RESPONSE OF AN ELASTIC LAYER OVER AN ELASTIC HALF-SPACE TO A POINT SOURCE

by

Terrance Gordon Barker

B.A., University of California at San Diego (1968)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE
OF MASTER OF SCIENCE

at the

Massachusetts Institute of Technology

June, 1970

Signature of Author . . . .
Department of Earth and Planetary
Sciences, June 4, 1970

Certified by . . . .
Thesis Supervisor

Accepted by . . . .
Chairman, Departmental Committee
on Graduate Students

ABSTRACT

Response of an elastic layer over an elastic
half-space to a point source
by
Terrance Gordon Barker

Submitted to the Department of Earth and Planetary
Sciences on June 4, 1970 in partial fulfillment of
the requirements for the degree of Master of Science

Synthetic seismograms are computed for a point source of compressional (P) or poloidal (SV) waves embedded within an elastic layer overlying an elastic half-space. A fast method which utilizes an approximation in Cagniard-de-Hoop technique is used in computation. The results are accurate for short ranges and low frequencies. A computer program evaluates the solution and plots the synthetic seismogram. These results are then compared with solutions to Lamb's problem and the one layer problem computed by the exact method. The comparisons are good.

Thesis Supervisor: M. Nafi Toksöz
Title: Associate Professor of Geophysics

ACKNOWLEDGEMENT

I would like to extend my sincere thanks to Professor Don Helmberger for his assistance and support in formulating and programming this problem. I would also like to thank Professor M. Nafi Toksöz for many fruitful discussions.

This work was supported by the United States Air Force Office of Aerospace Research and monitored by Air Force Cambridge Research Laboratories under Contract No. F 19628-68-C-0043. The author held a National Defense Education Act Fellowship, Title IV.

TABLE OF CONTENTS

INTRODUCTION

In this paper, a method is presented by which synthetic seismograms for a layer over a half-space can be computed. The method can be easily extended to more complicated models. It has the advantages of requiring little computer time, and being accurate for low as well as high frequencies. In the formulation the generalized ray approach, and the Cagniard-de-Hoop method are utilized to put the solutions into operational form. An improvement is made on the "high frequency approximation" (see Helmberger, 1967) to achieve accurate results for low frequencies. The computations are done for both compressional and shear sources. Plots are shown displaying the effects of combining the two sources.

Low frequency body waves are less affected by inhomogeneities in the crust than are those with high frequencies. They are not as sensitive to irregularities in the vicinity of the receiver. Long period seismograms are thus "cleaner", and give less distorted information about gross properties of the crust, and about the source of the waves. It is therefore important to be able to synthesize long period records quickly.

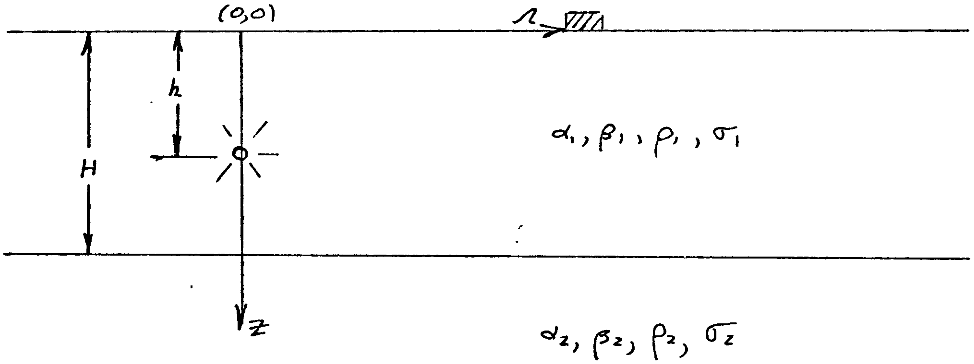
Many sources of seismic activity produce P and S waves simultaneously. The records obtained from that activity contain the responses to both source types. In order to interpret the records correctly, the effects must be distinguished. It is for this reason that we included both P and SV sources in this study. A source of SH waves could be added also, but was not because SH waves do not interact with P and SV waves.

Harkrider (1964) has presented a method to compute surface wave records at long distances. Helmberger (1965) has shown how to compute quickly high frequency records for long ranges. The method presented here is valid for high and low frequencies and short ranges. It can also be used for long ranges, but the surface wave response must be forfeited. However, long period body wave records are calculable at long ranges by this method.

In the next section we introduce the mathematics and the approximation used to bring the solution to operational form. The computer results and their implications are discussed in the third section. We have displayed the computer output following that discussion. A brief description of the program and a printout are included as an appendix.

THEORETICAL FORMULATION

A homogeneous, isotropic, linearly elastic slab of thickness H and infinite horizontal extent overlies a semi-infinite solid with different elastic properties. A point source of compressional (P) and poloidal (SV) waves is embedded within the slab a distance h from its upper surface. The upper surface is stress free. The layer and the half-space are welded together so that pressure and displacement are continuous across the interface. The problem will be formulated in circular cylindrical coordinates with the origin at the upper surface directly above the source. All variables are independent of the azimuthal coordinate.



We wish to compute the response at the surface. This is done by solving the linearized elastic wave equation by Laplace transform methods. The solution is then represented as an infinite sum of contributions from ray paths across the layer. Inversion is accomplished by Cagniard-de-Hoop techniques. The details of this approach have been displayed by several authors (Helmberger, 1967, and Phinney, 1965), as well as in Appendix I and will not be included in this section.

The once-transformed (in time) displacement for the n^{th} generalized ray from a step function source can be written

$$(1)\quad \bar{u}_{z,n}(r,0,s) = s\,\mathrm{Im}\int_0^{i\infty} p\,F_{z,n}(p)\,K_0(spr)\,e^{-sg_n(p)}\,dp$$

$$(2)\quad \bar{u}_{r,n}(r,0,s) = s\,\mathrm{Im}\int_0^{i\infty} p\,F_{r,n}(p)\,K_1(spr)\,e^{-sg_n(p)}\,dp$$

The subscript z refers to the vertical component, r refers to the radial component. $K_0(spr)$ and $K_1(spr)$ are modified Bessel functions. The functions $F_{z,n}$, $F_{r,n}$, and g_n are defined as follows:

$$F_{n,z}(p) = S\,\mathcal{R}_z(p)\,\mathcal{C}_n(p)$$

$$F_{n,r}(p) = S\,\mathcal{R}_r(p)\,\mathcal{C}_n(p)$$

where

$$S = [2\pi^2\eta_1(\lambda_1+2\mu_1)]^{-1} \qquad \text{for P source}$$

$$S = [2\pi^2\eta_1'\mu_1]^{-1} \qquad \text{for SV source}$$

with

$$\eta_1 = \left(\frac{1}{\alpha_1^2} - p^2\right)^{1/2}$$

$$\eta_1' = \left(\frac{1}{\beta_1^2} - p^2\right)^{1/2}$$

α_1 and β_1 are the P and S wave speeds in the slab, respectively. $\mathcal{R}_z(p)$ and $\mathcal{R}_r(p)$ are the receiver directivity functions which contain the response of the free surface due to an impinging P or SV wave. They are written out explicitly in the appendix. $\mathcal{L}_n(p)$ is the reflection function for the nth generalized ray. It is formed from the product of the reflection coefficients due to internal reflections within the layer. The reflection coefficients are also written out in the appendix. Finally, the function $g_n(p)$ is a measure of the time required to traverse vertically the nth ray path.

$$g_n(p) = (1 - HK_{ud} - \ )[(1-K_{sp})\eta_1' + K_{sp}\eta_1] + 1_s^{(n)}\eta_1' H + 1_p^{(n)}\eta_1 H$$

$$g_n(p) = (1 - HK_{ud} - h)[(1-K_{sp})\eta_1' + K_{sp}\eta_1] + l_s^{(n)}\eta_1' H + l_p^{(n)}\eta_1 H$$

where

$$K_{ud} = \begin{cases} 1 & \text{if, at the source, the ray is directed upward.} \\ 0 & \text{if, at the source, the ray is directed downward.} \end{cases}$$

$$K_{sp} = \begin{cases} 1 & \text{if the source emits P waves.} \\ 0 & \text{if the source emits S waves.} \end{cases}$$

and $1_s^{(n)}$ is the number of times the nth ray path traverses the layer as an SV wave, and $1_p^{(n)}$ is the number of traverses as a P wave.

