\omega) = 2\sum_{k=1}^{N} D_{k-1} e^{-H_k|\omega|} R_k e^{i(\theta_k + E_{k-1})\,sgn(\omega)} e^{-i\omega k\tau} . \qquad (28)$$

Examples and Discussions

Figure 3 shows a 4-fold surface to surface synthetic seismogram from a sonic log acquired at the Lusk area, Wyoming. A velocity versus time section plot is shown in Figure 4. In this synthetic seismogram, we deleted the source pulse since its amplitude is so large it obscures the reflections.

The computer execution time to make a 2.2 second synthetic seismogram for a surface source and a surface receiver from about 6,000 sonic data points (2 feet sampling interval) and 1 milli-second time sampling (2 milli-seconds for two-way travel time) is 1.5 seconds on the PDP-10.

Another useful application of this modeling program is to make vertical seismic profiles (VSP's). This is done by generating a whole series of synthetic seismograms, each one assuming successively greater geophone burial depth. From the attached computer program, which is written only for one source and one receiver position for each execution, it is easy to make VSP's. Figure 5 shows 121 synthetic seismograms for the Collins well, Wyoming. Each member of the set represents the wave field at a different burial depth.

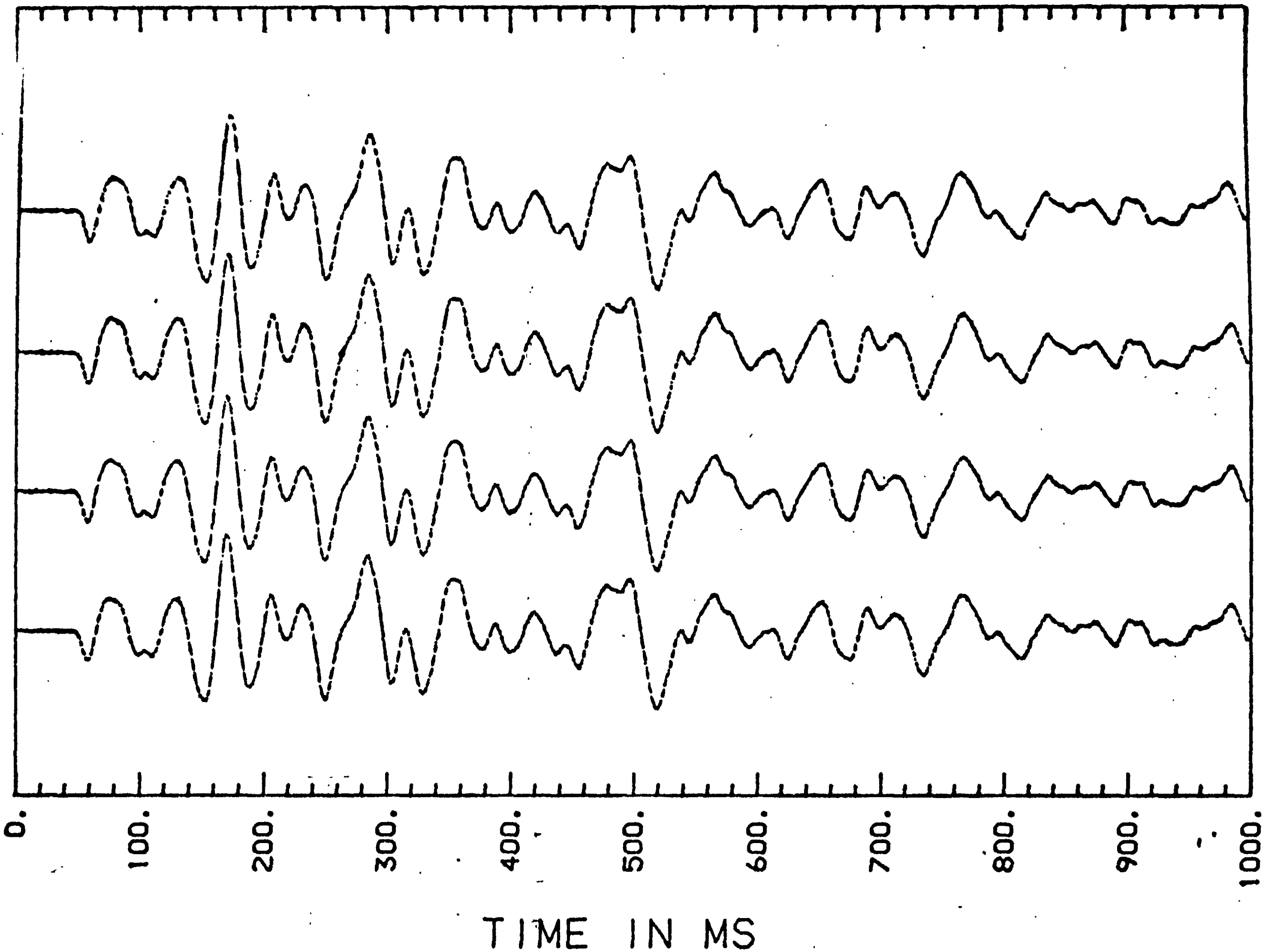


Figure 3. Four fold surface to surface synthethic seismogram for Lusk area, Wyoming.

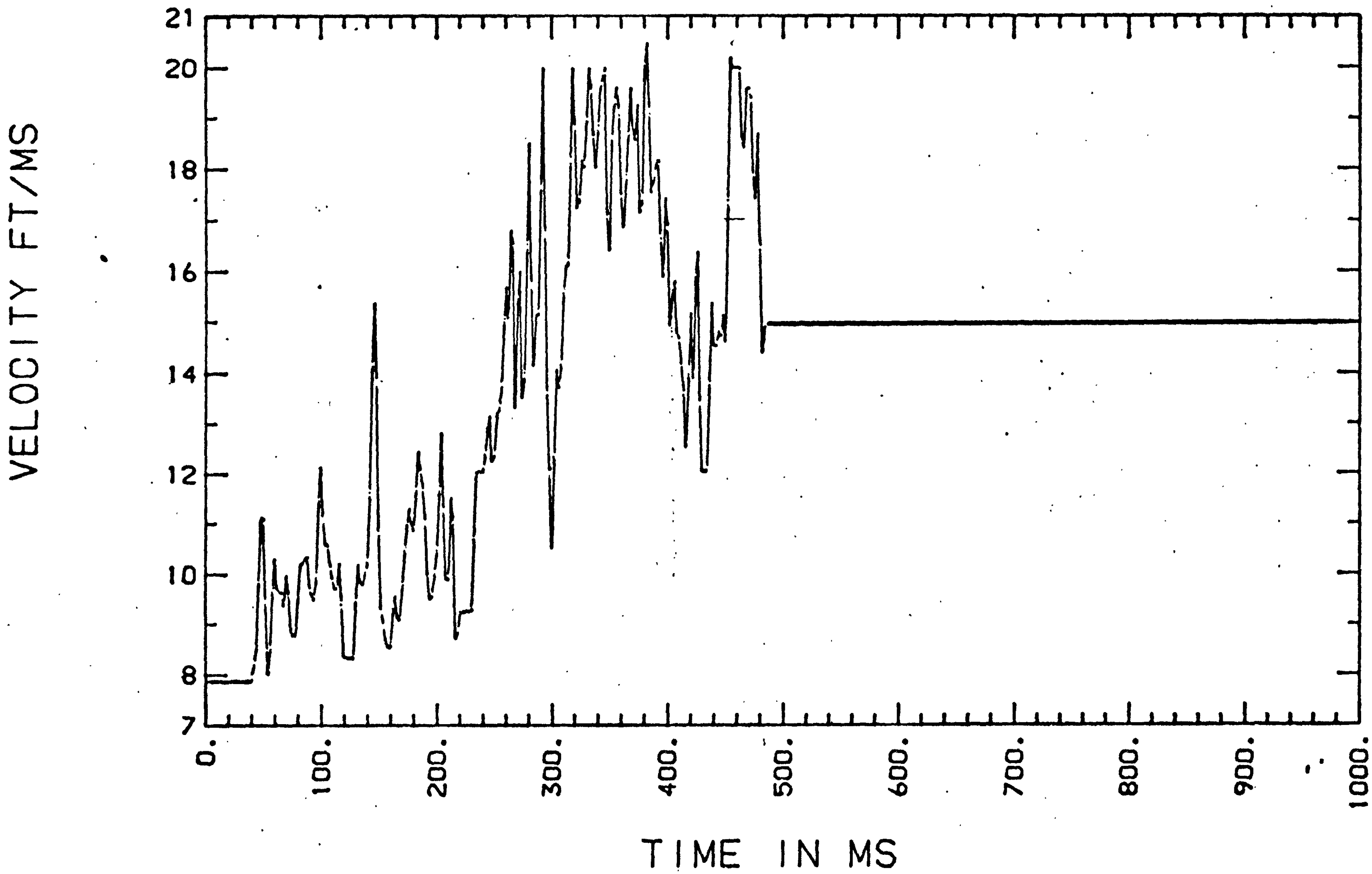


Figure 4 Velocity versus time at Lusk area, wyoming.

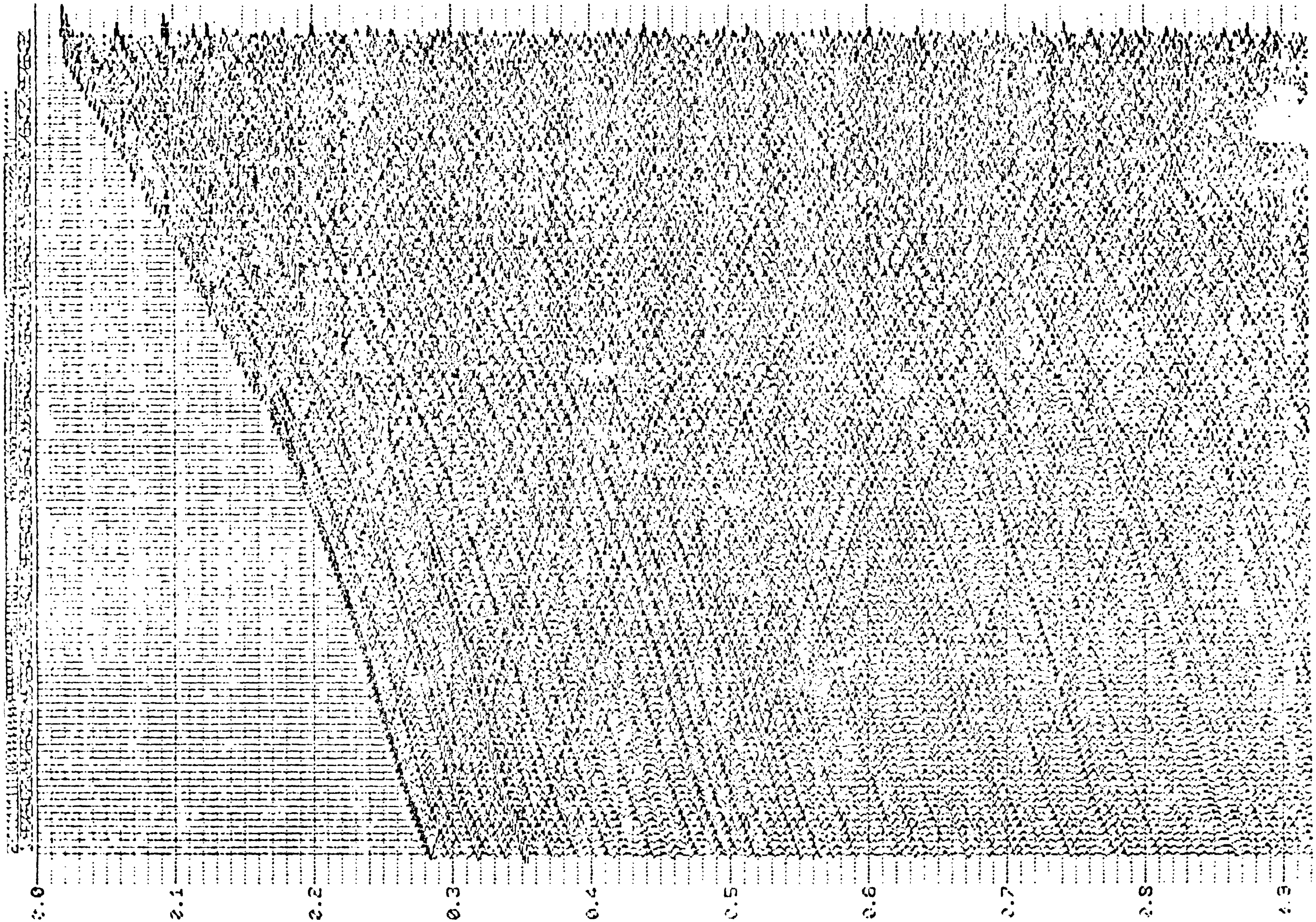


Figure 5. Vertical seismic profiles for Collins Well, Wyoming.

The phase and amplitude change due to a complex reflection coefficient was studied using a simple model without multiples. The model description is shown in Figure 6.

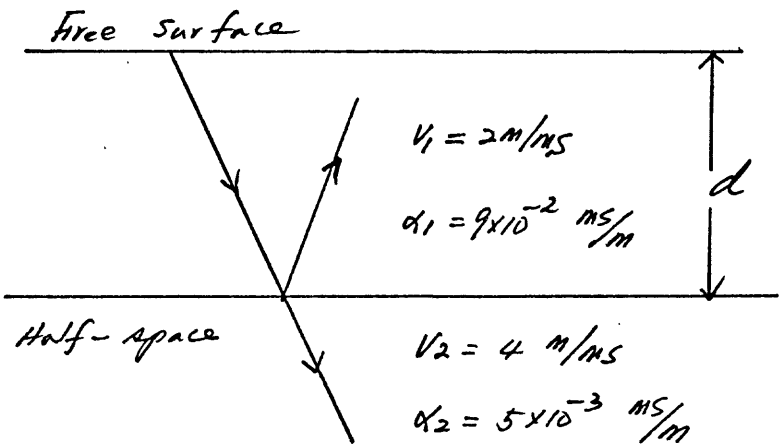


Figure 6. One-layered half-space attenuating model.

Let the input wave function be $P_T(t)$, which we take as

$$P_T(t) = 1 \quad \text{when } 0 \leq t \leq T$$
$$= 0 \quad \text{otherwise.}$$

The reflected pressure $U(t)$ at the free surface can be written, using Equation 28, as

$$U(t) = \frac{2R\cos\theta}{\pi}\left(\tan^{-1}\frac{t-\tau}{\beta} - \tan^{-1}\frac{t-\tau-T}{\beta}\right) - \frac{2R\sin\theta}{\pi}\ln\frac{\beta^2+(t-\tau)^2}{\beta^2+(t-\tau-T)^2}$$

where R is the amplitude of the complex reflection coefficient, θ is the phase angle of the reflection coefficient, $\tau = \frac{2d}{V_1}$, and $\beta = 2\alpha_1 d$.

Figure 7 shows the reflected wave form with varying layer depth "d", from 5 m to 200 m. From Figure 7, we can easily see that the phase distortion due to a complex reflection coefficient may be negligible unless "d" is small. When "d" is large, we can see only broadening of the wave form due to attenuation. So, in calculating a synthetic seismogram for many layered case, we may use a real reflection and transmission coefficients.

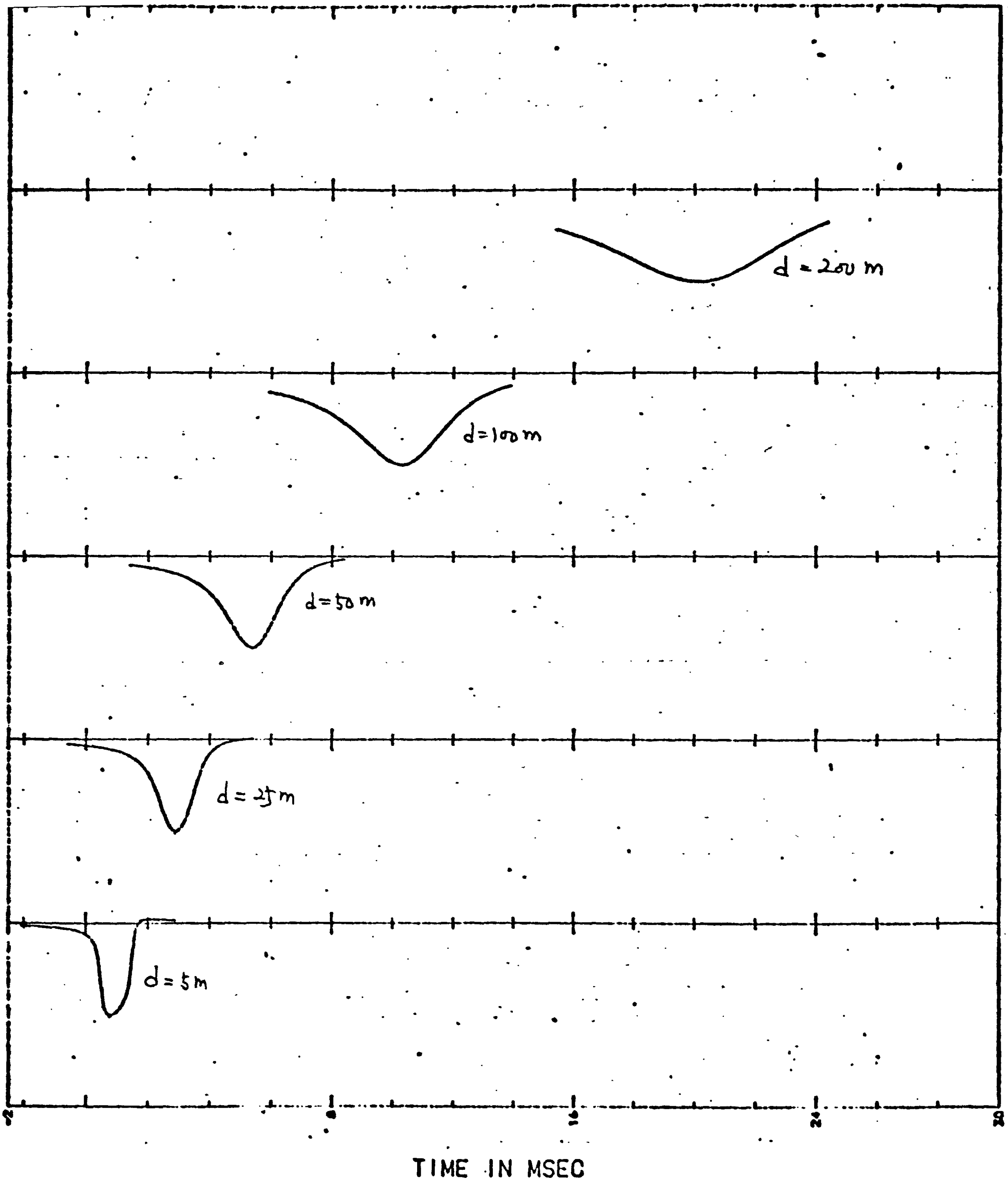


Figure 7. Amplitude and phase distortion of a reflected wave due to attenuation.

Although phase distortion may not be seen in a synthetic seismogram, it might be worth while to include the complex reflection and transmission coefficient in the following cases:

(1) To study a thin layer with high acoustic impedance and attenuation contrast with adjacent layers, especially when receiver and source are very close to the thin layer.

(2) Theoretical study.

Finite Difference Solution

The numerical solution of an elastic wave equation with a finite difference technique and high-speed digital computers has become a powerful seismic modeling tool. The analytic wave equation solution for an inhomogeneous, attenuating medium with multiples is very complicated. Even if we find the analytic solution, the numerical computation of the results is lengthy and complex. This is the reason we studied a finite difference scheme to solve the one-dimensional wave equation.

In this study, we included a perfect elastic medium, a Voigt Solid, and a medium whose attenuation is approximately dependent on the first power of frequency over any limited frequency range.

Since we are interested in many-layered earth models, we solved an inhomogeneous wave equation with two boundaries (one at a free surface and the other at the top of a half-space) instead of solving a homogeneous wave equation with many boundaries.

(A) Perfectly Elastic Medium

The inhomogeneous wave equation in a perfectly elastic medium is

$$\rho \frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x} \alpha \frac{\partial u}{\partial x} \tag{29}$$

where ρ is density, $\alpha = (\lambda + 2\mu)$, λ and μ are Lamé constant, x is vertical distance, and t is time.

We will replace the continuous function $u(x,t)$ with discrete samples of this function as

$$u(x,t) \rightarrow u(j\Delta x, n\Delta t) \triangleq u_j^n.$$

Using the central difference formula, we can show that

$$\frac{\partial}{\partial x}\left(\alpha \frac{\partial u}{\partial x}\right) \rightarrow \frac{\alpha_{j+\frac{1}{2}} u_{j+1}^n - (\alpha_{j+\frac{1}{2}} + \alpha_{j-\frac{1}{2}}) u_j^n + \alpha_{j-\frac{1}{2}} u_{j-1}^n}{(\Delta x)^2}$$

$$\frac{\partial^2 u}{\partial t^2} \rightarrow (u_j^{n+1} - 2u_j^n + u_j^{n-1})/(\Delta t)^2.$$

Therefore, Equation 29 can be written as the following difference equation

$$u_j^{n+1} = 2u_j^n - u_j^{n-1} + h_j\{\alpha_{j+\frac{1}{2}}\, u_{j+1}^n - (\alpha_{j+\frac{1}{2}} + \alpha_{j-\frac{1}{2}})u_j^n + \alpha_{j-\frac{1}{2}}\, u_{j-1}^n\}$$

where $h_j = \dfrac{(\Delta t)^2}{\rho_j(\Delta x)^2}$ (30)

Equation 30 is an explicit difference scheme, and we can compute the displacement at each grid point at time step ($n+1$) in terms of the displacement at the previous time steps (n) and ($n-1$).

The solution of Equation 30 has physical meaning only when the finite difference equation is stable. The sufficient stability condition of the above difference scheme may be derived by considering a homogeneous wave equation.

The stability condition for a homogeneous wave equation is

$$V\frac{\Delta x}{\Delta t} \leq 1,$$ where V is the velocity.

So, we used the following stability condition

$$V_{max}\frac{\Delta x}{\Delta t} \leq 1,$$ where Vmax is the highest velocity of the medium.

We often impose a free surface boundary condition. This requires that the stress must vanish, or at least be a given function of time when a source is located at the free boundary.

Therefore, at $x = 0$

$$\sigma_{xx} = \rho V^2 \frac{\partial u}{\partial x} = -p(t)$$

, where σ_{xx} is stress and $p(t)$ is a pressure source function.

Using an imaginary grid point at $x = -\Delta x$, we can write the above boundary condition as

$$\frac{u_3^n - u_1^n}{2\Delta x} = \frac{-p^n}{\rho_2 V_2^2},$$

where u_1^n is displacement at $x=-\Delta x$, and u_3^n is the displacement at $x=\Delta x$. Notice that when $p^n \to 0$, $u_3^n = u_1^n$, this is the free boundary condition.

At the interface between the two media, continuity of displacement and stress must be satisfied. Let u_h be the displacement in the half-space and since there is no incoming wave from the half-space, $u_h = Ae^{i\omega(t-\frac{x}{V_h})}$ where V_h is the velocity of the half-space. Hereafter the subscript h means half-space. The boundary conditions at $j=J$ (boundary at half-space) are

$$u_J^n = u_h$$

$$\sigma_J^n = \sigma_h$$

Therefore,

$$\rho_J V_J^2 \frac{\partial u_J^n}{\partial x} = \rho_h V_h^2 \frac{\partial u_h}{\partial x} = -\rho_h V_h \frac{\partial u_h}{\partial t}$$

$$= -\rho_h V_h \frac{\partial u_J^n}{\partial t}, \quad \text{because}$$

$$\frac{\partial u_h}{\partial x} = \frac{-i\omega}{V_h} Ae^{i\omega(t-\frac{x}{V_h})} = -\frac{1}{V_h}\frac{\partial u_h}{\partial t}.$$

Using a 3-point backward first derivative operator for $\frac{\partial u_J^n}{\partial x}$, the radiation boundary condition at $j=J$ can be written,

$$u_J^n = \frac{\rho_J V_J^2 (4u_{J-1}^n - u_{J-2}^n)}{2\Delta x\left(\frac{3\rho_J V_J^2}{2\Delta x} + \frac{\rho_h V_h}{\Delta t}\right)} + \frac{\rho_h V_h u_J^{n-1}}{\Delta t\left(\frac{3\rho_J V_J^2}{2\Delta x} + \frac{\rho_h V_h}{\Delta t}\right)}.$$

Notice that when

$$\rho_h V_h \gg \rho_J V_J,$$

$$u_J^n = u_J^{n-1}.$$

This is a rigid boundary condition, since initially all displacement is zero $(u_J' = 0.0)$.

When $\rho_h V_h \ll \rho_J V_J$,

$u_J^n = \frac{4}{3}(u_{J-1}^n - u_{J-2}^n)$, which is a free boundary condition.

(B) Voigt Solid

The inhomogeneous wave equation in a Voigt medium (White, 1965) is

$$\rho \frac{\partial^2 u}{\partial t^2} = \frac{\partial}{\partial x}\left(\alpha \frac{\partial u}{\partial x}\right) + \frac{\partial}{\partial x}\beta \frac{\partial^2 u}{\partial t \partial x} \tag{31}$$

Where ρ, α is the same as perfect elastic case and $\beta = \lambda' + 2\mu'$, which is attenuation term.

Using the same method applied in the perfectly elastic medium, the finite difference equation of the Equation 31 is

$$\begin{aligned}
&-\frac{h_J \beta_{J-\frac{1}{2}}}{2\Delta t} u_{J-1}^{n+1} + \left\{1 + \frac{h_J(\beta_{J+\frac{1}{2}} + \beta_{J-\frac{1}{2}})}{2\Delta t}\right\} u_J^{n+1} - \frac{h_J \beta_{J+\frac{1}{2}}}{2\Delta t} u_{J+1}^{n+1} \\
&= h_J \alpha_{J+\frac{1}{2}} u_{J+1}^n + \left\{2 - h_J(\alpha_{J+\frac{1}{2}} + \alpha_{J-\frac{1}{2}})\right\} u_J^n \\
&\quad + h_J \alpha_{J-\frac{1}{2}} u_{J-1}^n - \frac{h_J \beta_{J+\frac{1}{2}}}{2\Delta t} u_{J+1}^{n-1} \\
&\quad + \left\{\frac{h_J}{2\Delta t}(\beta_{J+\frac{1}{2}} + \beta_{J-\frac{1}{2}}) - 1\right\} u_J^{n-1} \\
&\quad - \frac{h_J}{2\Delta t} \beta_{J-\frac{1}{2}} u_{J-1}^{n-1}
\end{aligned} \tag{32}$$

where $h_J = \dfrac{(\Delta t)^2}{\rho_J (\Delta x)^2}$

Notice that when β is zero, Equation 32 is the same as Equation 30. When $\beta \neq 0$, the above equation is an implicit difference scheme, and its solution can be easily solved by the property of tri-diagonal matrix method (Richtmyer and Morton, 1967).

Balch (1970) showed that the stability condition is

$$V\frac{\Delta x}{\Delta t} \leqslant 1.0 \ .$$

The free boundary condition at $x=0$ is

$$u_1^{n+1} = u_3^{n+1} + (u_1^{n-1} - u_3^{n-1}) + \frac{2\beta_2 V_2^2 \Delta t}{(\lambda' + 2\mu')_2}(u_3^n - u_1^n) - \frac{4\Delta x \Delta t}{\beta_2} P^n .$$

The radiation boundary condition at $x = J\Delta x$ is

$$u_J^n = \frac{b}{(a+3b+3c)}(4u_{J-1}^n - u_{J-2}^n) + \frac{c}{(a+3b+3c)}(4u_{J-1}^n - u_{J-2}^n + u_{J-2}^{n-1} - 4u_{J-1}^{n-1} + 3u_J^{n-1}) + \frac{a}{(a+3b+3c)} u_J^{n-1} ,$$

where

$$a = \frac{\rho_h V_h}{\Delta t}$$

$$b = \frac{\rho_J V_J^2}{2\Delta x}$$

$$c = \frac{\beta_J}{2\Delta x \Delta t} .$$

(C) Linear with Frequency Medium

The following sets of equations satisfy the condition that a first power dependency on frequency could be approximated over any limited frequency range (White, 1965).

$$\rho \frac{\partial^2 u}{\partial t^2} = \frac{\partial \sigma_{xx}}{\partial x}$$
$$\sigma_{xx} + \gamma \frac{\partial \sigma_{xx}}{\partial t} = k \left(\frac{\partial u}{\partial x} + \beta \frac{\partial^2 u}{\partial x \partial t} \right) \tag{33}$$

Where $k = \lambda + 2\mu$

γ, β = attenuation constant.

In difference equation notation,

$$u_j^{n+1} = \frac{(\Delta t)^2}{\rho_j \Delta x} \left(\sigma_{j+\frac{1}{2}}^{n} - \sigma_{j-\frac{1}{2}}^{n} \right) + 2u_j^{n} - u_j^{n-1}$$
$$\sigma_{j+\frac{1}{2}}^{n} \left(1 + \frac{\gamma_{j+\frac{1}{2}}}{\Delta t}\right) = k_{j+\frac{1}{2}} \frac{u_{j+1}^{n} - u_j^{n}}{\Delta x} + \frac{\gamma_{j+\frac{1}{2}}}{\Delta t} \sigma_{j+\frac{1}{2}}^{n-1}$$
$$+ (k\beta)_{j+\frac{1}{2}} \frac{u_{j+1}^{n} - u_j^{n} - u_{j+1}^{n-1} + u_j^{n-1}}{\Delta x \cdot \Delta t} . \tag{34}$$

From the above coupled equation, we can compute explicitly the displacement at the (n+1) time step using the previous displacements and stresses.

The free boundary condition is

$$u_1^{n+1} = u_3^{n+1} - u_3^{n-1} + u_1^{n-1} + \frac{2\Delta t}{\beta_2} \left(u_3^{n} - u_1^{n} \right)$$
$$- \frac{4\Delta x}{\rho_2 V_2^2 \beta_2} \left\{ \gamma_2 \left(p_2^{n} - p_2^{n-1} \right) + \Delta t \, p_2^{n} \right\} .$$

The radiation boundary condition is

$$U_J^n = \frac{c}{Q}\left(4U_{J-1}^n - U_{J-2}^n\right) + \frac{d}{Q}\left(U_{J-2}^{n-1} - 4U_{J-1}^{n-1} + 4U_{J-1}^n - U_{J-2}^n + 3U_J^{n-1}\right) + \frac{b(1+2a)}{Q}U_J^{n-1} - \frac{ab}{Q}U_J^{n-2}$$

Where

$$a = \frac{c}{\Delta t}$$

$$b = \frac{\rho_J V_J}{\Delta t}, \quad c = \frac{\rho_J V_J^2}{2\Delta x}$$

$$d = \frac{\beta_J \rho_J V_J^2}{2\Delta x \Delta t}$$

$$Q = 3c + 3d + b(a+1)$$

Examples and Discussions

To study the accuracy and feasibility of an explicit difference scheme for the solution of an inhomogeneous perfectly elastic wave equation, we examined the following simple model, which consists of two different perfectly elastic bars in welded contact, Figure 8.

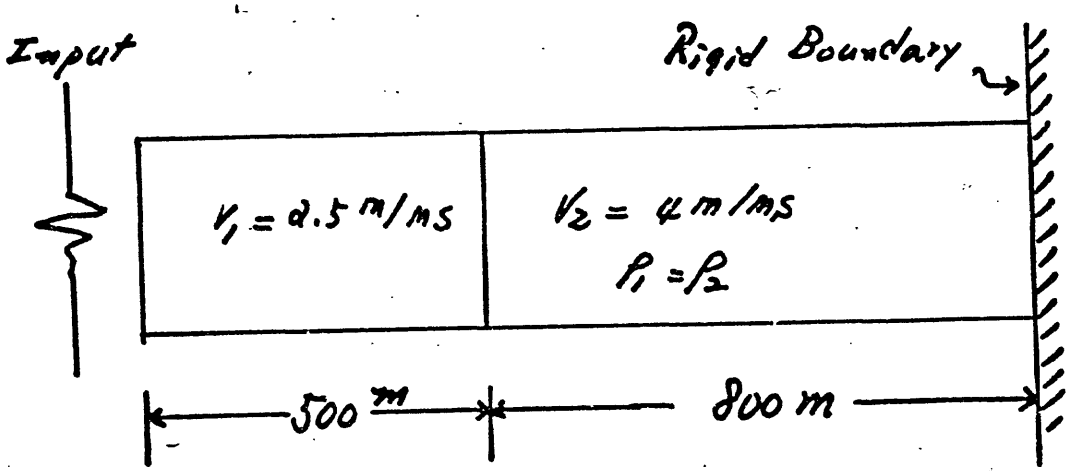


Figure 8. Geometry for one-dimensional perfectly elastic model.

The external boundary conditions for this model are the displacement field at one end and a rigid boundary at the other end. We computed the displacement field along the bar at $t = 150\ ms$, $t = 350\ ms$, and $t = 550\ ms$.

The solid line in Figure 9 represents the solution using a finite difference approach and the dots represent the analytic solution. The free surface reflected wave looks a little delayed. But overall agreement between two solutions is excellent.

To test the computer algorithm for the Voigt solid, we compared our result with a known analytic solution derived by Collins (1960). For a pressure impulse at the free surface of a homogeneous half-space, we calculated the velocity wave form at a dimensionless distance $X = 4$ with respect to dimensionless time T. The dimensionless distance and time are defined as

$$X = \frac{x}{\sqrt{\varepsilon}}, \quad T = \frac{t}{\varepsilon},$$

where

$$\varepsilon = \frac{\lambda' + 2\mu'}{\lambda + 2\mu}.$$

The points in Figure 10 were calculated from Collin's analytic solution, and the solid curve was computed by the finite difference method. The agreement between the two solutions is excellent.

To examine the computer program for the linear with frequency medium, we computed the attenuation of longitudinal wave versus frequency. Figure 11 shows the attenuation of longitudinal waves versus frequency for a semi-infinite medium whose longitudinal velocity is 7100 ft/sec; shear velocity is 2860 ft/sec γ is 0.345 ms, and $\beta = 0.375$ ms. This figure shows that attenuation is nearly proportional to the first power of frequency in the limited frequency range.

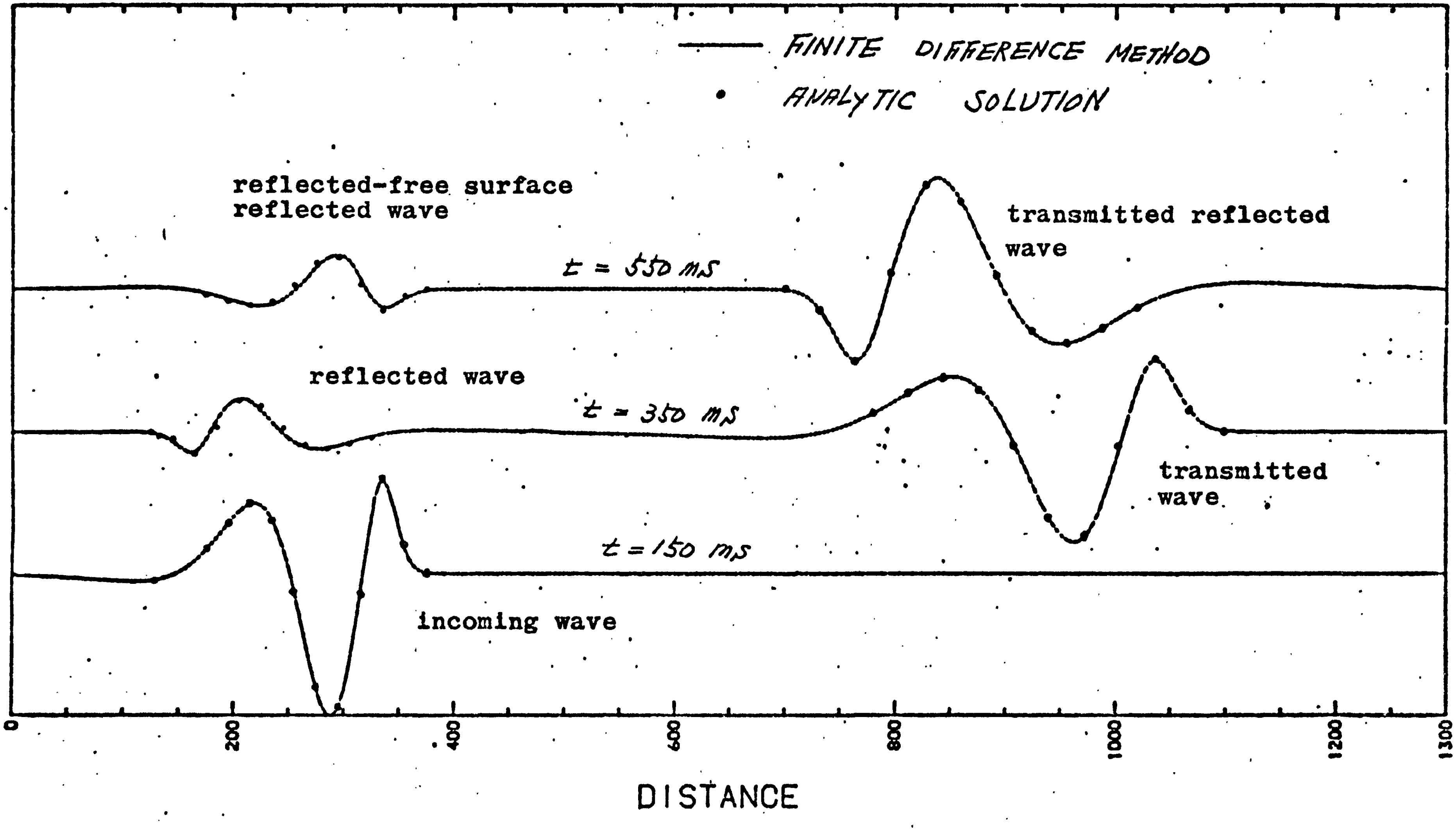


Figure 9. Comparision between an analytic solution and a finite difference solution for a perfectly elastic model.

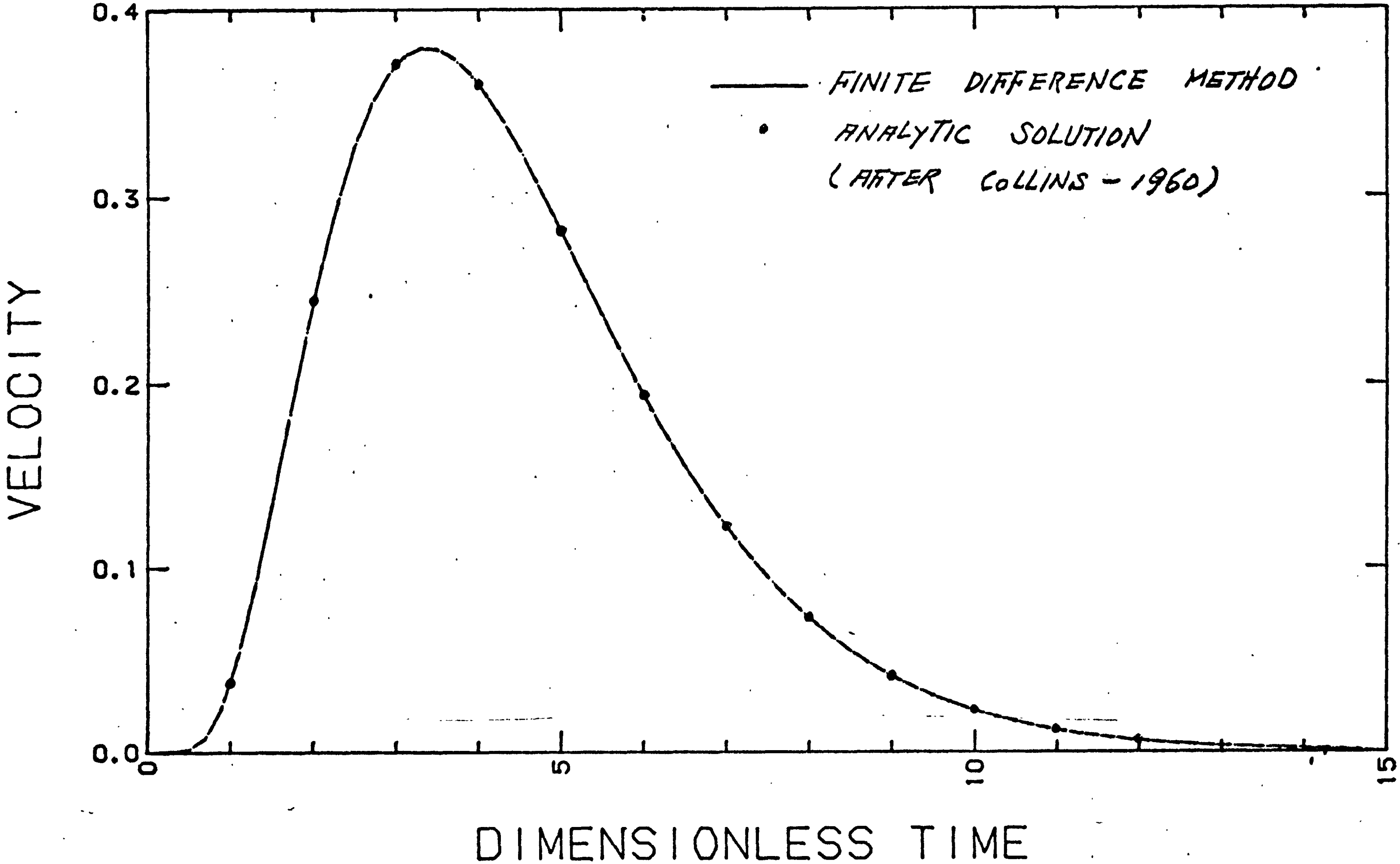


Figure10. Comparision between an analytic solution and finite difference solution for a Voigt solid.

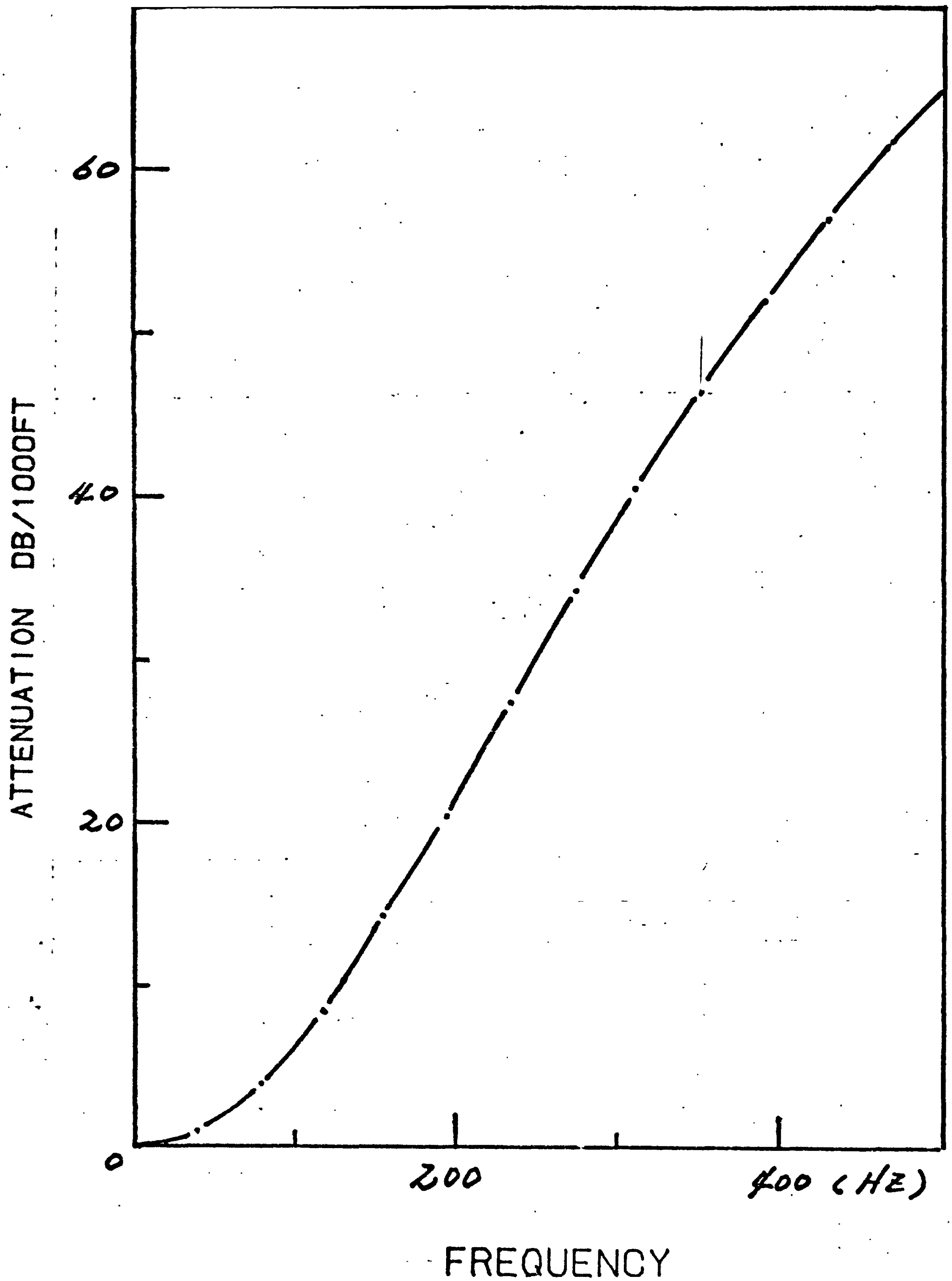


Figure 11. Attenuation of longitudinal waves versus frequency.