We have two approaches from which to choose in evaluating equations (1) and (2). The first is to invert the expressions analytically and evaluate the result. This is done with the Laplace inversion formulas

$$\mathcal{L}^{-1}\left\{K_0(spr)\, e^{-sg_n}\right\} = \frac{\mathcal{H}(t-pr-g_n)}{\left[(t-g_n)^2-p^2r^2\right]^{1/2}}$$

and

$$\mathcal{L}^{-1}\left\{K_1(spr)\, e^{-sg_n}\right\} = \frac{t-g_n}{pr}\cdot\frac{\mathcal{H}(t-pr-g_n)}{\left[(t-g_n)^2-p^2r^2\right]^{1/2}}$$

where $\mathcal{H}$ is the Heaviside step function. The displacements then become

$$(3)\quad u_{z,n}(r,0,t) = \frac{\partial}{\partial t}\, Re \int_{t_0}^{t_1} \frac{\mathcal{H}(t-\tau)}{\left[(t-\tau)(t-\tau+pr)\right]^{1/2}}\; p\, F_{z,n}(p)\, dp$$

$$(4)\quad u_{r,n}(r,0,t) = \frac{\partial}{\partial t}\, Re \int_{t_0}^{t} \frac{(t-g_n)\,\mathcal{H}(t-\tau)}{r\left[(t-\tau)(t-\tau+pr)\right]^{1/2}}\; F_{r,n}(p)\, dp$$

where we have set $\tau = pr + g_n$ and t_0 is the time at the beginning of the signal.

Equations (3) and (4) can now be evaluated by numerical quadrature. $p(\tau)$ is computed from the requirement that the argument of the Heaviside function be real (thus giving the contour of integration). The author has done this for Lamb's problem (elastic half-space with a free surface), and Pekeris, et al (1965) has used this method for a layer over a half-space. This method is time consuming, however. Since the upper limit of integration is included in the integrand, an integration from t_0 must be performed for every value of the upper limit. In addition to its time requirements, this method has the disadvantage of requiring repeated integrations past a strong pole in the complex plane. This is a source of numerical instabilities.

BRANCH POINT !?
Refracted arrivals

Our second choice is to approximate the modified Bessel functions with their asymptotic expansions, and then invert the results. Expressions for the Bessel functions valid for large values of their argument (spr) are (Dwight, 1961):

(5) $$K_0(spr) \approx \left(\frac{\pi}{2spr}\right)^{1/2} e^{-spr}\left[1 - \frac{1}{8}(spr)^{-1} + \frac{9}{128}(spr)^{-2} + \cdots\right]$$

(6) $$K_1(spr) \approx \left(\frac{\pi}{2spr}\right)^{1/2} e^{-spr}\left[1 + \frac{3}{8}(spr)^{-1} + \frac{3}{128}(spr)^{-2} + \cdots\right]$$

Substituting these expressions into (1) and (2), we have

(7) $$\bar{u}_{z,n}(r,0,s) \approx s\sqrt{\frac{\pi}{2}}\, dm \int_0^{i\infty} p\, F_{z,n}(p) \frac{e^{-s\tau}}{\sqrt{spr}} \left\{1 - \frac{1}{8}(spr)^{-1} + \frac{9}{128}(spr)^{-2} + \cdots\right\} dp$$

(8) $$\bar{u}_{r,n}(r,0,s) \approx s\sqrt{\frac{\pi}{2}}\, dm \int_0^{i\infty} p\, F_{r,n}(p) \frac{e^{-s\tau}}{\sqrt{spr}} \left\{1 + \frac{3}{8}(spr)^{-1} + \frac{3}{128}(spr)^{-2} + \cdots\right\} dp$$

By making use of the following inversion formulas

$$\mathcal{L}^{-1}\left[s^{1/2} e^{-s\tau} \bar{f}(s)\right] = \frac{\partial}{\partial t}\left[\frac{1}{\sqrt{\pi t}} * f(t)\right]$$

$$\mathcal{L}^{-1}\left[s^{-1/2} e^{-s\tau} \bar{f}(s)\right] = \frac{\partial}{\partial t}\left\{\frac{1}{\sqrt{\pi t}} * \left[\mathcal{H}(t) * f(t)\right]\right\}$$

$$\mathcal{L}^{-1}\left[s^{-3/2} e^{-s\tau} \bar{f}(s)\right] = \frac{\partial}{\partial t}\left\{\frac{1}{\sqrt{\pi t}} * \left[t * f(t)\right]\right\}$$

where * denotes convolution, equations (7) and (8) are

(9) $$u_{z,n}(r,0,t) = \frac{1}{\sqrt{2}}\, dm \frac{\partial}{\partial t}\left\{\frac{1}{\sqrt{t}} * \left[\sqrt{\frac{p}{r}} F_{n,z}(p) \frac{dp}{dt} - \frac{1}{8}\mathcal{H}(t) * \left(\frac{1}{\sqrt{pr^3}} F_{n,z}(p) \frac{dp}{dt}\right) + \right.\right.$$
$$\left.\left. + \frac{9}{128}\, t * \left(p^{-3/2} r^{-5/2} F_{n,z}(p) \frac{dp}{dt}\right)\right]\right\}$$

$$u_{r,n}(r,0,t) = \frac{1}{\sqrt{2}} \operatorname{Im} \frac{\partial}{\partial t}\left\{ \frac{1}{\sqrt{t}} * \left[-\sqrt{\frac{p}{r}}\, F_{n,r}(p)\frac{dp}{dt} - \frac{3}{8} H(t) * \left(\frac{1}{\sqrt{pr^3}} F_{n,r}(p)\frac{dp}{dt}\right) + \right.\right.$$
$$\left.\left. + \frac{3}{128}\, t * \left(p^{-3/2} r^{-5/2} F_{n,r}(p)\frac{dp}{dt}\right)\right]\right\} \qquad (10)$$

If we consider the lowest order term and do the inversion, we get the so-called high frequency approximation. As the name implies, this approximation is poor for low frequencies. It is also poor for short ranges. An improvement can be made by including terms that are higher order in 1/(spr) in (9) and (10). Again, the contour of integration is computed by requiring that τ be real, but now only one integration--a convolution--need be performed. This is considerably faster than the first alternative. We have done calculations employing this improved approximation, and present the results in the next section.

DISCUSSION OF RESULTS

The primary purpose of this investigation is to compute synthetic seismograms which have good low frequency response. Since surface waves have lower frequencies than body waves, they provide a means for testing. A program was already in existence which would compute exactly (i.e., by the first method of inversion described earlier) the response of a half-space to a buried compressional point source. A Rayleigh wave is produced at the surface by the impinging P wave. The approximate method was also used for this problem by allowing the program to consider only the ray which travels directly to the receiver with no internal reflections. The impulse response was convolved with a triangle whose base is 0.2 sec., and whose area is one (this applies to all the plots presented here). The results are shown in Figures 1-8 and the comparison appears good. In these calculations, the first two terms in the sums in equations (7) and (8) were used. It thus appears that two terms are sufficient. The displacement due to an SV source is also shown (Figures 9-12). The response begins with that part of the energy which has been refracted along the free surface as a P wave. The direct S arrives later and is followed soon by the Rayleigh wave.