One of the problems encountered in solving an inhomogeneous wave equation by a finite difference scheme is that the ratio $\Delta t/\Delta x$ is constant throughout the medium. The ratio $\Delta t/\Delta x$ controls the accuracy of the propagation of a given input wavelet in a finite difference method. If $(\Delta t/\Delta x)^{-1}$ is the same as the speed of the medium, the propagation error is zero-- this means that if a delta function of a displacement is introduced into a finite difference equation for an infinite one-dimensional medium, it propagates as a delta function without tails or precursors, as the theory predicts. However, if $\Delta x/\Delta t$ is not the same as medium speed, precursors and tails appear and the results are distorted. The larger $\Delta x/\Delta t$ compared with V (medium velocity) is, the more the propagation error is. This kind of error is more severe than the local truncation error and round-off error. Therefore, some method must be developed for the estimation of this propagation error.

The amplitude spectrum of the Fourier transform of a delta function is equal to one for all frequencies. Knowing this, we put a delta function displacement into our finite difference computer model and computed the amplitude spectrum of the resultant displacement at a certain point.

Figure 12 shows the amplitude spectrum of the displacement at $X = 200\Delta X$ with two values of ρ, which is defined as $\rho = V\Delta t/\Delta x$. Figure 13 shows the amplitude spectrum at $X = 100\Delta X$. Both figures clearly show that the larger ρ is, the more accurate the amplitude response is. Also the shorter the propagation distance is, the more accurate the response is.

Therefore, from a set of curves of this kind, we can estimate the maximum frequency which may be contained in an input wavelet, if the wavelet is to be propagated without much distortion. With this maximum frequency, a stability condition, and some criteria for the local truncation error of the difference operator, we can choose the optimum value of the $\Delta X/\Delta t$ ratio for a given numerical model.

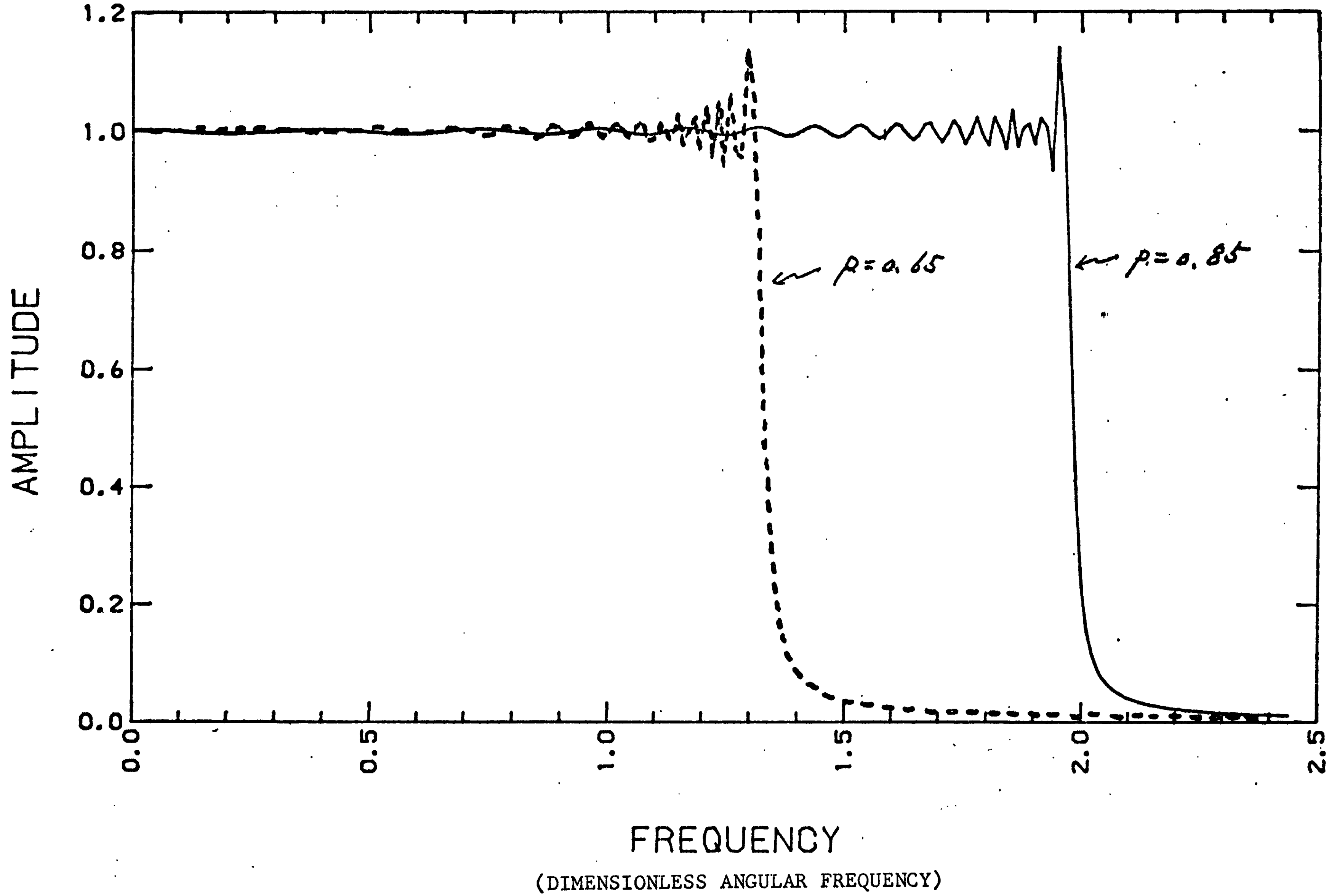


Figure 12. Amplitude spectrum of the displacement at x=200 Δx due to a unit delta function displacement input.

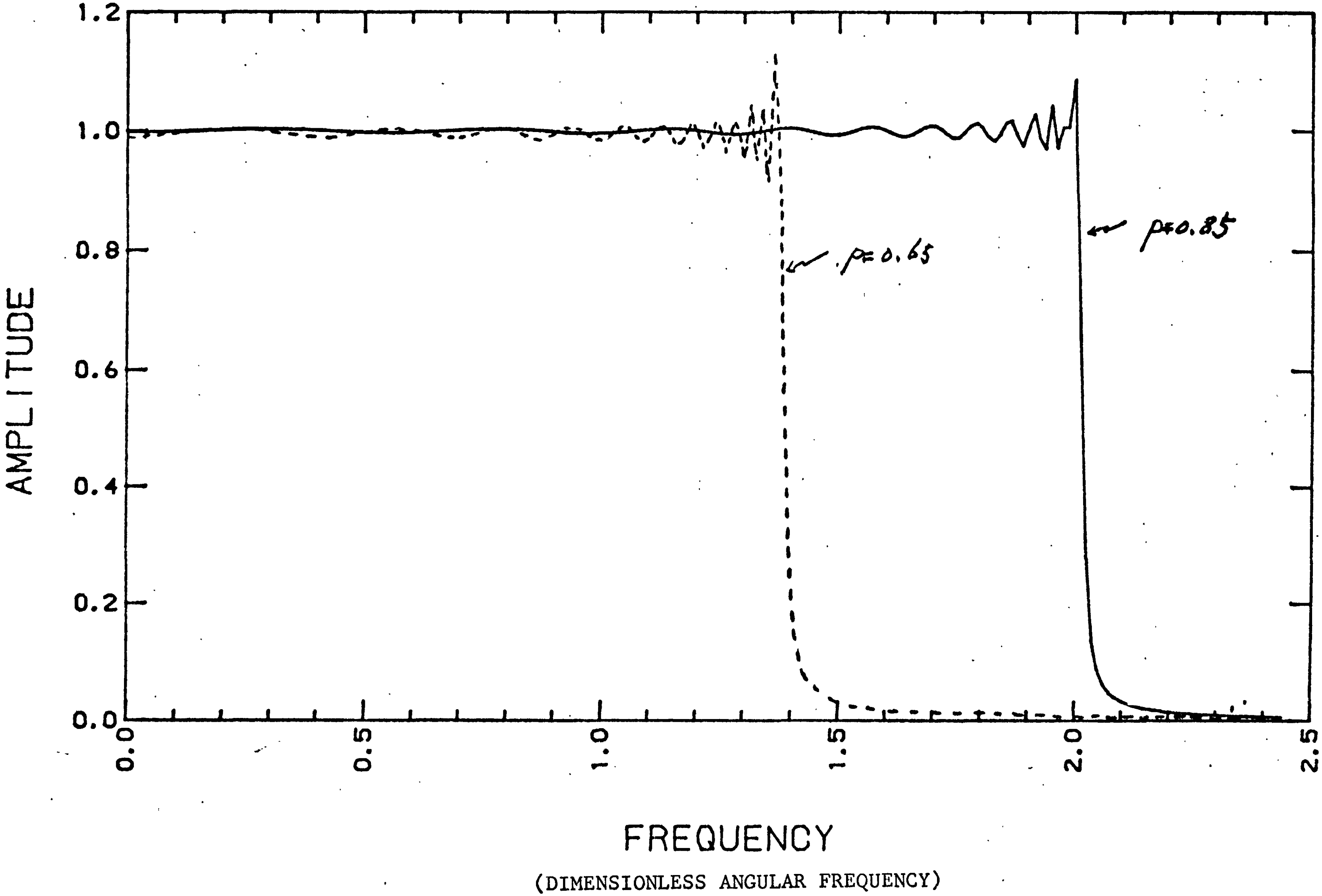


Figure 13. Amplitude spectrum of the displacement at x=100 Δx due to a unit delta function displacement input.

TWO-DIMENSION WAVE PROPAGATION

The previous chapter deals with one-dimensional wave propagation through horizontally layered media. This approach provides reasonable answers for many problems encountered in geophysical exploration. But it has limitations. For example, it cannot handle such problems as predicting the converted wave from a boundary, the geometrical spreading effect, or a non-normal incidence wave. Therefore, to make a realistic seismic model, it is necessary and important to study the solution of a two-dimensional elastic wave equation.

Presently, most of the two-dimensional modeling techniques use ray tracing. To calculate an arrival time for the seismic signal for a normal or nearly vertical ray path for an arbitrary subsurface, the ray theory may provide a reliable answer. Also this technique can solve some of the problems such as geometrical spreading, and mode conversion for a simple model. But present two-dimensional modeling techniques cannot fully solve the diffraction problem, all kinds of mode conversion, and true amplitude and shape of the seismic signal.

One way to eliminate the above problems, as far as theory is concerned, is to compute the entire elastic wave field by a finite difference method. Currently, numerous authors are studying the solution of an elastic wave equation by a finite difference method (Aboudi, 1971; Alford and others, 1974; Alterman and Aboudi, 1970; Alterman and Karal, 1969). Theoretically, a finite difference equation can give an exact solution of the elastic wave equation as the sampling interval both in space and time approach zero for any complex geological subsurface.

Thus, the purpose of this section is to study the feasibility and applicability of this finite difference approach. If subsurface structure is simple, it seems to be more reasonable to solve a homogeneous wave equation with appropriate boundary conditions at each boundary by a finite difference method. However, for a complex geological model to fit a boundary condition at each boundary is rather complicated. So we set out to solve an inhomogeneous elastic wave equation, which contains all interface boundary conditions in itself, by a finite difference scheme.

Difference Equation

Two-dimensional inhomogeneous ealstic wave equation in an orthogonal cartesian coordinates x and y can be written as

$$\rho \frac{\partial^2 u}{\partial t^2} = HU + \frac{\partial}{\partial x}(\lambda+2\mu)\frac{\partial u}{\partial x} + \frac{\partial \lambda}{\partial x}\frac{\partial w}{\partial y} + \frac{\partial \mu}{\partial y}\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial y}\right)$$

$$\rho \frac{\partial^2 w}{\partial t^2} = HW + \frac{\partial}{\partial y}(\lambda+2\mu)\frac{\partial w}{\partial y} + \frac{\partial \lambda}{\partial y}\frac{\partial u}{\partial x} + \frac{\partial \mu}{\partial x}\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial y}\right) \qquad (35)$$

$$HU = (\lambda+2\mu)\frac{\partial^2 u}{\partial x^2} + \lambda\frac{\partial^2 w}{\partial x \partial y} + \mu\left(\frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 w}{\partial x \partial y}\right) + \rho \overset{x}{F}$$

$$HW = (\lambda+2\mu)\frac{\partial^2 w}{\partial y^2} + \lambda\frac{\partial^2 u}{\partial x \partial y} + \mu\left(\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 u}{\partial x \partial y}\right) + \rho \overset{y}{F}$$

where

ρ : density

λ, μ :Lame constants

u : horizontal displacement

w : vertical displacement

$\overset{x}{F}, \overset{y}{F}$: x and y-component of body force.

Let

$$u(x,y,t) \rightarrow u(i\Delta x, j\Delta y, n\Delta t) \triangleq U^n_{i,j}.$$

Using a central difference approximation for all of the derivatives, we can write Equation 35 as the following coupled difference equation, assuming $\Delta x = \Delta y$ for simplicity:

$$\begin{aligned}
U^{n+1}_{i,j} &= \widetilde{HU} + 2U^n_{i,j} - U^{n-1}_{i,j} \\
&+ \frac{h_{i,j}}{4}\left[(\lambda+2\mu)_{i+1,j} - (\lambda+2\mu)_{i-1,j}\right]\left(U^n_{i+1,j} - U^n_{i-1,j}\right) \\
&+ \frac{h_{i,j}}{4}\left[\lambda_{i+1,j} - \lambda_{i-1,j}\right]\left(W^n_{i,j+1} - W^n_{i,j-1}\right) \\
&+ \frac{h_{i,j}}{4}\left(\mu_{i,j+1} - \mu_{i,j-1}\right)\left(W^n_{i+1,j} - W^n_{i-1,j} + \right. \\
&\left. U^n_{i,j+1} - U^n_{i,j-1}\right)
\end{aligned} \tag{36}$$

$$\begin{aligned}
W^{n+1}_{i,j} &= \widetilde{HW} + 2W^n_{i,j} - W^n_{i,j} \\
&+ \frac{h_{i,j}}{4}\left[(\lambda+2\mu)_{i,j+1} - (\lambda+2\mu)_{i,j-1}\right]\left(W^n_{i,j+1} - W^n_{i,j-1}\right) \\
&+ \frac{h_{i,j}}{4}\left[\lambda_{i,j+1} - \lambda_{i,j-1}\right]\left(U^n_{i+1,j} - U^n_{i-1,j}\right) \\
&+ \frac{h_{i,j}}{4}\left(\mu_{i+1,j} - \mu_{i-1,j}\right)\left(W^n_{i+1,j} - W^n_{i-1,j} + \right. \\
&\left. U^n_{i,j+1} - U^n_{i,j-1}\right)
\end{aligned} \tag{37}$$

$$\widetilde{HU} = h_{i,j}(\lambda+2\mu)_{i,j}\left(U^n_{i+1,j} - 2U^n_{i,j} + U^n_{i-1,j}\right) +$$

$$\frac{h_{i,j}}{4}(\lambda+\mu)_{i,j}\left(W^n_{i+1,j+1} - W^n_{i-1,j+1} - W^n_{i+1,j-1} + W^n_{i-1,j-1}\right) +$$

$$h_{i,j}\mu_{i,j}\left(U^n_{i,j+1} - 2U^n_{i,j} + U^n_{i,j-1}\right)$$

$$+ (\Delta t)^2 \overset{x}{F}{}^n_{i,j}$$

$$\widetilde{HW} = h_{i,j}(\lambda+2\mu)_{i,j}\left(W^n_{i,j+1} - 2W^n_{i,j} + W^n_{i,j-1}\right) +$$

$$\frac{h_{i,j}}{4}(\lambda+\mu)_{i,j}\left(U^n_{i+1,j+1} - U^n_{i-1,j+1} - U^n_{i+1,j-1} + U^n_{i-1,j-1}\right) +$$

$$h_{i,j}\mu_{i,j}\left(W^n_{i+1,j} - 2W^n_{i,j} + W^n_{i-1,j}\right)$$

$$+ (\Delta t)^2 \overset{y}{F}{}^n_{i,j}$$

where $h_{i,j} = (\Delta t)^2/(\rho_{i,j}\,\Delta x^2)$.

The finite difference Equations 36 and 37 are explicit, so that the displacement at the (n+1)-th time step can be computed using the previous n-th and (n-1)-th time step displacements and a body force term.

The difference equations have physical meaning only when they are stable. A sufficient stability condition may be derived by considering the homogeneous difference equation. Aboudi(1971) showed that the difference scheme is stable when

$$\Delta t \leq \frac{\Delta x}{\sqrt{\alpha^2+\beta^2}}$$

where α is p-wave velocity and β is s-wave velocity. So, in the inhomogeneous case, we can determine by the maximum value of $\sqrt{\alpha^2+\beta^2}$ of the medium.

The free boundary condition at y=0 is

$$\sigma_{xy} = \mu\left(\frac{\partial w}{\partial x} + \frac{\partial u}{\partial y}\right) = T_x$$

$$\sigma_{yy} = \lambda\left(\frac{\partial u}{\partial x} + \frac{\partial w}{\partial y}\right) + 2\mu\frac{\partial w}{\partial y} = T_y$$

where T_x and T_y are given surface tractions and σ_{xy} and σ_{yy} are components of stress.

As in the one-dimensional case, we introduced an imaginary grid point at $y = -\Delta y$ to compute the first derivative. In difference equation notation, the above boundary condition can be written as

$$u_{i,1}^n = u_{i,3}^n + \frac{\Delta y}{\Delta x}\left(w_{i+1,2}^n - w_{i-1,2}^n\right) - \frac{2\Delta y}{\mu} T_x^n$$

$$w_{i,1}^n = \frac{\lambda}{\lambda+2\mu}\frac{\Delta y}{\Delta x}\left(u_{i+1,2}^n - u_{i-1,2}^n\right) + w_{i,3}^n - \frac{2\Delta y}{\lambda+2\mu} T_y^n. \qquad (38)$$

When there is no surface traction, Equation 38 serves as a free boundary condition.

At the edge of the model, we provided a rigid boundary condition, which will produce unwanted "artificial" reflections. We tried to implement radiation boundary condition, but have as yet not been successful.

In the inhomogeneous finite difference formulation, every elastic constant (velocity and density) must be a continuous function of grid points. Thus, all discontinuous variation of elastic constants must be changed into continuous ones. If the continuous variation of elastic constants is sufficiently abrupt in the inhomogeneous formulation, we can approximately treat this continuous interface as a discontinuous interface. Figure 14 shows how we treat this discontinuous interface in the inhomogeneous difference scheme.

Most of the authors (Alford and others, 1974, Alterman and Karal, 1969) used an analytic solution in an infinite

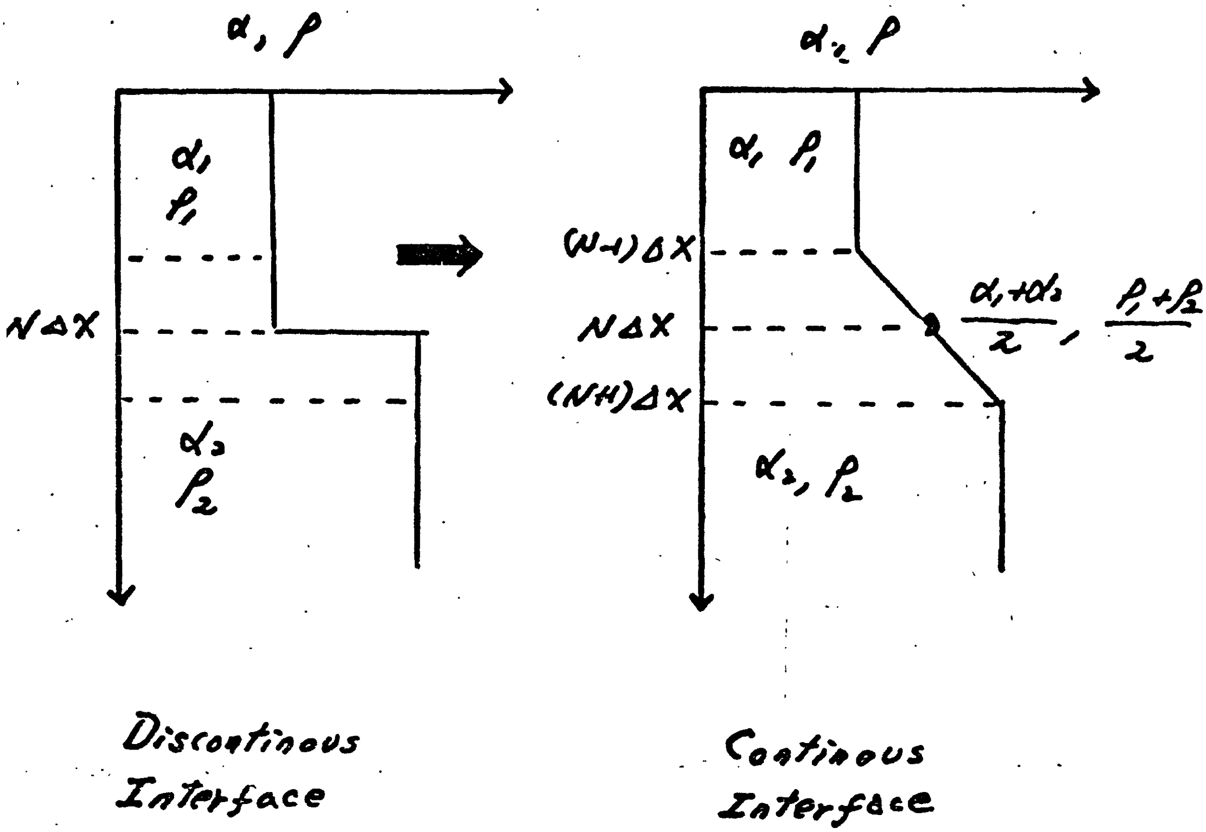


Figure 14. Elastic constant transition for inhomogeneous formulation.

medium around a source region to initiate an elastic disturbance. Alterman and Karal (1969) discussed how to fit a boundary condition around the source using an analytic solution. In this study, we used an initial disturbance by a finite difference method, which has been fully explained by Aboudi (1971). So we will not discuss this simulation of seismic source in a finite difference scheme here.

Grid Dispersion Relation

When we perform numerical calculations of wave propagation, using a finite difference equation, a propagating pulse on a discrete grid shows dispersion (Alford and others, 1974). This phenomena, called grid dispersion, can be examined by considering phase velocity as a function of frequency. The following derivation of grid dispersion for the displacement is based upon the plane wave propagation in a whole space by a finite difference scheme.

Let $\vec{D}$ be the plane wave solution such that

$$\vec{D} = \vec{D}_0 \, e^{i(\omega t - kx\cos\theta - ky\sin\theta)} \qquad (39)$$

We substitute Equation 39 into Equations 36 and 37, retaining only the homogeneous terms. Then,

$$\begin{aligned} 4\vec{D}_0 \sin^2\left(\frac{\omega \Delta t}{2}\right) = {} & 4[A]\varepsilon^2 \vec{D}_0 \sin^2\left(\frac{\Delta x \cos\theta k}{2}\right) + \\ & 4[B]\varepsilon^2 \vec{D}_0 \sin^2\left(\frac{\Delta x \cos\theta k}{2}\right) + \\ & [C]\vec{D}_0 \varepsilon^2 \sin(\Delta x \cos\theta k) \sin(\Delta x \sin\theta k) \end{aligned}$$

where

$$[A] = \begin{bmatrix} \alpha^2 , & 0 \\ 0 , & \beta^2 \end{bmatrix} \qquad [B] = \begin{bmatrix} \beta^2 , & 0 \\ 0 , & \alpha^2 \end{bmatrix}$$

$$[C] = \begin{bmatrix} 0 , & \alpha^2-\beta^2 \\ \alpha^2-\beta^2 , & 0 \end{bmatrix} \qquad \varepsilon = \frac{\Delta t}{\Delta x}$$

Let

$$\frac{k\Delta x \cos\theta}{2} = p \qquad \frac{k\Delta x \sin\theta}{2} = q$$

$$S_1 = \varepsilon^2(\alpha^2 \sin^2 p + \beta^2 \sin^2 q)$$

$$S_1 = \varepsilon^2(\alpha^2 \sin^2 p + \beta^2 \sin^2 q)$$
$$S_2 = \varepsilon^2(\beta^2 \sin^2 p + \alpha^2 \sin^2 q)$$
$$S_3 = \varepsilon^2(\alpha^2 - \beta^2) \sin p \sin q \cos p \cos q .$$

Then, to get non-trivial solution for $\vec{P_0}$, the following determinant should be 0.

$$\begin{vmatrix} \sin^2\left(\frac{\omega_0 t}{2}\right) - S_1 , & S_3 \\ S_3 , & \sin^2\left(\frac{\omega_0 t}{2}\right) - S_2 \end{vmatrix} = 0 \qquad (40)$$

The Equation 40 can be written as

$$\left(\sin^2 \frac{\omega_0 t}{2} - E_1\right)\left(\sin^2 \frac{\omega_0 t}{2} - E_2\right) = 0 .$$

The solution E_1 or E_2 is

$$E_1 = \frac{(S_1 + S_2) + \sqrt{(S_1 - S_2)^2 + 4S_2^2}}{2}$$

$$E_2 = \frac{(S_1 + S_2) - \sqrt{(S_1 - S_2)^2 + 4S_2^2}}{2} .$$

Therefore,

$$\frac{\omega_0 t}{2} = \sin^{-1}\sqrt{E_1} \quad \text{or} \quad \sin^{-1}\sqrt{E_2} . \qquad (41)$$

The Equation 41 provides us with a relation between grid dispersion and the sampling interval. Examining the limiting case (i.e., $\Delta x \to 0$, $\Delta t \to 0$), $\omega_0 t/2 = \sin^{-1}\sqrt{E_1}$ gives us a grid dispersion relation for the longitudinal wave and $\omega_0 t/2 = \sin^{-1}\sqrt{E_2}$ gives us a grid dispersion relation for the shear wave.

Grid dispersion relation for the longitudinal wave is as follows:

$$Q_\alpha = \frac{C_p}{C_0} = \frac{\omega}{\alpha k} = \frac{2 \sin^{-1}\sqrt{E_1}}{\alpha k \Delta t}$$

where C_0 is the phase velocity at zero frequency and C_p is the phase velocity at frequency ω .

Define

$$p = \alpha \frac{\Delta t}{\Delta x}$$

Then,

$$Q_\alpha = \frac{G \sin^{-1}\sqrt{E_1}}{p\pi}$$

where G is the number of grid points per wave length.

Like the p-wave grid dispersion relation, the s-wave grid dispersion relation is

$$Q_\beta = \frac{\omega}{\beta k} = \frac{G \sin^{-1}\sqrt{E_2}}{p\pi}$$

Examples and Discussions

Grid dispersion is one of the potential sources of trouble in finite difference calculations. So it is very important to study this kind of error in numerical modeling by finite difference schemes.

Figure 15 and Figure 16 show the normalized p-wave phase velocity for different propagation angles as a function of grid points per wave length, where

Q_α : p-wave phase velocity/zero frequency p-wave phase velocity
G : number of grid points/wave length
θ : propagation angle with respect to the grid

$$p = \alpha \Delta t / \Delta x$$
$$r = \beta^2 / \alpha^2 .$$

From Figures 15 and 16, we can say that:

(1) The larger the value of p becomes, the smaller the grid dispersion error becomes.

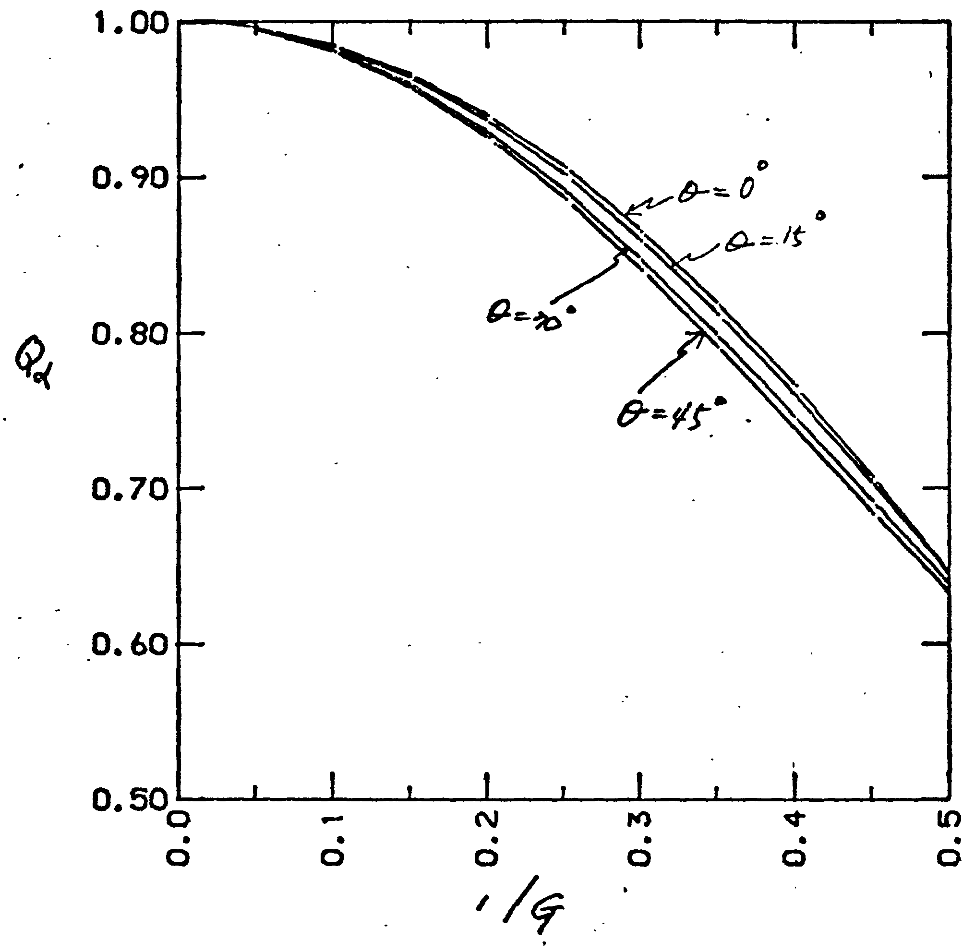


Figure 15. P-wave grid dispersion relation for p=0.3 and r=0.

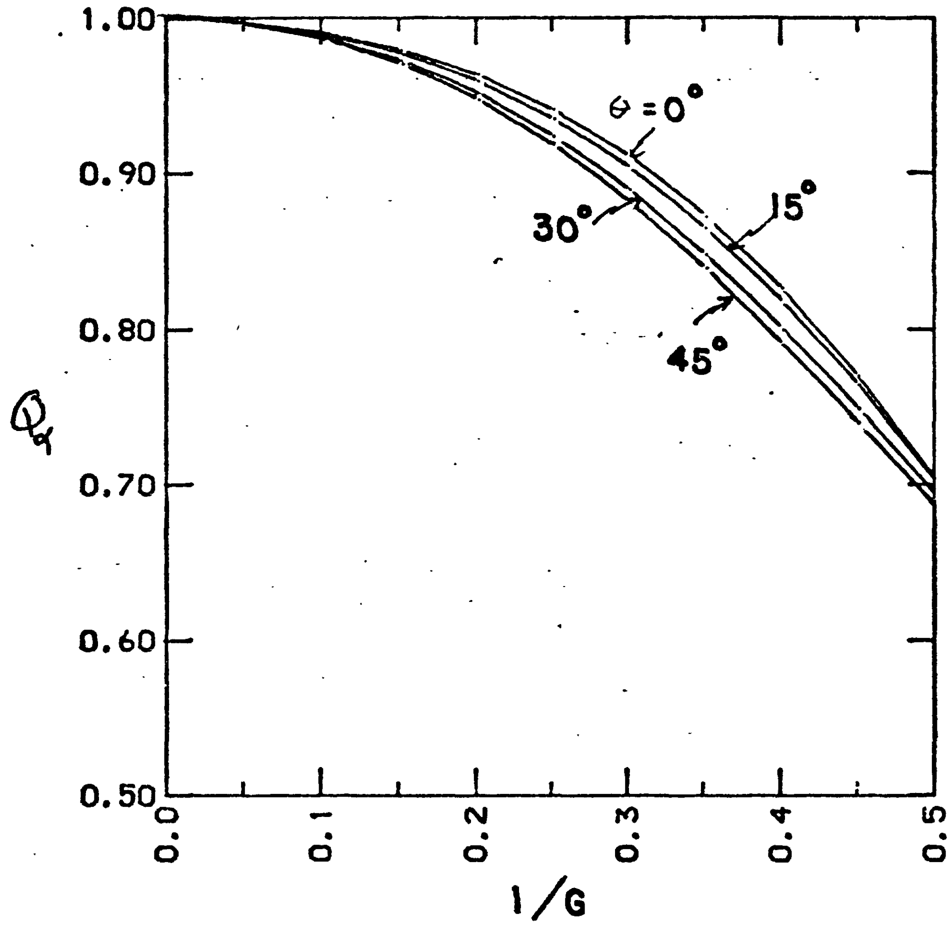


Figure 16. P-wave grid dispersion relation for p=0.7 and r=0.

(2) If 1/G is less than 0.05, this type of error is almost independent of p, and its phase velocity is nearly that of zero frequency p-wave phase velocity.

(3) Propagation angle is a minor factor for the p-wave grid dispersion error.

Figures 17 and 18 show the grid dispersion relation for the s-wave propagation, where

Q_β : s-wave phase velocity/zero frequency s-wave phase velocity.

From Figure 17 and Figure 18, we observe that:

(1) As the value of $\Delta x/\Delta t$ approaches the s-wave velocity, the s-wave grid dispersion error becomes smaller.

(2) If 1/G is less than 0.05, this type of error is almost independent of p, and its phase velocity is nearly that of the zero frequency s-wave phase velocity.

(3) The grid dispersion is largely dependent on the angle of propagation.

Therefore, for a given $\Delta x/\Delta t$ ratio, which must satisfy the stability condition, if the value of 1/G is nearly 0.05 for both p-wave and s-wave, we may not expect severe grid dispersion errors in a finite difference scheme.

A point source approximation in a finite difference scheme is described by Aboudi (1971). Using his temporal and spacial dependence of a point source, we compared the displacement field in a whole space computed by an analytic solution with a finite difference solution. Figure 19 shows a radial displacement at $x = 5\Delta x$, and $y = 20\Delta x$. The consistant discrepancy between two solutions is caused by the finite sampling interval and the ratio of grid size to the pulse width of the input source function. Actually, in this finite difference scheme, any theoretical point source is approximated by an extended source (in this example, the source region is extended by $4\Delta y$ and $4\Delta x$ in x- and y-direction respectively, which is the cloest numerical approximation to a point source). For the further discussions of these kind of extended source compared with point sources, the readers may consult Alterman and Aboudi (1970).

Figure 20 shows the radial displacement at $x = 10\Delta x$, $\theta = 0^\circ$ and $\theta = 37^\circ$ in an elastic whole space. One of the displacement is reversed in sign. We cannot see any noticeable differences between these two displacements. So we can say that, in this particular example, the angular dependence of the propagation error of a symmetric point source is

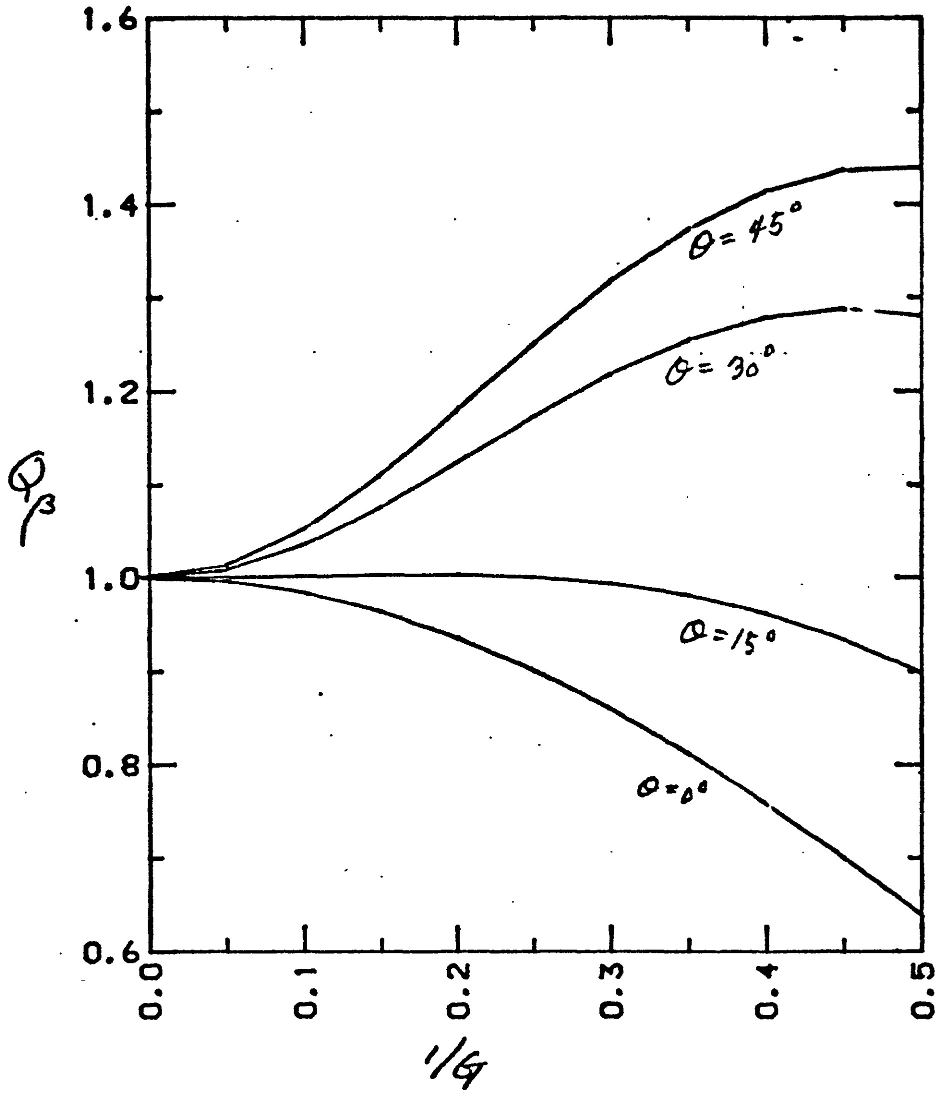


Figure 17. S-wave grid dispersion relation for p=0.3 and r=0.16.

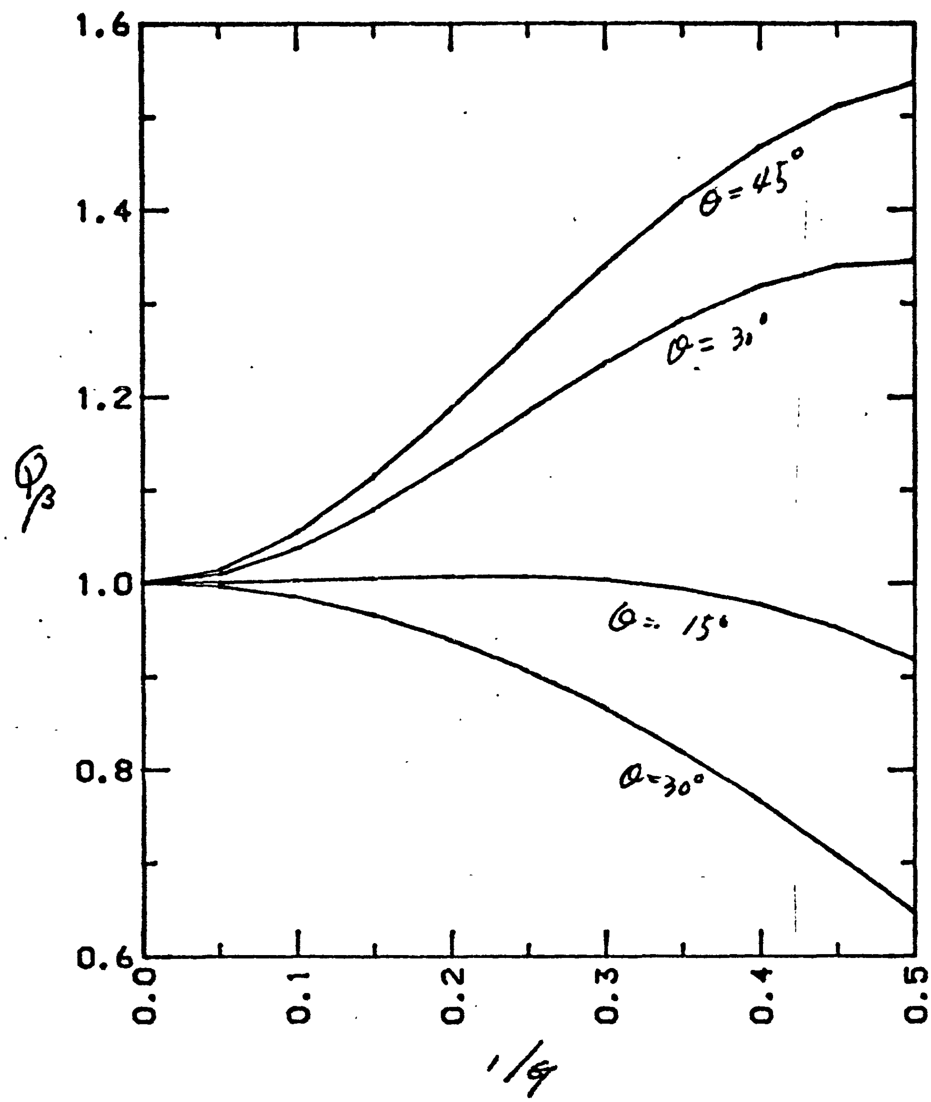


Figure 18. S-wave grid dispersion relation for p=0.7 and r=0.16.

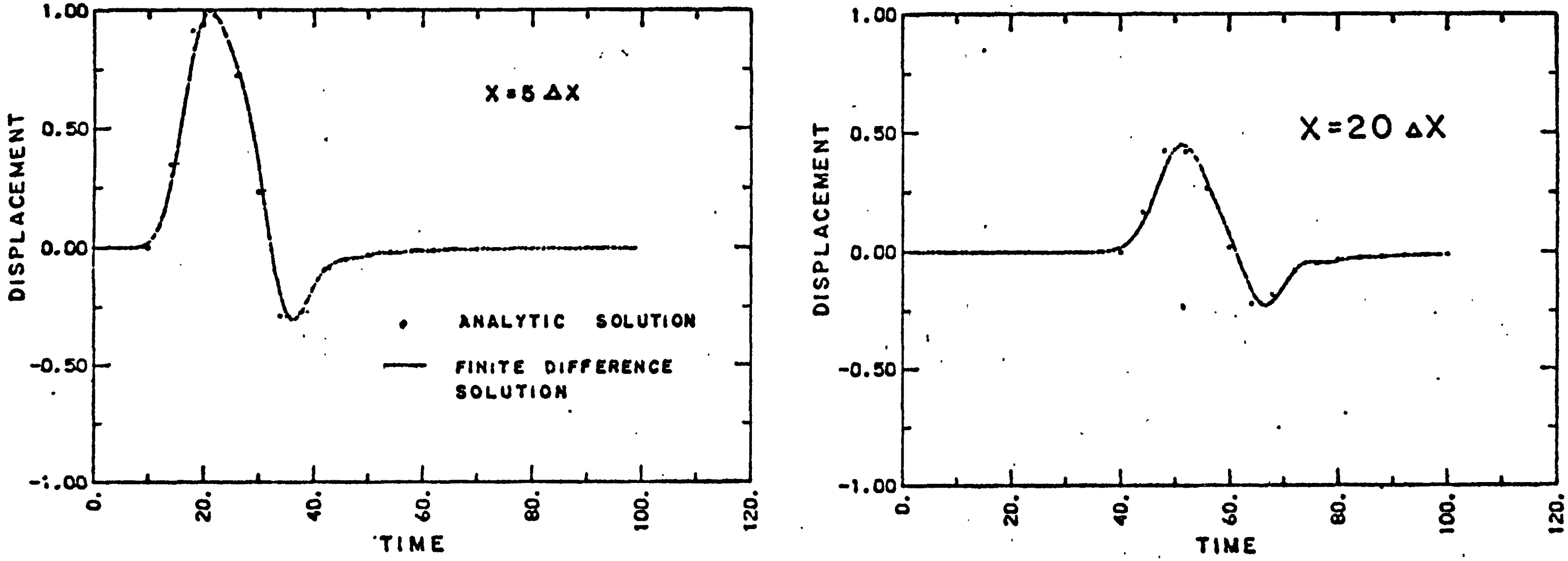


Figure 19. Radial displacement at x=5ΔX and x=20Δx in a whole space.

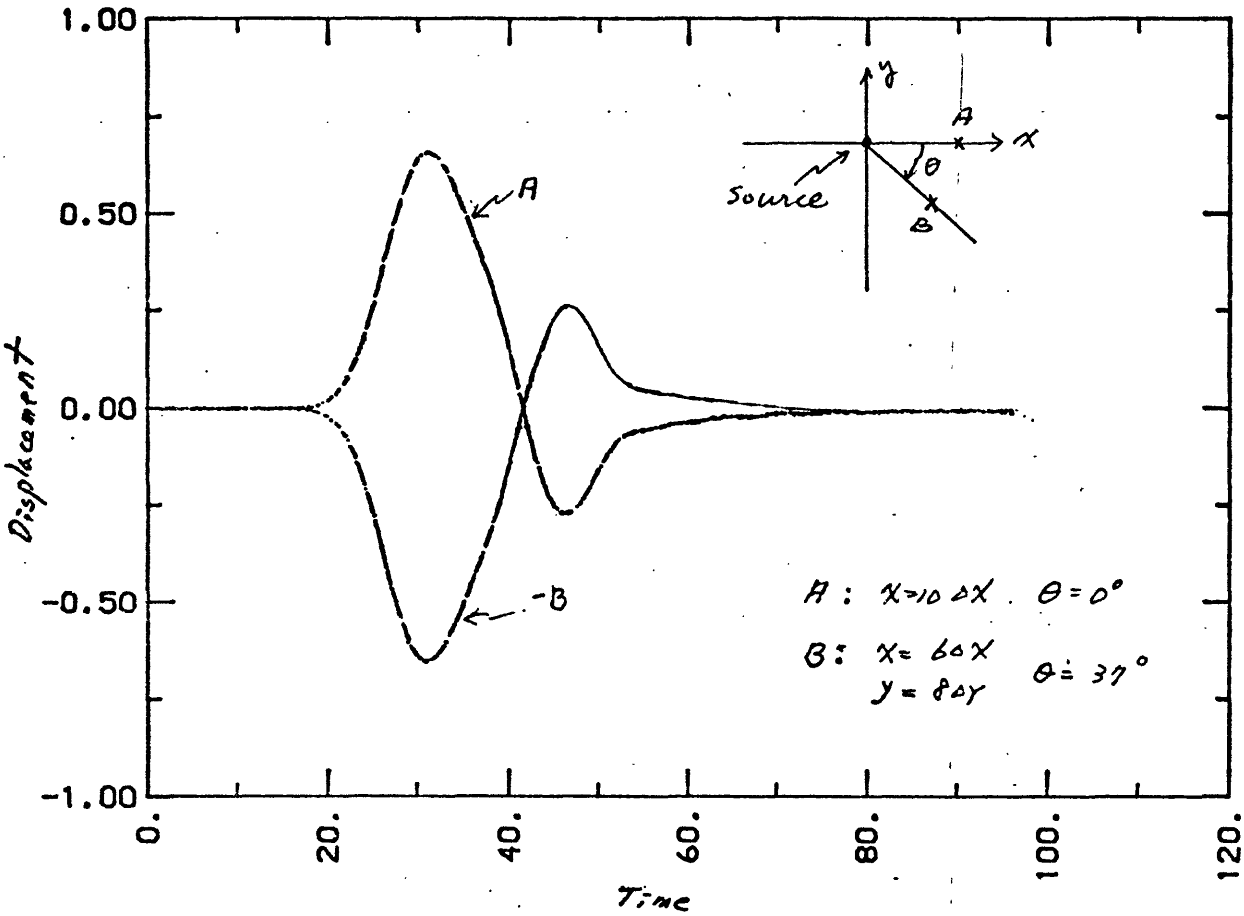


Figure 20. Radial displacement at $\theta = 0^\circ$ and $\theta = 37^\circ$. The negative value at $\theta = 37^\circ$ was plotted for the comparison of the two displacements.

negligible compared with the discrepancy between the analytic solution and the finite difference solution. In general, it is difficult to compare these two types of error. But it confirms that propagation angle is a minor factor for the p-wave grid dispersion error.

Figure 21 shows vertical and horizontal displacements in an elastic half-space at $x=20\Delta x$, y=0. An explosive source is located at x=0, $y=20\Delta x$. We can clearly see two distinct arrivals; a direct compressional wave (p) and Rayleigh wave (R). We can expect a surface generated shear wave in addition to p and R. But, this arrival time is near the Rayleigh wave arrival time, so it is not seen clearly in this example.

The geometry and parameters of the vertical fault model are shown in Figure 22. Figure 23 shows the vertical displacement on the free surface at various geophone positions. Each trace is normalized separately, and the normalization values are shown at the edge of the Figure 23. Using ray theory, the arrival times for different wave types are also shown in Figure 23, where

P: direct p-wave arrival
D: diffracted p-wave arrival
PP: bottom reflected p-wave arrival.

The amplitude decay of the direct p-wave arrivals follows the $1/\sqrt{r}$ law, the theoretical prediction. The quantitative analysis of the later arrivals are difficult, so we did not investigate the amplitude of the later arrivals. However, the arrival times are in good agreement with ray-theoretical arrival times.

Figure 24 shows the vertical displacement at the free surface for a vertical fault model, whose parameters are exactly the same as the parameters of Figure 22 except the upper medium is a Poisson's solid (i.e., $\lambda=\mu$) instead of a fluid.

Like Figure 23, each trace is normalized separately, and its normalization value is shown at the edge of that figure. Compared with Figure 23, in addition to P,PP,D, we can see the large amplitude surface wave after the first p-wave arrival and converted s-wave(reflected at the top of the fault). Due to the surface wave development along the free surface, its amplitude decay is quite different from Figure 23.

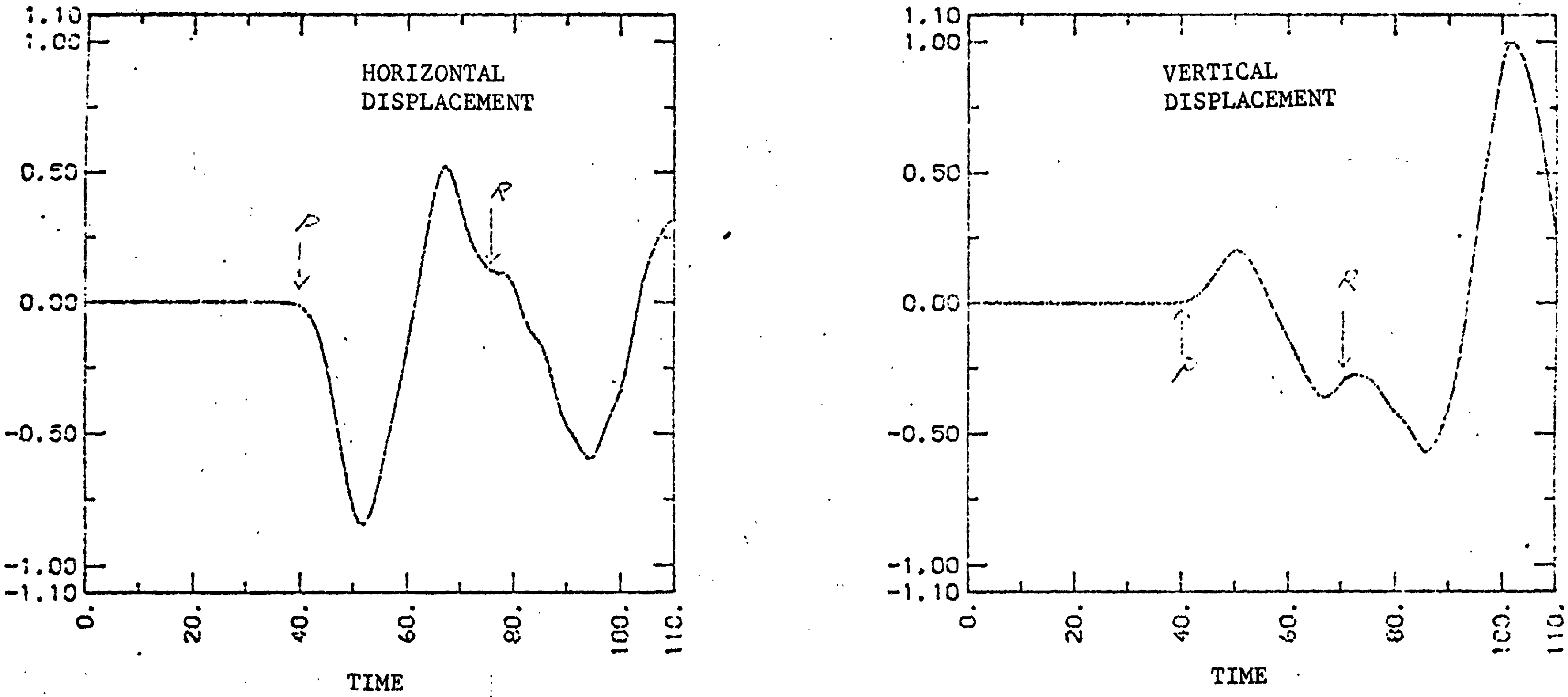


Figure21. Vertical and horizontal displacements on the free surface of an elastic half-space.

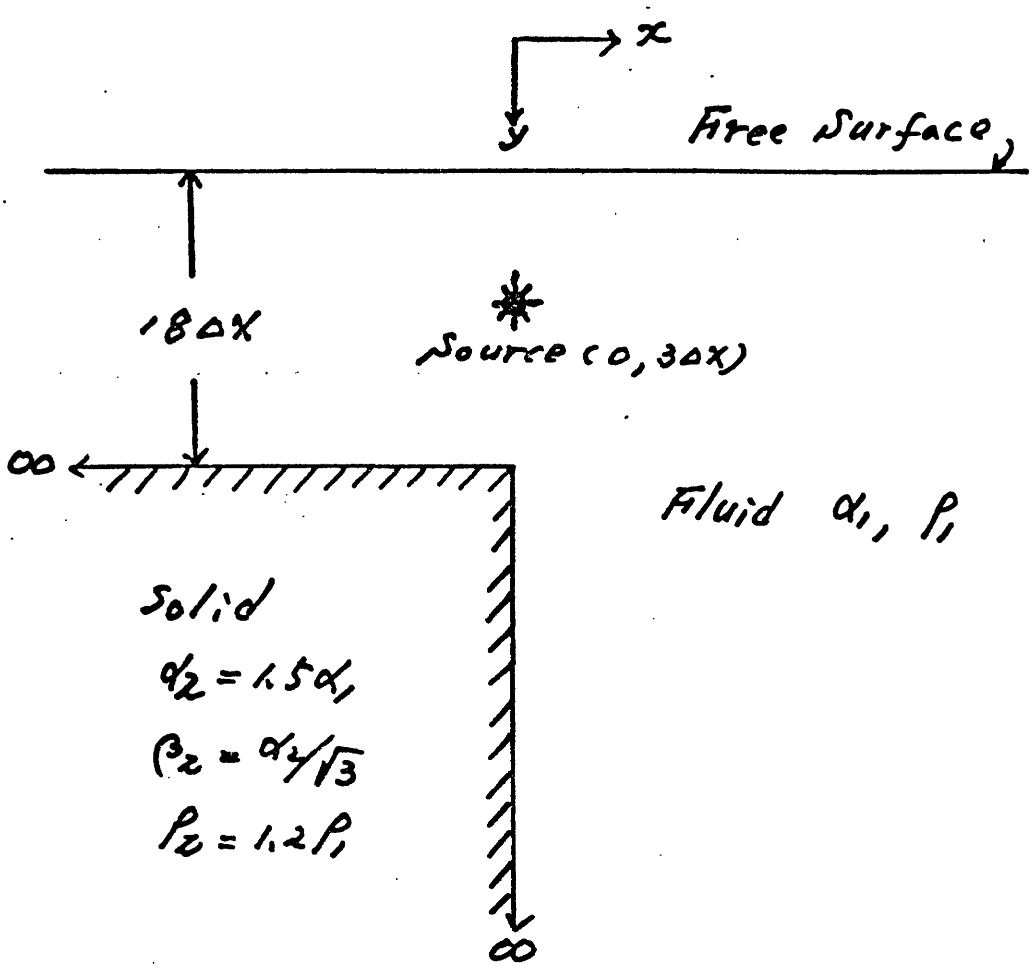


Figure 22. Geometry for a vertical fault model.

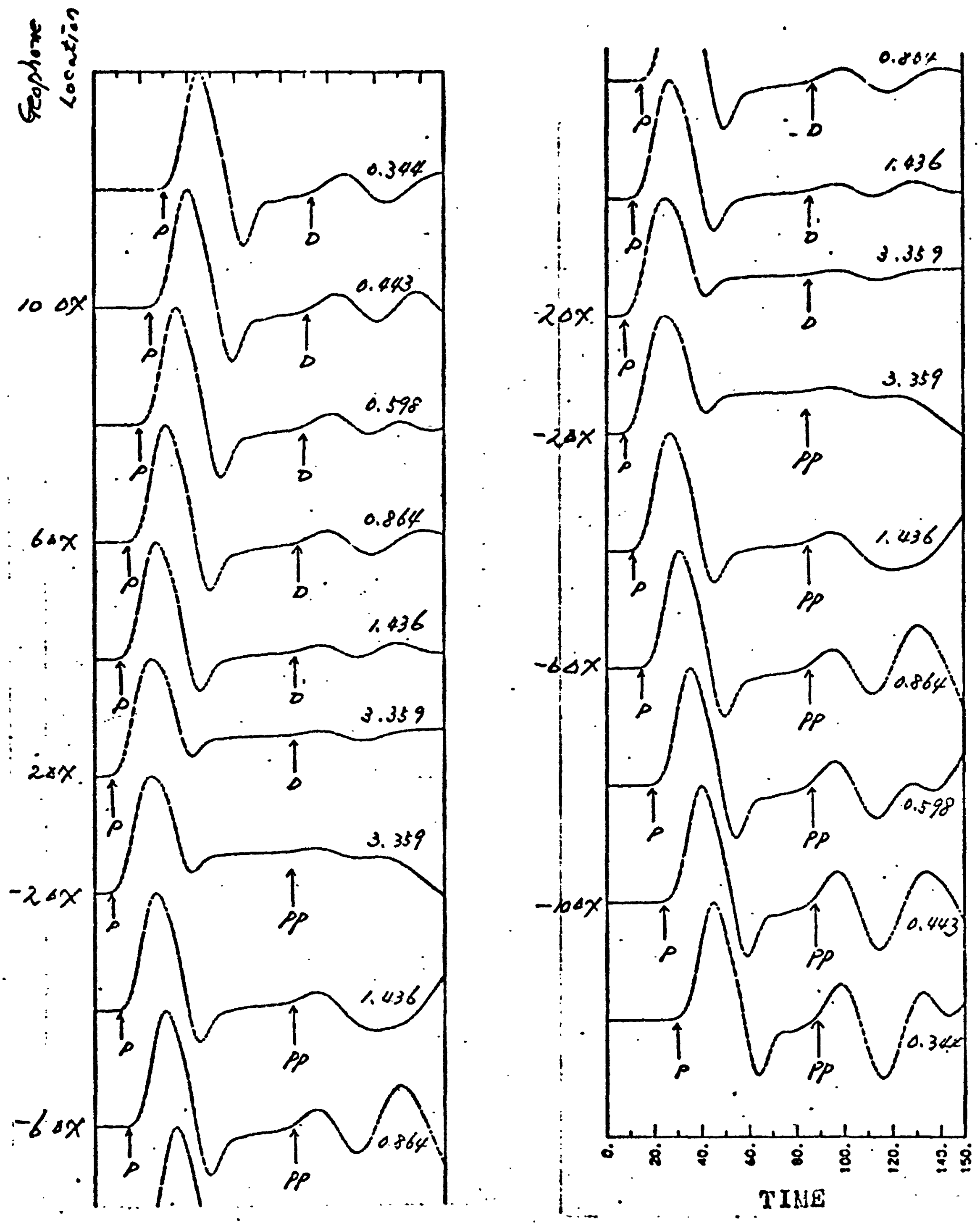


Figure23. Vertical displacement on the free surface for a vertical fault model, embedded in a fluid.

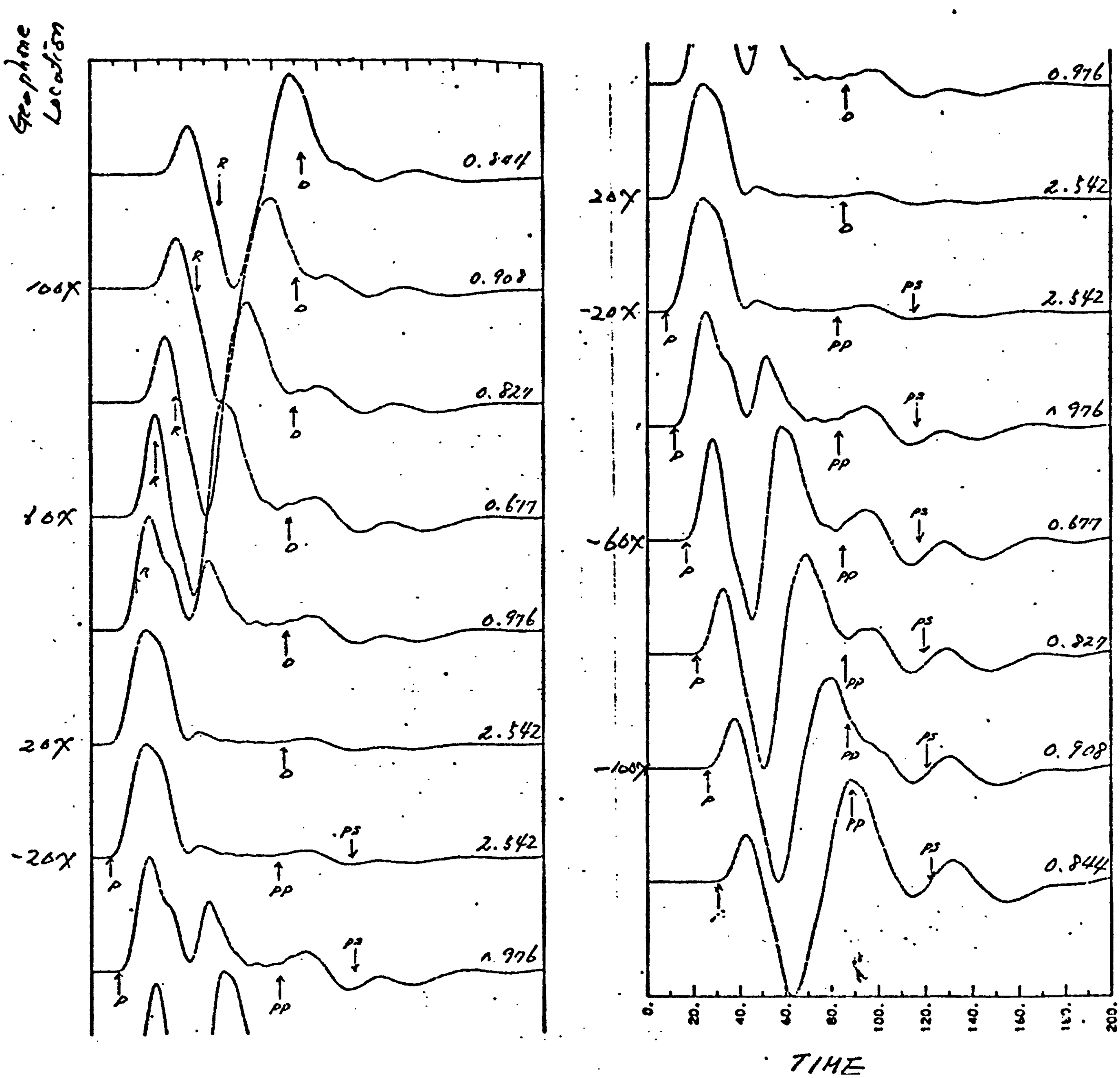


Figure 24. Vertical displacement on the free furface for a vertical fault model embedded in a solid.

Figure 25 shows the geometry for a one-layered half-space model. The vertical and in-line horizontal motions at the free surface are shown in Figures 26a and Figure 26b respectively, with ray-theoretical travel times indicated for selected arrivals. The converted s-wave (PS-reflection) is distinctly visible on both horizontal and vertical motions. The PS-reflection at short ranges is stronger on the horizontal motion than on vertical motion. The propagation of PS-reflection at long ranges is faster than expected, which is caused by the grid dispersion effect. We can see the dispersed wave train at the tail of the direct p-wave arrival, particularly on the horizontal motion. The sampling interval used in this computation is $\Delta t = 2$ ms, and $\Delta x = 25$ ft. Using these values, p=0.48 for the upper medium. We did not study the frequency content of the propagating pulse. So it is difficult to estimate, quantitatively, how much the grid dispersion effect on the dispersive wave train might be. However, the dispersed wave train is caused possibly by the grid dispersion effect. This grid dispersion error, like other types of error, might be increased as the number of iterations increases. On the vertical motion, the phase change of the totally reflected p-wave (beyond the critical angle) is also visible.

Due to the truncation of the model, we can see the artificial reflections around 0.5 sec in Figures 26a and 26b. This is the one of the major problems in using finite difference schemes for a realistic numerical modeling. Unless a radiation boundary condition at the edge of the model is implemented, we must always expect these unwanted artificial reflections.

Figure 27 shows the geometry of a localized inhomogeneity embedded in an elastic half-space. The p-wave velocity of the half-space is 7000 ft/sec and p-wave velocity of the inhomogeneity is 4000 ft/sec. Vertical and horizontal motions on the free surface are shown in Figure 28a and Figure 28b respectively with the identifiable reflections around the epicenter. The later part of the seismogram is severely contaminated by the arificial reflections from the edge of the model. The strong horizontal motion at short ranges, around 0.32 second, is quite comparable with the vertical motion. This strong horizontal motion is a ghost diffracted shear wave. The extension of the p-wave reflection from the top layer beyond the ray-theoretical reflection limit is p-wave diffraction.

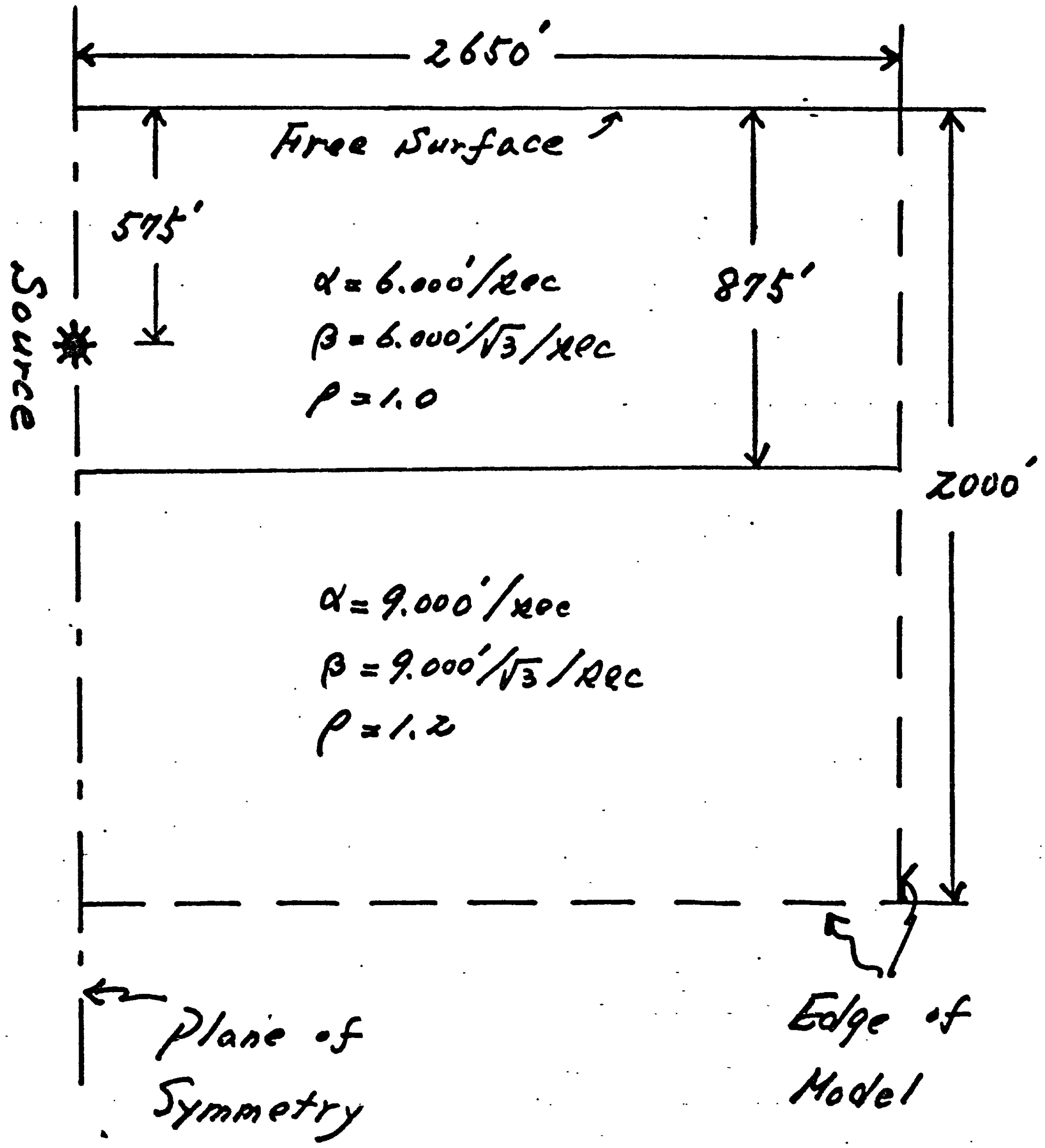
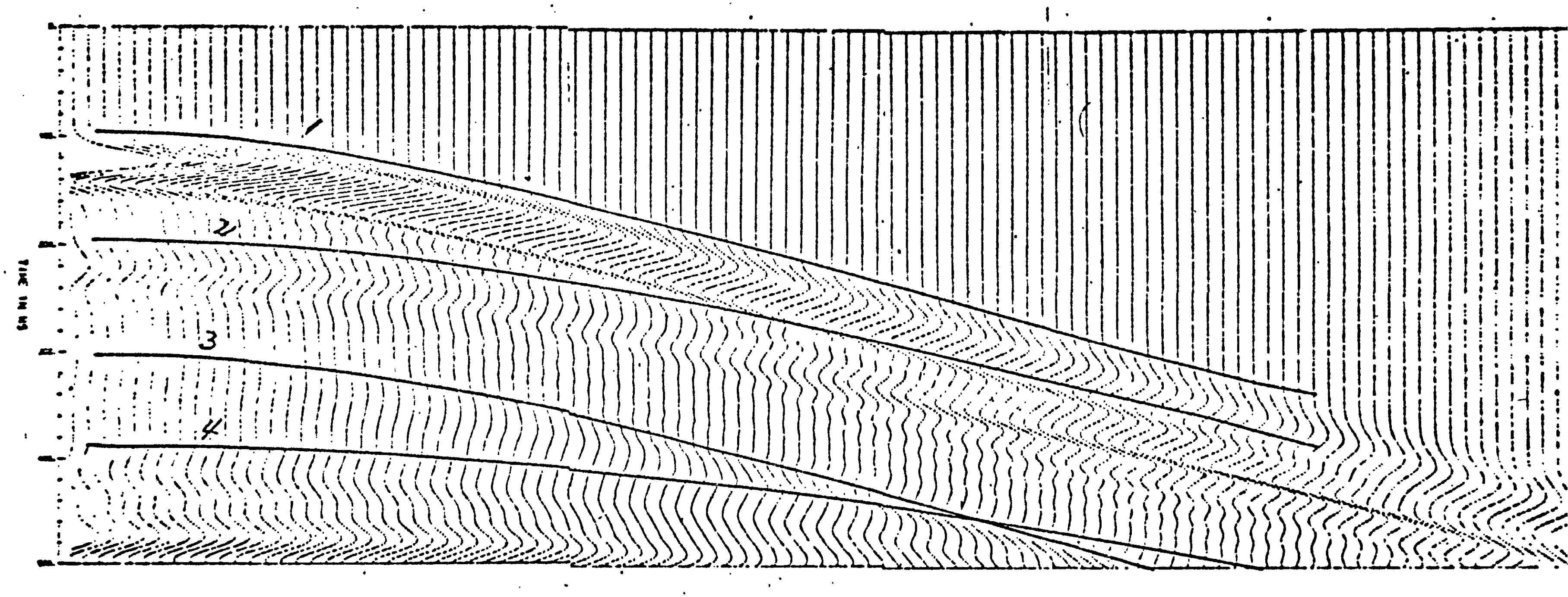


Figure 25. Geometry for one-layered half-space model.



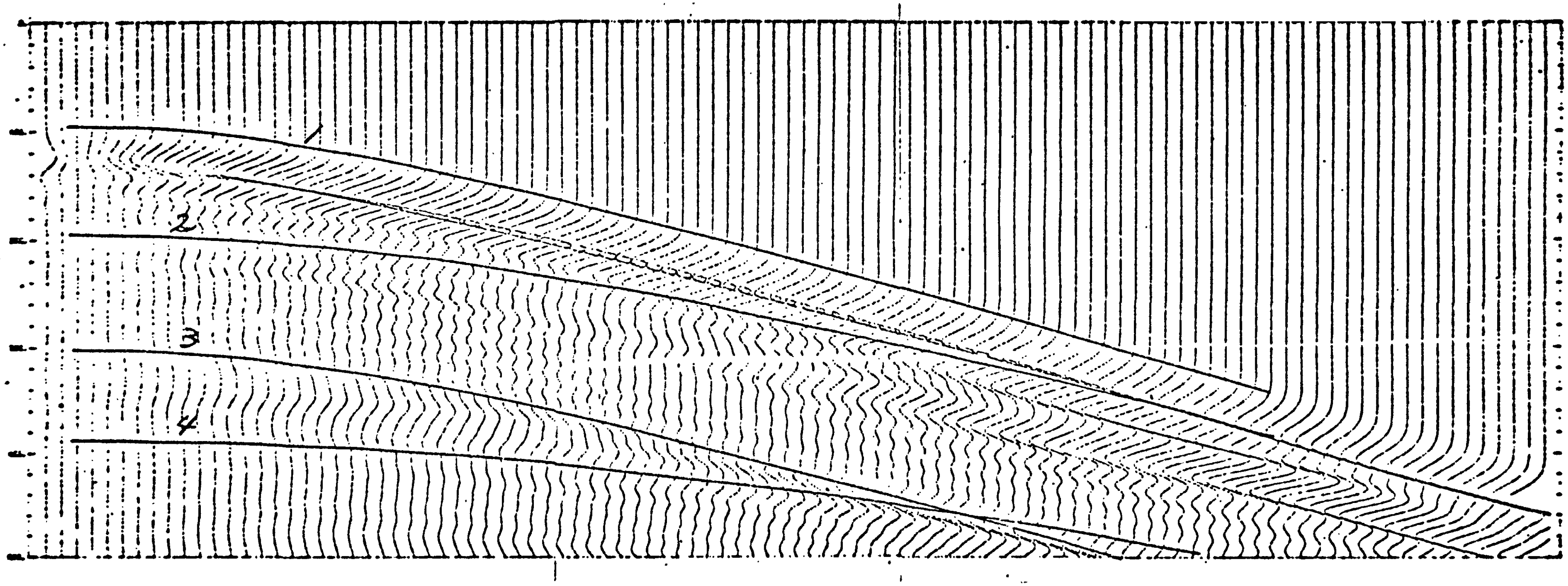
#1 Direct p-wave.

#2 Reflected p-wave.

#3 Ghost p-wave reflection.

#4 Ghost s-wave reflection.

Figure 26a. Vertical displacement on the free surface for one-layered half-space model.



#1 Direct p-wave.

#2 Reflected p-wave.

#3 Ghost p-wave reflection.

#4 Ghost s-wave reflection.

Figure 26b. In-line horizontal motion on the free surface for one-layered half-space model.

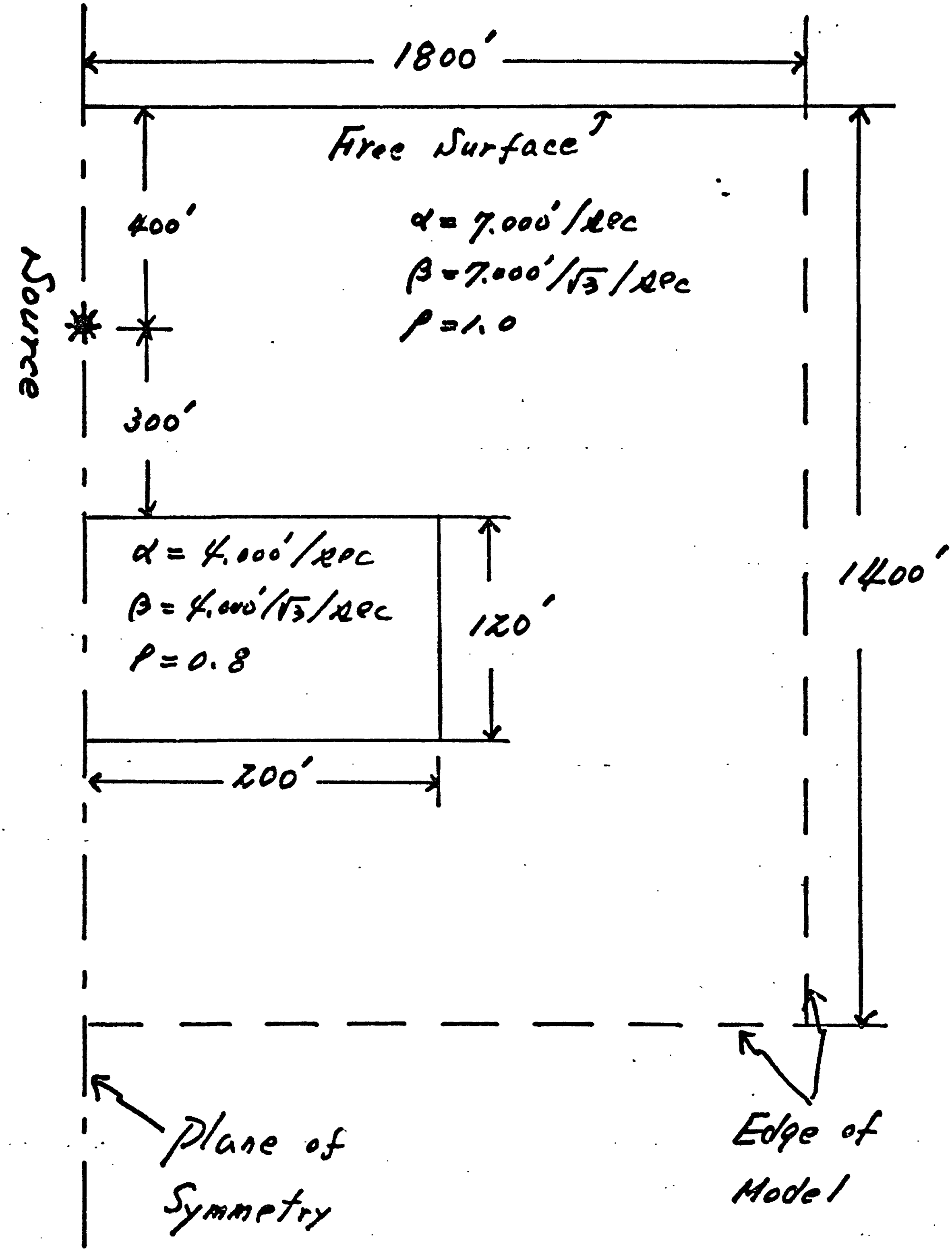
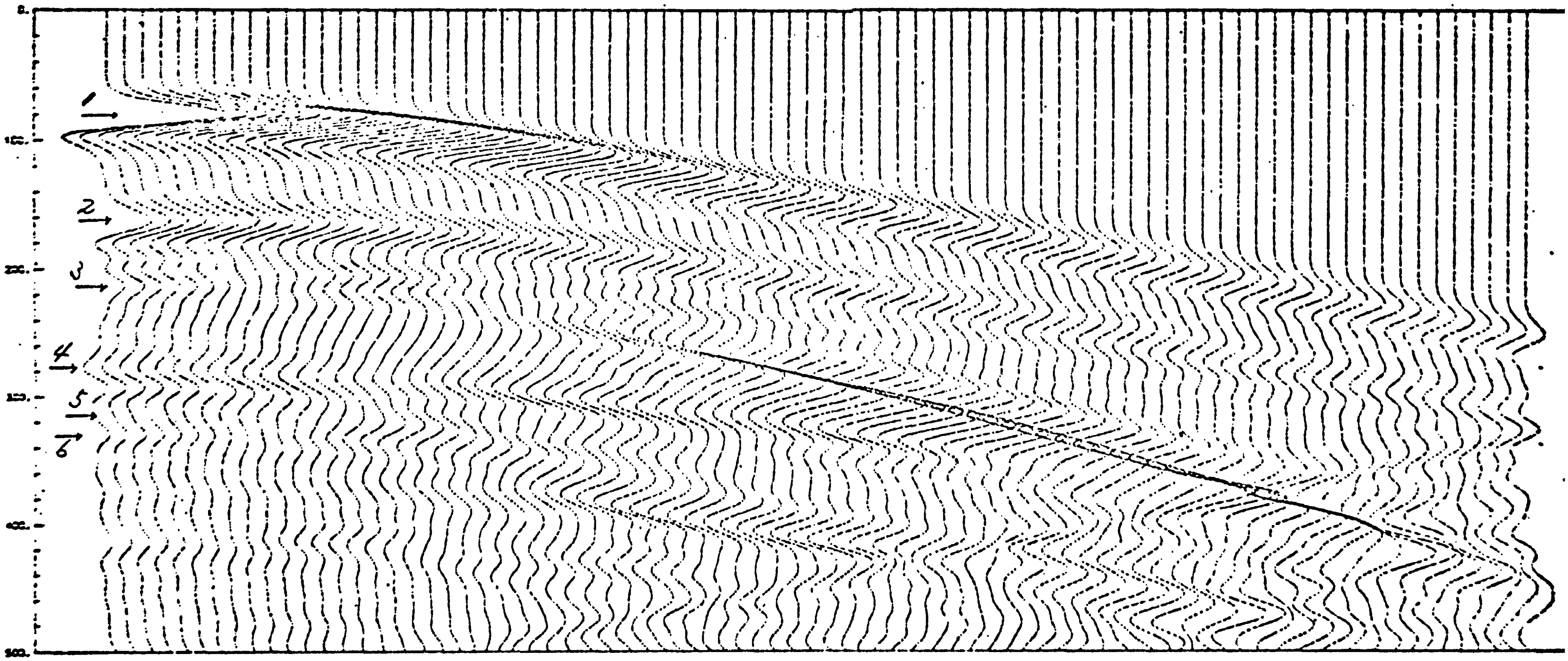


Figure 27. Geometry for localized inhomogeneity model.



#1 Direct p-wave.

#2 P-wave reflection (top layer).

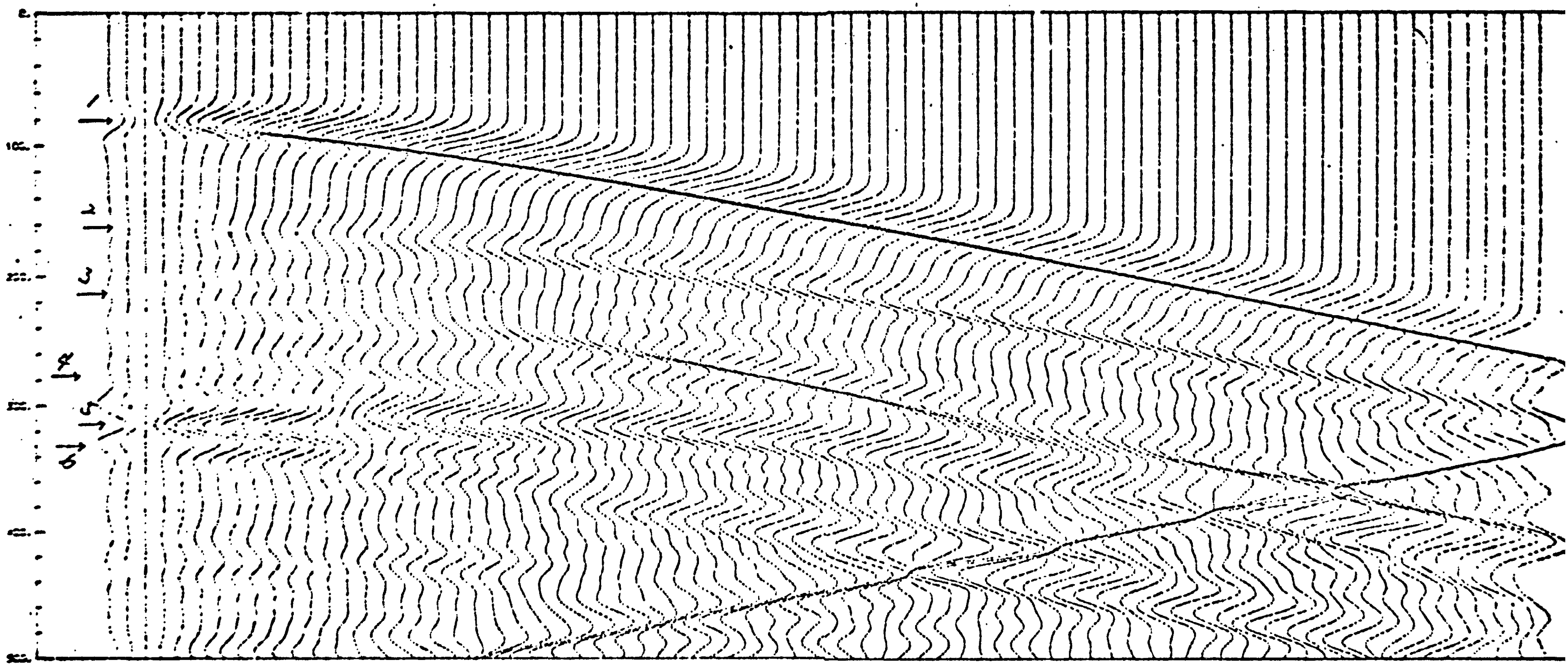
#3 P-wave reflection (bottom layer).