The remaining models are layered. As a further test of our method, a comparison was made with the results of Pekeris, _et al_ (1965), where they used the first inversion technique. The comparison is shown for two ranges and is seen to be good (Figures 13-15, 21). The reader will notice that the record for the approximate program is shorter than the other. This is because ray theory characteristically diverges rapidly from the actual values at some time from the beginning of the record, and the plot was terminated where a strong drift began. The divergence normally occurs near the time of arrival of the interfacial trapped wave, and there is a pole on the real p axis which is responsible for the contribution of the interface wave. Since more reflection coefficients are involved in the calculation of the interface wave than for the Rayleigh wave, the effect of the pole is multiplied. If a sufficient number of rays are not evaluated in the vicinity of that value of p, the result will not converge on the true value. Pekeris, et al, (1965) used about five times the number of rays that we used, and although this many rays could have easily been included in our computations, the extra expense would not have enhanced our goal of showing the accuracy of our results. It is also interesting to note that the ratio $\frac{H-h}{r}$, the contour comes closer to the pole, and the seismogram diverges more rapidly.

This model (labeled Model I in Table 1) was embedded with an SV source also. The plots (Figures 17, 18, 23, 24) show a prominent Rayleigh wave following the direct S wave closely. For r = 10, the vertical displacement has a long quiet period preceding several large signals due to S-multiples and the Rayleigh wave.

Model I was used again with the source moved to within 0.05 km. of the surface. The expected growth of the Rayleigh wave with respect to the body waves for smaller values of $\frac{h}{r}$ occurred. These plots (Figures 29-34) were not labeled with the ray arrival times because the behaviour of the Rayleigh wave was the predominant feature.

The shear wave velocity in the layer was then lowered to 0.756. This raised Poisson's ratio to 0.3, making the layer less rigid. This is Model II. The P source record shows a much stronger P_3 arrival than Model I (Figure 35). For the S source, there is an additional peak near the direct S arrival due to the beginnings of the interface wave (Figure 37). The arrival times are marked on the plots so that the reader may make further comparisons.

Model III was constructed by changing Poisson's ratio to 0.35 in both the layer and the half-space (Figures 41-48). The response to the S source is changed little. For the P source, the amplitude of the Rayleigh wave is diminished. The effect of P_3 and P_4 is increased and the downward spike due to P_2S is enlarged by the changes in Poisson's ratio.

In Figures 27, 28, 47, and 48 the amplitude of the shear source is multiplied by 0.2 to show the effect of changing the ratio of P to S source strengths. Compare these with Figures 25, 26, 45, and 46 respectively (where there has been no multiplication).

In all the calculations done here, the Rayleigh wave amplitude is greater for an S source than for a P source.

It would seem that an SV source is a more efficient generator of Rayleigh waves. Mathematically, this occurs because the contour path lies closer to the real axis for the direct S wave than for the P wave.

TABLE 1

Models used for computation

The models referred to in the text are the following (velocities in km/sec, and densities in gm/cc). The parameters were not chosen to represent real earth values, but to use in comparing with existing calculations.

Model I

$\alpha_1 = 1.73, \quad \beta_1 = 1.00, \quad \rho_1 = 1.00, \quad \sigma_1 = 0.25$

$\alpha_2 = 1.90, \quad \beta_2 = 1.10, \quad \rho_2 = 1.65, \quad \sigma_2 = 0.25$

Model II

$\alpha_1 = 1.73, \quad \beta_1 = 0.76, \quad \rho_1 = 1.17, \quad \sigma_1 = 0.30$

$\alpha_2 = 1.90, \quad \beta_2 = 1.10, \quad \rho_2 = 1.65, \quad \sigma_2 = 0.25$

Model III

$\alpha_1 = 2.16, \quad \beta_1 = 1.00 \quad \rho_1 = 1.00, \quad \sigma_1 = 0.35$

$\alpha_2 = 2.29, \quad \beta_2 = 1.10, \quad \rho_2 = 1.65, \quad \sigma_2 = 0.35$

In all these cases, the thickness H of the layer is one (1.) km. Unless otherwise noted, the source depth is 0.5 km.

COMPUTER PLOTS

In the figure captions, the first entry refers to the component (r or z), the second to the range, the third to the model (I, II, or III), and the fourth to the source type (P, S or C for the combination of P and S). The P and S source strengths are equal unless otherwise noted. The source depth is 0.5 km unless otherwise noted. The letters P, S, PS, etc. refer to the arrival times of rays with specified number of multiple reflections in the layer.

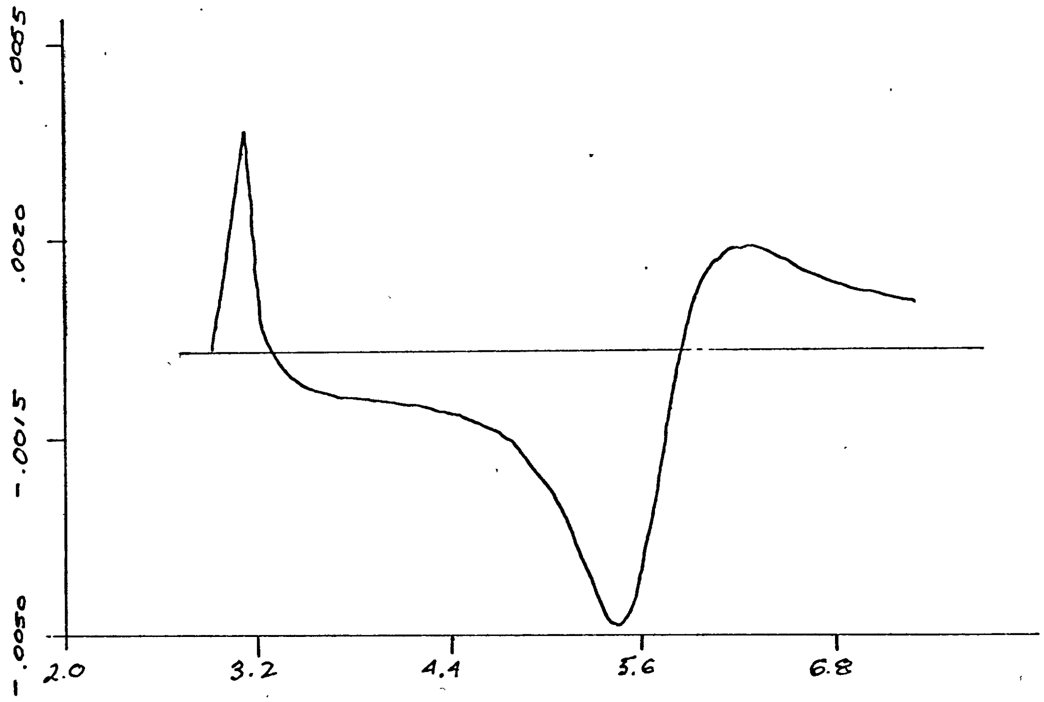


Fig. 1: z,5,I,P Lamb's problem by approximate program.