#4 Ghost p-wave reflection(top layer).

#5 Ghost PS-diffraction.

#6 Ghost p-wave reflection(bottom layer).

Figure 28a. Vertical displacement on the free surface for localized inhomogeneous model.



#1 Direct p-wave.

#2 P-wave reflection (top layer).

#3 P-wave reflection (bottom layer).

#4 Ghost p-wave reflection(top layer).

#5 Ghost PS-diffraction.

#6 Ghost p-wave reflection (bottom layer)

Figure 28b. In-line horizontal displacement on the free surface for localized inhomogeneous model.

THIN BED REFLECTION

The reflection and transmission amplitudes of an incident longitudinal wave due to a thin bed is of considerable interest in stratigraphic oil trap studies. The purpose of this section is to exhibit the calculated energy partition of the incident seismic signal due to a thin bed by varying the angle of incidence and frequency.

Theory

The basic theory for this problem can be found elsewhere (Bath, 1968, Haskell, 1953). We shall merely summarize the final results here and make some important definitions.

Figure 29 illustrates the problem.

Let

α_i: longitudinal wave velocity of the i-th medium

β_i: shear wave velocity of the i-th medium

ρ_i: density of the i-th medium

ϕ_i': dilatational displacement potential of the i-th medium for the down-going wave

ϕ_i'': dilatational displacement potential of the i-th medium for the up-going wave

ψ_i': rotational (i.e. shear wave) displacement potential of the i-th medium for the down-going wave

ψ_i'': rotational displacement potential of the i-th medium for the up-going wave.

From the Haskell's matrix method, we can compute all ϕ_i', ϕ_i'', ψ_i', and ψ_i'' in terms of incident potential ϕ_1'.

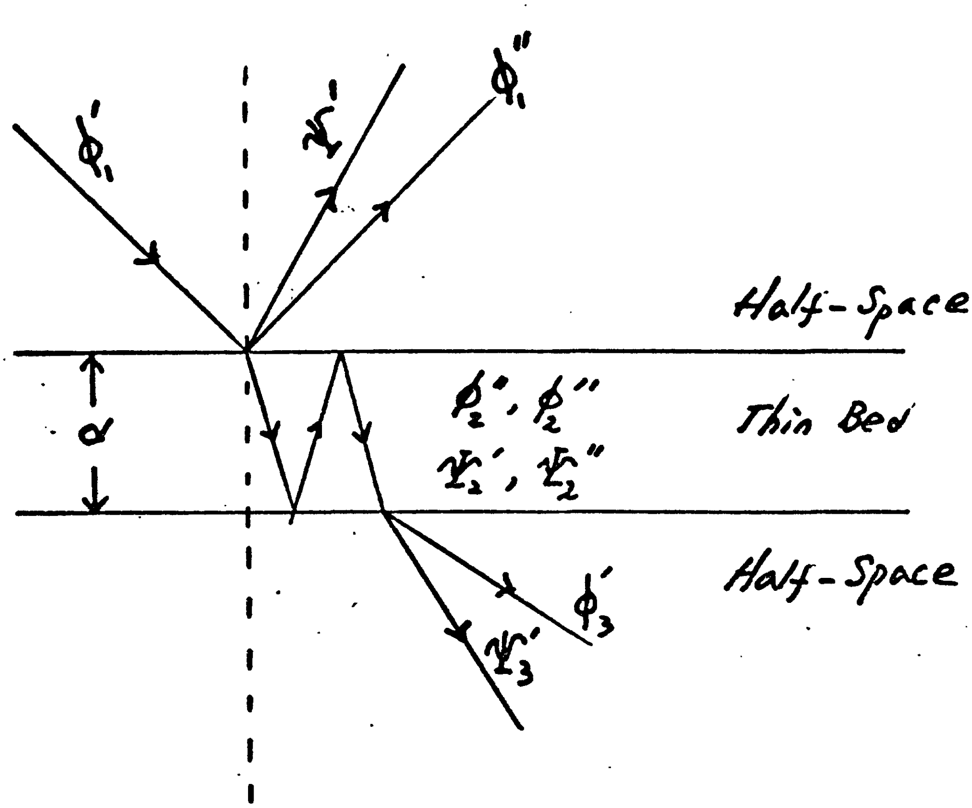


Figure 29. Thin bed reflection and transmission.

Define

$$R_p = \frac{\phi_1''}{\phi_1'}$$

$$R_s = \frac{\psi_1''}{\phi_1'}$$

$$T_p = \frac{\phi_3'}{\phi_1'}, \quad T_s = \frac{\psi_3'}{\phi_1'}$$

Then the energy partition between incident longitudinal wave and transmitted or reflected wave can be written by the following way.

$$E_r^p = R_p R_p^*$$

$$E_r^s = 2\left(\frac{\beta_1}{\alpha_1}\right)^2\left(\frac{\gamma_{\beta 1}}{\gamma_{\alpha 1}}\right) R_s R_s^*$$

$$E_t^p = \left(\frac{\rho_3}{\rho_1}\right)\left(\frac{\alpha_3}{\alpha_1}\right)^2\left(\frac{\gamma_{\alpha 3}}{\gamma_{\alpha 1}}\right) T_p T_p^*$$

$$E_t^s = 2\left(\frac{\rho_3}{\rho_1}\right)\left(\frac{\beta_3}{\alpha_1}\right)^2\left(\frac{\gamma_{\beta 3}}{\gamma_{\alpha 1}}\right) T_s T_s^*$$

where A^* means the complex conjugate of A, E means the energy ratio, whose superscription identifies the wave type (p or s-wave) and subscription differentiates between the reflected (r) or tranmitted (t), c is horizontal phase velocity, and

$$\gamma_{\alpha m} = \left[\left(\frac{c}{\alpha_m}\right)^2 - 1\right]^{\frac{1}{2}} \quad \text{for} \quad c > \alpha_m$$

$$= -i\left[1 - \left(\frac{c}{\alpha_m}\right)^2\right]^{\frac{1}{2}} \quad \text{for} \quad c < \alpha_m$$

$$\gamma_{\beta m} = \left[\left(\frac{c}{\beta_m}\right)^2 - 1\right]^{\frac{1}{2}} \quad \text{for} \quad c > \beta_m$$

$$= -i\left[1 - \left(\frac{c}{\beta_m}\right)^2\right]^{\frac{1}{2}} \quad \text{for} \quad c < \beta_m .$$

The above energy ratio is not an instantaneous energy ratio, it is the ratio averaged over one period.

Examples and Discussions

Figures 30 through 33 show the square root of the energy ratio as a function of varying bed thickness, using the following parameters (Table 1) for the computation.

Parameters \ Medium	1	2	3
p-wave Velocity m/sec	3090	3750	3000
s-wave Velocity m/sec	1784	2004	1708
Density Kg/m³	2420	2330	2400

Table 1. Parameters of a thin bed-A.

In these figures 1 represents bed thickness with $1=4d/\lambda$ and λ is the wavelength for normal incidence in the thin bed. When the angle of incidence is less than the p-wave critical angle (55°) between medium 1 and medium 2, we can clearly see the interference effect. For the normal incidence case, when 1 is an even number, the reflection coefficient is equal to the usual coefficient of reflection from the boundary between medium 1 and medium 3, just as if they were in direct contact with one another. Thus we can say that a layer, an integral number of half wave lengths thick, has no effect on the incident wave.

Figure 34 through 37 show the ratio of the square root of the reflected p-wave and s-wave energy to the incident p-wave energy as a function of frequency. The parameters used for this computation are shown in Table 2.

Parameters \ Medium	1	2	3
p-wave velocity m/sec	3939	3000	3787
s-wave Velocity m/sec	2211	1604	2126
Density Kg/m³	2650	2250	2550

Table 2. Parameters of a thin bed-B.

In these figures "d" represents bed thickness and θ represents angle of incidence of the incident wave. We can see how the interference pattern varies with incident pulse frequency.

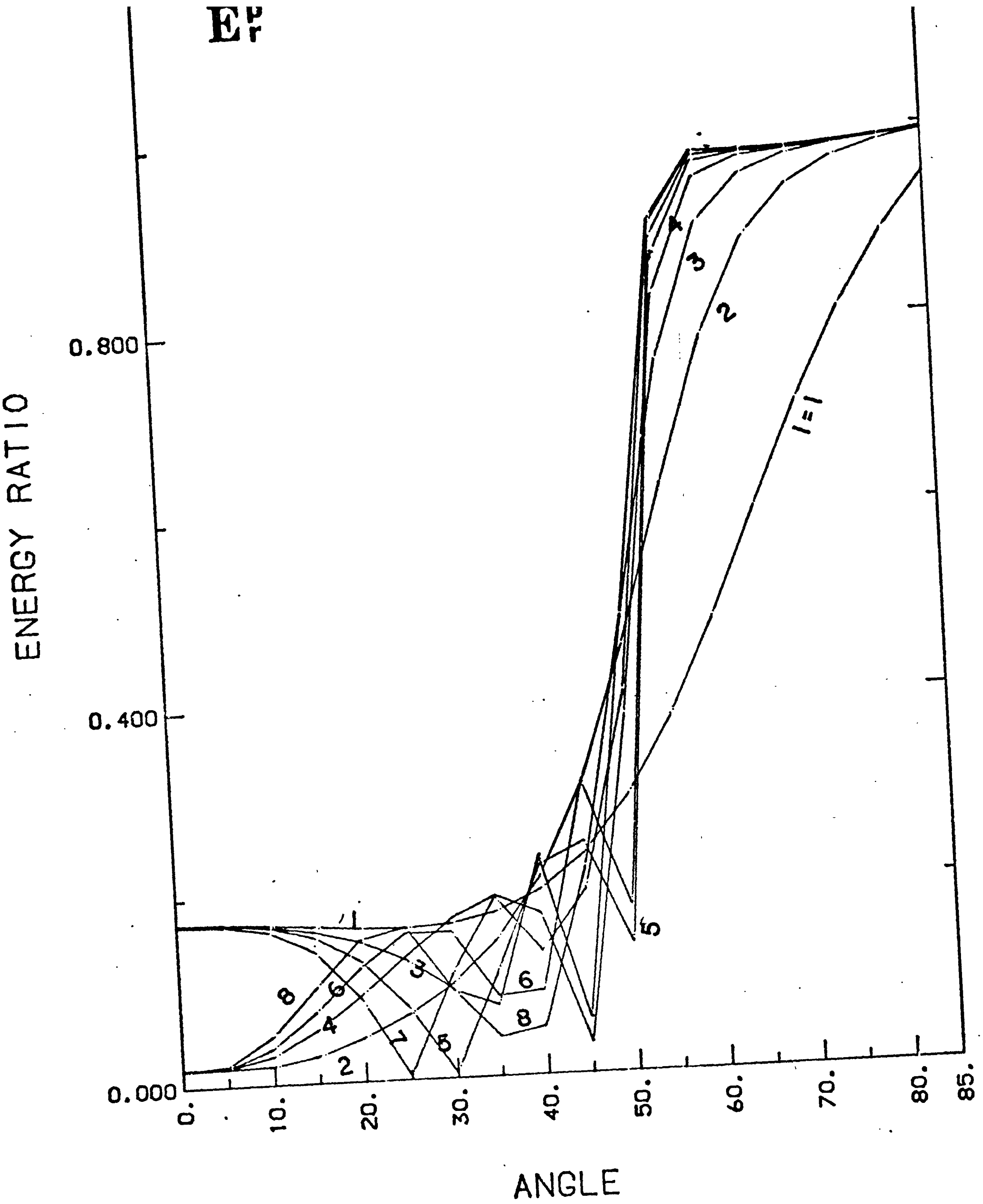


Figure 30. Energy partition for the reflected p-wave with varying bed thickness as a function of incidence angle.

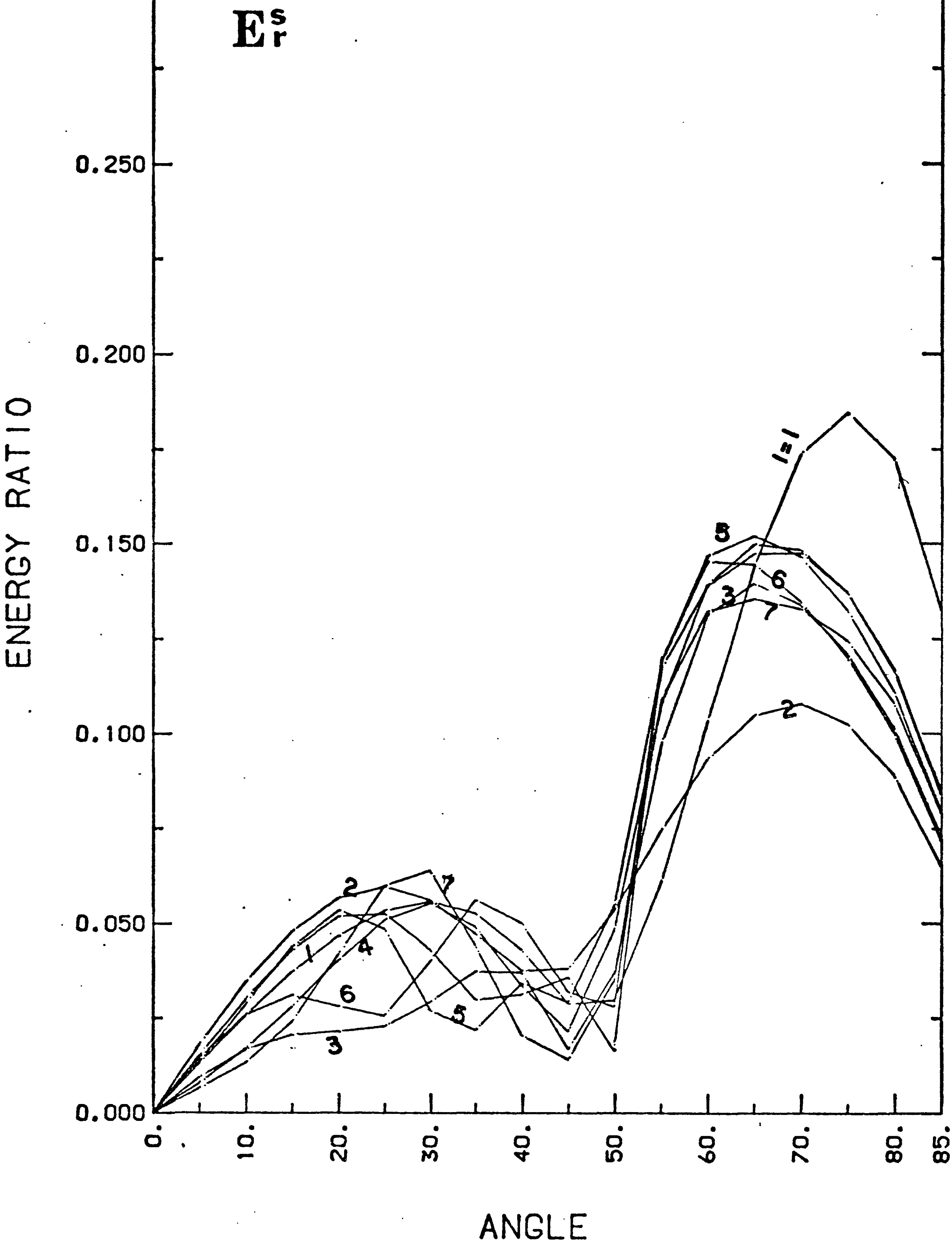


Figure 31. Energy partition for the reflected s-wave with varying bed thickness as a function of incidence angle.

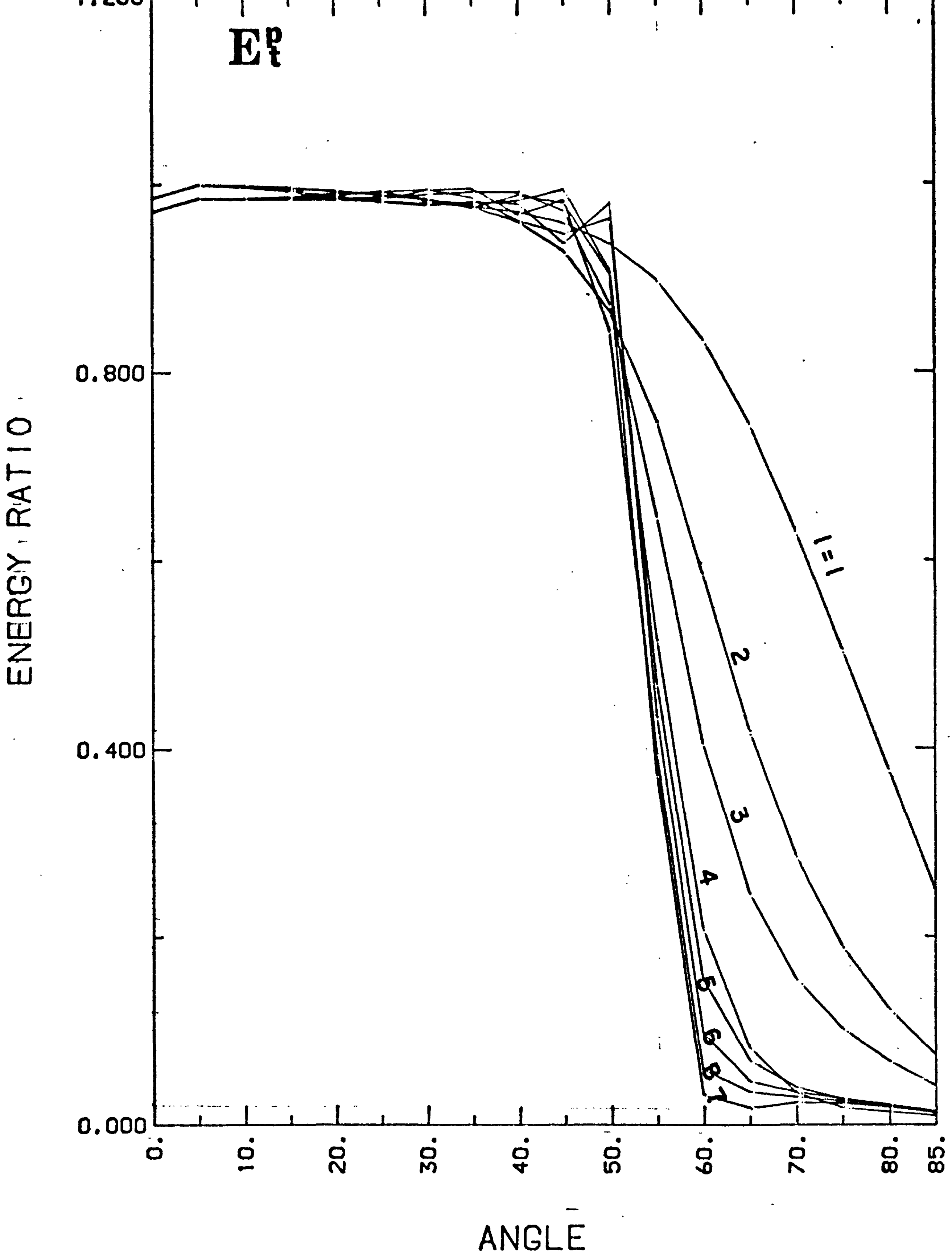


Figure 32. Energy partition for the transmitted p-wave with varying bed thickness as a function of incidence angle.

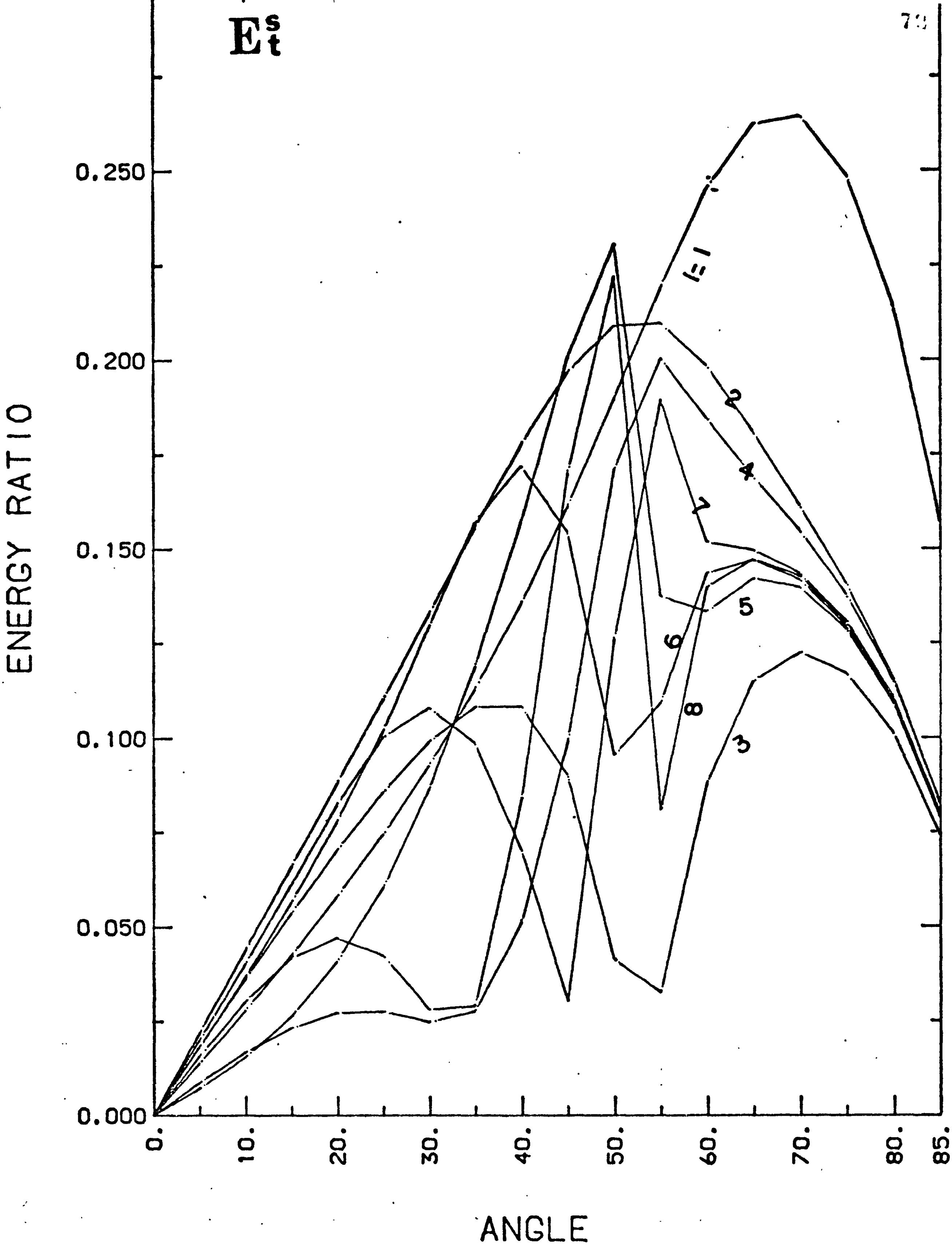


Figure 33. Energy partition for the transmitted s-wave with varying bed thickness as a function of incidence angle.

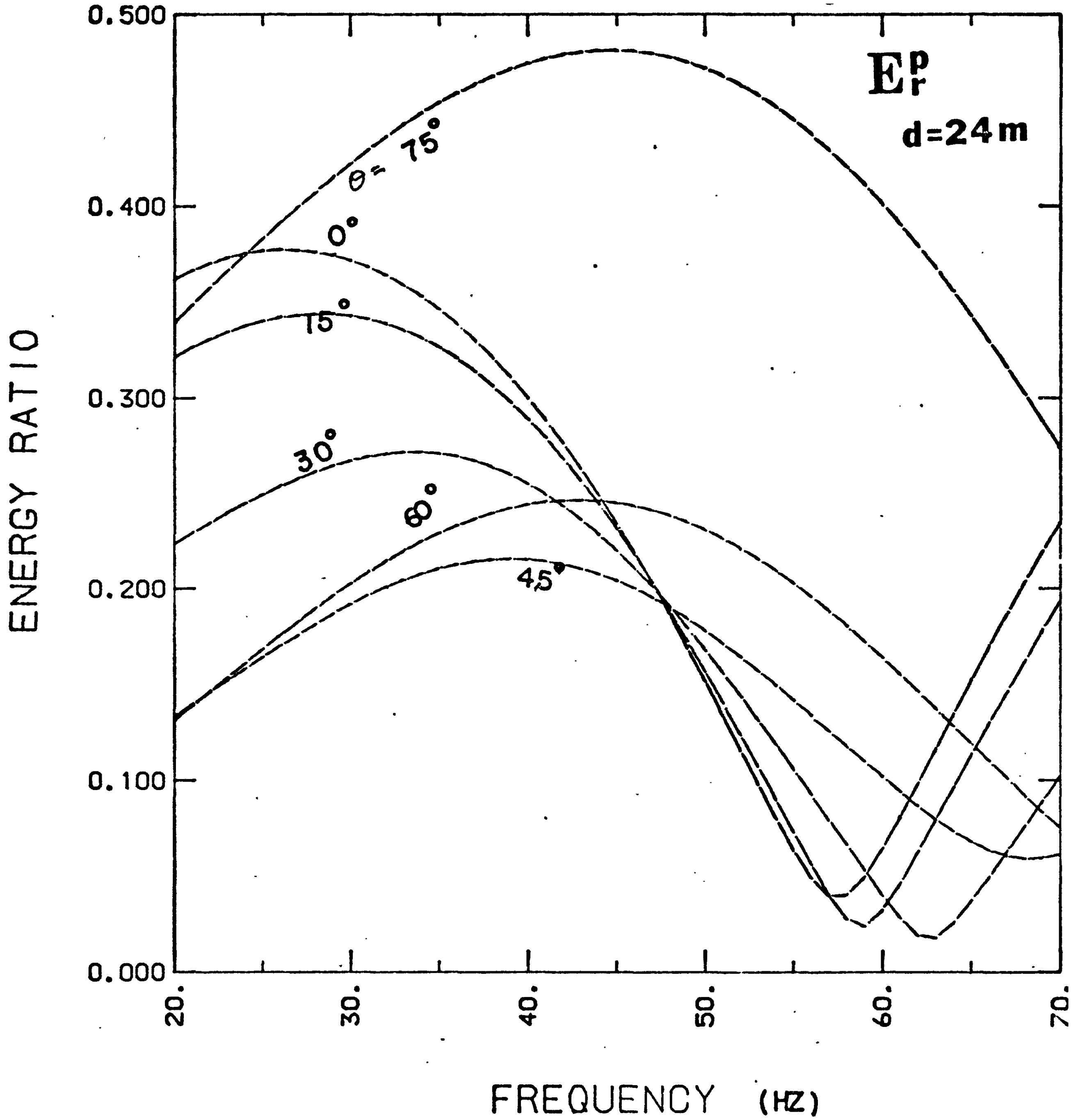


Figure 34. Energy partition for the reflected p-wave with varying angle of incidence as a function of frequency.

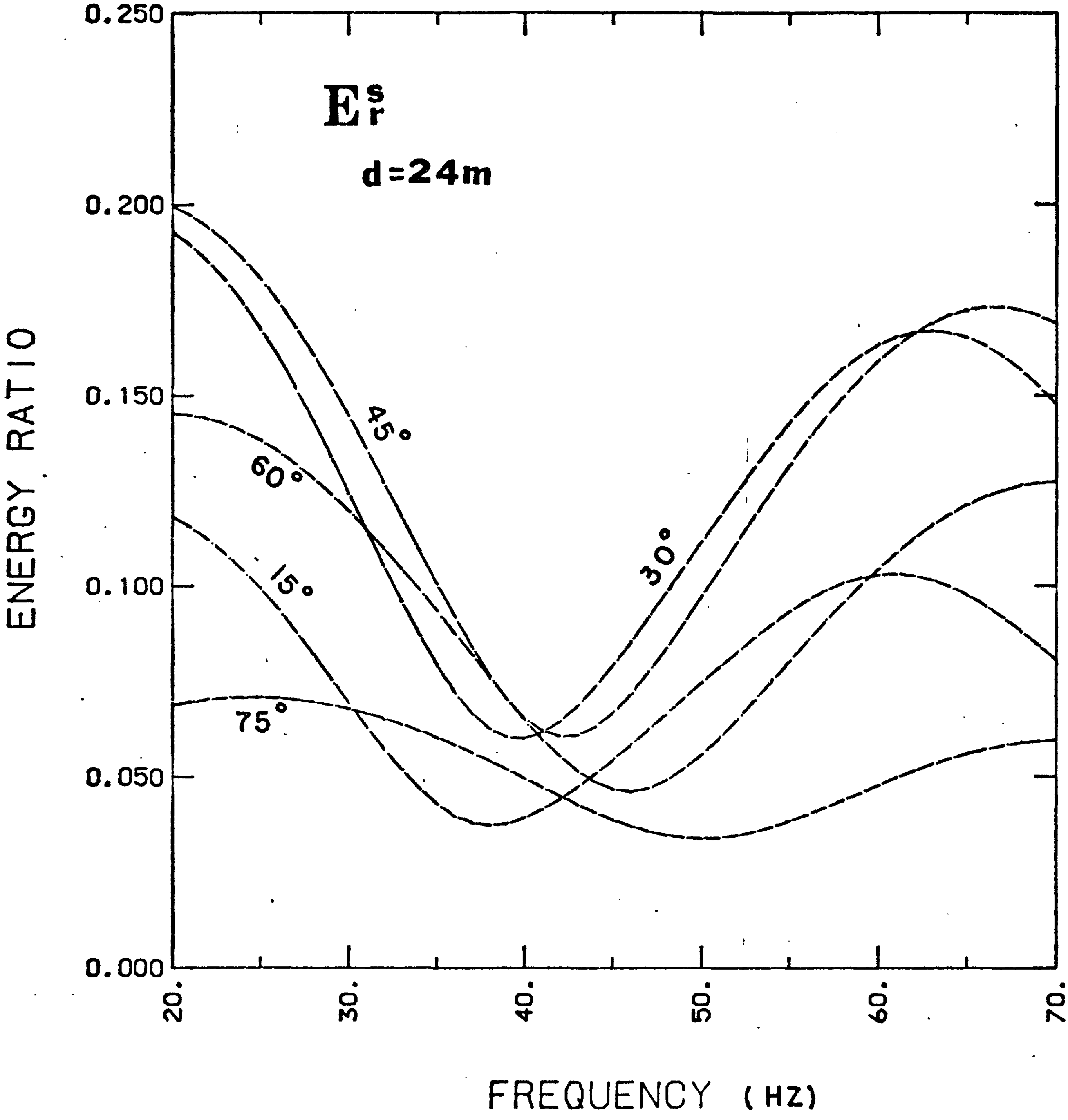


Figure 35. Energy partition for the reflected s-wave with varying angle of incidence as a function of frequency.

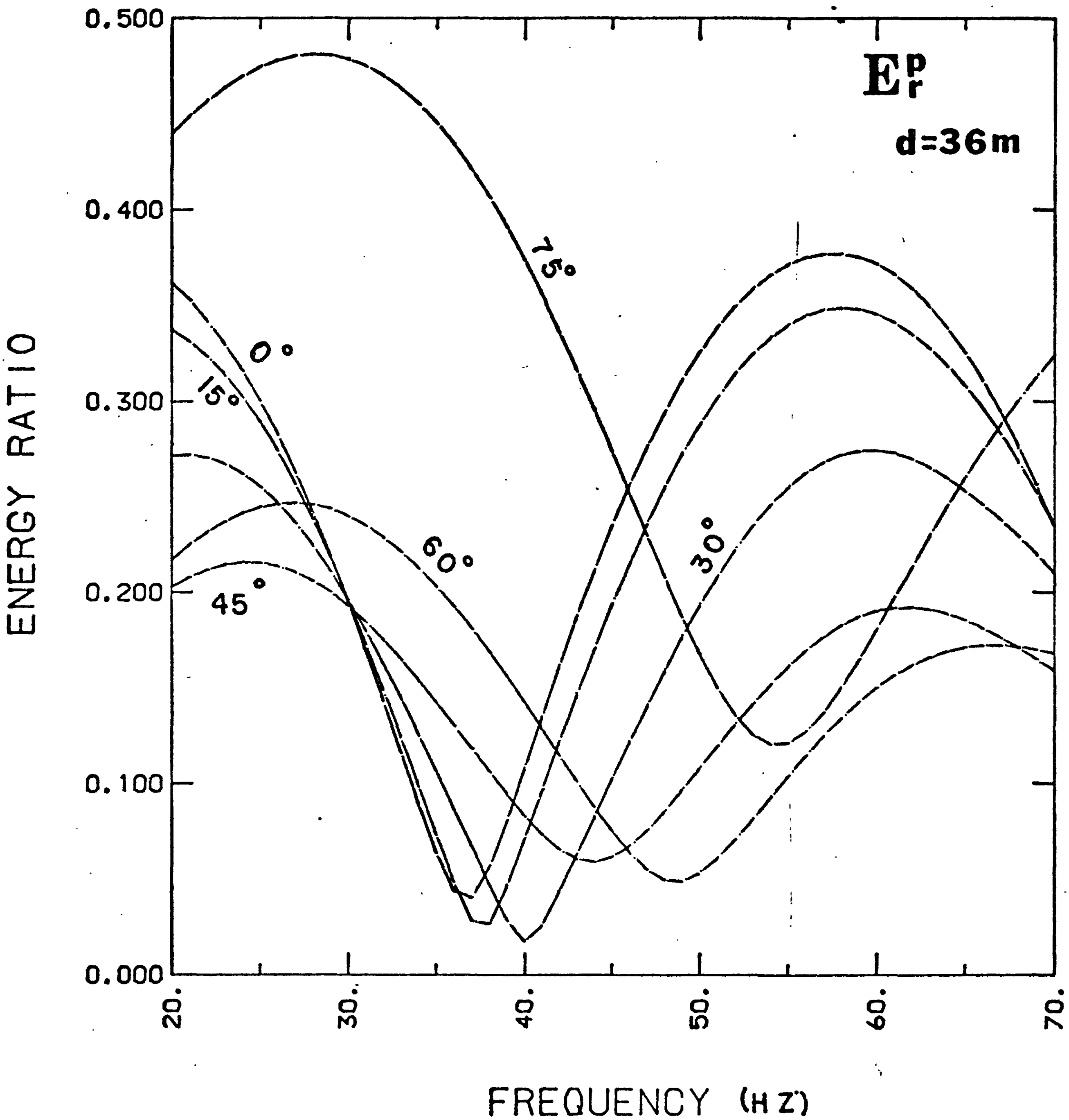


Figure 36. Energy partition for the reflected p-wave with varying angle of incidence as a function of frequency.

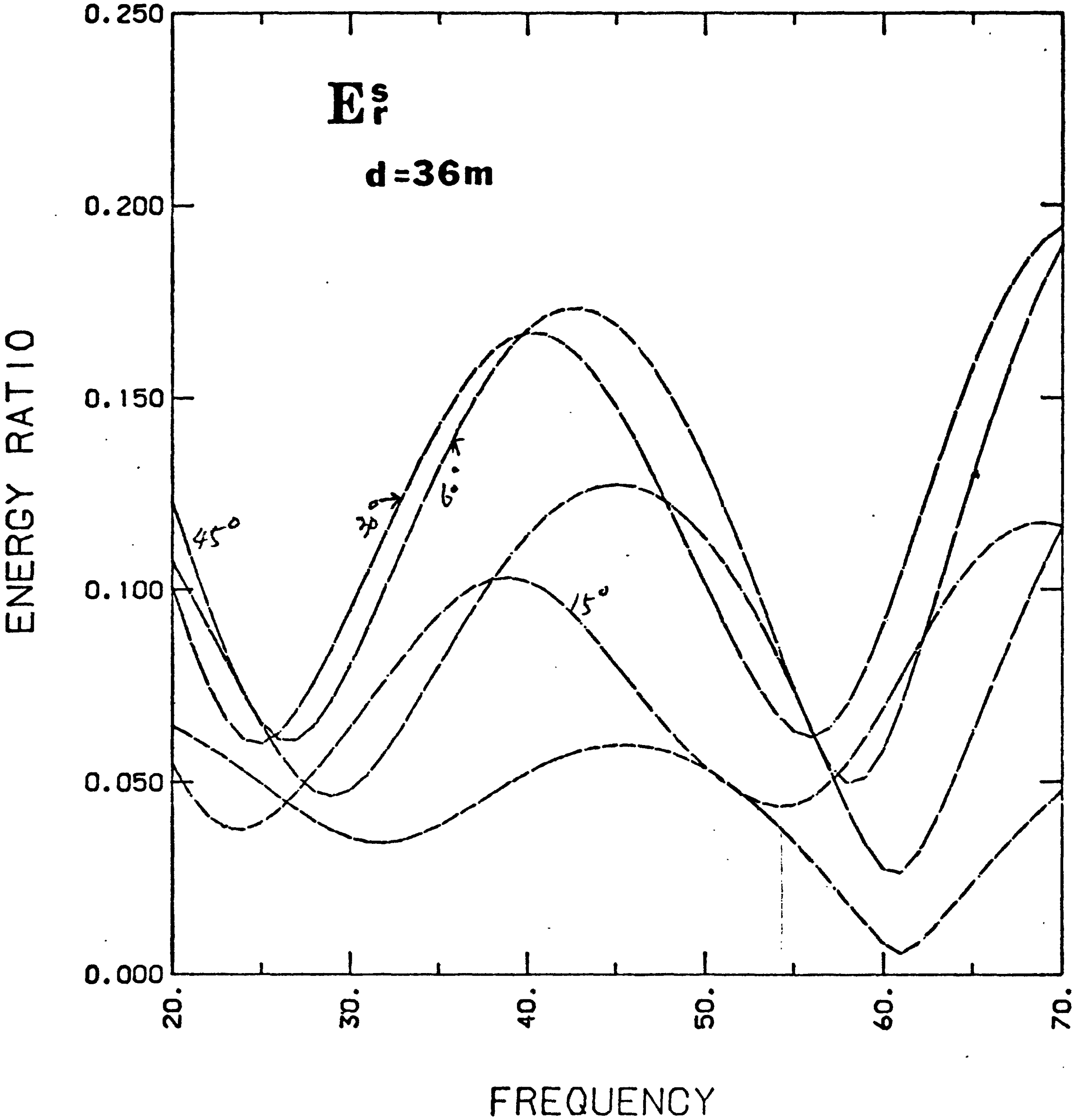


Figure 37. Energy partition of the reflected s-wave with varying angle of incidence as a function of frequency.

If we can separate a direct wave from a reflected wave, it is possible to measure the amplitude ratio between two waves in the frequency domain and compare with the calculated reflection coefficients, which may provide a thin bed characteristic. But much of our analysis of field measurement indicates that it is difficult to separate a reflected wave from a direct wave. In this case, we cannot directly compare the reflection coefficients with frequency. So it seems to be more useful to calculate a waveform in the time domain to study the effect of a thin bed or a packet of thin beds.

CONCLUSIONS

An analytic solution of an one-dimensional wave equation for a horizontally layered elastic medium provided a good computer algorithm to generate synthetic seismogram for arbitrary source and detector locations. By a slight modification of the appended computer program, vertical seismic profiles can be easily generated.

The phase and amplitude distortion of a reflected seismic wave, caused by complex reflection and transmission coefficients, may be ignored in generating synthetic seismograms.

Finite difference approaches for the solution of one-dimensional, inhomogeneous, attenuating media may be a good and powerful technique for one-dimensional modeling. For simple models, the solutions by finite difference schemes are in good agreement with analytic solutions. We successfully implemented a radiation boundary condition in the one-dimensional difference formulation. The main problem in applying finite difference approaches to realistic modeling will be the computer execution time, since we may use very small sampling intervals to reduce the propagation error in each inhomogeneous region.

In making two-dimensional seismic models, finite difference approaches may be a useful tool. Even if there are some erroneous arrival times, particularly converted shear waves, and dispersive tails in the wave propagation, we can identify all types of elastic waves in synthetic seismograms. Also we can see the phase change beyond the critical angle reflection and diffractions. For complex subsurface geologic models, particularly in studying amplitude and shape of the reflected and transmitted seismic signals and in studying shear motions, finite difference schemes of the inhomogeneous two-dimensional elastic wave equation seem to be the most pertinent method.

Wide angle reflection coefficients and interference patterns of a thin bed may be used to find characteristic parameters (thickness and elastic constants) of a thin bed. But it seems to be more promising in analyzing the effect of a thin bed or a packet of thin beds to calculate seismic wave form in the time domain rather than calculating reflection coefficients or interference patterns.

APPENDICES

APPENDIX A

Computer Program and User's Manual for Synthethic Seismogram for Horizontally Layered Perfectly Elastic Half-space

This computer program calculates a synthethic seismogram for a horizontally layered elastic half-space using an analytic solution. The input cards consist of directive cards whose format and forms are specified in this program and other data cards (velocity, density, pressure wavelet, check shot data). In this program there are 8 directive cards to read parameters and to execute the program. The format of this directive cards is (A5,5X,7F10.3).

Let's devide an 80-column computer card into 8 10-column fields and denote each field as DF_i ($i=1\sim 8$).

(1) Directive POSIT
This directive card provides general parameters.

DF1: POSIT, directive.
DF2: Ing, see comment for definition.
DF3: ISONIC, see comment for definition.
DF4: Shot depth in feet.
DF5: Detector depth in feet.
DF6: Sampling interval of time in milli-second.
DF7: Sampling interval of depth in feet.
DF8: Number of samples of the output, the length of the seismogram will be 2.0*DF6*DF8 milli-seconds.

Comment:

ING: Control number for a source and detector location.
ING=1, Both source and detector on the free surface.
ING=2, Source at free surface and detector buried.
ING=3, Source buried and detector at free surface.
ING=4, Both source and detector buried.
ISONIC: Control number for velocity input.
ISONIC=1, Input velocity is interval transit time.
ISONIC=2, Input velocity is real velocity.

(2) Directive FORMT
After this directive card a format card must be followed.

DF1: FORMT, directive.
DF2: Number of data per card.

(3) Directive VELOG

This directive card provide parameters for the velocity input and can accept next deck of velocity input cards whose format is described on a previous format card and number of input data per card is described on FORMT card.

DF1: VELOG, directive
DF2: SCALV, see comment for definition.
DF3: IGOV, see comment for definition.
DF4: TREF, see comment for definition.
DF5: Depth of the first data point in feet.
DF6: Surface velocity in ft/sec/1000.
DF7: Velocity of half-space in ft/ms.
DF8: FACT, see comment for definition.

Comment:

IGOV: Control number for the over-burden velocity.
IGOV=1, Constant velocity from the free surface to the first data point.
IGOV=2, Linear change of velocity from the free surface to the first data point.
IGOV=3, Velocity data is given from the free surface.

In case that input velocity is velocity rather than sonic log, DF4, DF8 can be any number. DF2 is a scale factor to convert the dimension of input velocity into ft/ms. If the dimension of input velocity is ft/sec, then DF2 is 0.001. If input velocity is interval transit time (sonic log), it must be converted into velocity by using the following equation:

$$v = c/(a*DT + T)$$

where v is velocity in ft/ms, DT is the interval transit time, and c,a, and T are constant to make correct velocity in ft/ms. In directive card,

DF2 = c
DF4 = T
DF8 = a.

(4) Directive DENST

This directive card provides parameters for density input and can accept next deck of input density cards. If there is no density information, this card can be ignored.

DF1: DENST, directive.
DF2: Scale factor to make all density in same dimension, DF2 can't be 0.
DF3: Same as DF3 field of directive VELOG except that it applies to the density.
DF4: Depth of the first data point.
DF5: Surface density.
DF6: Density of half-space.

(5) Directive PULST
This directive card has no parameters and can accept the e next deck of input pressure wavelet cards. If an impulsive seismogram is desired, this card must be ignored.

DF1: PULST,directive.