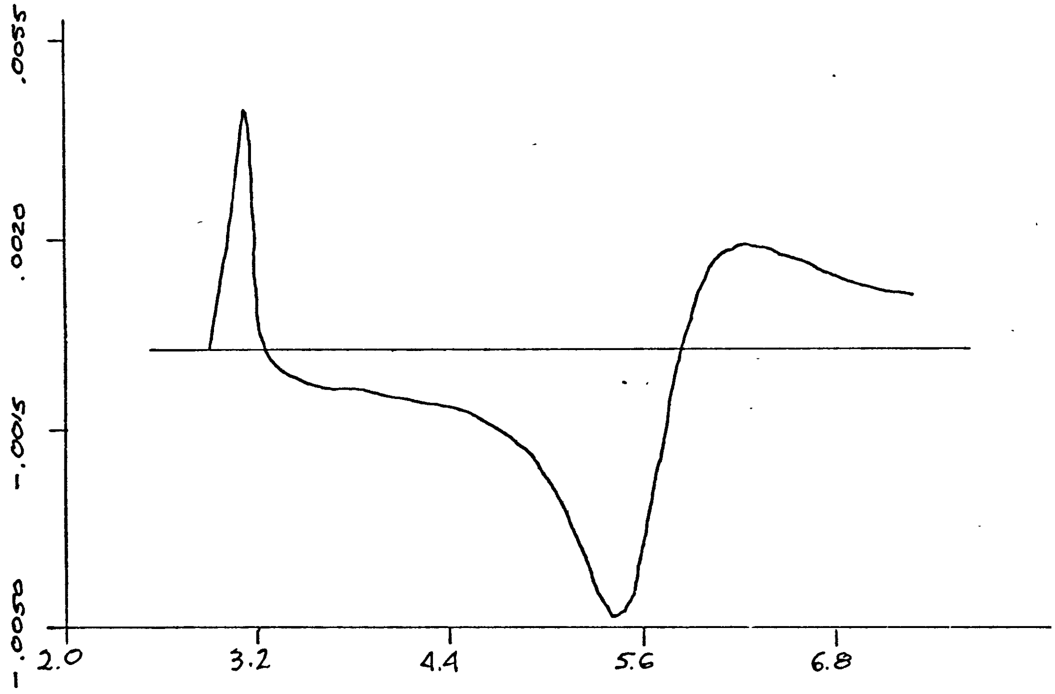


Fig. 2: z,5,I,P Lamb's problem by exact program.

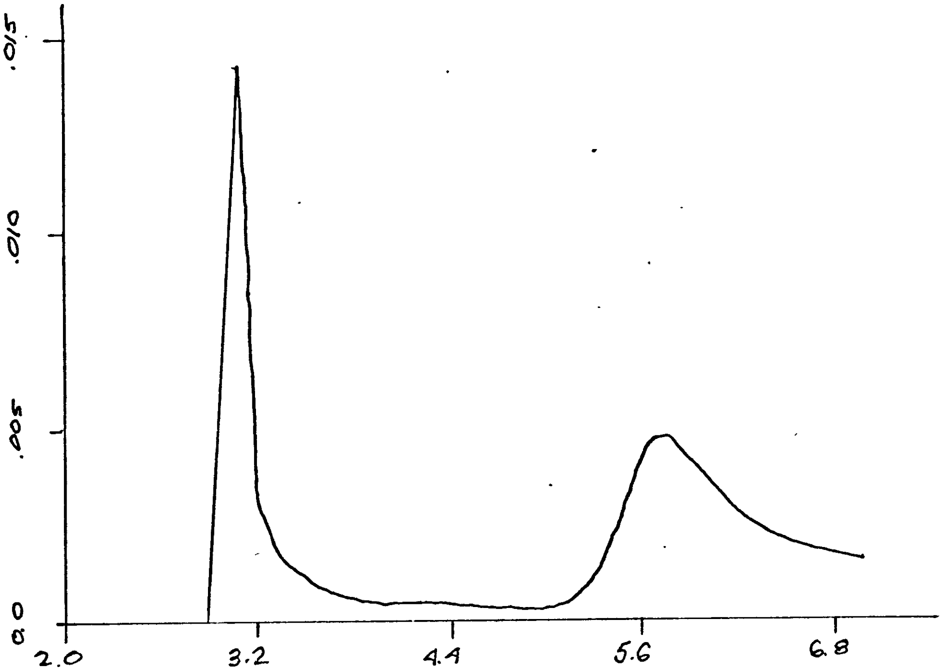


Fig. 3: r,5,I,P Lamb's problem by approximate program.

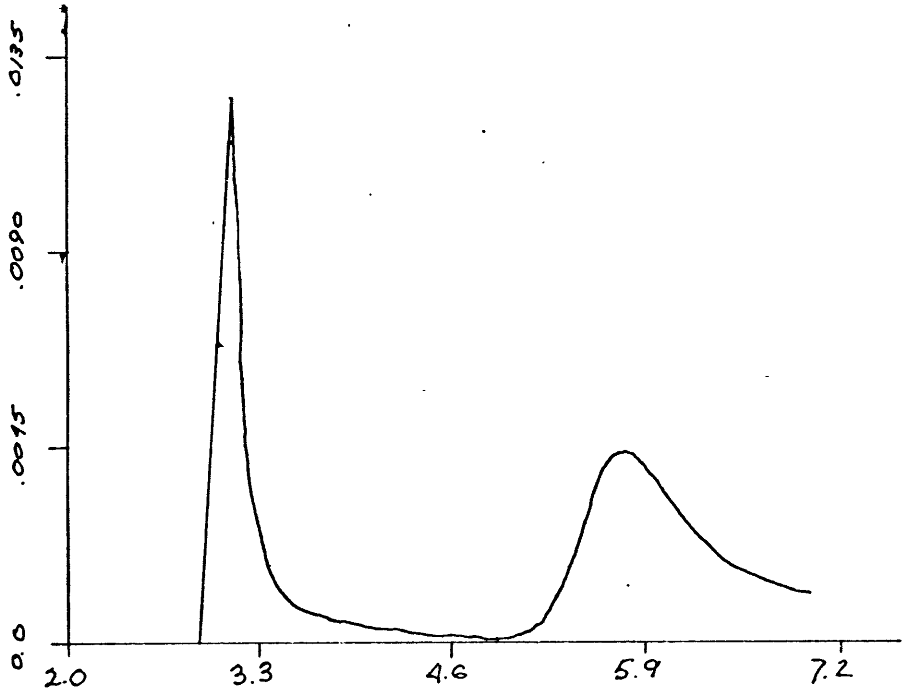


Fig. 4: r,5,I,P Lamb's problem by exact program.

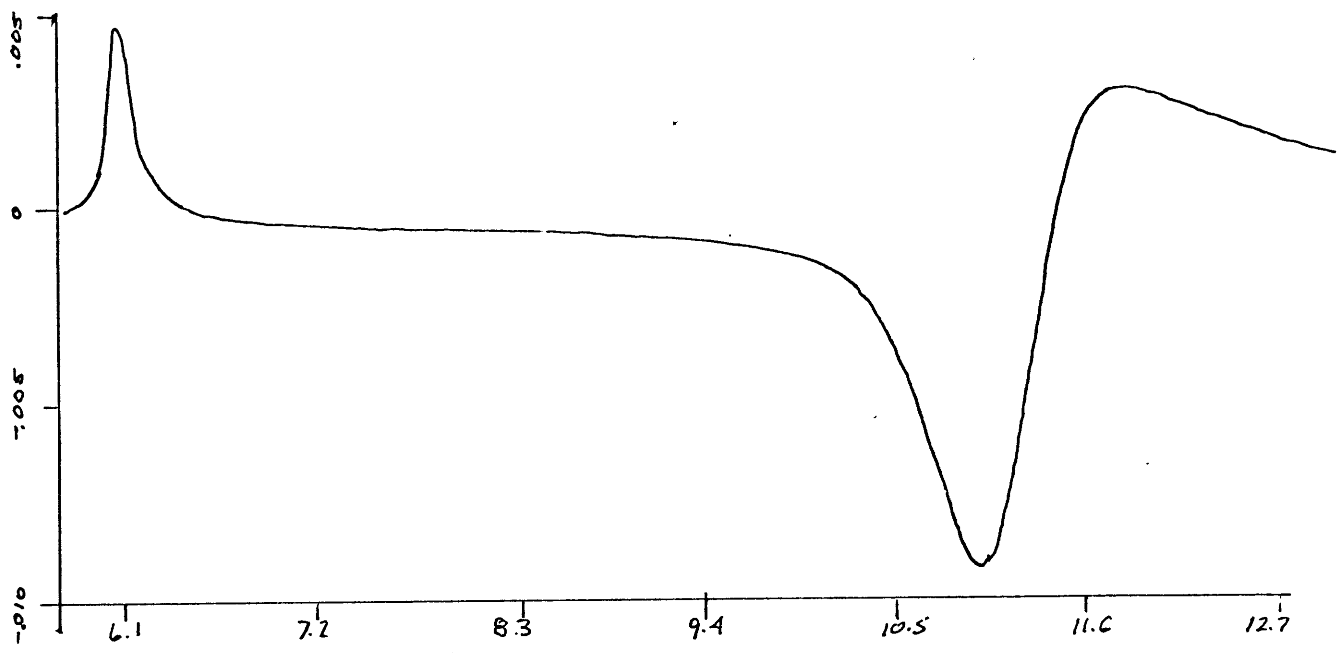


Fig. 5: z,10,I,P Lamb's problem by approximate program.

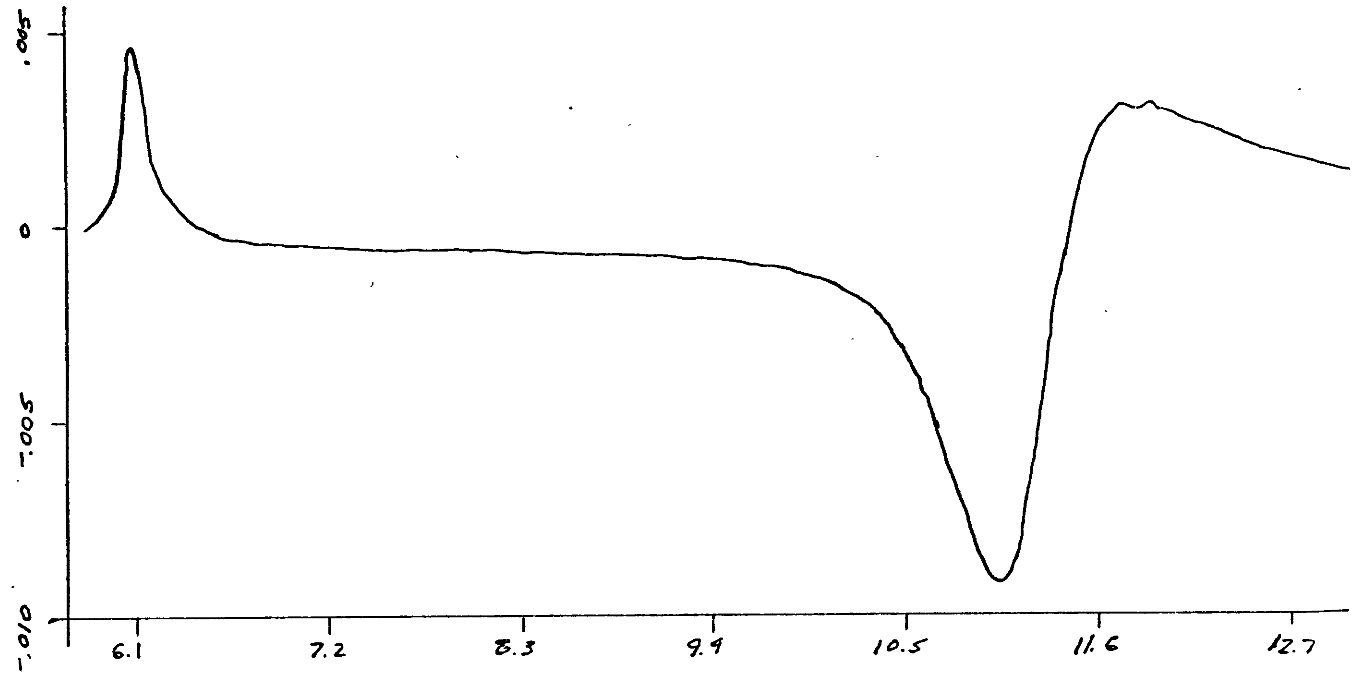


Fig. 6: z,10,I,P Lamb's problem by exact program.

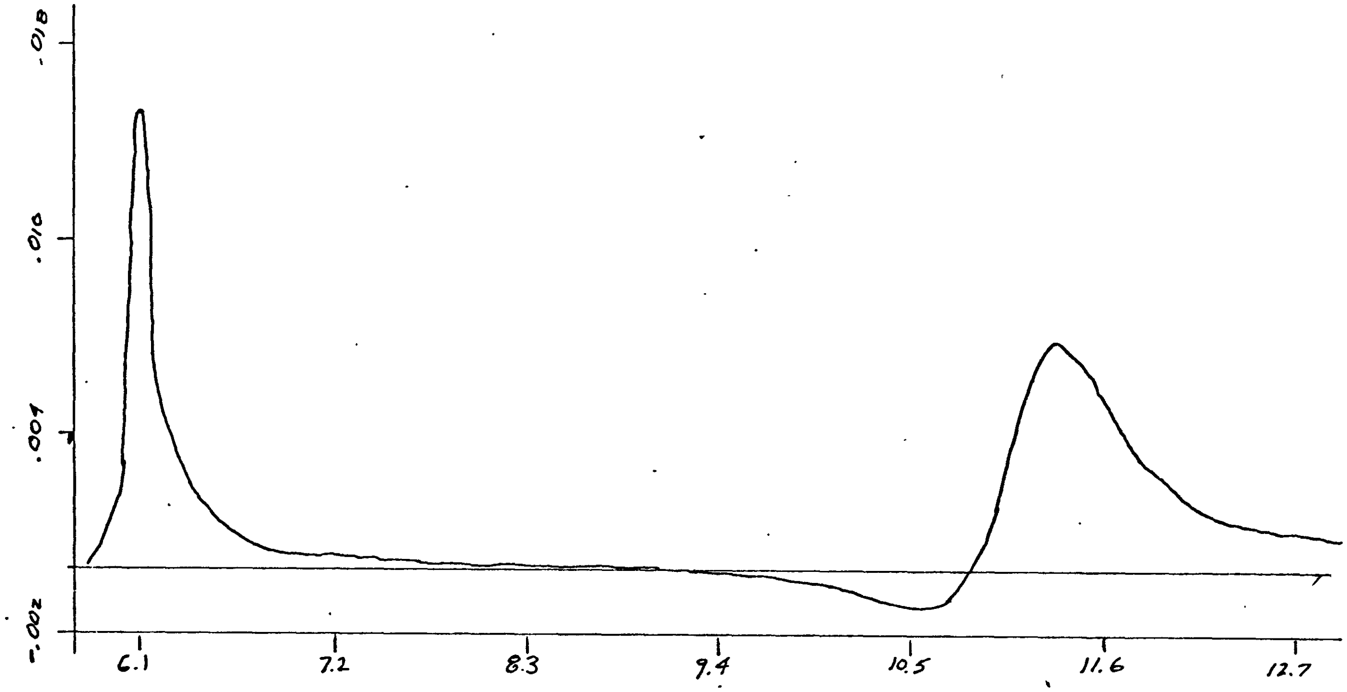


Fig. 7: r,10,I,P Lamb's problem by approximate program.