(6) Directive SHOTP
This is necessary to read check shot data, if available. If check shot data are available, two SHOTP cards are needed: One to read check shot depth in feet and another one to read check shot time in ms. If there are no ckeck shot data, this card should be ignored.

DF1: Shotp, directive.
DF2: 1 or 2, see comment

Comment:

If you want to read check shot depth after directive SHOTP, yoy must put 1 in DF2 field. If you want to read check shot time after SHOTP, you must put 2 in DF2 field. The first check shot point must be at the free surface- that is check shot depth is 0 feet and check shot time is 0 ms.

(7) Directive DEVIC
This is necessary to assign logical units for input-output devices.

DF1: DEVIC, directive.
DF2: Logical unit for input.
DF3: Logical unit for output(line printer).
DF4: Logical unit for output(other than line printer for further processing).

Comment:

Pre-assigned value of this card is
DF2=2,
DF3=3,
DF4=12.

The format for logical unit DF4 is (4X,F10.0,6X,E20.8,3X,F10.0,6X,E20.8), where first field is time and second field is velocity.

(8) Directive BEXIT
This directive card must be included to execute the computer program. This card must be the end of input deck cards.

DF1: BEXIT, directive.

Further illustrations:

After 'FORMT' card, a format card must be followed, whose format is (16A5). This format card must include a signal space between two field characters. The following symbols are legal signals.

* ..... Stop code. This signal is an end of data and te terminates reading of the data.

S ..... Skip code. The field of characters preceeding this code and all remaining field on the card are skipped.

D ..... Omit code. The field of characters preceeding this code is omitted.

(blank) .... The preceeding data value is stored in an array.

The followings are examples of valid format card;
(F9.2,1A), (F10.0,1A).

After 'VELOG', 'PULST','DENST','SHOTP' card, there must be followed an input data deck velocity, pressure wavelet, density density, check shot data respectively, whose format is described on the preceeding format card which appears after 'FORMT' card.

When there is no VELOG directive card, the program terminates the job. Therefore, the minimum number of directive cards to execute this program are 4 (FORMT, POSIT,VELOG,EEXIT), and other directives are optional.

Example:

Given that:
(a) Input velocity is sonic log such that

v=1000/(DT+40), and check shot data and density data are not available.

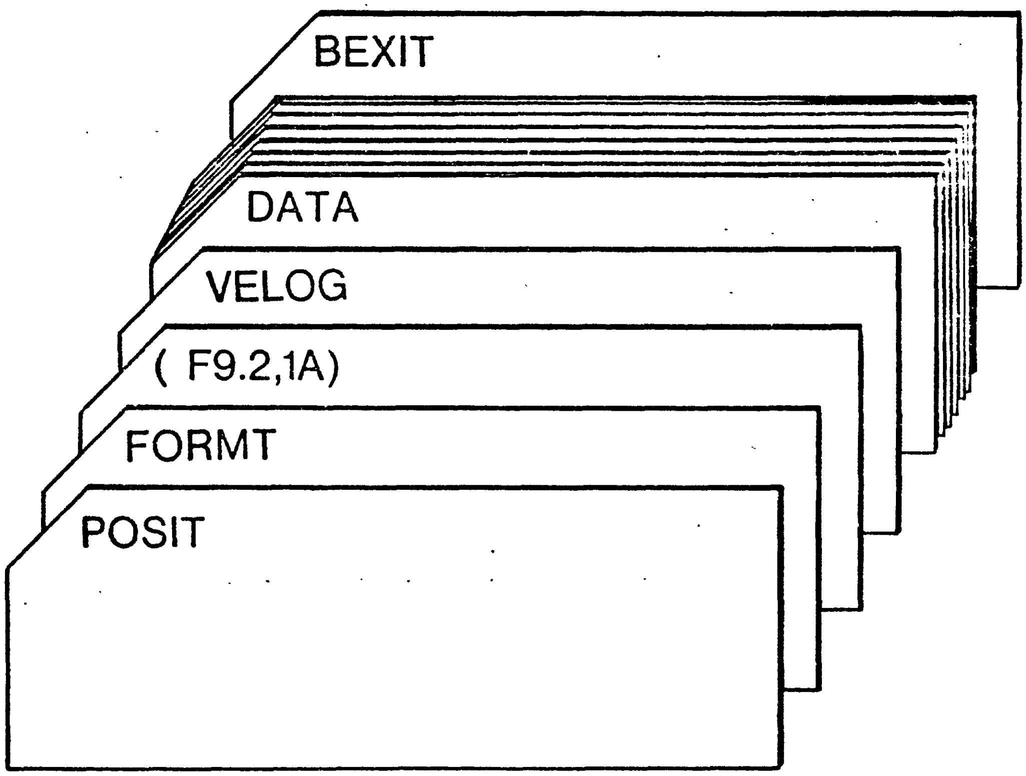
(b) Your first sonic input data is at depth 200 ft and over-burden velocity is assumed to be constant.

(c) Half-space velocity is 20 ft/ms and sampling interval DT=1 ms, DX=2 ft, and both shot source and detector located at the free surface.

From above information, we are going to make one-second long impluse response of layered half-space. The top of Figure A-1 shows the directive cards and the bottom of Figure A-1 shows how to put together input data cards.

DF1	DF2	DF3	DF4	DF5	DF6	DF7	DF8
POSIT	1.0	1.0	0.0	0.0	2.0	2.0	500.
FORMT	2						
VELOG	1000.	1.0	40.	200.	—	20.0	1.0
BEXIT							

DIRECTIVE CARDS



SEQUENCE OF INPUT CARDS

Figure A-1. Preparation of input cards for a synthetic seismogram.

```
C**************************************************
C     THIS PROGRAM COMPUTES SYNTHETIC SEISMOGRAM(VELOCITY RESPONSE)
C     FOR AN HORIZONTALLY LAYERED PERFECT ELASTIC HALF-SPACE.
C     ARGUMENT DEFINITIONS:
C         ING    =CONTROL NUMBER FOR SOURCE AND DECTOR LOCATION.
C             ING=1 SOURCE AT FREE SURFACE AND DETECTOR AT FREE SURFACE.
C             ING=2 SOURCE AT FREE SURFACE AND DETECTOR BURIED.
C             ING=3 SOURCE BURIED AND DETECTOR AT FREE SURFACE.
C             ING=4 SOURCE BURIED AND DETECTOR BURIED.
C         IGOV : CONTROL NUMBER FOR THE OVERBURDEN VELOCITY INFORMATION.
C           IGOV=1 ASSUME CONSTANT VELOCITY FROM THE SURFACE TO THE FIRST
C                  DATA POINT(INPUT DATA).
C           IGOV=2 LINEAR INTERPOLATION OF VELOCITY FROM THE SURFACE TO
C                  THE FIRST DATA POINT.
C           IGOV=3 VELOCITY DATA IS GIVEN FROM THE SURFACE.
C           *THEREFORE, WHEN YOUR INPUT IS START FROM 1000FEET BELOW THE SURFACE
C            AND ASSUME CONSTANT VELOCITY FOR 0 - 1000FEET, THEN IGOV=1.
C         ISONIC: CONTROL NUMBER FOR VELOCITY INFORMATION OF THE INPUT.
C           ISONIC=1 INPUT VELOCITY IS INTERVAL TRANSIT TIME(SONIC LOG).
C           ISONIC=2 INPUT VELOCITY IS REAL VELOCITY.
C         SPOINT: DEPTH SOURCE LOCATED IN FEET.
C         DPOINT: DEPTH DETECTOR LOCATED IN FEET.
C         DT     : SAMPLING INTERVAL IN TIME SECTION...ONE-WAY TRAVEL TIME.
C                  DIMENSION OF DT MUST BE MILLI-SECOND.
C         DZ     : SAMPLING INTERVAL OF DEPTH IN FEET, MUST BE IN FEET.
C         NOUT: THE LENGTH OF THE SEISMOGRAM.
C         SCALV: (1) WHEN ISONIC=2, SCALE FACTOR TO MAKE CORRECT DIMENSION
C                    OF VELOCITY OF INPUT IN FEET/MS. YOUR INPUT DATA IS FEET/MS,
C                    SCALV=1.0.
C                (2) WHEN ISONIC=1, SCALE FACTOR TO MAKE CORRECT DIMENSION OF
C                    VELOCITY FROM THE SONIC LOG BY THE FOLLOWING FORMULAR.
C                V=S/(C*T+TR), WHERE V=VELOCITY IN FT/MS,T=INTERVAL TRANSSIT
C                TIME ,S,TR,C ARE SCALE FACTOR FOR THE CORRECT VELOCITY.
C                S=SCALV.
C         TREF: TR IN THE ABOVE EQUATION,
C         FACT: C IN THE ABOVE EQUATION
C         DREFV: THE DEPTH IN FEET OF THE FIRST DATA POINT OF VELOCITY INPUT.
C         DREFD: THE DEPTH IN FEET OF THE FIRST DATA POINT OF DENSITY.
C         VSUR: SURFACE VELOCITY IN FT/MS.
C         VBASE: VELOCITY OF THE HALF-SPACE IN FT/MS.
C         DSUR: DENSITY OF THE FREE SURFACE.
C         DVASE: DENSITY OF THE HALF-SPACE.
C         SCALD: SCALE FACTOR FOR THE DENSITY.IT CAN BE ANY NUMBER EXCEPT ZERO.
C         ISEL: CONTROL NUMBER FOR THE CHECK SHOT DATA .
C           ISEL=1 INPUT DATA IS DEPTH IN FEET WHERE CHECK SHOT DATA IS GIVEN.
C           ISEL=2 INPUT IS CHECK SHOT TIME IN MS WHERE CHECK SHOT DATA IS GIVEN.
C           * FIRST CHECK SHOT POINT IS ON FREE SURFACE.
C**************************************************
          LOGICAL ERRIN
          DIMENSION LIBARY(8),PARA(7),FMT(16)
          DIMENSION CTIME(5000),DVEL(5000),R(5000),SMEGA(3000)
          DIMENSION CHECKT(50),CHECKD(50),PULSE(200)
          DIMENSION TVEL(1500),DEN(1500),SPOSI(1500),TEMP(1500)
          COMMON /BK1/IN,IOUT,ITAPE
          COMMON /BK3/DT,SPOINT,DPOINT
          COMMON /BK4/IDEC,ISOUR,NOUT,NPUL,FRONT,NTMAX,ISIGN,ING
```

```
          DATA LIBARY/5HPUSIT,5HFORMT,5HVELOG,5HDENST,5HPULST,
       1 5HDEVIC,5HBEXIT,5HSHUTP/
          NL=8
C**************************************************
C     PRESET PARAMETERS.
C**************************************************
          DATA MAXINT/10000/
          DATA IN,IOUT,ITAPE/2,3,12/
          DATA DT,DZ,SPOINT,DPOINT,PULSE(1)/1.0,2.0,0.0,0.0,1.0/
          DATA ICHECK,ISIGN,ISONIC,NOUT,NPUL,ING/2,2,2,999,1,1/
           ITERM=1
C**************************************************
C   READ DIRECTIVE CARDS
C**************************************************
1100      CALL MONITR(LIBARY,NL,PARA,INDEX)
          GO TO(701,702,703,704,705,706,707,708),INDEX
C**************************************************
C   READ PARAMETER CARD.
C**************************************************
701       ING=PARA(1)
          ISONIC=PARA(2)
          SPOINT=PARA(3)
          DPOINT=PARA(4)
          DT=PARA(5)
          DZ=PARA(6)
          NOUT=PARA(7)
          GO TO 1100
702       NPC=PARA(1)
C**************************************************
C   READ FORMAT CARD
C**************************************************
          READ(IN,1110)FMT
1110      FORMAT(16A5)
          GO TO 1100
C**************************************************
C   READ PARAMETER CARD FOR THE VELOCITY
C**************************************************
703        SCALV=PARA(1)
          IGOV=PARA(2)
          TREF=PARA(3)
          DREFV=PARA(4)
          VSUR=PARA(5)
          VBASE=PARA(6)
          FACT=PARA(7)
C**************************************************
C   READ VELOCITY INPUT.
C**************************************************
           CALL READIN(DVEL,NVEL,MAXINT,NPC,FMT,ERRIN)
          IF(ERRIN) CALL EXIT
          ITERM=2
          GO TO 1100
C**************************************************
C   READ PARAMETER CARD FOR THE DENSITY.
C**************************************************
704          SCALD=PARA(1)
          IGOD=PARA(2)
```

```
              DREFD=PARA(3)
              DSUR=PARA(4)
              DVASE=PARA(5)
      C*************************************************
      C   READ DENSITY INPUT
      C*************************************************
              CALL READIN(R,NDEN,MAXINT,NPC,FMT,ERRIN)
              IF(ERRIN) CALL EXIT
              ISIGN=1
              GO TO 1100
      C*************************************************
      C   READ INPUT PULSE
      C*************************************************
      705      CALL READIN(PULSE,NPUL,MAXINT,NPC,FMT,ERRIN)
              IF(ERRIN) CALL EXIT
              GO TO 1100
      C*************************************************
      C   READ INPUT, OUTPUT DEVICE LOGICAL UNIT.
      C*************************************************
      706      IN=PARA(1)
              IOUT=PARA(2)
              ITAPE=PARA(3)
              GO TO 1100
      708      ISEL=PARA(1)
              ICKECK=1
              GO TO(801,802),ISEL
      C*************************************************
      C   READ CHECK SHOT DATA...DEPTH
      C*************************************************
      801      CALL READIN(CHECKD,NSHOT,MAXINT,NPC,FMT,ERRIN)
              IF(ERRIN) CALL EXIT
              GO TO 1100
      C*************************************************
      C   READ CHECK SHOT DATA...TIME
      C*************************************************
      802      CALL READIN(CHECKT,NSHOT,MAXINT,NPC,FMT,ERRIN)
              IF(ERRIN) CALL EXIT
              GO TO 1100
      707      IF(ITERM.NE.1) GO TO 709
              WRITE(IOUT,711)
      711     FORMAT(//,4X,'NO VELOCITY INFORMATION...JOB ABORTED!.')
              CALL EXIT
      709      CONTINUE
      C*************************************************
      C      CHECK THE INPUT VELOCITY(SONIC OR VELOCITY)
      C*************************************************
              GO TO(714,715),ISONIC
      C*************************************************
      C   COMPUTES VELOCITY FROM THE SONIC LOG
      C*************************************************
      714      CALL SONVEL(DVEL,FACT,SCALV,TREF,NVEL)
              SCALV=1.0
      715      GO TO(716,716,717),IGOV
      C*************************************************
      C   COMPUTES OVER-BURDEN VELOCITY.
      C*************************************************
```

```
716     CALL MOVE(VSUR,DREFV,DVEL,NVEL,DZ,IGOV)
717     GO TO(718,719),ISIGN
718     GO TO(811,811,812),IGOD
C***********************************************
C   COMPUTES OVER-BURDEN DENSITY.
C***********************************************
811     CALL MOVE(DSUR,DREFD,SNEGA,NDEN,DZ,IGOD)
812      NVEL=AMIN0(NVEL,NDEN)
C***********************************************
C   COMPUTES CUMULATIVE TIME
C***********************************************
719      CALL CUMUT(CTIME,DVEL,NVEL,DZ,SCALV,CHECKT,CHECKD,ICHECK,MSHOT)
C***********************************************
C   COMPUTES INTERVAL VELOCITY IN EQUAL TIME SECTION.
C***********************************************
        CALL VELTIM(DVEL,CTIME,NVEL,TVEL,NTMAX,DT)
C***********************************************
C   COMPUTES INTERFACE NUMBER FOR THE SHOT AND DETECTOR POSITION.
C***********************************************
        CALL POSIT(TVEL,SPOINT,DT,ISOUR)
         CALL POSIT(TVEL,DPOINT,DT,IDEC)
        GO TO(813,814),ISIGN
C***********************************************
C   COMPUTES INTERVAL DENSITY FOR THE EQUAL TIME SECTION.
C***********************************************
813     CALL VELTIM(R,CTIME,NVEL,DEN,NTMAX,DT)
814      NTMAX=NTMAX+1
        TVEL(NTMAX)=VBASE
C***********************************************
C   LIST INTERVAL VELOCITY IN TIME SECTION
C***********************************************
        CALL OUTLIS(0.,DT,ISIGN,IGOV,SPOINT,DPOINT,VBASE,DVEL,TEMP,
      1 TVEL,NTMAX,2)
        GO TO(815,818),ISIGN
815      DEN(NTMAX)=DBASE
        DO 10 I=1,NTMAX
C***********************************************
C   COMPUTES ACOUSTIC IMPEDANCE
C***********************************************
10      TVEL(I)=TVEL(I)*DEN(I)
818     NTMAX=NTMAX-1
C***********************************************
C   COMPUTES REFLECTION COEFFICIENT
C***********************************************
        CALL REFL(TVEL,NTMAX,R)
        NTMAX=NTMAX+1
C***********************************************
C   COMPUTES SOME CONSTANT
C***********************************************
        CALL FRT(ING,TVEL,ISOUR,IDEC,FRONT,RTIME,R)
         GO TO (111,111,222,222),ING
C***********************************************
C   SYNTHETHIC SEISMOGRAM FOR THE SURFACE SOURCE
C***********************************************
111     CALL SURFAC(R,CTIME,DVEL,TVEL,DEN,TEMP,PULSE)
        GO TO 3000
```

```
C*************************************************
C   SYNTHETHIC SEISMOGRAM FOR THE BURIED SOURCE.
C*************************************************
222     CALL BURIED(R,CTIME,SNEGA,TVEL,DEN,DVEL,TEMP,PULSE,SPOSI)
3000    CONTINUE
C*************************************************
C   LIST VELOCITY RESPONSE
C*************************************************
        CALL OUTLIS(RTIME,DT,ISIGN,IGOV,SPOINT,DPOINT,VBASE,DEN,R,CTIME,
     1 NOUT,1)
        STOP
        END
```

```
COMMON BLOCKS

/BK1/(+3)
IN      +0      IOUT    +1      ITAPE   +2
/BK3/(+3)
DT      +0      SPOINT  +1      DPOINT  +2
/BK4/(+10)
IDEC    +0
ISOUR   +1      NOUT    +2      NPUL    +3      FRONT   +4      NTMAX   +5
ISIGN   +6      ING     +7

SUBPROGRAMS CALLED

OUTLIS  REFL    CUMUT   AMIN0.  POSIT   SONVEL
SURFAC  FRT     MONITR  MOVE    READIN  EXIT
BURIED  VELTIM

SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*ICHECK 1       *NDEN   2        CHECKD 3       *RTIME  65       TVEL   66      *NVEL   3022
 CTIME  3023     LIBARY 14633   *DZ     14643    CHECKT 14644    ERRIN  14726   *NL     14727
*NPC    14730    FMT    14731   *DREFD  14751    SNEGA  14752   *DVASE  22642   *ISEL   22643
*INDEX  22644   *SCALV  22645    SPOSI  22646    PARA   25602   *VSUR   25611    PULSE  25612
*IGOD   26122   *MSHOT  26123   *ISONIC 26124    DVEL   26125    .S0000 37735   *DSUR   37736
*TREF   37737    TEMP   37740    R      42674   *VBASE  54504   *ITERM  54505    DEN    54506
*IGOV   57442   *I      57443   *FACT   57444   *ICKECK 57445   *SCALD  57446   *MAXINT 57447
*DBASE  57450   *DREFV  57451

TEMPORARIES


MAIN.   [ NO ERRORS DETECTED ]
```

```
OUTLIS  LN47TL.FOR       FORTRAN V.4A(230) /KI   1-OCT-75        22:26   PAGE 1

              SUBROUTINE OUTLIS(RTIME,DT,ISIGN,IGO,SPOINT,DPOINT,VBASE,TIME,
          1    DEP,VEL,NMAX,IFLAG)
      C--------------SUBROUTINE OUTLIS------------------
      C        THIS SUBROUTINE LIST THE OUTPUT.
      C-------------------------------------------------
              COMMON /BK1/IN,IOUT,ITAPE
              DIMENSION TIME(1),VEL(1),DEP(1)
              DTT=DT*2.
              DTC=RTIME-DTT
              DO 10 I=1,NMAX
              DTC=DTC+DTT
      10      TIME(I)=DTC
              GO TO(111,112),IFLAG
      111     WRITE(IOUT,100)
      100     FORMAT(1H1,//,8X,'SYNTHETHIC SEISMOGRAM'//,
          1    8X,'OUTPUT IS VELOCITY RESPONSE')
              GO TO(12,13),ISIGN
      12      WRITE(IOUT,200)
      200     FORMAT(//,8X,'DENSITY INFORMATION IS GIVEN')
              GO TO 20
      13      WRITE(IOUT,300)
      300     FORMAT(//,8X,'DENSITY IS ASSUMED TO BE CONSTANT')
              GO TO 20
      112      WRITE(IOUT,400)
      400     FORMAT(1H1,//,8X,'INTERVAL VELOCITY.....FEET/MILLI-SECOND')
              GO TO(21,22,20),IGO
      21      WRITE(IOUT,500)
      500        FORMAT(//,8X,'OVERBURDEN VELOCITY IS ASSUMED TO BE CONSTANT')
              GO TO 20
      22      WRITE(IOUT,600)
      600      FORMAT(//,8X,'OVERBURDEN VELOCITY IS ASSUMED TO BE RAMP')
      20      WRITE(IOUT,700)SPOINT,DPOINT,VBASE
      700      FORMAT(//,8X,'SOUECE POSITION=',F6.0,2X,'FEET'/,
          1    8X,'DETECTOR POSITION=',F6.0,2X,'FEET'/,
          2   6X,'VELOCITY OF HALF SPACE=',F12.6,'FT/MS'///)
              GO TO(210,220),IFLAG
      210     WRITE(IOUT,900)
      900     FORMAT(11X,'TIME',15X,'VELOCITY',11X,'TIME',15X,'VELOCITY'//)
              WRITE(IOUT,800)(TIME(I),VEL(I),I=1,NMAX)
              WRITE(ITAPE,800)(TIME(I),VEL(I),I=1,NMAX)
      800     FORMAT(4X,F10.0,6X,E20.8,3X,F10.0,6X,E20.8)
              RETURN
      220     DEP(1)=0.0
              DO 30 I=2,NMAX
      30      DEP(I)=DEP(I-1)+DT*VEL(I-1)
              WRITE(IOUT,1000)
      1000    FORMAT(14X,'TIME',7X,'DEPTH',6X,'VELOCITY',10X,'TIME',8X,'DEPTH',
          1    6X,'VELOCITY'//)
              WRITE(IOUT,1100)(TIME(I),DEP(I),VEL(I),I=1,NMAX)
      1100    FORMAT(13X,F6.0,5X,F6.0,3X,F12.6,7X,F6.0,7X,F6.0,3X,F12.6)
              RETURN
              END
```

```
              SUBROUTINE MONITR(LIBARY,NL,PARA,INDEX)
      C-------------SUBROUTINE MONITR-------------------
      C   THIS SUBROUTINE MONITORS CONTROL CARD INPUT AND RETURNS,,
      C   THE INDEX OF THE CONTROL CARD NAME PLACED IN THE LIBARY ARRAY.
      C   ARGUMENT DEFINITIONS,,
      C     NL     = THE NUMBER OF NAMES IN LIBARY
      C     PARA   = THE ARRAY OF REAL PARAMETERS TO BE READ FROM THE CARD.
      C     INDEX = INDEX OF THE CONTROL CARD NAME FOUND IN THE LIBARY
      C               TO BE RECOGNIZES BY THE MONITOR
      C-------------------------------------------------
              DIMENSION LIBARY(NL),PARA(7)
              COMMON /BK1/IN,IOUT,ITAPE
              DATA NMAX/7/
              NUM=NMAX
              READ(IN,41)NAME,(PARA(I),I=1,NUM)
      41      FORMAT(A5,5X,7F10.3)
              DO 51 J=1,NL
              IF(NAME.NE.LIBARY(J)) GO TO 51
              INDEX=J
              RETURN
      51      CONTINUE
              WRITE(IOUT,61)NAME,(PARA(K),K=1,NUM)
      61        FORMAT(1H0,18H ERROR DECECTION -//,24H ILLEGAL CONTROL CARD...//,
             1  1H0,A5,5X,7F10.0)
              CALL EXIT
              END
```

```
COMMON BLOCKS

/BK1/(+3)
IN       +0       IOUT     +1       ITAPE    +2

SUBPROGRAMS CALLED

EXIT

SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*K        1       *NAME    2       LIBARY 3      *NUM     4      *NL      5      *INDEX  6
*J        7        PARA    10      .S0002 11      .S0001  12      .S0000  13      .I0002  14
 .I0001  15       *I       16      *NMAX  17      .I0000  20

TEMPORARIES

 .MON16 43

MONITR  [ NO ERRORS DETECTED ]
```

```
            SUBROUTINE READIN(A,NA,NMAX,NPC,FMT,ERROR)
C--------------SUBROUTINE READIN------------------
C    THIS SUBROUTINE READS A ONE DIMENSIONAL ARRAY IN VARIABLE FORMAT.
C    THE NUMBER OF VALUES PER CARD AND THE FORMAT ARE SPECIFIED IN THE
C    SUBROUTINE ARGUMENT. EACH FIELD OF CHARACTERS IS SEPARATED BY A
C    SIGNAL SPACE. THE FOLLOWING SYMBOLS ARE LEGAL SIGNALS.
C      (BLANK)            THE PRECEEDING DATA IS STORED IN AN ARRAY.
C                         READING OF THE DATA.
C       S     SKIP CODE. THE FIELD OF CHARACTERS PRECEEDING THIS CODE AND
C                         ALL REMAINING FIELDS ON THE CARD ARE SKIPPED.
C       D     OMIT CODE. THE FIELD OF CHARACTERS PRECEEDING THIS CODE IS
C                         OMITTED.
C       *     STOP CODE. THIS IS THE END OF THE DATA AND TERMINATES READING
C                         OF THE INPUT DATA.
C    ARGUMENT DEFINITIONS..
C      A     = ARRAY CONTAINING DATA READ FROM CARDS.
C      NA    = NUMBER OF ELEMENT IN ARRAY A.
C      NMAX = MAXIMUM NUMBER OF DATA VALUES ALLOWED.
C      NPC  = NUMBER OF DATA VALUES PUNCHED PER CARD.
C      FMT    = FORMAT FOR DATA CARDS.
C      ERROR= ERROR MESSAGE RETURNED- .TRUE. IF AN ERROR PRESENT
C------------------------------------------------
            LOGICAL ERROR
            INTEGER BLANK,STOP,OMIT,SIGNAL(20)
            DIMENSION WORD(20),FMT(16),A(1)
            COMMON /BK1/IN,IOUT,ITAPE
            DATA MESS/5HERROR/,BLANK,STOP,SKIP,OMIT/1H ,1H*,1HS,1HD/
            ERROR=.FALSE.
            IGO=1
      10    NA=0
      99    READ(IN,FMT)(WORD(I),SIGNAL(I),I=1,NPC)
            DO 60 J=1,NPC
            ICHEC=SIGNAL(J)
            IF(ICHEC.EQ.BLANK) GO TO 50
            IF(ICHEC.EQ.STOP) GO TO 40
            IF(ICHEC.EQ.SKIP) GO TO 99
            IF(ICHEC.EQ.OMIT) GO TO 60
            NUM=NA/NPC+1
            WRITE(IOUT,88)MESS,NUM
      88    FORMAT(1H0,10X,A6//,4X,'ILLEGAL SIGNAL IN INPUT ON DATA CARD',I4)
            ERROR=.TRUE.
            GO TO 60
            IF(NA.GT.NMAX) GO TO 70
      40   IGO=2
      50    NA=NA+1
            A(NA)=WORD(J)
            GO TO(60,90),IGO
      60    CONTINUE
            GO TO 99
      70    WRITE(IOUT,89)MESS,NMAX
      89    FORMAT(1H0,10X,A6//,8X,29HINPUT IN EXCESS OF ALLOWABLE ,I4)
            ERROR=.TRUE.
            GO TO 10
      90     RETURN
            END
```

```
      SUBROUTINE FRT(ING,TVEL,ISOUR,IDEC,FRONT,RTIME,R)
C--------------SUBROUTINE FRT--------------------
C  THIS SUBROUTINE COMPUTES THE CONSTANT TERM FOR VELOCITY RESPONSE.
C  ARGUMENT DEFINITIONS
C       SPOINT=SOURCE POSITION.......FEET.
C       DPOINT=DETECTOR POSITION.........FEET.
C       ISOUR =INTERFACE NUMBER SOURCE LOCATED
C       IDEC  =INTERFACE NUMBER DETECTOR LOCATED.
C       FRONT =CONSTANT TERM.
C       ING   =CONTROL NUMBER FOR SOURCE AND DECTOR LOCATION.
C          ING=1 SOURCE AT FREE SURFACE AND DETECTOR AT FREE SURFACE.
C          ING=2 SOURCE AT FREE SURFACE AND DETECTOR BURIED.
C          ING=3 SOURCE BURIED AND DEC
C          ING=4 SOURCE BURIED AND DETECTOR BURIED.
C---------------------------------------------------
      DIMENSION R(1),TVEL(1)
      COMMON /BK3/DT,SPOINT,DPOINT
      GO TO(111,222,333,444),ING
111   RTIME=0.0
      FRONT=1.0/TVEL(1)
      RETURN
222    RTIME=DT*IDEC
      IDEC1=IDEC-1
      FRONT=1.0/TVEL(1)
      CALL TRANS(SUM,1,IDEC1,R)
      FRONT=FRONT*SUM
      RETURN
333   RTIME=DT*ISOUR
      ISOUR1=ISOUR-1
      FRONT=1.0
      CALL TRANS(SUM,1,ISOUR1,R)
      FRONT=FRONT/SUM
      RETURN
444    IDEC1=IDEC-1
      IDEC2=IDEC+1
      ISOUR1=ISOUR-1
      ISOUR2=ISOUR+1
      IF(SPOINT.LE.DPOINT) GO TO 80
      RTIME=DT*(ISOUR-IDEC)
      FRONT=0.5*TVEL(1)/TVEL(IDEC)
      IFIST1=1
      ILAST1=IDEC1
      IFIST2=IDEC
      ILAST2=ISOUR1
      CALL TRANS(SUM,IFIST1,ILAST1,R)
      FRONT=FRONT/(SUM*SUM)
      CALL TRANS(SUM,IFIST2,ILAST2,R)
      FRONT=FRONT/SUM
      RETURN
80    RTIME=DT*(IDEC-ISOUR)
      CALL TRANS(SUM,ISOUR,IDEC1,R)
      FRONT=SUM*0.5
       RETURN
      END
```

```
      SUBROUTINE BURIED(R,D11,D12,E11,E12,SNEGA,TEMP,PULSE,SPOSI)
C------------SUBROUTINE BURIED--------------
C  THIS SUBROUTINE COMPUTES VELOCITY RESPONSE FOR BURIED SOURCE.
C  ARGUMENT DEFINITIONS
C              * SEE SUBROUTINE SURFAC *
C-------------------------------------------------------
      COMMON /BK4/IDEC,ISOUR,NOUT,NPUL,FRONT,NTMAX,ISIGN,ING
     DIMENSION R(1),D11(1),D12(1),SNEGA(1),TEMP(1),PULSE(1),SPOSI(1)
      DIMENSION E11(1),E12(1)
      NTMAX1=NTMAX-1
      IDEC1=IDEC-1
      IDEC2=IDEC+1
      ISOUR1=ISOUR-1
      ISOUR2=ISOUR+1
      IF(IDEC.GT.ISOUR) GO TO 8000
      IIN=1
      IF(ING.EQ.3) GO TO 531
      CALL MATRIX(R,1,IDEC1,D11,D12,E11,E12)
      TEMP(1)=D11(1)
      TEMP(2)=D11(2)-D12(2)-D12(IDEC)
      IF(IDEC.LT.2) GO TO 531
      TEMP(IDEC)=D11(2)-D12(IDEC)-D12(2)
      TEMP(IDEC2)=D11(1)
      IF(IDEC1.LT.3) GO TO 531
      DO 432 I=3,IDEC1
432   TEMP(I)=D11(I)-D12(I)+D11(IDEC1-I+3)-D12(IDEC1-I+3)
531   CALL MATRIX(R,1,ISOUR1,D11,D12,E11,E12)
      SPOSI(1)=D12(1)
      SPOSI(2)=D12(2)
      IF(ISOUR.LT.2) GO TO 532
      SPOSI(ISOUR)=D11(2)+D12(ISOUR)
      SPOSI(ISOUR2)=D11(1)
      IF(ISOUR.LT.4) GO TO 532
      DO 433 I=3,ISOUR1
433   SPOSI(I)=D12(I)+D11(ISOUR1-I+3)
532   SNEGA(1)=D11(1)
      IF(ISOUR.LT.3) GO TO 444
      DO 438 I=2,ISOUR1
438   SNEGA(I)=D11(I)+D12(ISOUR-I+2)
444   SNEGA(ISOUR)=D12(2)
      CALL MATRIX(R,1,NTMAX1,D11,D12,E11,E12)
      D11(NTMAX)=0.0
C
      CALL CONV(D12,NTMAX,SNEGA,ISOUR,R,L1OUT)
      CALL CONV(D11,NTMAX,SPOSI,ISOUR2,SNEGA,L2OUT)
      R(L2OUT)=0.
      DO 435 I=1,L2OUT
435   SPOSI(I)=R(I)-SNEGA(I)
      IF(ING.EQ.3) GO TO 10
      CALL CONV(SPOSI,L2OUT,TEMP,IDEC2,SNEGA,L3OUT)
      DO 777 I=ISOUR2,L3OUT
      SNEGA(IIN)=SNEGA(I)
777   IIN=IIN+1
      L3OUT=L3OUT-ISOUR
      D11(NTMAX)=0.0
      DO 436 I=1,NTMAX
```

```
436     SPOSI(I)=D11(I)-D12(I)
C
        CALL PCONV(D11,NOUT,SNEGA,L3OUT,SPOSI,NTMAX,PULSE,NPUL)
        DO 38 I=1,NOUT
38      D11(I)=D11(I)*FRONT
        GO TO 3000
10      DO 58 I=1,NTMAX
58      SNEGA(I)=D11(I)-D12(I)
        DO 888 I=ISOUR2,L2OUT
        SPOSI(IIN)=SPOSI(I)
888    IIN=IIN+1
        L2OUT=L2OUT-ISOUR
        CALL PCONV(D11,NOUT,SPOSI,L2OUT,SNEGA,NTMAX,PULSE,NPUL)
        IF(ISIGN.EQ.3) GO TO 3000
        DO 59 I=1,NOUT
59      D11(I)=D11(I)*FRONT
        GO TO 3000
8000    CALL MATRIX(R,1,ISOUR1,D11,D12,E11,E12)
        SPOSI(1)=D11(1)
        SPOSI(2)=D11(2)-D12(2)+D12(ISOUR)
        SPOSI(ISOUR)=-D11(2)-D12(ISOUR)+D12(2)
        SPOSI(ISOUR2)=-D11(1)
        IF(ISOUR1.LT.3) GO TO 632
        DO 633 I=3,ISOUR1
633     SPOSI(I)=D11(I)-D11(ISOUR1-I+3)-D12(I)+D12(ISOUR1-I+3)
632     IMAX=NTMAX1-IDEC+1
        CALL MATRIX(R,IDEC,IMAX,D11,D12,E11,E12)
        DO 640 I=1,IMAX
640      TEMP(I)=D11(I)+D12(IDEC+I)
        CALL CONV(SPOSI,ISOUR2,TEMP,IMAX,SNEGA,L3OUT)
        CALL MATRIX(R,1,NTMAX1,D11,D12,E11,E12)
        D11(NTMAX)=0.
        DO 641 I=1,NTMAX
641      SPOSI(I)=D11(I)-D12(I)
        CALL PCONV(D11,NOUT,SNEGA,L3OUT,SPOSI,NTMAX,PULSE,NPUL)
        DO 437 I=1,NOUT
437      D11(I)=D11(I)*FRONT
3000    RETURN
        END


COMMON BLOCKS

/BK4/(+10)
IDEC    +0      ISOUR   +1      NOUT    +2      NPUL    +3      FRONT   +4
NTMAX   +5      ISIGN   +6      ING     +7

SUBPROGRAMS CALLED

PCONV   CONV    MATRIX

SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*NTMAX1  1      *IDEC2   2      E12      3      *IDEC1   4      D11      5      *IIN     6
 SNEGA   7       SPOSI   10     .S0007   11      .S0006  12     PULSE    13     *IMAX    14
```

```
      SUBROUTINE SURFAC(R,D11,D12,E11,E12,TEMP,PULSE)
C--------------SUBROUTINE SURFAC---------------------
C  THIS SUBROUTINE COMPUTES THE VELOCITY RESPONSE FOR SURFACE SOURCE.
C  ARGUMENT DEFINITIONS
C      IDEC  =INTERFACE NUMBER DETECTOR LOCATED.
C      ING   =CONTROL NUMBER FOR SOURCE AND DECTOR LOCATION.
C         ING=1 SOURCE AT FREE SURFACE AND DETECTOR AT FREE SURFACE.
C         ING=2 SOURCE AT FREE SURFACE AND DETECTOR BURIED.
C         ING=3 SOURCE BURIED AND DETECTOR AT FREE SURFACE.
C         ING=4 SOURCE BURIED AND DETECTOR BURIED.
C      NOUT =LENGTH OF THE OUTPUT ARRAY D11.
C      NTMAX=THE NUMBER OF LAYER + HALF-SPACE
C      PULSE=DIGITIZED INPUT WAVE FUNCTION.
C      NPUL =LENGTH OF THE ARRAY PULSE
C      ISIGN = CONTROL NUMBER FOR DENSITY.
C       ISIGN=1 DENSITY INFORMATION IS GIVEN.
C       ISIGN=2 DENSITY IS ASSUMED TO BE CONSTANT.
C--------------------------------------------------
      DIMENSION R(1),D11(1),D12(1),DEN(1),TEMP(1),PULSE(1),SPOSI(1)
      DIMENSION E11(1),E12(1)
      COMMON /BK4/IDEC,ISOUR,NOUT,NPUL,FRONT,NTMAX,ISIGN,ING
      NTMAX1=NTMAX-1
      GO TO(33,44),ING
33    CALL MATRIX(R,1,NTMAX1,D11,D12,E11,E12)
      D11(NTMAX)=0.0
      DO 30 K=1,NTMAX
      R(K)=D11(K)-D12(K)
30    D12(K)=D11(K)+D12(K)
      CALL PCONV(D11,NOUT,D12,NTMAX,R,NTMAX,PULSE,NPUL)
      DO 34 I=1,NOUT
34    D11(I)=D11(I)*FRONT
      GO TO 3000
C   SYNTHETHIC SEISMOGRAM FOR BURIED DETECTOR AND SURFACE SOURCE
44      IDEC1=IDEC-1
      IDEC2=IDEC+1
      IMAX=NTMAX-IDEC
      CALL MATRIX(R,IDEC,IMAX,D11,D12,E11,E12)
      DO 60 K=1,IMAX
60    TEMP(K)=D11(K)+D12(K+IDEC)
      CALL MATRIX(R,1,NTMAX1,D11,D12,E11,E12)
      D11(NTMAX)=0.0
      DO 70 K=1,NTMAX
70    D12(K)=D11(K)-D12(K)
      CALL PCONV(D11,NOUT,TEMP,IMAX,D12,NTMAX,PULSE,NPUL)
      DO 49 I=1,NOUT
49    D11(I)=D11(I)*FRONT
3000  RETURN
      END
```

COMMON BLOCKS

```
/BK4/(+10)
IDEC    +0      ISOUR   +1      NOUT    +2      NPUL    +3      FRONT   +4
NTMAX   +5      ISIGN   +6      ING     +7
```

```
      SUBROUTINE POSIT(Y,DPOINT,DT,IPOINT)
C---------------SUBROUTINE POSIT------------------
C       THIS SUBROUTINE DETERMINE ARRAY(LAYER NUMBER) INDEX FOR
C       DETECTOR AND SOURCE.
C       ARGUMENT DEFINITIONS:
C       IPOINT: THE LAYER INTERFACE NUMBER EITHER DECTOR OR SOURCE LOCATED
C       DPOINT: THE ACTUAL DEPTH EITHER DECTOR OR SOURCE LOCATED.
C       Y: INTERVAL VELOCITY FOR EQUAL TIME SECTION
C       DT: TIME INCREMENT IN Y.
C--------------------------------------------------
      DIMENSION Y(1)
      M=1
      SUM=0.0
99    SUM=SUM+DT*Y(M)
      IF(DPOINT-SUM)1,2,3
2     IPOINT=M
      RETURN
3     M=M+1
      GO TO 99
1     TEMP=SUM-Y(M)*DT
      D1=SUM-DPOINT
      D2=DPOINT-TEMP
      IF(D1.GT.D2) IPOINT=M-1
      IF(D1.LE.D2) IPOINT=M
      RETURN
      END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
*IPOINT 1     *DT     2     *D2     3     Y     4     *M     5     *D1     6
*SUM    7     *TEMP   10    *DPOINT 11
```

TEMPORARIES

POSIT [NO ERRORS DETECTED]

```
      SUBROUTINE VELTIM(X,T,NT,Y,ND,DT)
C-------------SUBROUTINE VELTIM-----------------
C     THIS SUBROUTINE COMPUTES INTERVAL VELOCITY FOR EQUAL TIME
C      SECTION FROM VELOCITY FOR EQUAL DEPTH.
C     ARGUMENT DEFINITIONS:
C       T: CUMULATIVE TIME (T(1)=0.0)
C       X: INTERVAL VELOCITY FOR EQUAL DISTANCE CALCULATED FROM SONIC LOG
C       Y: NEW INTERVAL VELOCITY FOR EQUAL TIME SECTION
C       NT: TOTAL NUMBER IN ARRAY X.
C       ND: TOTAL NUMBER IN ARRAY Y.
C       DT: TIME INCREMENT IN Y.
C-----------------------------------------------
      DIMENSION X(1),Y(1),T(1)
      Y(1)=X(1)
      ND=IFIX(T(NT)/DT)
      M=2
      DT1=0.0
      DO 100 I=2,ND
      DT1=DT+DT1
99    CRT=DT1-T(M)
      IF(CRT.LE.0.0) GO TO 55
      M=M+1
      GO TO 99
55    Y(I)=X(M)+(X(M)-X(M-1))*CRT/(T(M)-T(M-1))
100   CONTINUE
      ND=ND-1
      DO 10 I=1,ND
10    Y(I)=(Y(I)+Y(I+1))/2.0
      ND=ND+1
      Y(ND)=X(NT)
      RETURN
      END
```

SUBPROGRAMS CALLED

IFIX.

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

*DT1	1	T	2	*DT	3	Y	4	*M	5	*CRT	6		
.S0001	7	.S0000	10	X	11	*ND	12	*I	13	*NT	14		

TEMPORARIES

.VEL16 15

VELTIM [NO ERRORS DETECTED]

```
      SUBROUTINE SONVEL(V,FACT,SDZ1,TREF,NMAX)
C-------------SUBROUTINE SONVEL-----------------
C    THIS SUBROUTINE CONVERTS INVERVAL TRANSIT TIME INTO
C     INTERVAL VELOCITY.
C     ARGUMENTS DEFINITIONS:
C      V:INTERVAL TRANSIT TIME AND INTERVAL VELOCITY
C      SDZ1,FACT,TREF: CONSTANTS FOR MAKING UCRRECT VELOCITY AND DIMENSION
C      NMAX:THE LENGTH OF ARRAY V.
C----------------------------------------------
      DIMENSION V(1)
      DO 40 I=1,NMAX
40    V(I)=SDZ1/(FACT*V(I)+TREF)
      RETURN
      END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
 V      1      .S0000 2      *SDZ1   3      *TREF   4      *I      5      *NMAX   6
*FACT   7
```

TEMPORARIES

SONVEL [NO ERRORS DETECTED]

```
      SUBROUTINE MOVE(SUR,DREF,V,NT,DZ,IGO)
C--------------SUBROUTINE MOVE--------------------
C     THIS SUBROUTINE COMPUTES OVER-BURDEN VELOCITY OR DENSITY BY
C     LINEAR INPERPOLATION OR CONSTANT VALUE. THE INPUT AND OUTPUT ARE STORED AT
C     THE SAME ARRAY NEME V.
C     ARGUMENT DEFINITIONS:
C       SUR: FREE SURFACE VALUE...INPUT.
C       DREF: FIRST DATA VALU
C       DREF: DEPTH OF THE FIRST DATA POINT IN FEET.
C       V: VELOCITY OR DENSITY.
C       NT: THE LENGTH OF ARRAY V.
C       DZ: SAMPLING INTERVAL IN FEET(DEPTH).
C       IGO: CONTROL NUMBER WHICH REPRESENTS,
C         IGO=1...CONSTANT VALUE FOR OVER-BURDEN MEDIUM.
C         IGO=2 LINEAR INTERPOLATION OF OVER-BURDEN  MEDIUM.
C-------------------------------------------------
      DIMENSION V(1)
      FIR=V(1)
      IREF=IFIX(DREF/DZ)
      IREF1=IREF+1
      IF(IREF.LT.1) RETURN
      NMAX=NT+IREF
      DO 10 I=NMAX,IREF1,-1
10    V(I)=V(I-IREF)
      GO TO(41,42),IGO
41    DO 20 I=1,IREF
20    V(I)=FIR
      NT=NMAX
      RETURN
42    SLOPE=(FIR-SUR)/DREF
      DZ2=-DZ
      DO 30 I=1,IREF
      DZ2=DZ+DZ2
30    V(I)=SUR+SLOPE*DZ2
      NT=NMAX
      RETURN
      END
```

SUBPROGRAMS CALLED

IFIX.