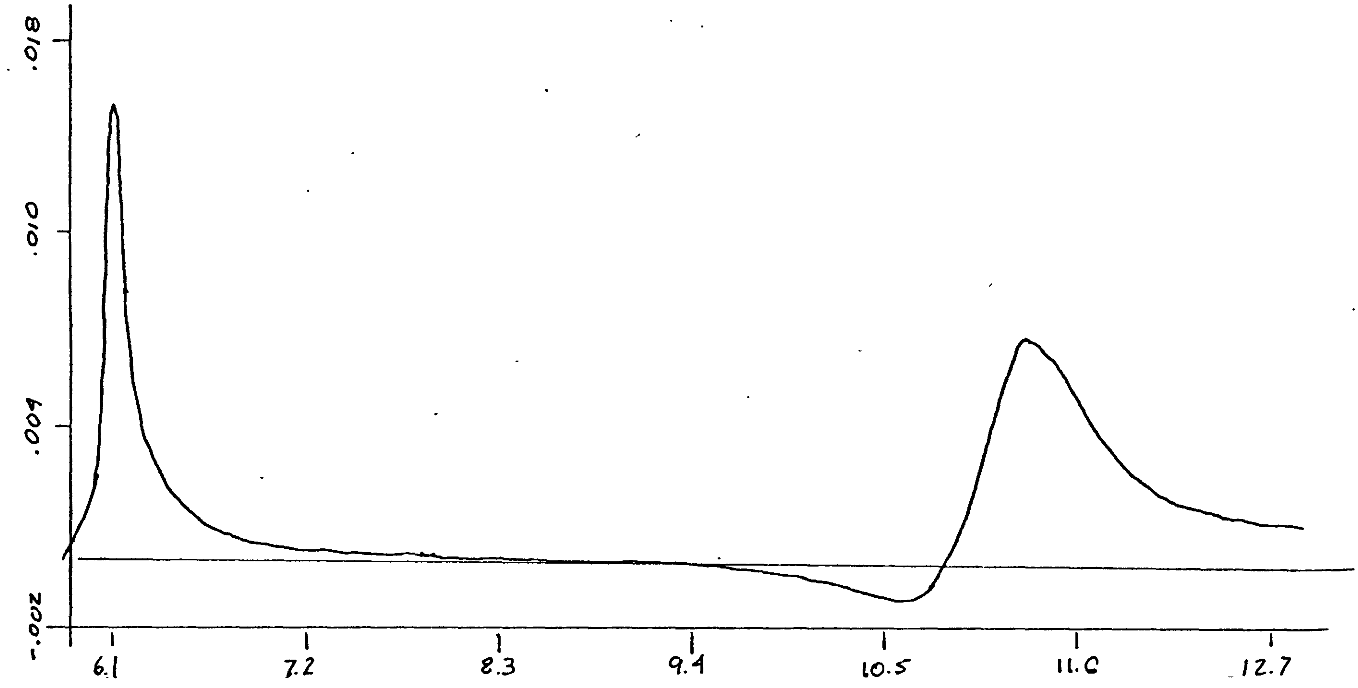


Fig. 8: r,10,I,P Lamb's problem by exact program.

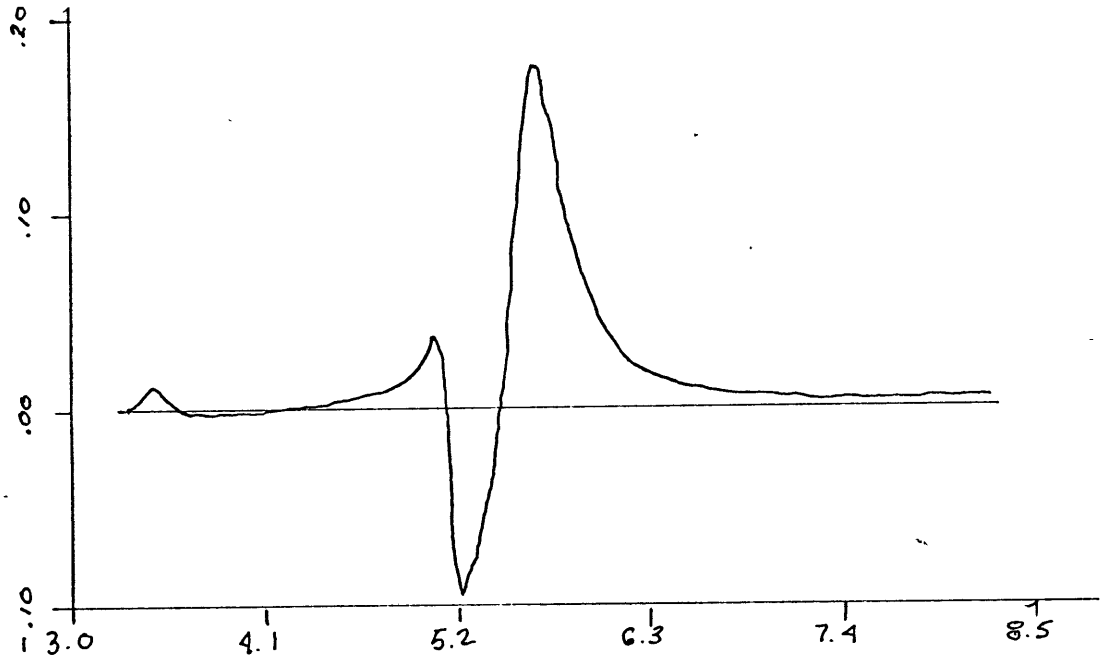


Fig. 9: z,5,I,S Lamb's problem

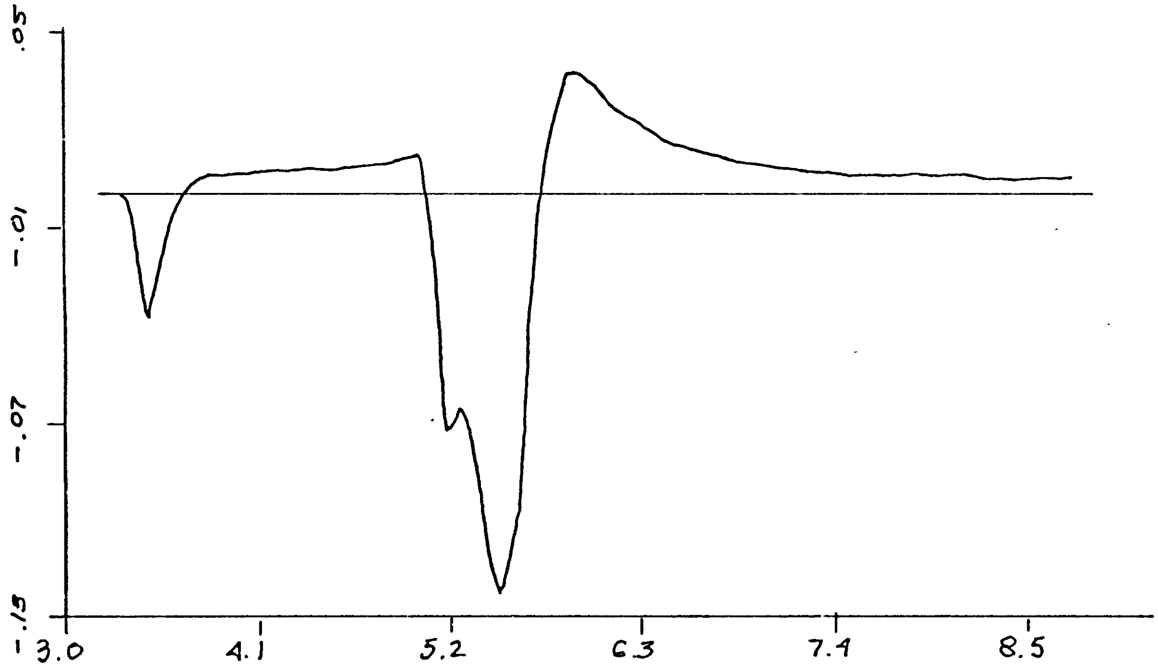


Fig. 10: r,5,I,S

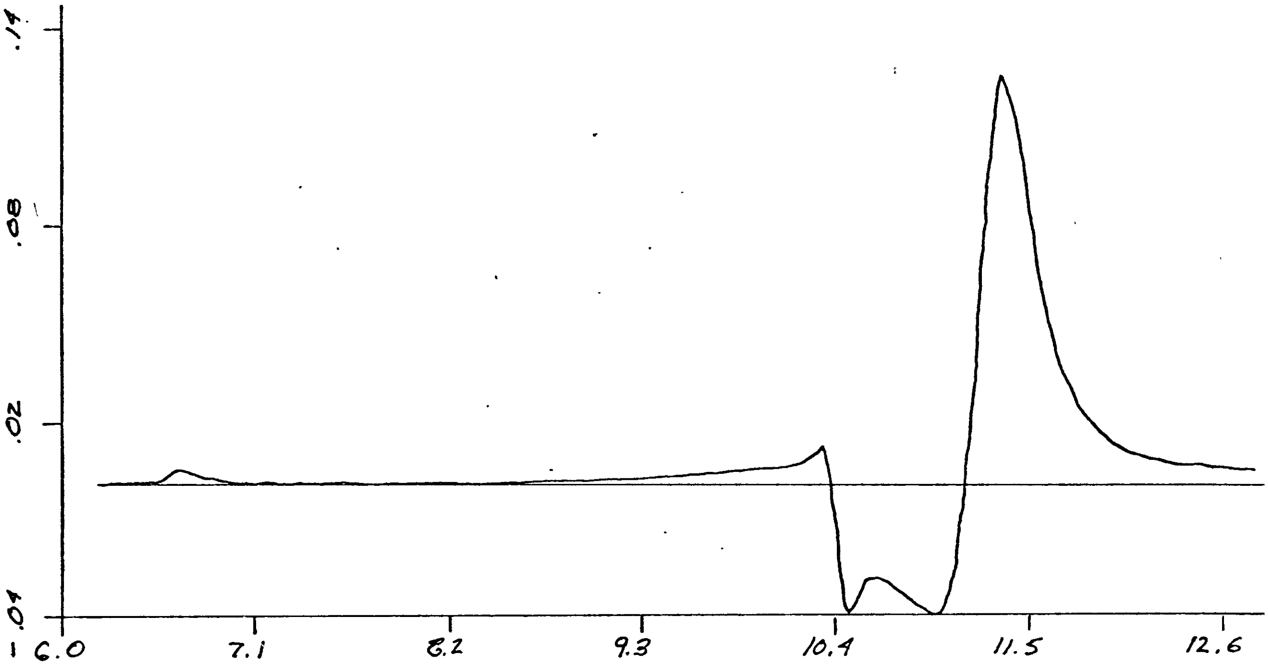


Fig. 11: z,10,I,S Lamb's problem

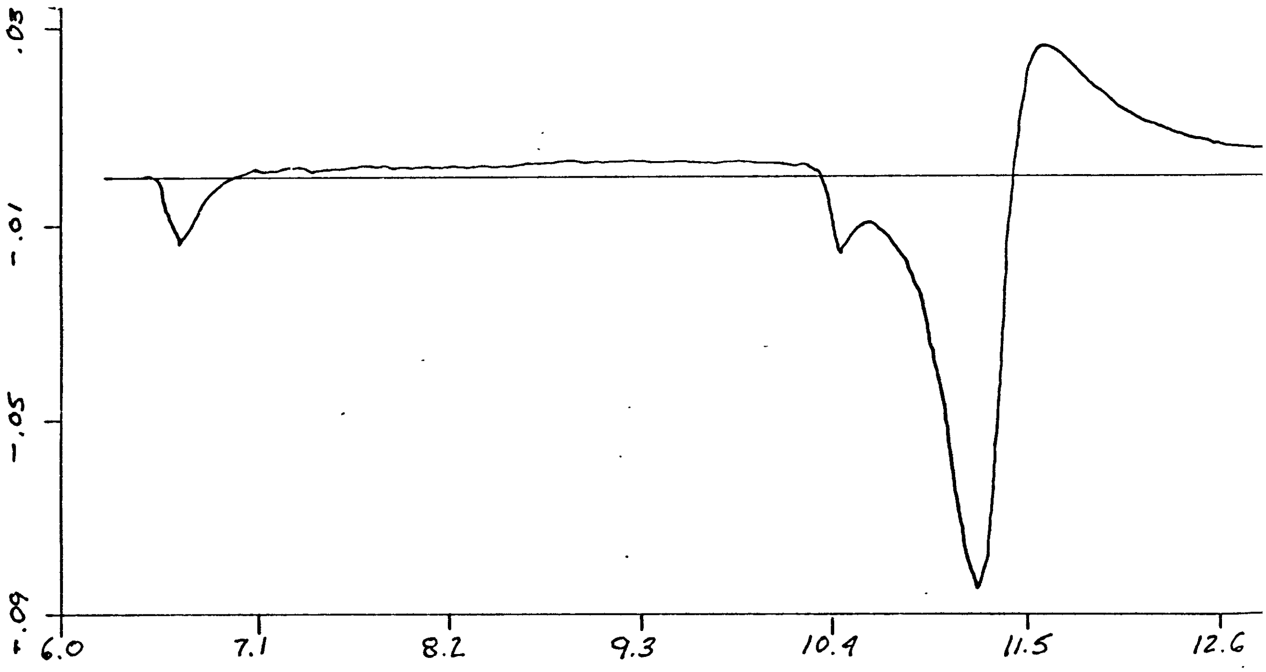


Fig. 12: r,10,I,S Lamb's problem

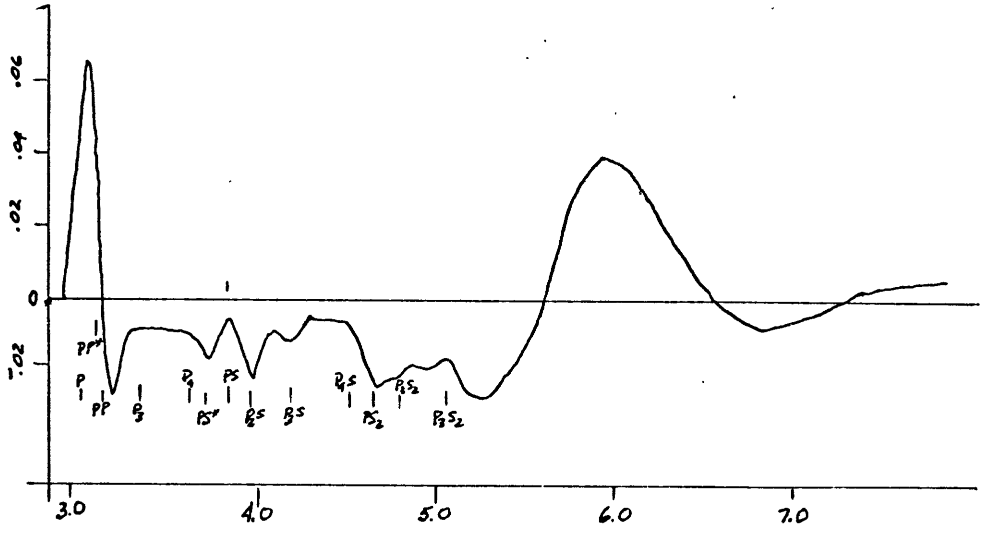


Fig. 13: z,5,I,P Exact solution according to Pekeris,et, al.

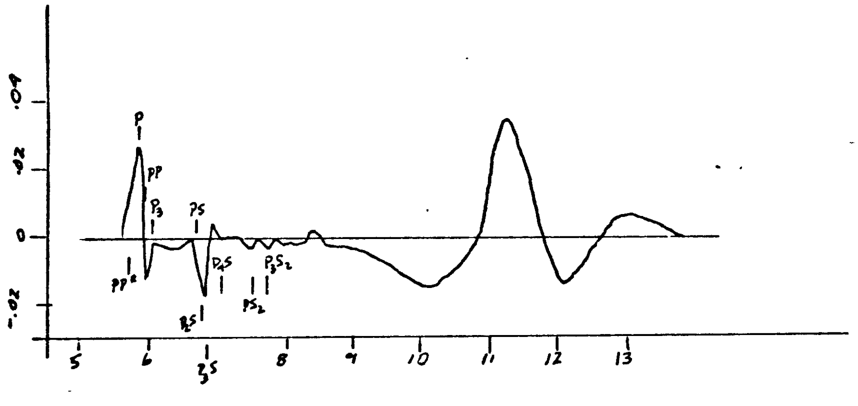


Fig. 14: z,10,I,P Exact solution according to Pekeris, et. al.