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

*IGO	1	*DREF	2	*DZ	3	*DZ2	4	*IREF1	5	*SLOPE	6
V	7	*IREF	10	*FIR	11	.S0002	12	.S0001	13	.S0000	14
*I	15	*NMAX	16	*NT	17	*SUR	20				

TEMPORARIES

.MOV16 21

MOVE   [NO ERRORS DETECTED]

```
          SUBROUTINE CUMUT(T,V,NT,DZ,SCALE,CHECK,CHECKD,ICHECK,IMAX)
C--------------SUBROUTINE CUMUT-------------------
C       THIS SUBROUTINE COMPUTES CUMULATIVE TIME FOR WAVE TO PROPAGATE FRO-
C       SURFACE TO  INSIDE THE HALF SPACE.
C        ARGUMENT DEFINITIONS
C         ICHECK: =1 CHECK SHOT DATA IS GIVEN
C                 =2 CHECK SHOT DATA IS NOT GIVEN
C         DZ: DEPTH INCREMENT
C         T: CUMULATIVE TIME
C         V: INTERVAL VELOCITY FOR EQUAL DEPTH
C         NT: DIMENSION OF V.
C-------------------------------------------------
          DIMENSION V(1),T(1),CHECK(1),CHECKD(1)
          T(1)=0.0
          DZ=DZ*SCALE
          DO 10 I=2,NT
          TEMP=DZ/V(I-1)
10        T(I)=T(I-1)+TEMP
          GO TO(100,200),ICHECK
100          IMAX1=IMAX+1
          CHECK(IMAX1)=T(NT)
          DO 20 I=1,IMAX
          M1=IFIX(CHECKD(I)/DZ)+1
          M2=IFIX(CHECKD(I+1)/DZ)+1
          CONST=(CHECK(I+1)-CHECK(I))/(T(M2)-T(M1))
          DO 30 M=M1,M2
30        T(M)=CONST*(T(M)-T(M1))+CHECK(I)
20        CONTINUE
200        RETURN
         END

SUBPROGRAMS CALLED

IFIX.

SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*ICHECK 1       CHECKD 2        T       3       *DZ     4       *M2     5       *IMAX1  6
 V      7       *M1    10       *M      11      *IMAX   12      .S0002 13       .S0001 14
.S0000 15        CHECK 16       *SCALE  17      *CONST  20      *TEMP   21      *I     22
*NT    23

TEMPORARIES

 .CUM16 24

CUMUT   [ NO ERRORS DETECTED ]
```

```
          SUBROUTINE CONV(A,LA,B,LB,C,LC)
C---------------SUBROUTINE CONV-------------------
C       THIS SUBROUTINE CONVOLVES TWO FUNCTIONS.
C         A,B: TWO FUNCTIONS TO BE CONVOLVED.
C         C: RESULTANT FUNCTION.
C         LA: THE LENGTH OF ARRAY A.
C         LB: THE LENGTH OF ARRAY B.
C         LC: THE LENGTH OF ARRAY C.
C-------------------------------------------------
          DIMENSION A(LA),B(LB),C(LC)
          LC=LA+LB-1
          DO 10 I=1,LC
10        C(I)=0.0
          DO 1 I=1,LA
          DO 1 J=1,LB
          K=I+J-1
1         C(K)=C(K)+A(I)*B(J)
          RETURN
          END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
 .I0010 1     *K      2     B      3     *J      4     A      5     .S0002 6
 .S0001 7     .S0000 10    *LC    11     .I0007 12     .I0006 13    .I0005 14
 .I0004 15    *LB    16     .I0003 17    .I0002 20     .I0001 21    *I     22
 .I0000 23     C     24    *LA    25
```

TEMPORARIES

```
 .CON16 26
```

CONV [NO ERRORS DETECTED]

```
REFL     LN47TL.FOR       FORTRAN V.4A(230) /KI    1-OCT-75         22:26    PAGE 1

               SUBROUTINE REFL(Z,NMAX,R)
        C--------------SUBROITINE REFL-------------------
        C  THIS SUBROUTINE COMPUTES NEGATIVE REFELCTION COEFFICIENTS.
        C     ARGUMENT DEFINITIONS:
        C        R: REFLECTION COEFFICIENT
        C        Z: IMPEDANCE
        C------------------------------------------------
               DIMENSION Z(1),R(1)
               DO 20 I=1,NMAX
        20     R(I)=(Z(I)-Z(I+1))/(Z(I)+Z(I+1))
               RETURN
               END

SUBPROGRAMS CALLED


SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

 Z       1         .S0000 2         R       3         *I       4         *NMAX    5

TEMPORARIES


REFL     [ NO ERRORS DETECTED ]
```

```
      SUBROUTINE MATRIX(R,ING,MT,D11,D12,E11,E12)
C--------------SUBROUTINE MATRIX-----------------
C     THIS SUBROUTINE COMPUTES THE MULTIPLICATION OF THE LAYER
C     MATRIX BY A ITERATIVE SCHEME.
C     ARGUMENT DEFINITIONS:
C      R: REFLECTION COEFFICIENT
C      D11,D12,E11,E12: WORKING ARRAY.
C      ING: STARTING LAYER MATRIX TO BE MULTIPLIED.
C      MT: TOTAL NUMBER OF MATRIX TO BE MULTIPLIED.
C-----------------------------------------------
      DIMENSION D11(1),D12(1),E11(1),E12(1),R(1)
      D11(1)=1.0
     D12(ING+1)=R(ING)
     RT=D12(ING+1)
      D12(ING)=0.0
      IF(MT.LT.2) GO TO 70
      DO 30 I=2,MT
     RC=R(ING+I-1)
     I1=I-1
     DO 10 K=2,I1
     D11(K)=E11(K)+E12(ING+I-K+1)*RC
10    D12(ING+K)=E12(ING+K)+E11(I-K+1)*RC
     D11(I)=RT*RC
     D12(ING+I)=RC
     DO 20 K=1,I
      KM=K+ING
      E11(K)=D11(K)
20    E12(KM)=D12(KM)
30    CONTINUE
      RETURN
70    D11(2)=0.0
      IF(MT.LT.1) D12(ING+1)=0.0
      RETURN
      END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
*K      1     *KM     2      E12    3     *MT     4      D11    5     *RT    6
 D12    7     *ING    10    *RC     11    .S0002 12      .S0001 13    .S0000 14
 R      15    *I1     16    *I      17     E11   20
```

TEMPORARIES

MATRIX [NO ERRORS DETECTED]

```
        SUBROUTINE TRANS(SUM,NIN,NLAST,R)
C--------------SUBROUTINE TRANS-------------------
C      THIS SUBROUTINE COMPUTES MULTIPLICATION OF TRANSMISSION COEFFICIENT.
C      ARGUMENT DEFINITIONS:
C        R: REFLECTION COEFFICIENT.
C        NIN: STARTING LAYER TO BE MULTIPLIED.
C        NLAST: ENDING LAYER TO BE MULTIPLIED.
C-------------------------------------------------
        DIMENSION R(1)
        SUM=1.0
        DO 10 I=NIN,NLAST
10      SUM=SUM*(1.0+R(I))
        RETURN
        END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
*SUM    1        .S0000 2       *NIN    3        R      4       *I      5       *NLAST  6
```

TEMPORARIES

TRANS [NO ERRORS DETECTED]

```
      SUBROUTINE PCONV(Y,NOUT,A,NA,B,NB,X,NS)
C--------------SUBROUTINE PCONV------------------
C    THIS SUBROUTINE COMPUTES POLYNOMIAL CONVOLUTION OF THE FOLLOWING
C    TYPE   Y(Z)=A(Z)*S(Z)/B(Z) USING RECURSIVE RELATION.
C      ARGUMENT DEFINITIONS
C       Y: OUTPUT ARRAY
C       X: INPUT SOURCE FUNCTION
C       A,B: THE COEFFICIENT OF THE TRANSMISSION MATRIX
C       NOUT: DIMENSION OF AOUTPUT ARRAY
C       NS: DIMENSION OF X
C       NA: DIMENSION OF A
C       NB: DIMENSION OF B
C      NOTICE THAT B(1) SHOULD BE 1.0.
C-----------------------------------------------
       DIMENSION Y(1),A(1),B(1),X(1)
       NTOT=NOUT-1
       IF(NS.EQ.1) GO TO 77
       CALL CONV(A,NA,X,NS,Y,NTEMP)
       NTEMP2=NTEMP+1
       DO 10 I=NTEMP2,NOUT
10     Y(I)=0.0
       GO TO 88
77     DO 20 I=1,NA
20     Y(I)=A(I)
       NA2=NA+1
       DO 30 I=NA2,NOUT
30     Y(I)=0.0
88     NA1=NA-1
       NB1=NB-1
       DO 60 K=1,NB1
       DO 60 IK=1,K
       KT=K+1
60     Y(KT)=Y(KT)-B(IK+1)*Y(KT-IK)
       DO 70 K=NB,NTOT
       DO 70 IK=1,NB1
       KT=K+1
70     Y(KT)=Y(KT)-B(IK+1)*Y(KT-IK)
       RETURN
       END
```

SUBPROGRAMS CALLED

CONV

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
*NA1     1      *NS      2      *NTOT    3      *NB      4      *K       5      *IK      6
 B       7      *NTEMP  10      *NA     11      *NA2    12       Y      13      *NTEMP2 14
 .S0006 15       .S0005 16       .S0004 17       .S0003 20       A      21       .S0002 22
 .S0001 23       .S0000 24      *NOUT   25       X      26      *NB1    27      *I      30
*KT     31
```

APPENDIX B

Computer Program and User's Manual for One-dimensional Finite Difference Scheme

This computer program computes particle displacements in an one-dimensional inhomogeneous medium by a finite difference scheme. This program calculates displacement fields for the following three types of earth material:

(1) Perfectly elastic material.
(2) Voigt solid.
(3) Solid whose attenuation varies approximately linear with frequency.

To solve a free boundary condition, an imaginary grid point is included at x=-DX, where Dx is the spacial sampling interval. We implemented a radiation boundary condition under the assumption that half-space is perfectly elastic.

The pre-assigned logical units for the input-output devices are:

2= Input logical unit.
3= Output logical unit for line printer.
12= Output logical unit except line printer for the further processing.

The format of logical unit 12 is the same as logical unit 3 except that there are no headings.

Figure B-1 shows how to prepare input cards to run this program. Each BLK_i contains parameters of the model and each BLK_i will be read by a format described previous FMT (format) card. The definition of each BLK_i are written below.

BLK1

This block contains DT,DZ,T,NITER,NGEO,IDENS,IMODE,VEHL, DENH, NMAX, IGEOP sequentially.

1. DT: Sampling interval of time.
2. DX; Sampling interval of distance.
3. T: The width of source function. In this program, we used the following function as a sourcr function:

$$f(t) = e^{-\alpha t^2}$$

The width T is defined as

$$T = \sqrt{\frac{2}{\alpha}}$$

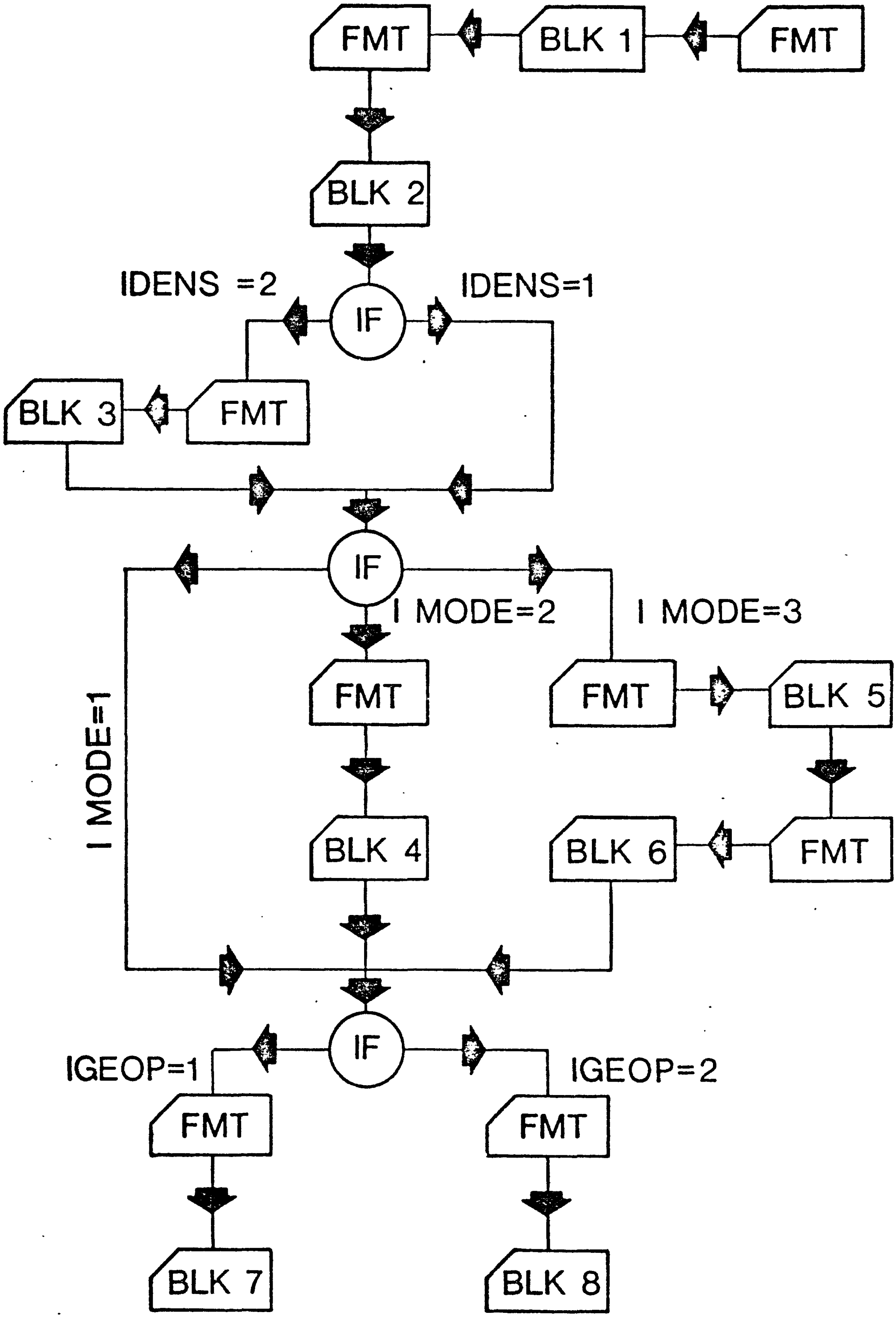


Figure B-1. Preparation of input cards for one-dimensional finite difference schemes.

4. NITER: Number of iteration.
5. NGEO: Number of output geophone(arraY).
6. IDENS: Control number for density,
   IDENS=1, Density is assumed to be constant.
   IDENS=2, Density data are given.
7. IMODE: Control number for the model type,
   IMODE=1, Perfectly elastic.
   IMODE=2, Voigt solid.
   IMODE=3, Linear with frequency solid.
8. VELH: Velocity of half-space.
9. DENH: Density of half-space.
10. NMAX: Number of input data (velocity,density, etc)
11. IGEOP: Control number for the output,
    IGEOP=1, Output are regularly spaced.
    IGEOP=2, Output are irregularly spaced.

BLK2

This block contains the p-wave velocity of the model.

BLK3

This block contains the density of the model.

BLK4

This block contains the attenuation term of the Voigt solid. The equation of motion for the Voigt solid is:

$$\rho \frac{\partial^2 u}{\partial t^2} = (\lambda+2\mu)\frac{\partial^2 u}{\partial x^2} + (\lambda'+2\mu')\frac{\partial^2 u}{\partial t \partial x}$$

where u is displacement, ρ is density, and $\lambda, \mu, \lambda', \mu'$ are Lame' constants. The attenuation input is defined as

$$\frac{\lambda'+2\mu'}{\rho}.$$

BLK5

This block contains the second attenuation term for the linear with frequency solid. See BLK6.

BLK6

This block contains the first attenuation term for the linear with frequency solid. The constitutive equation for this material can be written as

$$P_{xx} + \alpha \frac{\partial P_{xx}}{\partial t} = (\lambda+2\mu)\frac{\partial u}{\partial x} + (\lambda+2\mu)\beta \frac{\partial^2 u}{\partial t \partial x}$$

where α is the first attenuation term and β is the second attenuation term.

BLK7

This block contains the information about regularly spaced output array.

1. IDECS: The starting index number for the output.
2. IDECL: The ending index number for the output.
3. IDEINC: Index interval between output arrays.

BLK8

This block contains the information about the irregularly spaced geophones. So the input of this block is the index numbers of the output.

```
C***************************************************
C***************************************************
C       THIS PROGRAM COMPUTRS PARTICLE DISPLACEMENT IN AN ONE-DIMENSIONAL
C       INHOMOGENEOUS MEDIUM BY A FINITE DIFFERENCE SCHEME. THIS PROGRAM
C       CALCULATES DISPLACEMENT FIELD FOR THE FOLLOWING THREE TYPE OF
C       EARTH MATERIAL.
C         (1) PERFECT ELASTIC.
C         (2) VOIGT SOLID.
C         (3) SOLID WHOSE ATTENUATION VARIES LINEARY WITH FREQUENCY.
C       THE DISPLACEMENT SOURCE IS LOCATED ON THE FREE SURFACE.
C       RADIATION BOUNDARY CONDITION IS IMPLEMENTED.
C       LOGICAL UNIT OF THE INPUT-OUTPUT ARE:
C         2= INPUT.
C         3= OUTPUT (LINE PRINTER)
C         12= OUTPUT (MAGETIC TAPE OR DISK FILE).
C       THE HALF-SPACE IS ASSUMED TO BE PERFECT ELASTIC.
C       TO SOLVE THE FREE BOUNDARY CONDITION AN IMAGINARY MESH POINT IS
C       INCLUED( INDEX=1, ACTUAL FREE BOUNDARY IS LOCATED AT INDEX=2).
C       ARGUMENT DEFINITIONS:
C        IMODE: CONTROL NUMBER FOR THE TYPE OF THE MEDIUM.
C           =1: PERFECT ELASTIC.
C           =2: VOIGT SOLID.
C           =3: LINEAR WITH FREQUENCY SOLID.
C        IDENS: CONTROL NUMBER FOR DENSITY.
C           =1: DENSITY IS ASSUMED TO BE CONSTANT.
C           =2: DENSITY INFORMATION IS GIVEN.
C        IGEOP: CONTROL NUMBER FOR THE OUTPUT GEOPHONE ARRAY.
C           =1: REGULARY SPACED OUTPUT.
C           =2: IRREGULARY SPACED GEOPHONE GROUP.
C        DT: TIME INCREMENT.
C        DX: SAMPLING INTERVAL IN DISTANCE.
C        NITER: NUMBER OF ITERATION.
C        IDECS: STARTING INDEX FOR THE OUTPUT.
C        IDECL: ENDING INDEX FOR THE OUTPUT.
C        IDEINC: INTERVAL BETWEEN GEOPHONE OF THE OUTPUT.
C        T: THE WIDTH OF THE SOURCE FUNCTION.
C        NGEO: NUMBER OF THE OUTPUT ARRAY.
C        NMAX: TOTAL NUMBER OF INPUT ARRAY(VEL OR DEN,..).
C        VELH: VELOCITY OF HALF-SPACE.
C        DENH: DENSITY OF HALF-SPACE
C        VEL: P-WAVE VELOCITY.
C        DEN: DENSITY.
C        PULSE: INPUT DISPLACEMENT PULSE
C        IGEO: ARRAY WHICH CONTAINS THE INDICES FOR THE OUTPUT( SELECTED GEOPHONE)
C***************************************************
         DIMENSION VEL(200),DEN(200),A(200),B(200),C(200),PULSE(100)
         DIMENSION TEMP(200),ATEMP(200),ALPA(200),BETA(200),S(10)
         DIMENSION E(10),F(10),G(200),IGEO(10),FMT(16)
         COMMON /INOUT/IN,IOUT,ITAPE,IDECL,IDECS,IDEINC
         COMMON /BK1/DT,DX,TEND,T
         COMMON /BK2/VEL1,VELB,VELH,DEN1,DENB,DENH,ALPA1,ALPAB,BETA1,
       1 BETAB
         COMMON /BK3/IMODE,IGEOP,IDENS,NPUL,NMAX,NMED,N2,NGEO
         COMMON /BK4/SUM1,CON1,CON2,CON3,CON4
         DATA IN,IOUT,ITAPE/2,3,12/
         READ(IN,100)FMT
```

```
100     FORMAT(16A5)
C**************************************************
C   READ COMMON PARAMETERS
C**************************************************
        READ(IN,FMT)DT,DZ,T,NITER,NGEO,IDENS,IMODE,VELH,DENH/NMAX,IGEOP
        READ(IN,100)FMT
C**************************************************
C   READ VELOCITY
C**************************************************
        READ(IN,FMT)(VEL(I),I=1,NMAX)
        VEL1=VEL(1)
        VELB=VEL(NMAX)
        GO TO (11,12),IDENS
12      CONTINUE
        READ(IN,100)FMT
C**************************************************
C   READ DENSITY
C**************************************************
        READ(IN,FMT)(DEN(I),I=1,NMAX)
        DEN1=DEN(1)
        DENB=DEN(NMAX)
        GO TO 15
11      DEN1=1.0
        DENB=1.0
15      GO TO (13,14,14),IMODE
14      CONTINUE
        READ(IN,100)FMT
C**************************************************
C   READ ATTENUATION PARAMETER
C**************************************************
        READ(IN,FMT)(BETA(I),I=1,NMAX)
        BETA1=BETA(1)
        BETAB=BETA(NMAX)
        IF(IMODE.EQ.2) GO TO 13
C**************************************************
C   READ ATTENUATION PARAMETER
C**************************************************
        READ(IN,FMT)(ALPA(I),I=1,NMAX)
        ALPA1=ALPA(1)
        ALPAB=ALPA(NMAX)
13      CONTINUE
        GO TO(31,32),IGEOP
31      READ(IN,100)FMT
C**************************************************
C   READ OUTPUT PARAMETERS FOR THE REGULARY SPACED GEOPHONE
C**************************************************
        READ(IN,FMT)IDECS,IDECL,IDEINC
        GO TO 33
32        READ(IN,100)FMT
C**************************************************
C   READ OUTPUT PARAMETERS FOR ARBITRARY SPACED GEOPHONE
C**************************************************
        READ(IN,FMT)(IGEO(I),I=1,NGEO)
33      NMED=NMAX-1
        TEND=DT*NITER
        N2=NMAX-2
```

```
C***********************************************
C   LIST INPUT PARAMETERS
C***********************************************
        CALL LIST(IMODE,IDENS,DT,DX,T,NITER)
C***********************************************
C***********************************************
C   CHECK THE STABILITY CONDITION.
C***********************************************
        RATIO=DT/DX
        VMAX=VEL(1)
        DO 66 I=2,NMAX
        DMAX=VEL(I)
66      IF(DMAX.GT.VMAX) VMAX=DMAX
        CRIT=VMAX*RATIO
        IF(CRIT.LT.1.0) GO TO 67
        WRITE(IOUT,200)
200      FORMAT(//,4X,'ERROR DETECTED.....UNSTABLE!')
        CALL EXIT
67      CONTINUE
C***********************************************
C   COMPUTES INPUT PRESSURE FUNCTION
C***********************************************
        CALL GAUSS(PULSE,DT,T,NPUL)
C***********************************************
C   COMPUTES CONSTANTS FOR THE BOUNDARY CONDITIONS
C***********************************************
        CALL BOUND(SUM1,CON1,CON2,CON3,CON4)
C***********************************************
C   COMPUTES ARRAY FOR THE ITERATION
C***********************************************
        CALL COEF(VEL,ALPA,BETA,DEN,A,C,R,E,F,G)
        GO TO(81,82,83),IMODE
81      CONTINUE
C***********************************************
C   PERFECT ELASTIC
C***********************************************
        CALL ELAST(VEL,DEN,PULSE,A,C,R,TEMP,IGEO)
        GO TO 987
82      CONTINUE
C***********************************************
C   VOIGT SOLID
C***********************************************
        CALL VOIGT(VEL,BETA,DEN,ATEMP,TEMP,E,F,G,C,R,A,PULSE,S,IGEO)
        GO TO 987
83     CONTINUE
C***********************************************
C   LINEAR WITH FREQUENCY
C***********************************************
        CALL LINER(VEL,ALPA,BETA,DEN,A,C,R,G,TEMP,ATEMP,PULSE,IGEO)
987     STOP
        END
```

COMMON BLOCKS

/INOUT/(+6)

```
      SUBROUTINE BOUND(SUM1,CON1,CON2,CON3,CON4)
C--------------SUBROUTINE BOUND------------------
C     THIS SUBROUTINE COMPUTES APPROPRIATE CONSTANTS FOR THE
C     FREE BOUNDARY AND RADIATION BOUNDARY CONDITIONS.
C     ARGUMENT DEFINITIONS:
C      VEL1: P-WAVE VELOCITY AT FREE SURFACE.
C      VELB: P-WAVE VELOCITY AT THE BOUNDARY OF A HALF-SPACE.
C      VELH: P-WAVE VELOCITY OF A HALF-SPACE.
C      DEN1,DENB,DENH: DENSITIES.
C      ALPA1,ALPAB,BETA1,BETAB: ATTENUATION TERMS.
C-------------------------------------------------
      COMMON /BK2/VEL1,VELB,VELH,DEN1,DENB,DENH,ALPA1,ALPAB,BETA1,
     1 BETAB
      COMMON /BK1/DT,DX,TEND,T
      COMMON /BK3/IMODE,IGEOP,IDENS,NPUL,NMAX,NMED,N2,NGEO
      IFLAG=1
      GO TO(10,11,12),IMODE
10    CONTINUE
      A=DENH*VELH/DT
      B=DENB*VELB*VELB/(2.0*DX)
      COM=A+3.0*B
      CON1=B/COM
      CON2=A/COM
      RETURN
11    CONTINUE
      SUM1=2.0*DT*VEL1*VEL1/BETA1
      A=DENH*VELH/DT
      B=DENB*VELB*VELB/(2.0*DX)
      C=BETAB*DENB/(2.0*DX*DT)
      COM=A+3.0*B+3.0*C
      CON1=A/COM
      CON2=B/COM
      CON3=C/COM
      RETURN
12    CONTINUE
      SUM1=2.0*DT/BETA1
      SUM2=1.0/3.0
      A=ALPAB/DT
      B=DENH*VELH/DT
      C=DENB*VELB*VELB/(2.0*DX)
      D=BETAB*C/DT
      Q=(1.0+A)*B+3.0*C+3.0*D
      CON1=C/Q
      CON2=D/Q
      CON3=B*(1.0+2.0*A)/Q
      CON4=A*B/Q
      RETURN
      END
```

COMMON BLOCKS

```
/BK2/(+12)
VEL1    +0     VELB    +1     VELH    +2     DEN1    +3     DENB    +4
DENH    +5     ALPA1   +6     ALPAB   +7     BETA1   +10    BETAB   +11
/BK1/(+4)
```

```
      SUBROUTINE COEF(VEL,ALPA,BETA,DEN,A,C,R,E,F,G)
C--------------SUBROUTINE COEF--------------------
C     THIS SUBROUTINE COMPUTES ARRAYS FOR THE ITERATION
C-----------------------------------------------
      DIMENSION VEL(1),ALPA(1),BETA(1),DEN(1),A(1),C(1),R(1),E(1)
      DIMENSION F(1),G(1)
      COMMON /BK1/DT,DX,TEND,T
      COMMON /BK3/IMODE,IGEOP,IDENS,NPUL,NMAX,NMED,N2,NGEO
      RAMDA=(DT*DT)/(DX*DX)
      RAMDA=RAMDA/2.0
      RAMDT=RAMDA/(2.0*DT)
      RCONST=RAMDA
      TCONST=RAMDT
      SUM1=DT+DT/DX
      DTDX=DT*DX
      DO 10 I=1,NMAX
10    VEL(I)=VEL(I)*VEL(I)
      GO TO(21,22,23),IMODE
C**************************************************
C   CHECK DENSITY INFORMATION
C**************************************************
21    GO TO(11,12),IDENS
C**************************************************
C   DENSITY IS CONSTANT
C**************************************************
11    DO 20 I=2,NMED
      VELN=(VEL(I)+VEL(I-1))*RAMDA
      VELP=(VEL(I)+VEL(I+1))*RAMDA
      A(I)=VELP
      C(I)=VELN
20    R(I)=2.0-VELN-VELP
      RETURN
C**************************************************
C   DENSITY INFORMATION IS GIVEN.
C   CHANGE THE VELOCITY INTO ELASTIC MODULUS
C**************************************************
12    DO 70 I=1,NMAX
70    VEL(I)=VEL(I)*DEN(I)
      DO 30 I=2,NMED
      RAMDA=RCONST/DEN(I)
      VELP=(VEL(I)+VEL(I+1))*RAMDA
      VELN=(VEL(I)+VEL(I-1))*RAMDA
      C(I)=VELN
      A(I)=VELP
30    R(I)=2.0-VELN-VELP
      RETURN
22    GO TO(31,32),IDENS
31    DO 35 I=2,NMED
      ATTN=(BETA(I)+BETA(I-1))*RAMDT
      ATTP=(BETA(I)+BETA(I+1))*RAMDT
      VELN=(VEL(I)+VEL(I-1))*RAMDA
      VELP=(VEL(I)+VEL(I+1))*RAMDA
      A(I)=ATTN
      C(I)=ATTP
      R(I)=1.0+ATTN+ATTP
      E(I)=VELP
```

```
        F(I)=2.0-(VELP+VELN)
 35     G(I)=VELN
        RETURN
 32     DO 60 I=1,NMAX
        VEL(I)=VEL(I)*DEN(I)
 60     BETA(I)=BETA(I)*DEN(I)
        DO 50 I=2,NMED
        RAMDA=RCONST/DEN(I)
        RAMDT=TCONST/DEN(I)
        ATTN=(BETA(I)+BETA(I-1))*RAMDT
        ATTP=(BETA(I)+BETA(I+1))*RAMDT
        VELN=(VEL(I)+VEL(I-1))*RAMDA
        VELP=(VEL(I)+VEL(I+1))*RAMDA
        A(I)=ATTN
        C(I)=ATTP
        B(I)=1.0+ATTN+ATTP
        E(I)=VELP
        F(I)=2.0-(VELP+VELN)
 50     G(I)=VELN
        RETURN
 23     GO TO(41,42),IDENS
 41     DO 40 I=1,NMED
        ATT=(ALPA(I)+ALPA(I+1))/2.0
        BTT=(BETA(I)+BETA(I+1))/2.0
        VVV=(VEL(I)+VEL(I+1))/2.0
        CCC=1.0+ATT/DT
        A(I)=SON1
        G(I)=VVV/(DX*CCC)
        C(I)=VVV*BTT/(CCC*DTDX)
        R(I)=ATT/(DT*CCC)
 40     CONTINUE
        RETURN
 42     DO 45 I=1,NMED
        ATT=(ALPA(I)+ALPA(I+1))/2.0
        BTT=(BETA(I)+BETA(I+1))/2.0
        VVV=(VEL(I)*DEN(I)+VEL(I+1)*DEN(I+1))/2.0
        CCC=1.0+ATT/DT
        A(I)=SON1/DEN(I)
        G(I)=VVV/(DX*CCC)
        C(I)=VVV*BTT/(CCC*DTDX)
        R(I)=ATT/(DT*CCC)
 45     CONTINUE
        RETURN
        END
```

COMMON BLOCKS

```
/BK1/(+4)
DT      +0      DX      +1      TEND    +2      T       +3
/BK3/(+10)
IMODE   +0      IGEOP   +1
IDENS   +2      NPUL    +3      NMAX    +4      NMED    +5      N2      +6
NGEO    +7
```

```
GAUSS    LN47TP.FOR       FORTRAN V.4A(230) /KI    1-OCT-75        22:25    PAGE 1

               SUBROUTINE GAUSS(F,DT,W,NMAX)
        C-------------SUBROUTINE GAUSS-----------------
        C      THIS SUBROUTINE COMPUTES TIME DEPENDENCE OF SOURCE FUNCTION
        C      USING GAUSSIAN FUNCTION(BODY FORCE).
        C      ARGUMENT DEFINITIONS.
        C         F: SOURCE TIME FUNCTION.
        C         DT: SAMPLING INTERVAL OF TIME.
        C         NMAX: THE LENGTH OF ARRAY F.
        C         W: PARAMETER DETERMING THE WIDTH OF THE SOURCE FUNCTION.
        C---------------------------------------------------------
               DIMENSION F(1)
               TST=W*1.6
               NMAX=IFIX(TST*2.0/DT)+1
               ALPA=2.0/(W*W)
               T=-TST-DT
               DO 10 I=1,NMAX
               T=T+DT
               TT=T*T*ALPA
        10     F(I)=EXP(-TT)
               RETURN
               END

SUBPROGRAMS CALLED

IFIX.
EXP.

SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*W        1         *T        2        *ALPA    3        *DT      4        *TST     5         .S0000 6
*I        7         *NMAX     10       *TT      11        F       12

TEMPORARIES

 .GAU16 13        .Q0000 14

GAUSS    [ NO ERRORS DETECTED ]
```

```
        SUBROUTINE MATRO(D,R,C,A,E,S,N1,N2,NMAX)
C--------------SUBROUTINE MATRO-------------------
C     THIS SUBROUTINE SOLVES TRI-DIAGONAL MATRIX.
C-------------------------------------------------
        DIMENSION D(1),R(1),C(1),A(1),S(1),E(1)
        E(N1)=D(N1)/R(N1)
        DO 10 I=N2,NMAX
10      E(I)=(D(I)+A(I)*E(I-1))/C(I)
        NMAX1=NMAX-1
        DO 20 I=NMAX1,N1,-1
20      E(I)=E(I)+E(I+1)*S(I)
        RETURN
        END
```

```
SUBPROGRAMS CALLED


SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*N1      1       E       2     S     3     D     4     A       5     .S0001 6
*NMAX1   7       .S0000 10     R    11     *I   12     *NMAX  13     C     14
*N2     15

TEMPORARIES


MATRO   [ NO ERRORS DETECTED ]
```

```
      SUBROUTINE ELAST(VEL,BETA,PULSE,A,C,R,TEMP,IGEO)
C--------------SUBROUTINE ELAST-------------------
C     THIS SUBROUTINE COMPUTES DISPLACEMENT DUE TO A PRESSURE SOURCE
C     ON A FREE SURFACE IN PERFECT ELASTIC MEDIUM.
C     ARGUMENT DEFINITIONS:
C      TEMP: DISPLACEMENT AT PRESENT( N-TH) TIME STEP
C      VEL: DISPLACEMENT AT (N-1)-TH TIME STEP
C      BETA: DISPLACEMENT AT (N-2)-TH TIME STEP
C------------------------------------------------
      DIMENSION VEL(1),BETA(1),A(1),C(1),R(1),TEMP(1),PULSE(1)
      DIMENSION IGEO(1)
      COMMON /BK1/DT,DX,TEND,T
      COMMON /BK3/IMODE,IGEOP,IDENS,NPUL,NMAX,NMED,N2,NGEO
      COMMON /BK4/SUM1,CON1,CON2,CON3,CON4
C************************************************
C   INITIALIZATION
C************************************************
      DO 40 I=1,NMAX
      VEL(I)=0.0
40    BETA(I)=0.0
      NST=3
      NP=1
      TIME=0.0
888   IF(NP.GT.NPUL) GO TO 42
      TEMP(2)=PULSE(NP)
      GO TO 41
42    NST=2
C************************************************
C   COMPUTES CURRENT DISPLACEMENT
C************************************************
41    DO 50 I=NST,NMED
50    TEMP(I)=A(I)*VEL(I+1)+R(I)*VEL(I)+C(I)*VEL(I-1)-BETA(I)
C************************************************
C   FREE BOUNDARY CONDITION
C************************************************
      TEMP(1)=TEMP(3)
C************************************************
C   RADIATION BOUNDARY CONDITION
C************************************************
      TEMP(NMAX)=CON1*(4.0*TEMP(NMED)-TEMP(N2))+CON2*VEL(NMAX)
C************************************************
C   CHANGING THE ARRAY FOR THE NEXT TIME STEP
C************************************************
      DO 60 I=1,NMAX
      BETA(I)=VEL(I)
60    VEL(I)=TEMP(I)
C************************************************
C   LIST THE DISPLACEMENT
C************************************************
      CALL LISTP(TEMP,TIME,IGEO,IGEOP,NGEO)
      IF(TIME.GT.TEND) RETURN
      TIME=TIME+DT
      NP=NP+1
      GO TO 888
      END
```

```
      SUBROUTINE VOIGT(VEL,BETA,COM,ATEMP,TEMP,E,F,G,C,R,A,PULSE,
     1 S,IGEO)
C--------------SUBROUTINE VOIGT------------------
C     THIS SUBROUTINE COMPUTES DISPLACEMENT DUE TO A PRESSURE SOURCE ON
C     FREE SURFACE IN VOIGT SOLID.
C     ARGUMENT DEFINITIONS:
C       TEMP: DISPLACEMENT AT PRESENT( N-TH) TIME STEP
C       VEL: DISPLACEMENT AT (N-1)-TH TIME STEP
C       BETA: DISPLACEMENT AT (N-2)-TH TIME STEP
C-----------------------------------------------
      DIMENSION VEL(1),BETA(1),ATEMP(1),TEMP(1),COM(1),E(1),F(1)
      DIMENSION G(1),C(1),R(1),A(1),PULSE(1)
      DIMENSION S(1),IGEO(1)
      COMMON /BK3/IMODE,IGEOP,IDENS,NPUL,NMAX,NMED,N2,NGEO
      COMMON /BK4/SUM1,CON1,CON2,CON3,CON4
      COMMON /BK1/DT,DX,TEND,T
C***********************************************
C   INITIALIZATION
C***********************************************
      DO 10 I=1,NMAX
      VEL(I)=0.0
10    BETA(I)=0.0
      TERM=0.0
      NP=1
      TIME=DT
      ING=1
      NST1=3
      NST2=4
C***********************************************
C   COMPUTES THE COEFFICIENT OF THE TRI-DIAGONAL MATRIX
C***********************************************
      ATEMP(3)=C(3)/R(3)
      DO 50 I=NST2,NMED
      COM(I)=R(I)-A(I)*ATEMP(I-1)
50    ATEMP(I)=C(I)/COM(I)
888   IF(NP.GT.NPUL) GO TO 40
      DISP=PULSE(NP)
      TEMP(2)=DISP
777    QTERM=TERM+SUM1*(VEL(3)-VEL(1))
      IF(ING.EQ.2) DISP=QTERM
      DO 21 I=NST1,NMED
      S(I)=E(I)*VEL(I+1)+F(I)*VEL(I)+G(I)*VEL(I-1)-C(I)*BETA(I+1)-
     1 (2.0-R(I))*BETA(I)-A(I)*BETA(I-1)
21    CONTINUE
      S(NST1)=S(NST1)+DISP*A(NST1)
C***********************************************
C   SOLVE THE TRI-DIAGONAL MATRIX
C   COMPUTES CURRENT DISPLACEMENT
C***********************************************
      CALL MATRO(S,R,COM,A,TEMP,ATEMP,NST1,NST2,NMED)
C***********************************************
C   FREE BOUNDARY CONDITION
C***********************************************
      TEMP(1)=TEMP(3)+QTERM
C***********************************************
C   RADIATION BOUNDARY CONDITION
```

```
C***************************************************
      TEMP(NMAX)=CON2*(4.0*TEMP(NMED)-TEMP(N2))+CON3*(4.0*TEMP(NMED)-
     1 TEMP(N2)+3.0*VEL(NMAX)-4.0*VEL(NMED)+VEL(N2))+CON1*VEL(NMAX)
C***************************************************
C   CHANGING THE ARRAY FOR THE NEXT TIME STEP
C***************************************************
      DO 30 I=1,NMAX
      BETA(I)=VEL(I)
30    VEL(I)=TEMP(I)
      TERM=BETA(1)-BETA(3)
C***************************************************
C   LIST THE DISPLACEMENT
C***************************************************
      CALL LISTP(TEMP,TIME,IGEO,IGEOP,NGEO)
      IF(TIME.GT.TEND) RETURN
      TIME=TIME+DT
      NP=NP+1
      IF(ING.EQ.2) GO TO 777
      GO TO 888
40    NST1=2
      NST2=3
      ING=2
      CXT=C(2)+A(2)
      ATEMP(2)=CXT/R(2)
      DO 20 I=3,NMED
      COM(I)=R(I)-A(I)*ATEMP(I-1)
20    ATEMP(I)=C(I)/COM(I)
      GO TO 777
      END
```

COMMON BLOCKS

```
/BK3/(+10)
IMODE   +0     IGEOP   +1     IDENS   +2     NPUL   +3     NMAX   +4
NMED    +5     N2      +6     NGEO    +7
/BK4/(+5)
SUM1    +0     CON1    +1     CON2    +2
CON3    +3     CON4    +4
/BK1/(+4)
DT      +0     DX      +1     TEND    +2     T      +3
```

SUBPROGRAMS CALLED

```
MATRU   LISTP
```

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
*NST1   1    *QTERM  2    IGEO    3    COM     4    VEL     5    BETA    6
*DISP   7    E       10   S       11   *CXT    12   G       13   PULSE   14
*ING    15   .S0004  16   .S0003  17   A       20   .S0002  21   .S0001  22
*TERM   23   .S0000  24   *NP     25   ATEMP   26   *TIME   27   R       30
 TEMP   31   *NST2   32   *I      33   F       34   C       35
```

```
      SUBROUTINE LINER(SECD,FIRD,SECPP,FIRPP,A,C,R,G,TEMPD,TEMPP,
     1 PULSE,IGEO)
C--------------SUBROUTINE LINER------------------
C     THIS SUBROUTINE COMPUTES DISPLACEMENT DUE TO A PRESSURE SOURCE
C     ON FREE SURFACE IN A SOLID WHOSE ATTENUATION VARIES LINEARY
C     WITH FREQUENCY.
C     ARGUMENT DEFINITIONS:
C      TEMPD: DISPLACEMENT AT PRESENT (N-TH) TIME STEP
C      SECD: DISPLACEMENT AT (N-1)-TH TIME STEP
C      FIRD: DISPLACEMENT AT (N-2)-TH TIME STEP
C      TEMPP: PRESSURE AT N-TH TIME STEP
C      SECPP: PRESSURE AT (N-1)-TH TIME STEP
C      FIRPP: PRESSURE AT (N-2)-TH TIME STEP
C-----------------------------------------------
      DIMENSION SECD(1),FIRD(1),SECPP(1),FIRPP(1),A(1),C(1),R(1),G(1)
       DIMENSION TEMPD(1),TEMPP(1),PULSE(1),IGEO(1)
       COMMON /BK3/IMODE,IGEOP,IDENS,NPUL,NMAX,NMED,N2,NGEO
       COMMON /BK1/DT,DX,TEND,T
       COMMON /BK2/VEL1,VELB,VELH,DEN1,DENB,DENH,ALPA1,ALPAB,BETA1,
     1 BETAB
       COMMON /BK4/SUM1,CON1,CON2,CON3,CON4
C***********************************************
C   INITIALIZATION
C***********************************************
       DO 10 I=1,NMAX
       SECD(I)=0.0
       FIRD(I)=0.0
       SECPP(I)=0.0
10     FIRPP(I)=0.0
       NP=1
       TIME=DT
       NST=3
       NSTP=2
999    IF(NP.GT.NPUL) GO TO 44
       TEMPD(2)=PULSE(NP)
       GO TO 56
44     NST=2
56     CONTINUE
C***********************************************
C   COMPUTES CURRENT DISPLACEMENT
C***********************************************
       DO 40 I=NST,NMED
40     TEMPD(I)=A(I)*(SECPP(I)-SECPP(I-1))+SECD(I)+SECD(I)-FIRD(I)
C***********************************************
C   RADIATION BOUNDARY CONDITION
C***********************************************
       TEMPD(NMAX)=CON1*(4.0*TEMPD(NMED)-TEMPD(N2))+CON2*(SECD(N2)-
     1 4.0*SECD(NMED)+4.0*TEMPD(NMED)+3.0*SECD(NMAX)-TEMPD(N2))+
     2 CON3*SECD(NMAX)-CON4*FIRD(NMAX)
C***********************************************
C   FREE BOUNDARY CONDITION
C***********************************************
       TEMPD(1)=TEMPD(3)
C***********************************************
C   COMPUTES CURRENT PRESSURE
C***********************************************
```

```
66      CONTINUE
        DO 60 I=NSTP,NMED
60      TEMPP(I)=G(I)*(TEMPD(I+1)-TEMPD(I))+C(I)*(TEMPD(I+1)-TEMPD(I)+
     1 SECD(I)-SECD(I+1))+R(I)*SECPP(I)
        TEMPP(1)=-TEMPP(2)
C**********************************************
C   CHANGING THE ARRAY FOR THE NEXT TIME STEP
C**********************************************
        DO 55 I=1,NMAX
        FIRD(I)=SECD(I)
        SECD(I)=TEMPD(I)
         FIRPP(I)=SECPP(I)
        SECPP(I)=TEMPP(I)
55      CONTINUE
C**********************************************
C   LIST THE DISPLACEMENT
C**********************************************
        WRITE(12,300)TEMPD(12),TEMPD(22),TEMPD(42),TEMPD(82),TEMPD(162)
300     FORMAT(5E16.6)
        WRITE(3,400)TIME,TEMPD(13),TEMPD(123),TEMPD(N2)
400     FORMAT(2X,F10.5,3E20.8)
        IF(TIME.GT.TEND) RETURN
        NP=NP+1
        TIME=TIME+DT
        GO TO 999
        END
```

```
COMMON BLOCKS

/BK3/(+10)
IMODE   +0      IGEOP   +1      IDENS   +2      NPUL    +3      NMAX    +4
NMED    +5      N2      +6      NGEO    +7
/BK1/(+4)
DT      +0      DX      +1      TEND    +2
T       +3
/BK2/(+12)
VEL1    +0      VELB    +1      VELH    +2      DEN1    +3      DENB    +4
DENH    +5      ALPA1   +6      ALPAB   +7      BETA1   +10     BETAB   +11
/BK4/(+5)
SUM1    +0
CON1    +1      CON2    +2      CON3    +3      CON4    +4

SUBPROGRAMS CALLED


SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

 FIRD   1       %IGEO           FIRPP   2       SECD    3       SECPP   4       *NSTP   5
 G      6        PULSE  7       .S0003  10      *NST    11      A       12      .S0002  13
.S0001  14       .S0000 15      *NP     16       TEMPP  17      *TIME   20      R       21
*I      22       C      23       TEMPD  24
```

```
      SUBROUTINE LIST(IMODE,IDENS,DT,DX,T,NITER)
C----------------SUBROUTINE LIST-----------------------------------------
C      THIS SUBROUTINE LIST PARAMETERS OF THE MODEL.
C-----------------------------------------------------------------------
      COMMON /INOUT/IN,IOUT,ITAPE,IDECL,IDECS,IDEINC
      GO TO(11,12,13),IMODE
11    WRITE(IOUT,100)
100   FORMAT(//,4X,'PERFECT ELASTIC MEDIUM')
      GO TO 14
12    WRITE(IOUT,200)
200   FORMAT(//,4X,'VOIGT SOLID')
      GO TO 14
13    WRITE(IOUT,300)
300   FORMAT(//,4X,'LINEAR WITH FREQUENCY')
14    GO TO(15,16),IDENS
15    WRITE(IOUT,400)
400   FORMAT(/,4X,'DENSITY IS ASSUMED TO BE CONSTANT')
      GO TO 23
16    WRITE(IOUT,500)
500   FORMAT(/,4X,'DENSITY INFORMATION IS GIVEN')
23    CONTINUE
      WRITE(IOUT,600)DT,DX,T,NITER
600   FORMAT(///,8X,'TIME     INCREMENT=',F10.5,/,
     1 8X,'DISTANCE INCREMENT=',F10.5,/,
     2 8X,'TEMPORAL SOURCE WIDTH=',F10.5,/,
     3 8X,'NUMBER OF ITERATION=',I5,/,
     4 8X,'TIME',4X,'X-INDEX',10X,'DISPLACEMENT'
      RETURN
      END
```

```
      SUBROUTINE LISTP(A,TIME,IGEO,IGEOP,NGEO)
c-----------------------------------------------------------------------
C     THIS SUBROUTINE LIST THE OUTPUT GEOPHONE DISPLACEMENT
C     IN LINE-PRINTER AND MAGNETIC TAPE.
C-----------------------------------------------------------------------
      DIMENSION A(1),IGEO(1)
      COMMON /INOUT/IN,IOUT,ITAPE,IDECL,IDEINC
      GO TO(11,12),IGEOP
11    DO 10 I=IDECS,IDECL,IDEINC
      WRITE(IOUT,400)TIME,I,A(I)
10    WRITE(ITAPE,400)TIME,I,A(I)
400   FORMAT(2X,F10.5,2X,I6,4X,E20.8)
      RETURN
12    INN=1
40    IT=IGEO(INN)
      WRITE(IOUT,400)TIME,IT,A(IT)
      WRITE(ITAPE,400)TIME,IT,A(IT)
      IF(INN.GT.NGEO) RETURN
      INN=INN+1
      GO TO 40
      END
```

APPENDIX C

Computer Program and User's Manual for Two-dimensional Finite Difference Scheme

This computer program computes vertical and horizontal displacements of an elastic wave in a two-dimensional orthogonal cartesian coordinate system. In making a general computer program for an two-dimensional, inhomogeneous wave equation by a finite difference scheme, large core-memory is required, because we must store all elastic constants in addition to the 4 working arrays at each grid point. So we selected three types of model to reduce this large memory requirement. This program can handle the following three types of model:

(1) One-layered elastic half-space.
(2) Vertical fault in an elastic half-space.
(5) Localized arbitrary shaped inhomogeniety extended in the horizontal direction in an elastic half-space.

The geometry of the vertical fault and localized inhomogeneity model is left-justified, which means the discontinuity of half-space starts from the left and ends at right side of model(see Figure C-2 and Figure C-3).

Throughout the computer program, we used the same sampling interval in x- and y-direction. Also we used an imaginary grid line at y=-DY to solve a free boundary condition.

The pre-assigned input-output logical units are:
2= Input logical unit.
3= Output logical unit for line printer.
12= Output logical unit except line printer.
The format of logical unit 12 is the same as logical unit 3 except that there is no heading.

Figure C-1 shows how to prepare input cards to execute this program. Each BLK_i contains parameters of the model and each BLK_i will be read by a format described previous FMT (format) card. The definitions of each block are written below.

BLK1

The BLK1 contains DT, DX, T, TX, ALPA1, BETA1, DEN1, DEN2, IGEO, NGEO sequentially.

1. DT: Sampling interval in time.
2. DX: Sampling interval of distance.
3. T: Parameter determine the width of temporal source function (see source function).
4. TX: Parameter which controls the extent of the source region (see source function).

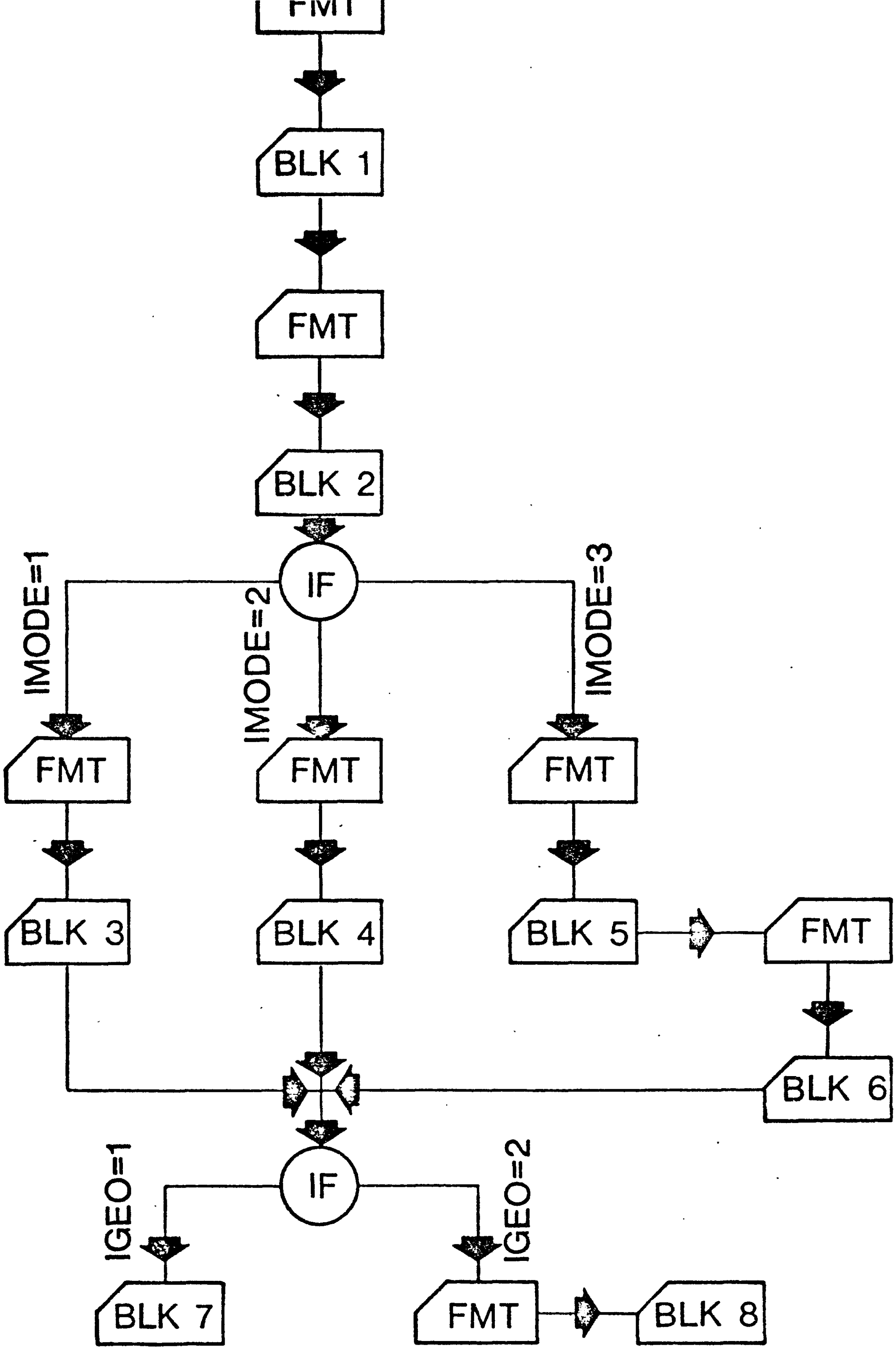


Figure C-1. Preparation of input cards for two-dimensional finite difference schemes.

3. T: Parameter which determines the width of temporal source function (see Source function).
4. TX: Parameter which controls the extent of source region (see Source function).
5. ALPA1: P-wave velocity of the upper medium.
6. BETA1: S-wave velocity of the upper medium.
7. DEN1: Density of the upper medium.
8. IGEO: Control number for the output.
   IGEO=1, Either vertical or horizontal index of the output arrays is fixed and output arrays are regularly spaced.
   IGEO=2, Output arrays are irregularly spaced.
9. NGEO: Number of the output geophone.

BLK2

1. NITER: Number of iteration.
2. IMODE: Control number for the model type:
   IMODE=1, One-layered half-space.
   IMODE=2, Vertical fault model.
   IMODE=3, Localized inhomogeneity model.
3. ISYM: Control number for the symmetry of the model.
   ISYM=1, Symmetrical model.
   ISYM=2, Asymmetrical model.
4. Imax: Horizontal dimension of the model.
5. JMAX: Vertical dimension of the model.
6. IST: Parameter for the source region (see source function).
7. IFN: Parameter for the source region.
8. JST: Parameter for the source region.
9. JFN: Parameter for the source region.

BLK3

1. ALPA2: P-wave velocity of the lower medium.
2. BETA2: S-wave velocity of the lower medium.
3. DEN2: Density of the lower medium.
4. JINT: Vertical index number of the horizontal interface (see Figure C-2).

BLK4

1. Alpa2: P-wave velocity of the fault medium.
2. BETA2: S-wave velocity of the fault medium.
3. DEN2: Density of the fault medium.
4. JINT: Vertical index number of the horizontal Interface (see Figure C-2).
5. IINT: Horizontal index number of the vertical interface (see Figure C-2).

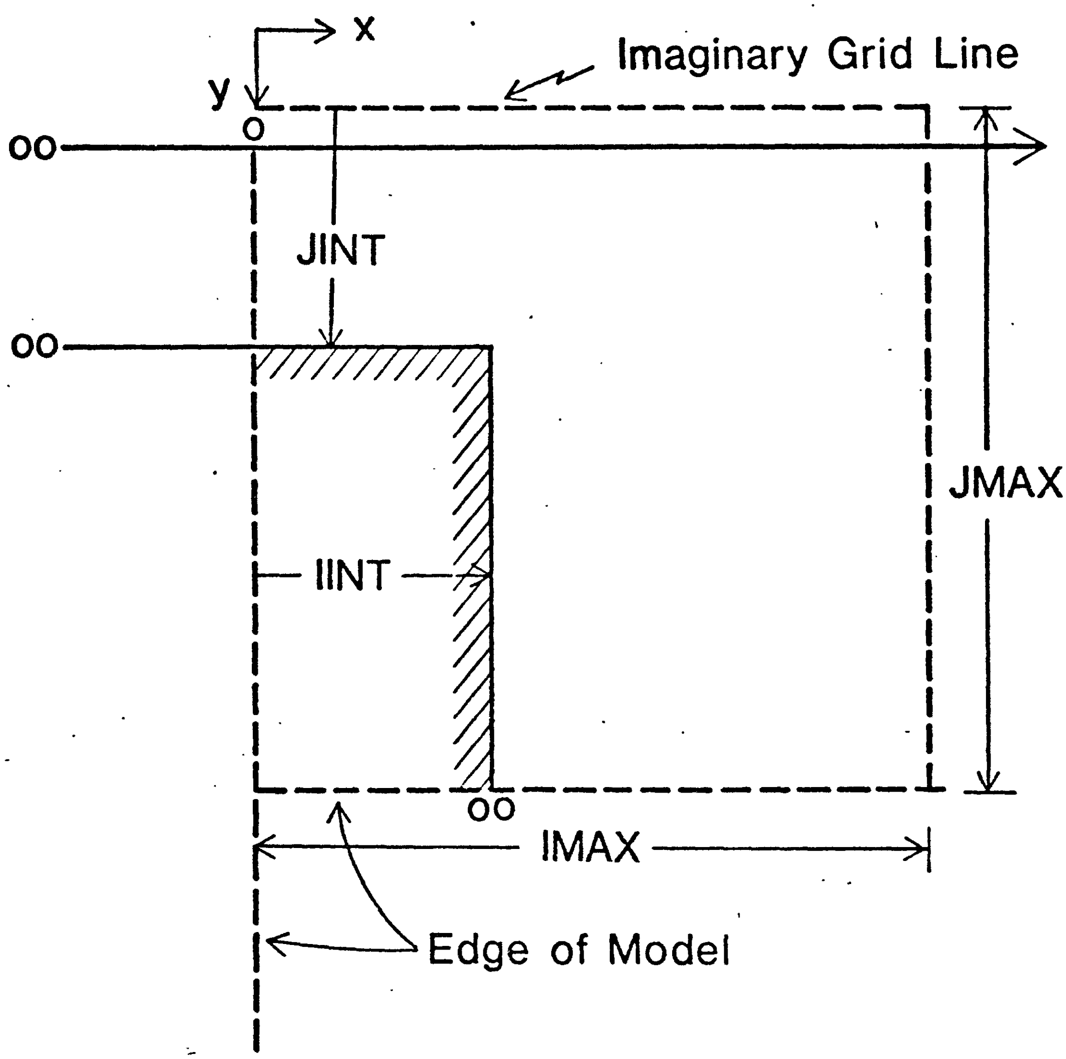


Figure C-2. Definition of parameters for vertical fault model.

BLK5

1. JTOP: Vertical index which specifies the upper boundary of the localized inhomogeneity model.
2. JBOT: Vertical index number which specifies the lower boundary of the inhomogeneous region.
3. IRIG: Horizontal index which specifies the right boundary of the inhomogeneous region.

For further illustration, see Figure C-3.

BLK6

1. ALPA: P-wave velocity inside the inhomogeneous region.
2. BETA: S-wave velocity inside the inhomogeneous region.
3. DEN: Density inside the inhomogeneous region.

These parameters will be read as:

((ALPA(I,J),BETA(I,J),DEN(I,J),I=1,ITOTX),J=1,JTOTY).

BLK7

1. Idecs: The starting x- or y-index number for the ir- regularly spaced geophones.
2. IDECL: The ending x- or y-index number for the regularly spaced geophone.
   (IDECS IDECL)
3. IDEINC: Index interval between output array.
4. ICONS: Fixed index number for the output.
5. ITERM: Control number for the fixed index of the output.
   ITERM=1, Horizontal index is fixed.
   ITERM=2, Vertical index is fixed.

BLK8

This block contains the information about irregularly spaced output.

1. ICOUL: Array which contains the horizontal index number for the irregularly spaced output.
2. JROW: Array which contains the vertical index number for the irregularly spaced output.

These arrays will be read as

(ICOUL(I),JROW(I),I=1,NGEO)

Source function

Before reading this section, the users are recommended to read Aboudi's paper (1971) for clarity.

As an input source function, we used a body force. Let the body force be defined as

$$F_i = f_i(x,y)\, g(t)$$

where F_i is body force. It is obvious that $f_i(x,y)$ is the

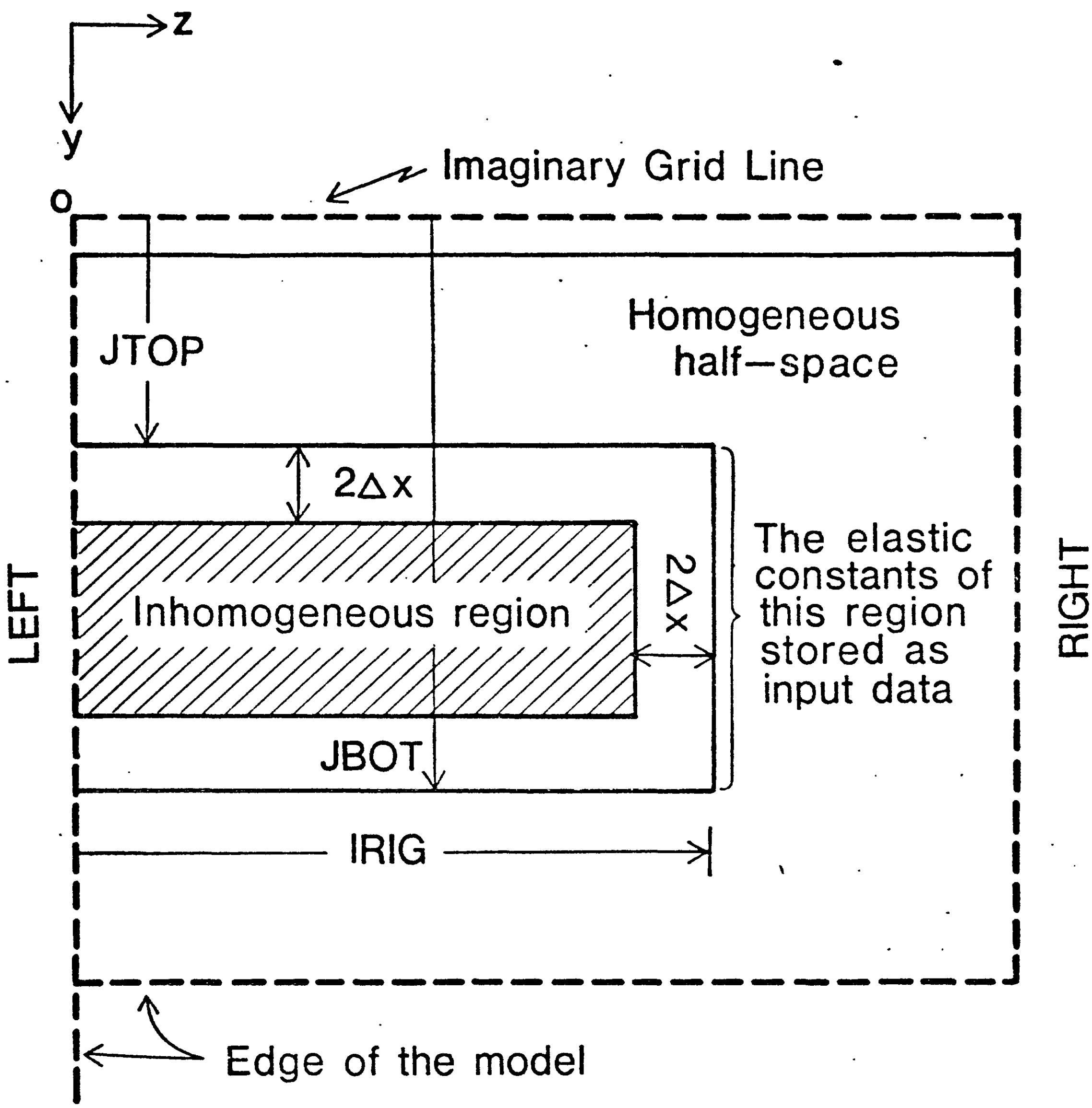


Figure C-3. Definition of parameters for localized inhomogeneity model.

function determines the spacial dependence of body force and $g(t)$ is the time dependence of the source function. We used a symmetrical source function, so the spacial extent and shape of $f_x(x,y)$ in x- and y-direction are the same.

Let the center of the source be located at (nDX,mDX) and the spacial dependence is shown in the Figure C-4. The spacial extent of this function is 4DX. But at x=(n±4)DX, the function values are zero. So we defined spacial extent TX as shown in Figure C-4. The smallest possible source extent is TX=DX, whixh is the cloest approximation of the point source in this program. For this example, input parameters for the source region are:

TX=3DX,
IST=n-3,
IFN=n+3,
JST=m-3,
JFN=m+3.

In this program there are limitation for the source region:

JST or IST $\geq$ 3,
JFN or IFN $\leq$ JMAX-2 or Imax-2.

For the temporal dependence of the source function, we used the following function:

$$g(t) = e^{-\alpha t^2}.$$

The width of the temporal source width T is defined as the following relation

$$T = \sqrt{\frac{2}{\alpha}}.$$

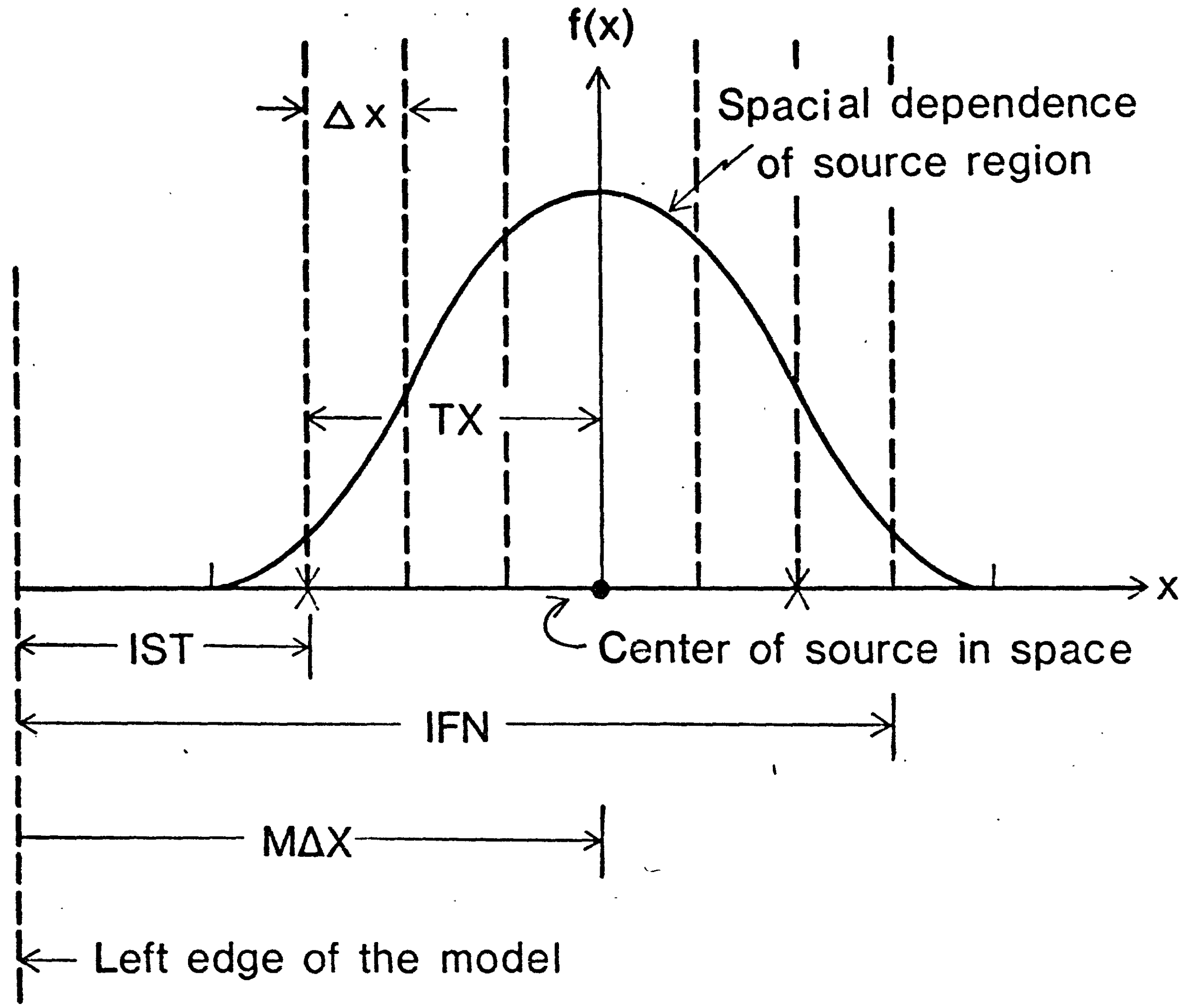


Figure C-4. Description of source function.

```
C********************************************************
C********************************************************
C********************************************************
C********************************************************
C********************************************************
C********************************************************
C********************************************************
C********************************************************
C       THIS FORTRAN 4 COMPUTER PROGRAM CALCULATES VERTICAL AND
C       HORIZONTAL DISPLACEMENTS OF A TWO-DIMENSIONAL ELASTIC MODEL
C       IN AN ORTHOGONAL CARTESIAN COOREDINATE BY SOLVING AN INHOMOGENEOUS
C       ELASTIC WAVE EQUATION WITH FINITE DIFFERENCE SCHEME.
C       THIS PROGRAM CAN HANDLE THE FOLLOWING THREE MODEL TYPE;
C           (1).. ONE-LAYERED ELASTIC HALF-SPACE.
C           (2).. VERTICAL FAULT IN AN ELASTIC HALF-SPACE.
C           (3).. LOCALIZED ARBITRARY SHAPED INHOMOGENEITY EXTENDED IN
C                   HORIZONTAL DIRECTION IN AN ELASTIC HALF-SPACE.
C       THE SAME SAMPLING INTERVAL IN HORIZONTAL AND VERTICAL DIRECTION IS USED.
C       THE SOURCE FUNCTION IS EXPLOSIVE SOURCE. THE SOURCE IS NOT
C       A POINT SOURCE IN A MATHEMATICAL SENSE, BUT RATHER VOLUMETRIC SOURCE.
C       MAKING THE SAMPLING INTERVAL SMALLER, WE CAN BETTER APPOXIMATE POINT
C       SOURCE PROBLEM. THIS SOURCE CAN BE PLACED ANY POINT IN AN
C       ELASTIC MEDIUM EXCEPT THE OUTER BOUNDARY OF THE MODEL.
C       THE INPUT PARAMETERS CAN BE TRANSFERRED FROM A CARD READER, DISK FILE
C       OR TELETYPE BY ASSINGING THE DEVICE'S NAME. THE OUTPUT IS PRINTED
C       IN LINE PRINTER AND STORED IN A MAGNETIC TAPE FOR THE FURTHER PROCESS.
C       LOGICAL UNITS OF THE INPUT-OUTPUT ARE:
C         2= INPUT
C         3= LINE PRINT OUTPUT.
C         8= OUTPUT IN A MAGNETIC TAPE.
C       ARGUMENT DEFINITIONS.
C       ALPA1,BETA1,DEN1: LONGITUDINAL VELOCITY, SHEAR VELOCITY AND
C                         DENSITY OF THE UPPER MEDIUM RESPECTIVELY.
C       ALPA2,BETA2,DEN2: LONGITUDINAL VELOCITY, H
C       ALPA2,BETA2,DEN2: LONGITUDINAL VELOCITY,SHEAR VELOCITY AND DENSITY
C                           OF THE LOWER OR FAULT MEDIUM RESPECTIVELY.
C       ALPA,BETA,DEN: LONGITUDINAL VELOCITY, SHEAR VELOCITY AND DENSITY
C                     OF THE INHOMOGENEITY RESPECTIVELY.
C       NITER: NUMBER OF ITERATION IN TIME.
C       IMAX,JMAX: HORIZONTAL AND VERTICAL DIMENSION OF THE MODEL RESPECTIVELY.
C       IST: THE STARTING HORIZONTAL INDEX OF THE SOURCE REGION..NON-ZERO VALUE OF
C       THE SOURCE FUNCTION BEGINS AT HORIZONTAL-INDEX=IST.
C       IFN: THE ENDING HORIZONTAL INDEX OF THE SOURCE REGION.
C       JST: THE STARTING VERTICAL INDEX OF THE SOURCE REGION.
C       JFN: THE ENDING VERTICAL INDEX OF THE SOURCE REGION.
C       JINT: VERTICAL INDEX OF THE VERTI
C       JINT: VERTICAL INDEX OF THE HORIZONTAL INTERFACE.
C       IINT: HORIZONTAL INDEX OF THE VERTICAL INTERFACE.
C       JTOP: UPPER VERTICAL INDEX OF THE INHOMOGENEOUS REGION.
C       JBOT: LOWER VERTICAL INDEX OF THE INHOMOGENEOUS REGION.
C       IRIG: RIGHT HORIZONTAL INDEX OF THE INHOMOGENEOUS REGION.
C           IMODE: CONTROL NUMBER FOR THE MODEL TYPE;
C              =1: ONE-LAYERED HALF-SPACE MODEL.
C              =2: VERTICAL FAULT MODEL.
C              3=: LOCALIZED INHOMOGENEOUS ARBITRARY SHAPED MODE.
C           ISYM: CONTROL NUMBER FOR THE SYMMETRY OF THE MODEL.
```

```
C              =1: SYMMETRICAL MODEL.
C              =2: ASYMMETRICAL MODEL.
C            IGEO: CONTROL NUMBER FOR THE OUT GEOPHONE.
C              =1: EITHER VERTICAL OR HORIZONTAL INDEX IS FIXED AND OUTPUT SI
C                  PEGULARY SPACED.
C              =2: RANDOM( OR SELECTED) IRREGULARY SPACED OUTPUT.
C            ITERM: CONTROL NUMBER FOR THE FIXED INDEX OF THE OUPPUT.
C               =1: HORIZONTAL INDEX IS FIXED.
C               =2: VERTICAL INDEX IS FIXED.
C*****************************************************
C*****************************************************
      DIMENSION A(93,72),B(93,72),C(93,72),D(93,72)
      DIMENSION FX(5,5),FY(5,5),ALPA(25,11),BETA(25,11),DEN(25,11)
      DIMENSION ICOUL(50),JROW(50)
      DIMENSION PULSE(100),GX(10),GY(10),FMT(16)
      COMMON /GEOPH/IGEO,ITERM,IDECS,IDECL,NGEO,IDEINC,ICONS
      COMMON /INOUT/IN,IOUT,ITAPE
      COMMON /BK1/IDX,JDY,SON1,SON2,DT,NPUL,ISYM,ISET,IST,IFN,
     1 JST,JFN,NITER
      COMMON /BK2/CON1,CON2,CON3,PON1,PON2,PON3,TON1,TON2,TON3
      COMMON /BK3/C1,C2,C3,S1,S2,S3,T1,T2,T3
      COMMON /BK4/DY1,DY2,DY3,CX1,CX2,CX3,SX1,SX2,SX3,TX1,TX2,
     1 TX3,DX1,DX2,DX3
      DATA IN,IOUT,ITAPE/2,3,12/
      READ(IN,100)FMT
100    FORMAT(16A5)
C-----------------------------------------------------
C   RAED COMMON PARAMETERS FOR ALL MODELS.
C-----------------------------------------------------
      READ(IN,FMT)DT,DX,ALPA1,BETA1,DEN1,I,TX,IGEO,NGEO
      READ(IN,100)FMT
      READ(IN,FMT)NITER,IMODE,ISYM,IMAX,JMAX,IST,IFN,JST,JFN
      GO TO(11,12,13),IMODE
11    READ(IN,100)FMT
C-----------------------------------------------------
C   READ PARAMETERS FOR LOWER MEDIUM
C-----------------------------------------------------
      READ(IN,FMT)ALPA2,BETA2,DEN2,JINT
      GO TO 29
12    READ(IN,100)FMT
C-----------------------------------------------------
C   READ PARAMETERS FOR FAULT MEDIUM.
C-----------------------------------------------------
      READ(IN,FMT)ALPA2,BETA2,DEN2,JINT,IINT
      GO TO 29
13    READ(IN,100)FMT
C-----------------------------------------------------
C   READ PARAMETERS FOR THE INHOMOGENEOUS MEDIUM.
C-----------------------------------------------------
      READ(IN,FMT)JTOP,JBOT,IRIG
      ITOTX=IRIG
      JTOTY=JBOT-JTOP+1
      READ(IN,100)FMT
      DO 30 J=1,JTOTY
30    READ(IN,FMT)(ALPA(I,J),BETA(I,J),DEN(I,J),I=1,ITOTX)
C-----------------------------------------------------
```

```
C   RECOGNIZE THE OUTPUT GEOPHONE PATTERN.
C-----------------------------------------------------
29      GO TO(41,42),IGEO
41      READ(IN,300)IDECS,IDECL,IDEINC,ICONS,ITERM
300     FORMAT(5I5)
        GO TO 19
42      READ(IN,100)FMT
        READ(IN,FMT)(ICOUL(I),JROW(I),I=1,NGEO)
C-----------------------------------------------------
C     CHECK THE STABILITY CONDITION.
C-----------------------------------------------------
19      GO TO(14,14,15),IMODE
14      VMAX1=ALPA1*ALPA1+BETA1*BETA1
        VMAX2=ALPA2*ALPA2+BETA2*BETA2
        VMAX=AMAX1(VMAX1,VMAX2)
        GO TO 18
15        VMAX=ALPA1*ALPA1+BETA1*BETA1
        DO 20 I=1,ITOTX
        DO 20 J=1,JTOTY
        DMAX=ALPA(I,J)*ALPA(I,J)+BETA(I,J)*BETA(I,J)
20      IF(DMAX.GT.AMAX) VMAX=DMAX
18      RATIO1=DT/DZ
        RATIO2=1.0/SQRT(VMAX)
        IF(RATIO1.LT.RATIO2) GO TO 17
        WRITE(IOUT,200)
200      FORMAT(//,4X,'ERROR DETECTED..UNSTABLE')
        CALL EXIT
C-----------------------------------------------------
C   COMPUTATION OF THE SOURCE FUNCTION.
C-----------------------------------------------------
17      CALL GAUSS(PULSE,DT,T,NPUL)
        CALL SPAC(GX,GY,TX,DX,XSPA)
        ICOUNT=IFN-IST+1
        JCOUNT=JFN-JST+1
        DO 22 I=1,ICOUNT
       ITEM=I+1
        DO 22 J=1,JCOUNT
       JTEM=J+1
       FX(I,J)=GY(ITEM)*GX(JTEM)
22     FY(I,J)=GX(ITEM)*GY(JTEM)
C-----------------------------------------------------
C   INITIALIZATION OF THE DISPLACEMENTS.
C-----------------------------------------------------
        DO 10 I=1,IMAX
        DO 10 J=1,JMAX
        A(I,J)=0.0
        B(I,J)=0.0
        C(I,J)=0.0
10      D(I,J)=0.0
        IF(ISYM.EQ.1) ISET=IST+IFN-1
        CALL LIST(IMODE,ALPA1,ALPA2,BETA1,BETA2,DEN1,DEN2,DX,TX,T)
C-----------------------------------------------------
C     COMPUTES SOME CONSTANTS.
C-----------------------------------------------------
        RAMDA=DT*DT/(DX*DX)
        RDT=RAMDA/4.0
```

```
            CON1=ALPA1*ALPA1*RAMDA
            CON2=(ALPA1*ALPA1-BETA1*BETA1)*RAMDA/4.0
            CON3=BETA1*BETA1*RAMDA
            SON2=(ALPA1*ALPA1-2.0*BETA1*BETA1)/(ALPA1*ALPA1)
            IDX=IMAX-1
            JDY=JMAX-1
            IF(IMODE.EQ.3) GO TO 32
          ALPAB=(ALPA1+ALPA2)/2.0
          BETAB=(BETA1+BETA2)/2.0
            DENB=(DEN1+DEN2)/2.0
            PON1=ALPAB*ALPAB*RAMDA
            PON2=(ALPAB*ALPAB-BETAB*BETAB)*RDT
            PON3=BETAB*BETAB*RAMDA
            TON1=ALPA2*ALPA2*RAMDA
            TON2=(ALPA2*ALPA2-BETA2*BETA2)*RAMDA/4.0
            TON3=BETA2*BETA2*RAMDA
            CALL BOUND(BETA1,BETA1,BETAB,DEN1,DEN1,DENB,RDT,C1)
            CALL BOUND(ALPA1,ALPA1,ALPAB,DEN1,DEN1,DENB,RDT,C2)
            CALL BOUND(BETA1,BETAB,BETA2,DEN1,DENB,DEN2,RDT,S1)
            CALL BOUND(ALPA1,ALPAB,ALPA2,DEN1,DENB,DEN2,RDT,S2)
            CALL BOUND(BETAB,BETA2,BETA2,DENB,DEN2,DEN2,RDT,T1)
            CALL BOUND(ALPAB,ALPA2,ALPA2,DENB,DEN2,DEN2,RDT,T2)
            S3=S2-2.0*S1
            C3=C2-2.0*C1
            T3=T2-2.0*T1
            IF(IMODE.EQ.1) GO TO 43
            CALL BOUND(BETA1,BETAB,BETAB,DEN1,DENB,DENB,RDT,DY1)
            CALL BOUND(ALPA1,ALPAB,ALPAB,DEN1,DENB,DENB,RDT,DY2)
            DY3=DY2-2.0*DY1
            CX1=-C1
            CX2=-C2
            CX3=-C3
            SX1=-S1
            SX2=-S2
            SX3=-S3
            TX1=-T1
            TX2=-T2
            TX3=-T3
            DX1=-DY1
            DX2=-DY2
            DX3=-DY3
C-----------------------------------------------
C     VERTICAL FAULT MODEL.
C-----------------------------------------------
            CALL FAULT(A,B,C,D,IMAX,JMAX,PULSE,FX,FY,JINT,IINT,ICOUL,JROW)
            GO TO 999
C-----------------------------------------------
C     ONE-LAYERED HALF-SPACE MODEL.
C-----------------------------------------------
43          CALL LAYER(A,B,C,D,IMAX,JMAX,PULSE,FX,FY,JINT,ICOUL,JROW)
            GO TO 999
C-----------------------------------------------
C     LOCALIZED INHOMOGENEOUS MODEL.
C-----------------------------------------------
C***********************************************
C     CHANGE VELOCITIES INTO LAME'S CONSTANTS AND DENSITY INTO
```

```
      C    (DT*DT)/(DX*DX*DENSITY)
      C**************************************************
      32       DO 90 I=1,ITOTX
               DO 90 J=1,JTOTY
               CC=DEN(I,J)*BETA(I,J)*BETA(I,J)
               BB=DEN(I,J)*ALPA(I,J)*ALPA(I,J)
               ALPA(I,J)=BB-2.0*CC
               BETA(I,J)=CC
      90       DEN(I,J)=RAMDA/DEN(I,J)
               CALL INHOMO(A,B,C,D,IMAX,JMAX,PULSE,FX,FY,ALPA,BETA,DEN,
             1 JTOP,JBOT,IRIG,ITOTX,JTOTY,ICOUL,JROW)
      999      STOP
               END
```

```
COMMON BLOCKS

/GEOPH/(+7)
IGEO     +0      ITERM    +1      IDECS    +2      IDECL    +3      NGEO     +4
IDEINC   +5      ICONS    +6
/INOUT/(+3)
IN       +0      IOUT     +1      ITAPE    +2
/BK1/(+15)
IDX      +0      JDY      +1
SON1     +2      SON2     +3      DT       +4      NPUL     +5      ISYM     +6
ISET     +7      IST      +10     IFN      +11     JST      +12     JFN      +13
NITER    +14
/BK2/(+11)
CON1     +0      CON2     +1      CON3     +2      PON1     +3      PON2     +4
PON3     +5      TON1     +6      TON2     +7      TON3     +10
/BK3/(+11)
C1       +0      C2       +1
C3       +2      S1       +3      S2       +4      S3       +5      T1       +6
T2       +7      T3       +10
/BK4/(+17)
DY1      +0      DY2      +1      DY3      +2      CX1      +3
CX2      +4      CX3      +5      SX1      +6      SX2      +7      SX3      +10
TX1      +11     TX2      +12     TX3      +13     DX1      +14     DX2      +15
DX3      +16

SUBPROGRAMS CALLED

GAUSS    SQRT.   FAULT    AMAX1.  LAYER   INHOMO
SPAC     EXIT    BOUND    LIST

SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*DEN2    1       *JTOP    2       *T       3       *ITEM    4       *VMAX    5        ALPA    6
*JINT    431      BETA    432     *TX      1055    *ITOTX   1056    *BB      1057    *JMAX    1060
*DEN1    1061     B       1062    *JCOUNT  16132   *DMAX    16133   *DZ      16134   *AMAX    16135
 ICOUL   16136    FMT     16220    JROW    16240    GY      16322   *RATIO1  16334   *IMODE   16335
*KSPA    16336   *RATIO2  16337   *IRIG    16340   *IINT    16341   *J       16342    .S0007  16343
 .S0006  16344    PULSE   16345    .S0005  16511   *IMAX    16512    D       16513    .S0004  33563
 .S0003  33564    A       33565    .S0002  50635   *ICOUNT  50636    .S0001  50637    GX      50640
 .S0000  50652   *DX      50653   *DENB    50654   *JTOTY   50655   *JBOT    50656   *ALPAB   50657
```

```
      SUBROUTINE BOUND(AN,A,AP,DN,D,DP,RDT,CONST)
C--------------SUBROUTINE BOUND------------------
C     THIS SUBROUTINE COMPUTES DERIVATIVES OF LAME CONSTANT AT THE
C     INTERFACE OF ONE-LAYERED OR VERTICAL FAULT MODEL.
C     ARGUMENT DEFINITIONS.
C        AN,A,AP: P- OR S-WAVE VELOCITY.
C        DN,D,DP: DENSITY.
C        CONST: DERIVATIVE OF LAME CONSTANT/DENSITY.
C-----------------------------------------------
       CONST=AP*AP-AN*AN+A*A*(DP-DN)/D
       CONST=CONST*RDT
       RETURN
       END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

*DP	1	*AP	2	*DN	3	*AN	4	*D	5	*A	6
*CONST	7	*RDT	10								

TEMPORARIES

BOUND   [NO ERRORS DETECTED]

```
      SUBROUTINE LAYER(A,B,C,D,IMAX,JMAX,PULSE,FX,FY,JINT,
     1 ICOUL,JROW)
C--------------SUBROUTINE LAYER------------------
C     THIS SUBROUTINE COMPUTERS VERTICAL AND HORIZONTAL DISPLACEMENTS
C     FOR A ONE-LAYERED HALF-SPCAE MODEL
C-----------------------------------------------
      DIMENSION A(IMAX,JMAX),B(IMAX,JMAX),C(IMAX,JMAX),D(IMAX,JMAX)
      DIMENSION ICOUL(1),JROW(1)
      DIMENSION PULSE(1),FX(5,5),FY(5,5)
      COMMON /GEOPH/IGEO,ITERM,IDECS,IDECL,NGEO,IDEINC,ICONS
      COMMON /BK1/IDX,JDY,SON1,SON2,DT,NPUL,ISYM,ISET,IST,IFN,
     1 JST,JFN,NITER
      COMMON /BK2/CON1,CON2,CON3,PON1,PON2,PON3,TON1,TON2,TON3
      COMMON /BK3/C1,C2,C3,S1,S2,S3,T1,T2,T3
      JNEGA=JINT-2
      JINTM=JINT-1
      JINT1=JINT+1
      JINTP=JINT+2
      IDIMX=IMAX
      JDIMY=JMAX
      TIME=0.0
      NP=1
      TEND=DT*NITER
999   TIME=TIME+DT
C***********************************************
C   COMPUTATION FOR THE HOMOGENEOUS REGION
C***********************************************
      CALL ITER(A,B,C,D,IDIMX,JDIMY,CON1,CON2,CON3,2,IDX,2,JNEGA)
      CALL ITER(A,B,C,D,IDIMX,JDIMY,TON1,TON2,TON3,2,IDX,JINTP,JDY)
C***********************************************
C   COMPUTATION FOR THE HORIZONTAL INTERFACE.
C***********************************************
      CALL INTERY(A,B,C,D,CON1,CON2,CON3,C1,C2,C3,2,I2,
     1 JINTM,IDIMX,JDIMY)
      CALL INTERY(A,B,C,D,PON1,PON2,PON3,S1,S2,S3,2,I2,
     1 JINT,IDIMX,JDIMY)
      CALL INTERY(A,B,C,D,TON1,TON2,TON3,T1,T2,T3,2,I2,
     1 JINT1,IDIMX,JDIMY)
      IF(NP.GT.NPUL) GO TO 77
C***********************************************
C   COMPUTATION FOR THE SOURCE REGION
C***********************************************
      DO 52 I=IST,IFN
      ITEM=I-IST+1
      DO 52 J=JST,JFN
      JTEM=J-JST+1
      B(I,J)=B(I,J)+FX(ITEM,JTEM)*PULSE(NP)
52    D(I,J)=D(I,J)+FY(ITEM,JTEM)*PULSE(NP)
77    CONTINUE
C***********************************************
C   SYMMETRY CONDITION
C***********************************************
      GO TO(71,72),ISYM
71    DO 40 J=2,JMAX
      B(1,J)=-B(ISET,J)
40    D(1,J)=D(ISET,J)
```

```
C*************************************************
C    FREE BOUNDARY CONDITION.
C*************************************************
72      DO 30 I=2,IDX
        B(I,1)=B(I,3)+D(I+1,2)-D(I-1,2)
30      D(I,1)=D(I,3)+SON2*(B(I+1,2)-B(I-1,2))
        GO TO(83,84),ISYM
83      B(1,1)=-B(ISET,1)
        D(1,1)=D(ISET,1)
84      NP=NP+1
C*************************************************
C    LIST THE OUTPUT
C*************************************************
        CALL LISTP(B,D,IMAX,JMAX,IGEO,ITERM,IDECS,IDECL,NGEO,IDEINC,
     1 ICOUL,JRUM,ICONS,TIME)
C*************************************************
C    CHANGING THE ARRAY FOR THE NEXT TIME STEP,
C*************************************************
        DO 90 I=1,IMAX
        DO 90 J=1,JMAX
        TEMPB=B(I,J)
        TEMPD=D(I,J)
        B(I,J)=A(I,J)
        A(I,J)=TEMPB
        D(I,J)=C(I,J)
        C(I,J)=TEMPD
90      CONTINUE
        GO TO 999
        RETURN
        END
```

COMMON BLOCKS

```
/GEOPH/(+7)
IGEO     +0      ITERM    +1      IDECS    +2      IDECL    +3      NGEO     +4
IDEINC   +5      ICONS    +6
/BK1/(+15)
IDX      +0      JDY      +1      SON1     +2      SON2     +3
DT       +4      NPUL     +5      ISYM     +6      ISET     +7      IST      +10
IFN      +11     JST      +12     JFN      +13     NITER    +14
/BK2/(+11)
CON1     +0      CON2     +1
CON3     +2      PON1     +3      PON2     +4      PON3     +5      TON1     +6
TON2     +7      TON3     +10
/BK3/(+11)
C1       +0      C2       +1      C3       +2      S1       +3
S2       +4      S3       +5      T1       +6      T2       +7      T3       +10
```

SUBPROGRAMS CALLED

```
ITER     LISTP    INTERY
```

```
      SUBROUTINE FAULT(A,B,C,D,IMAX,JMAX,PULSE,FC,FY,JINT,IINT,
     1 ICOUL,JROW)
C--------------SUBROUTINE FAULT-------------------
C     THIS SUBROUTINE COMPUTES DISPLACEMENT FIELD FOR THE VERTICAL
C     FAULT MODEL IN THE HALF-SPACE.
C-------------------------------------------------
      DIMENSION ICOUL(1),JROW(1)
      DIMENSION A(IMAX,JMAX),B(IMAX,JMAX),C(IMAX,JMAX),D(IMAX,JMAX)
      DIMENSION PULSE(1),FX(5,5),FY(5,5)
      COMMON /GEOPH/IGEO,ITERM,IDECS,IDECL,NGEO,IDEINC,ICONS
      COMMON /BK1/IDX,JDY,SON1,SON2,DT,NPUL,ISYM,ISET,IST,IFN,
     1 JST,JFN,NITER
      COMMON /BK2/CON1,CON2,CON3,PON1,PON2,PON3,TON1,TON2,TON3
      COMMON /BK3/C1,C2,C3,S1,S2,S3,T1,T2,T3
      COMMON /BK4/DY1,DY2,DY3,CX1,CX2,CX3,SX1,SX2,SX3,TX1,TX2,
     1 TX3,DX1,DX2,DX3
      IDIMX=IMAX
      JDIMY=JMAX
      JNEGA=JINT-2
      JINTN=JINT-1
      JINT1=JINT+1
      JINTP=JINT+2
      IINTP=IINT+2
      IINT1=IINT+1
      IINTN=IINT-1
      INEGA=IINT-2
      TIME=0.0
      NP=1
      TEND=DT*NITER
  999 TIME=TIME+DT
C*************************************************
C   COMPUTATION FOR THE HOMOGENEOUS REGION
C*************************************************
      CALL ITER(A,B,C,D,IDIMX,JDIMY,CON1,CON2,CON3,2,IDX,2,JNEGA)
      CALL ITER(A,B,C,D,IDIMX,JDIMY,TON1,TON2,TON3,2,INEGA,JINTP,JDY)
      CALL ITER(A,B,C,D,IDIMX,JDIMY,CON1,CON2,CON3,IINTP,IDX,JINTN,JDY)
      CALL ITER(A,B,C,D,IDIMX,JDIMY,CON1,CON2,CON3,IINT1,IINT1,
     1 JINTN,JINTN)
C*************************************************
C   COMPUTATION FOR THE HORIZONTAL INTERFACE.
C*************************************************
      CALL INTERY(A,B,C,D,CON1,CON2,CON3,C1,C2,C3,2,IINT,JINTN,
     1 IDIMX,JDIMY)
      CALL INTERY(A,B,C,D,PON1,PON2,PON3,S1,S2,S3,2,IINTN,JINT,
     1 IDIMX,JDIMY)
     CALL INTERY(A,B,C,D,TON1,TON2,TON3,T1,T2,T3,2,INEGA,JINT1,
     1 IDIMX,JDIMY)
C*************************************************
C   COMPUTATION FUT THE VERTICAL INTERFACE
C*************************************************
     CALL INTERX(A,B,C,D,PON1,PON2,PON3,SX2,SX3,SX1,IINT,JINT1,
     1 JDY,IDIMX,JDIMY)
      CALL INTERX(A,B,C,D,TON1,TON2,TON3,TX2,TX3,TX1,IINTN,JINTP,
     1 JDY,IDIMX,JDIMY)
      CALL INTERX(A,B,C,D,CON1,CON2,CON3,CX2,CX3,CX1,IINT1,JINT,
     1 JDY,IDIMX,JDIMY)
```

```
C*************************************************
C   COMPUTATION FOR THE CORNER POINT
C*************************************************
          CALL CORNER(A,B,C,D,IDIMX,JDIMY,IINT,JINT,PON1,PON2,PON3,
       1 DY1,DX2,DX3,DY2,DY3,DX1)
          CALL CORNER(A,B,C,D,IDIMX,JDIMY,IINTH,JINT1,TON1,TON2,TON3,
       1 T1,TX2,TX3,T2,T3,TX1)
         IF(NP.GT.NPUL) GO TO 77
C*************************************************
C   COMPUTATION FOR THE SOURCE REGION
C*************************************************
         DO 52 I=IST,IFN
         ITEM=I-IST+1
          DO 52 J=JST,JFN
         JTEM=J-JST+1
         B(I,J)=B(I,J)+FX(ITEM,JTEM)*PULSE(NP)
52       D(I,J)=D(I,J)+FY(ITEM,JTEM)*PULSE(NP)
77       CONTINUE
          GO TO(71,72),ISYM
C*************************************************
C   SYMMETRY CONDITION
C*************************************************
71        DO 40 J=2,JMAX
          B(1,J)=-B(ISET,J)
40        D(1,J)=D(ISET,J)
C*************************************************
C   FREE BOUNDARY CONDITION.
C*************************************************
72        DO 30 I=2,IDX
          B(I,1)=B(I,3)+D(I+1,2)-D(I-1,2)
30        D(I,1)=D(1,3)+SQH2*(B(I+1,2)-B(I-1,2))
          GO TO(83,84),ISYM
83        B(1,1)=-B(ISET,1)
          D(1,1)=D(ISET,1)
84        NP=NP+1
C*************************************************
C   LIST THE OUTPUT
C*************************************************
          CALL LISTP(B,D,IMAX,JMAX,IGEO,ITERM,IDECS,IDECL,NGEO,IDEINC,
       1 ICOUL,JROW,ICONS,TIME)
          IF(TIME.GT.TEND) RETURN
C*************************************************
C   CHANGING THE ARRAY FOR THE NEXT TIME STEP.
C*************************************************
          DO 90 I=1,IMAX
          DO 90 J=1,JMAX
          TEMPB=B(I,J)
          TEMPD=D(I,J)
          B(I,J)=A(I,J)
          A(I,J)=TEMPB
          D(I,J)=C(I,J)
          C(I,J)=TEMPD
90        CONTINUE
          GO TO 999
          RETURN
          END
```

```
      SUBROUTINE INHOMO(A,B,C,D,IMAX,JMAX,PULSE,FX,FY,ALPA,BETA,DEN,
     1 JTOP,JBOT,IRIG,ITOTX,JTOTY,ICOUL,JROW)
C--------------SUBROUTINE INHOMO-----------------
C     THIS SUBROUTINE COMPUTERS DISPLACEMENT FIELD CAUSED BY A LOCALIZED
C     INHOMOGENEITY EMBEDDED IN AN ELASTIC HALF-SPACE.
C-----------------------------------------------
      DIMENSION ICOUL(1),JROW(1)
      DIMENSION A(IMAX,JMAX),B(IMAX,JMAX),C(IMAX,JMAX),D(IMAX,JMAX)
      DIMENSION ALPA(ITOTX,JTOTY),BETA(ITOTX,JTOTY),DEN(ITOTX,JTOTY)
      DIMENSION PULSE(1),FX(5,5),FY(5,5)
      COMMON /GEOPH/IGEO,ITERM,IDECS,IDECL,NGEO,IDEINC,ICONS
      COMMON /BK1/IDX,JDY,SON1,SON2,DT,NPUL,ISYM,ISET,IST,IFN,
     1 JST,JFN,NITER
      COMMON /BK2/CON1,CON2,CON3,PON1,PON2,PON3,TON1,TON2,TON3
      JBOTT=JBOT-1
      JTOPP=JTOP+1
      IRIGG=IRIG-1
      TIME=0.0
      NP=1
      TEND=DT*NITER
999   TIME=TIME+DT
C***********************************************
C   COMPUTATION FOR THE HOMOGENEOUS REGION
C***********************************************
      CALL ITER(A,B,C,D,IMAX,JMAX,CON1,CON2,CON3,2,IDX,2,JTOP)
      CALL ITER(A,B,C,D,IMAX,JMAX,CON1,CON2,CON3,2,IDX,JBOT,JDY)
C***********************************************
C   COMPUTATION FOR THE INHOMOGENEOUS REGION.
C***********************************************
      CALL ITERH(A,B,C,D,IMAX,JMAX,ALPA,BETA,DEN,ITOTX,JTOTY,
     1 2,IRIGG,JTOPP,JBOTT)
      IF(IRIG.GT.IDX) GO TO 789
      CALL ITER(A,B,C,D,IMAX,JMAX,CON1,CON2,CON3,
     1 IRIG,IDX,JTOPP,JBOTT)
789   CONTINUE
      IF(NP.GT.NPUL) GO TO 77
C ***********************************************
C   COMPUTATION FOR THE SOURCE REGION
C ***********************************************
      DO 52 I=IST,IFN
      ITEM=I-IST+1
      DO 52 J=JST,JFN
      JTEM=J-JST+1
      B(I,J)=B(I,J)+FX(ITEM,JTEM)*PULSE(NP)
52    D(I,J)=D(I,J)+FY(ITEM,JTEM)*PULSE(NP)
77    CONTINUE
      GO TO(71,72),ISYM
C***********************************************
C . SYMMETRY CONDITION
C***********************************************
71    DO 40 J=2,JMAX
      B(1,J)=-B(ISET,J)
40    D(1,J)=D(ISET,J)
C***********************************************
C   FREE BOUNDARY CONDIIION.
C***********************************************
```

```
72      DO 30 I=2,IDX
        B(I,1)=B(I,3)+D(I+1,2)-D(I-1,2)
30      D(I,1)=D(I,3)+SON2*(B(I+1,2)-B(I-1,2))
        GO TO(83,84),ISYM
83      B(1,1)=-B(ISET,1)
        D(1,1)=D(ISET,1)
84      NP=NP+1
C**********************************************
C   LIST THE OUTPUT
C**********************************************
        CALL LISTP(B,D,IMAX,JMAX,IGEO,ITERM,IDECS,IDECL,NGEO,IDEINC,
     1 ICOUL,JROW,ICONS,TIME)
        IF(TIME.GT.TEND) RETURN
C**********************************************
C   CHANGING THE ARRAY FOR THE NEXT TIME STEP.
C**********************************************
        DO 90 I=1,IMAX
        DO 90 J=1,JMAX
        TEMPB=B(I,J)
        TEMPD=D(I,J)
        B(I,J)=A(I,J)
        A(I,J)=TEMPB
        D(I,J)=C(I,J)
        C(I,J)=TEMPD
90      CONTINUE
        GO TO 999
        RETURN
        END
```

COMMON BLOCKS

```
/GEOPH/(+7)
IGEO     +0      ITERM    +1      IDECS    +2      IDECL    +3      NGEO     +4
IDEINC   +5      ICONS    +6
/BK1/(+15)
IDX      +0      JDY      +1      SON1     +2      SON2     +3
DT       +4      NPUL     +5      ISYM     +6      ISET     +7      IST      +10
IFN      +11     JST      +12     JFN      +13     NITER    +14
/BK2/(+11)
CON1     +0      CON2     +1
CON3     +2      PON1     +3      PON2     +4      PON3     +5      TON1     +6
TON2     +7      TON3     +10
```

SUBPROGRAMS CALLED

```
ITER     LISTP    ITERH
```

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
.I0013 1     .I0012 2     .I0011 3     .I0010 4     *IRIGG  5     *JTOP   6
*ITEM  7     .I0027 10    ALPA   11    .I0026 12    BETA    13    .I0025  14
.I0024 15    .I0023 16    *ITOTX 17    .I0022 20    .I0021  21    *JMAX   22
.I0020 23    B      24    *TEMPB 25    ICOUL  26    .I0033  27    JROW    30
.I0032 31    .I0031 32    .I0030 33    *IRIG  34    *J      35    PULSE   36
```

```
      SUBROUTINE ITER(A,B,C,D,IX,IY,CON1,CON2,CON3,I1,I2,J1,J2)
C--------------SUBROUTINE ITER--------------------
C     THIS SUBROUTINE COMPUTERS DISPLACEMENT FIELD FOR A HOMOGENEOUS
C     REGION BY FINITE DIFFERENCE SCHEME.
C     ARGUMENT DEFINITIONS.
C        A: HORIZONTAL DISPLACEMENT AT (M)-TH TIME STEP
C        B: HORIZONTAL DISPLACEMENT AT (M+1) OR (M-1)-TH TIME STEP.
C        C: VERTICAL DISPLACEMENT AT (M)-TH TIME STEP.
C        D: VERTICAL DISPLACEMENT AT (M+1) OR (M-1)-TH TIME STEP.
C        CON1,CON2,CON3: CONSTANT TERM FOR A HOMOGENEOUS REGION.
C--------------------------------------------------
       DIMENSION A(IX,IY),B(IX,IY),C(IX,IY),D(IX,IY)
      DO 70 I=I1,I2
       IP=I+1
      IN=I-1
       DO 70 J=J1,J2
       JP=J+1
       JN=J-1
      B(I,J)=A(I,J)+A(I,J)-B(I,J)+CON1*(A(IP,J)-A(I,J)-A(I,J)+A(IN,J))+
     1 CON2*(C(IP,JP)-C(IN,JP)-C(IP,JN)+C(IN,JN))+CON3*(A(I,JP)-A(I,J)-
     2 A(I,J)+A(I,JN))
      D(I,J)=C(I,J)+C(I,J)-D(I,J)+CON1*(C(I,JP)-C(I,J)-C(I,J)+C(I,JN))+
     1 CON2*(A(IP,JP)-A(IN,JP)-A(IP,JN)+A(IN,JN))+CON3*(C(IP,J)-C(I,J)-
     2 C(I,J)+C(IN,J))
70     CONTINUE
      RETURN
      END
```

```
SUBPROGRAMS CALLED


SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*JP      1      .I0013  2      .I0012  3      .I0011  4      .I0010  5      B       6
*J2      7      *IP    10      *JN    11      *J1    12      *CON3  13      *J     14
D       15      A      16      .S0001 17      .S0000 20      *CON2  21      *I2    22
*IN     23      *IY    24      .I0007 25      .I0006 26      .I0005 27      .I0004 30
*CON1   31      .I0003 32      .I0002 33      .I0001 34      *I1    35      *I     36
.I0000  37      C      40      *IX    41      .I0017 42      .I0016 43      .I0015 44
.I0014  45

TEMPORARIES

.ITE16  46

ITER    [ NO ERRORS DETECTED ]
```

```
      SUBROUTINE ITERH(A,B,C,D,IMAX,JMAX,ALPA,BETA,DEN,ITOTX,
     1 JTOTY,I1,I2,J1,J2)
C--------------SUBROUTINE ITERH-------------------
C     THIS SUBROUTINE COMPUTES DISPLACEMENT FIELD FOR AN INHOMOGENEOUS
C     REGION BY FINITE DIFFERENCE SCHEME.
C     ARGUMENT DEFINITIONS:
C        ALPA,BETA: LAME CONSTANTS.
C        DEN: DENSITY.
C      *ALSO SEE SUBROUTINE ITER FOR DEFINITIONS.
C--------------------------------------------------
      DIMENSION A(IMAX,JMAX),B(IMAX,JMAX),C(IMAX,JMAX),D(IMAX,JMAX)
      DIMENSION ALPA(ITOTX,JTOTY),BETA(ITOTX,JTOTY),DEN(ITOTX,JTOTY)
      IH=1
      DO 10 I=I1,I2
      IP=I+1
      IN=I-1
      IH=IH+1
      IHP=IH+1
      IHN=IH-1
      JH=1
      DO 10 J=J1,J2
      JH=JH+1
      JP=J+1
      JN=J-1
      JHP=JH+1
      JHN=JH-1
      CONST=DEN(IH,JH)/4.0
      CON1=DEN(IH,JH)*BETA(IH,JH)
      TEMP=DEN(IH,JH)*ALPA(IH,JH)
     CON2=(CON1+TEMP)/4.0
       CON3=CON1+CON1+TEMP
      B(I,J)=A(I,J)+A(I,J)-B(I,J)+CONST*(ALPA(IHP,JH)+BETA(IHP,JH)+
     1 BETA(IHP,JH)-ALPA(IHN,JH)-BETA(IHN,JH)-BETA(IHN,JH))*(A(IP,J)-
     2 A(IN,J))+CON3*(A(IP,J)-A(I,J)-A(I,J)+A(IN,J))+CONST*(ALPA(IHP,JH)
     3 -ALPA(IHN,JH))*(C(I,JP)-C(I,JN))+CON2*(C(IP,JP)-C(IN,JP)-C(IP,JN)
     4          +C(IN,JN))+CON1*(A(I,JP)-A(I,J)-A(I,J)+A(I,JN))+
     5 CONST*(BETA(IH,JHP)-BETA(IH,JHN))*(C(IP,J)-C(IN,J)+A(I,JP)-
     6 A(I,JN))
      D(I,J)=C(I,J)+C(I,J)-D(I,J)+CONST*(ALPA(IH,JHP)+BETA(IH,JHP)+
     1 BETA(IH,JHP)-ALPA(IH,JHN)-BETA(IH,JHN)-BETA(IH,JHN))*(C(I,JP)-
     2 C(I,JN))+CON3*(C(I,JP)-C(I,J)-C(I,J)+C(I,JN))+CONST*(ALPA(IH,JHP)
     3 -ALPA(IH,JHN))*(A(IP,J)-A(IN,J))+CON2*(A(IP,JP)-A(IN,JP)-A(IP,JN)
     4 +A(IN,JN))+CON1*(C(IP,J)-C(I,J)-C(I,J)+C(IN,J))+CONST*(
     5 BETA(IHP,JH)-BETA(IHN,JH))*(C(IP,J)-C(IN,J)+A(I,JP)-A(I,JN))
   10 CONTINUE
      RETURN
      END
```

SUBPROGRAMS CALLED

```
      SUBROUTINE INTERY(A,B,C,D,CON1,CON2,CON3,BOUN1,BOUN2,BOUN3,
     1 I1,I2,J,IDX,IDY)
C---------------SUBROUTINE INTERY------------------
C     THIS SUBROUTINE COMPUTES DISPLACEMENT AT HORIZONTAL
C     INTERFACES FOR ONE-LAYERED OR VERTICAL FAULT MODEL.
C     ARGUMENT DEFINITIONS.
C        A,B,C,D: SEE SUBROUTINE ITER.
C        CON1,CON2,CON3: CONSTANT TERM FOR A HOMOGENEOUS REGION.
C        BOUN1,BOUN2,BOUN3: CONSTANT TERM FOR AN INHOMOGENEOUS REGION.
C        J: VERTICAL INDEX NUMBER FOR THE HORIZONTAL INTERFACE.
C-------------------------------------------------
      DIMENSION A(IDX,IDY),B(IDX,IDY),C(IDX,IDY),D(IDX,IDY)
       JN=J-1
       JP=J+1
       DO 10 I=I1,I2
       IN=I-1
       IP=I+1
      B(I,J)=A(I,J)+A(I,J)-B(I,J)+CON1*(A(IP,J)-A(I,J)-A(I,J)+A(IN,J))+
     1 CON2*(C(IP,JP)-C(IN,JP)-C(IP,JN)+C(IN,JN))+CON3*(A(I,JP)-A(I,J)-
     2 A(I,J)+A(I,JN))+BOUN1*(C(IP,J)-C(IN,J)+A(I,JP)-A(I,JN))
      D(I,J)=C(I,J)+C(I,J)-D(I,J)+CON1*(C(I,JP)-C(I,J)-C(I,J)+C(I,JN))+
     1 CON2*(A(IP,JP)-A(IN,JP)-A(IP,JN)+A(IN,JN))+CON3*(C(IP,J)-C(I,J)-
     2 C(I,J)+C(IN,J))+BOUN2*(C(I,JP)-C(I,JN))+BOUN3*(A(IP,J)-A(IN,J))
 10    CONTINUE
       RETURN
       END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
*JP      1    .I0013  2    .I0012  3    .I0011  4    .I0010  5    *IDX     6
*BOUN3   7    B      10    *BOUN2 11    *IP    12    *IDY   13    *JN     14
*BOUN1  15    *CON3  16    *J     17    D      20    A      21    .S0000  22
*CON2   23    *I2    24    *IN    25    .I0007 26    .I0006 27    .I0005  30
.I0004  31    *CON1  32    .I0003 33    .I0002 34    .I0001 35    *I1     36
*I      37    .I0000 40    C      41    .I0017 42    .I0016 43    .I0015  44
.I0014  45
```

TEMPORARIES

```
.INT16 46    .Q0000 47    .Q0001 50
```

INTERY [NO ERRORS DETECTED]

```
         SUBROUTINE INTERX(A,B,C,D,CON1,CON2,CON3,BOUN1,BOUN2,BOUN3,
        1 I,J1,J2,IDX,IDY)
C---------------SUBROUTINE INTERX------------------
C       THIS SUBROUTINE COMPUTES DISPLACEMENT FIELD AT VERTICAL
C       INTERFACES FOR A VERTICAL FAULT MODEL.
C       ARGUMENT DEFINITIONS.
C          CON1,CON2,CON3: CONSTANT TERM FOR A HOMOGENEOUS REGION.
C          I: HORIZONTAL INDEX NUMBER FOR THE VERTICAL INTERFACE.
C          BOUN1,BOUN2,BOUN3: CONSTANT TERM FOR AN INHOMOGENEOUS REGION.
C        * ALSO SEE SUBROUTINE INTERY FOR DEFINITIONS.
C-------------------------------------------------
        DIMENSION A(IDX,IDY),B(IDX,IDY),C(IDX,IDY),D(IDX,IDY)
         IN=I-1
         IP=I+1
         DO 10 J=J1,J2
         JN=J-1
         JP=J+1
        B(I,J)=A(I,J)+A(I,J)-B(I,J)+CON1*(A(IP,J)-A(I,J)-A(I,J)+A(IN,J))+
       1 CON2*(C(IP,JP)-C(IN,JP)-C(IP,JN)+C(IN,JN))+CON3*(A(I,JP)-A(I,J)-
       2A(I,J))+BOUN1*(A(IP,J)-A(IN,J))+BOUN2*(C(I,JP)-C(I,JN))
        D(I,J)=C(I,J)+C(I,J)-D(I,J)+CON1*(C(I,JP)-C(I,J)-C(I,J)+C(I,JN))+
       1 CON2*(A(IP,JP)-A(IN,JP)-A(IP,JN)+A(IN,JN))+CON3*(C(IP,J)-C(I,J)-
       2 C(I,J)+C(IN,J))+BOUN3*(C(IP,J)-C(IN,J)+A(I,JP)-A(I,JN))
 10      CONTINUE
         RETURN
         END
```

```
SUBPROGRAMS CALLED


SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*JP       1     .I0013 2     .I0012 3     .I0011 4     .I0010 5     *IDX    6
*BOUN3    7     B      10    *BOUN2 11    *J2    12    *IP    13    *IDY    14
*JN       15    *BOUN1 16    *J1    17    *CON3  20    *J     21    D       22
 A        23    .S0000 24    *CON2  25    *IN    26    .I0007 27    .I0006  30
 .I0005   31    .I0004 32    *CON1  33    .I0003 34    .I0002 35    .I0001  36
 .I0000   37    *I     40    C      41    .I0017 42    .I0016 43    .I0015  44
 .I0014   45

TEMPORARIES

 .INT16 46      .Q0000 47     .Q0001 50     .Q0002 51

INTERX  [ NO ERRORS DETECTED ]
```

```
CORNER  LN47TN.FOR      FORTRAN V.4A(230) /KI    2-OCT-75         2139    PAGE 1

         SUBROUTINE CORNER(A,B,C,D,IDX,IDY,I,J,CON1,CON2,CON3,BOUN1,
       1 BOUN2,BOUN3,BOUN4,BOUN5,BOUN6)
C--------------SUBROUTINE CORNER------------------
C      THIS SUBROUTINE COMPUTES DISPLACEMENT AT THE CORNER OF
C      ARGUMENT DEFINITIONS.
C      THE VERTICAL FAULT MODEL.
C        * SEE SUBROUTINE INTERY AND INTERX FOR DEFINITIONS.
C-------------------------------------------------
        DIMENSION A(IDX,IDY),B(IDX,IDY),C(IDX,IDY),D(IDX,IDY)
         JN=J-1
         JP=J+1
         IN=I-1
         IP=I+1
        B(I,J)=A(I,J)+A(I,J)-E(I,J)+CON1*(A(IP,J)-A(I,J)-A(I,J)+A(IN,J))+
       1 CON2*(A(IP,JP)-A(IN,JP)-A(IP,JN)+A(IN,JN))+CON3*(C(IP,J)-C(I,J)-
       2 A(I,J)+A(I,JN))+BOUN1*(C(IP,J)-C(IN,J)+A(I,JP)-A(I,JN))
       3 +BOUN2*(A(IP,J)-A(IN,J))+BOUN3*(C(I,JP)-C(I,JN))
        D(I,J)=C(I,J)+C(I,J)-F(I,J)+CON1*(C(I,JP)-C(I,J)-C(I,J)+C(I,JN))+
       1 CON2*(C(IP,JP)-C(IN,JP)-C(IP,JN)+C(IN,JN))+CON3*(A(I,JP)-A(I,J)-
       2 C(I,J)+C(IN,J))+BOUN4*(C(I,JP)-C(I,JN))+BOUN5*(A(IP,J)-A(IN,J))+
       3 BOUN6*(C(IP,J)-C(IN,J)+A(I,JP)-A(I,JN))
         RETURN
         END
```

SUBPROGRAMS CALLED

SCALARS AND ARRAYS ["*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED]

```
*JP      1      .I0013 2      .I0012 3      .I0011 4      .I0010 5      *IDX     6
*BOUN3   7      B      10     *BOUN2  11     *IP     12     *IDY    13     *JN     14
*BOUN1  15      *CON3  16     *BOUN6  17     *J      20     D       21     A       22
*CON2   23      *IN    24     *BOUN5  25     .I0007 26      .I0006 27      .I0005 30
.I0004  31      *CON1  32     .I0003 33      .I0002 34      .I0001 35      .I0000 36
*I      37      *BOUN4 40     C       41     .I0017 42      .I0016 43      .I0015 44
.I0014  45
```

TEMPORARIES

```
.COR16 46      .Q0000 47      .Q0001 50      .Q0002 51      .Q0003 52      .Q0004 53
.Q0005 54
```

CORNER [NO ERRORS DETECTED]

```
GAUSS   LN47TH.FOR      FORTRAN V.4A(230) /KI   2-OCT-75        2139    PAGE 1

              SUBROUTINE GAUSS(F,DT,W,NMAX)
        C--------------SUBROUTINE GAUSS------------------
        C      THIS SUBROUTINE COMPUTES TIME DEPENDENCE OF SOURCE FUNCTION
        C      USING GAUSSIAN FUNCTION(BODY FORCE).
        C      ARGUMENT DEFINITIONS.
        C         F: SOURCE TIME FUNCTION.
        C         DT: SAMPLING INTERVAL OF TIME.
        C         NMAX: THE LENGTH OF ARRAY F.
        C         W: PARAMETER DETERMING THE WIDTH OG THE SOURCE FUNCTION.
        C--------------------------------------------------
              DIMENSION F(1)
              TST=W*1.6
              NMAX=IFIX(TST*2.0/DT)+1
              ALPA=2.0/(W*W)
              T=-TST-DT
              DO 10 I=1,NMAX
              T=T+DT
              TT=T*T*ALPA
        10    F(I)=EXP(-TT)
              RETURN
              END

SUBPROGRAMS CALLED

IFIX.
EXP.

SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*W      1       *T      2       *ALPA   3       *DT     4       *TST    5       .S0000 6
*I      7       *NMAX   10      *TT     11      F       12

TEMPORARIES

 .GAU16 13      .Q0000 14

GAUSS   [ NO ERRORS DETECTED ]
```

```
SPAC     LN47TN.FOR       FORTRAN V.4A(230) /KI    2-OCT-75         2:39    PAGE 1

              SUBROUTINE SPAC(Y,D,DT,T,NMAX)
        C--------------SUBROUTINE SPAC--------------------
        C       THIS SUBROUTINE CALCULATES THE SPACIAL DEPENDENCE OF SOURCE
        C       FUNCTION.THE SMALLEST SOURCE REGION IN THIS COMPUTER PROGRAM IS
        C       2DX*2DY,WHERE DX AND DY ARE SAMPLING INTERVAL IN SPACIAL DOMAIN.
        C       ARGUMENT DEFINITIONS.
        C          Y,D: FUNCTION WHICH CAN DETERMINE SPACIAL DEPENDENCE
        C                OF SOURCE FUNCTION.
        C          DT: SAMPLING INTERVAL ( =DX OR DY).
        C          T: PARAMETER WHICH CONTRL THE EXTENT OF THE SOURCE REGION.
        C          NMAX: THE LENGTH OF ARRAY Y OR D.
        C-----------------------------------------------------
              DIMENSION Y(1),D(1)
              T2=T*2.0
              T3=T*3.0
              T4=T*4.0
              NMAX=IFIX(T4/DT)+1
              A=0.5/(T*T)
              X=-DT
              DO 10 I=1,NMAX
              X=X+DT
              IF(X.GT.T) GO TO 21
              Y(I)=A*X*X
              D(I)=2.0*A*X
              GO TO 10
        21    IF(X.GT.T3) GO TO 22
              Y(I)=-A*(X-T2)*(X-T2)+1.0
              D(I)=-2.0*A*(X-T2)
              GO TO 10
        22    Y(I)=A*(X-T4)*(X-T4)
              D(I)=2.0*A*(X-T4)
        10    CONTINUE
              RETURN
              END

SUBPROGRAMS CALLED

IFIX.

SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*T        1       *DT       2       Y       3      *T4      4       D       5      *A      6
 .S0000   7       *X        10      *T3     11     *I       12      *NMAX   13     *T2     14

TEMPORARIES

 .SPA16 15

SPAC    [ NO ERRORS DETECTED ]
```

```
LISTP   LN47TN.FOR      FORTRAN V.4A(230) /KI   2-OCT-75        2:39    PAGE 1

              SUBROUTINE LISTP(B,D,IMAX,JMAX,IGEO,ITERM,IDECS,IDECL,NGEO,
            1 IDEINC,ICOUL,JROW,ICONS,TIME)
        C-------------SUBROUTINE LISTP-------------------
        C     THIS SUBROUTINE WRITES THE DISPLACEMENT OUTPUT IN LINE PRINTER AND
        C     IN A MAGNETIC TAPE.
        C------------------------------------------------
              DIMENSION B(IMAX,JMAX),D(IMAX,JMAX),ICOUL(1),JROW(1)
              COMMON /INOUT/IN,IOUT,ITAPE
              GO TO(11,12),IGEO
        11    GO TO(13,14),ITERM
        13    DO 10 J=IDECS,IDECL,IDEINC
              WRITE(IOUT,400)TIME,ICONS,J,B(ICONS,J),D(ICONS,J)
        10   WRITE(ITAPE,400)TIME,ICONS,J,B(ICONS,J),D(ICONS,J)
        400   FORMAT(2X,F10.5,2X,I6,2X,I6,4X,2E20.8)
              RETURN
        14    DO 20 I=IDECS,IDECL,IDEINC
              WRITE(IOUT,400)TIME,I,ICONS,B(I,ICONS),D(I,ICONS)
        20   WRITE(ITAPE,400)TIME,I,ICONS,B(I,ICONS),D(I,ICONS)
              RETURN
        12    INN=1
              JNN=1
        40    IT=ICOUL(INN)
              JT=JROW(INN)
              WRITE(IOUT,400)TIME,IT,JT,B(IT,JT),D(IT,JT)
             WRITE(ITAPE,400)TIME,IT,JT,B(IT,JT),D(IT,JT)
              IF(INN.GT.NGEO) RETURN
              INN=INN+1
              GO TO 40
              END

COMMON BLOCKS

/INOUT/(+3)
IN        +0        IOUT      +1        ITAPE     +2

SUBPROGRAMS CALLED



SCALARS AND ARRAYS [ "*" NO EXPLICIT DEFINITION - "%" NOT REFERENCED ]

*ICONS   1       *IDECL   2       *IGEO    3       *JNN     4       *JMAX    5        B       6
*IDEINC  7       *JT      10       ICOUL   11       JROW    12      *NGEO    13      *INN     14
*J       15      *IMAX    16       D       17      .S0001   20      .S0000   21      *IT      22
*TIME    23      .I0007   24      .I0006   25      *IDECS   26      .I0005   27      .I0004   30
*ITERM   31      .I0003   32      .I0002   33      .I0001   34      *I       35      .I0000   36

TEMPORARIES

 .LIS16 46

LISTP   [ NO ERRORS DETECTED ]
```

```
      SUBROUTINE LIST(IMODE,ALPA1,ALPA2,BETA1,BETA2,DEN1,DEN2,DX,TX,T)
C-----------------------------------------------------
C     THIS SUBROUTINE LIST PARAMETERS OF THE MODEL.
C-----------------------------------------------------
      COMMON /INOUT/IN,IOUT,ITAPE
      COMMON /BK1/IDX,JDY,SON1,SON2,DT,NPUL,ISYM,ISET,IST,IFN,
     1 JST,JFN,NITER
      GO TO (11,12,13),IMODE
11    WRITE(IOUT,100)
100   FORMAT(//,4X,'PRESENT MODEL IS ONE-LAYERED HALF-SPACE')
      GO TO 14
12    WRITE(IOUT,200)
200   FORMAT(//,4X,'PRESENT MODEL IS VERTICAL FAULT')
      GO TO 14
13    WRITE(IOUT,300)
300    FORMAT(//,4X,'PRESENT MODEL IS LOCALIZED INHOMOGENEITY')
14    GO TO(15,16),ISYM
15    WRITE(IOUT,400)
400    FORMAT(4X,'THIS MODEL IS SYMMETRICAL')
      GO TO 17
16    WRITE(IOUT,500)
500    FORMAT(4X,'THIS MODEL IS ASYMMETRICAL')
17    IC=(IFN+IST)/2
      JC=(JST+JFN)/2
      WRITE(IOUT,600)DT,DX,TX,T,NITER,IC,JC
600    FORMAT(///,8X,'TIME    INCREMENT=',F10.5,/,
     1 8X,'DISTANCE INCREMENT=',F10.5,/,
     2 8X,'SPACIAL SOURCE WIDTH=',F10.5,/,
     3 8X,'TEMPORAL SOURCE WIDTH=',F10.5,/,
     4 8X,'NUMBER OF ITERATION=',I5,/,
     5 8X,'CENTER OF SOURCE IS LOCATED AT(',I2,'DX,'I2,'DY)')
      WRITE(IOUT,700) ALPA1,BETA1,DEN1
700   FORMAT(///,8X,'ELASTIC PARAMETERS IN UPPER MEDIUM',/,
     1 8X,'LONGITUDINAL VELOSITY=',F12.5,/,
     2 8X,'SHEAR VELOCITY       =',F12.5,/,
     2 8X,'DENSITY              =',F12.5)
      GO TO(21,21,22),IMODE
21    WRITE(IOUT,800)ALPA2,BETA2,DEN2
800    FORMAT(//,8X,'ELASTIC PARAMETERS IN LOWER MEDIUM',/,
     1 8X,'LONGITUDINAL VELOSITY=',F12.5,/,
     2 8X,'SHEAR VELOCITY       =',F12.5,/,
     2 8X,'DENSITY              =',F12.5)
22    WRITE(IOUT,900)
900   FORMAT(////,6X,'TIME',4X,'X-INDEX',3X,'Y-INDEX',10X,
     1 'HORIZONTAL',10X,'VERTICAL',/,41X,'DISPLACEMENT',8X,
     2 'DISPLACEMENT',/)
      RETURN
      END
```

COMMON BLOCKS

```
/INOUT/(+3)
IN      +0      IOUT    +1      ITAPE   +2
/BK1/(+15)
IDX     +0      JDY     +1      SON1    +2
```

SELECTED REFERENCES

Aboudi, Jacob, 1971, Numerical simulation of seismic source: Geophysics, v. 36, p. 810-821.

Alford, R. M., Kelley, V. R., and Boore, D. M., 1974, Accuracy of finite-difference modeling of the acoustic wave equation: Geophysics, v. 39, p. 834-841.

Alterman, Z. S., and Aboudi, J., 1970, Source of finite extent, applied force and couple in an elastic half-space: Geophysical Journal, v. 21, p. 47-64.

Alterman, Z. S., and Karal, F. C., 1969, Propagation of elastic waves in layered media by finite difference methods: Seismol. Soc. America Bull., v. 58, p. 367-398.

Balch, A. H., and Smolka, F. R., 1970, Plane and spherical transient Voigt waves: Geophysics, v. 35, p. 745-761.

Bath, Markus, 1968, Mathematical aspects of seismology: Amsterdam, Elsevier.

Collins, Francis, 1960, Plane compressional Voigt waves: Geophysics, v. 25, p. 483-504.

Ewing, W. M., Jardetzky, W. S., and Press, Frank, 1957, Elastic waves in layered media: New York, McGraw-Hill.

Haskell, N. A., 1953, The dispersion of surface waves in multi-layered media: Seismol. Soc. America Bull., v. 43, p. 17-34.

Horton, C. N., 1959, A loss mechanism for the Pierre shale: Geophysics, v. 24, p. 667-680.

Munasinghe, M., and Farnell, G. W., 1973, Finite difference analysis of Rayleigh wave scattering at vertical discontinuities: Jour. Geophys. Research, v. 78, p. 2454-2466.

Peterson, R. A., Fillippone, W. R., and Coker, F. B., 1955, The synthesis of seismogram from well log data: Geophysics, v. 20, p. 516-538.

Richtmyer, R. D., and Morton, K. W., 1967, Difference methods for initial-value problems, New York, Interscience.

Sherwood, J. W. C., and Trorey, A. W., 1965, Minimum-phase and related properties of the response of a horizontally stratified obsorptive earth to plane acoustic waves: Geophysics, v. 30, p. 191-197.

Trorey, A. W., 1962, Theoretical seismogram with frequency and depth dependent absorption: Geophysics, v. 27, p. 766-785.

White, J. E., 1965, Seismic waves; radiation, transmission, and attenuation: New York, McGraw-Hill.

Weunschel, P. C., 1960, Seismogram synthesis including multiples and transmission parameters: Geophysics, v. 25, p. 106-129.