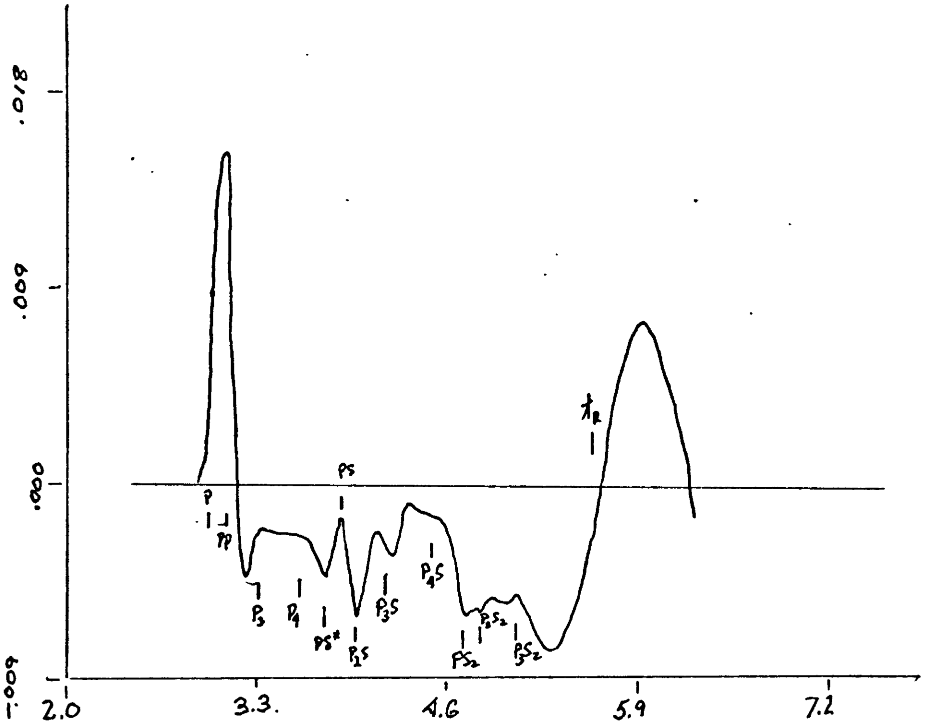


Fig. 15: z,5,I,P

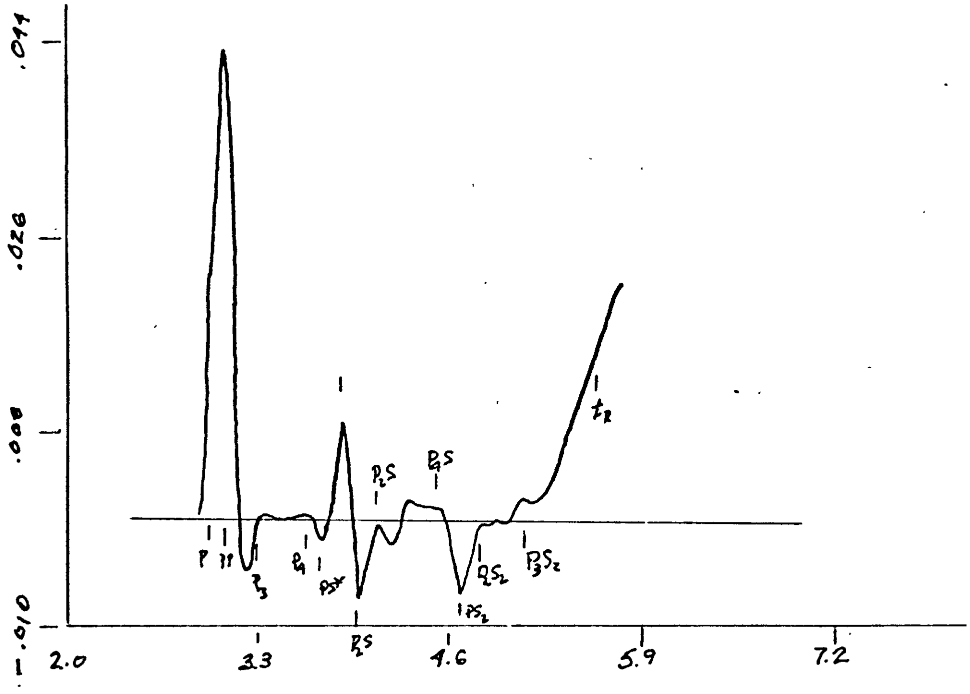


Fig. 16: r,5,I,P

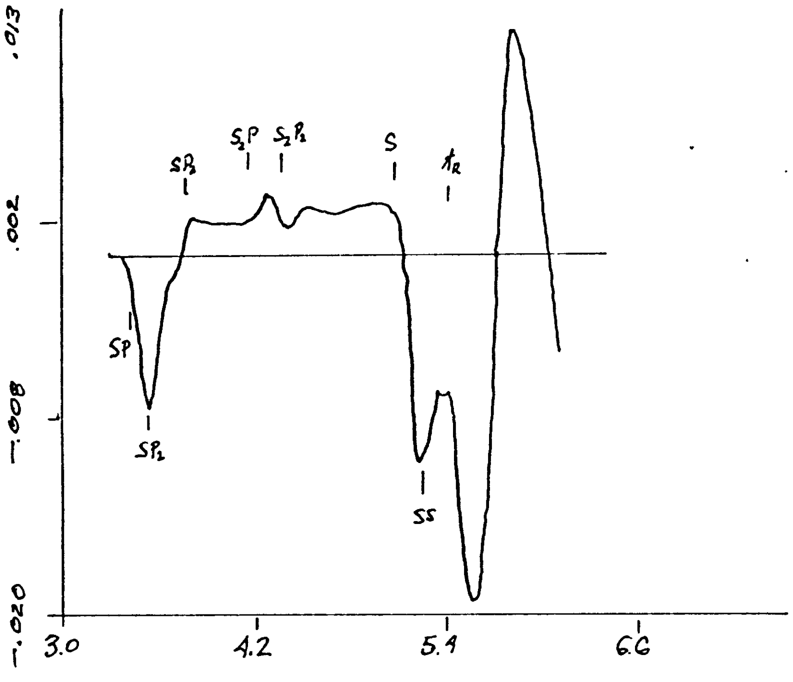


Fig. 17: r,5,I,S

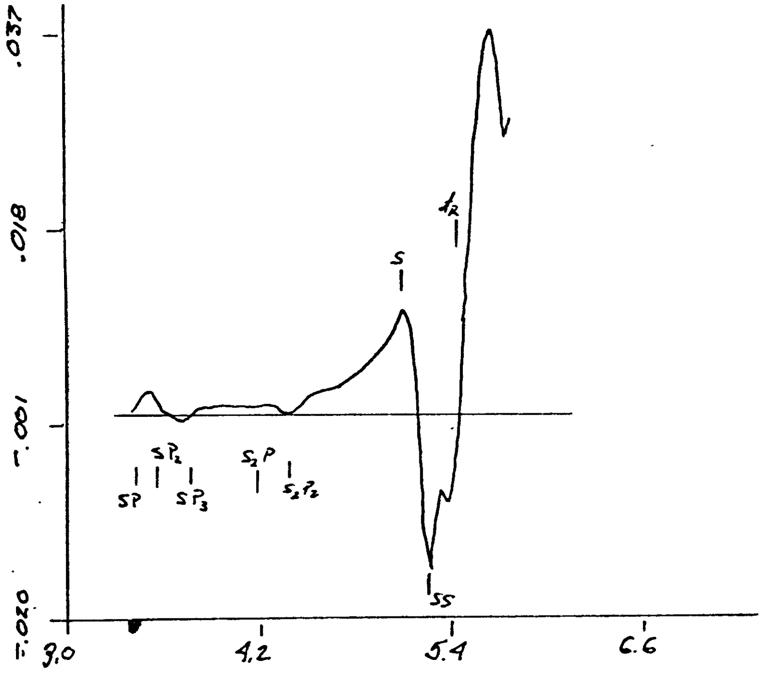


Fig. 18: z,5,I,S

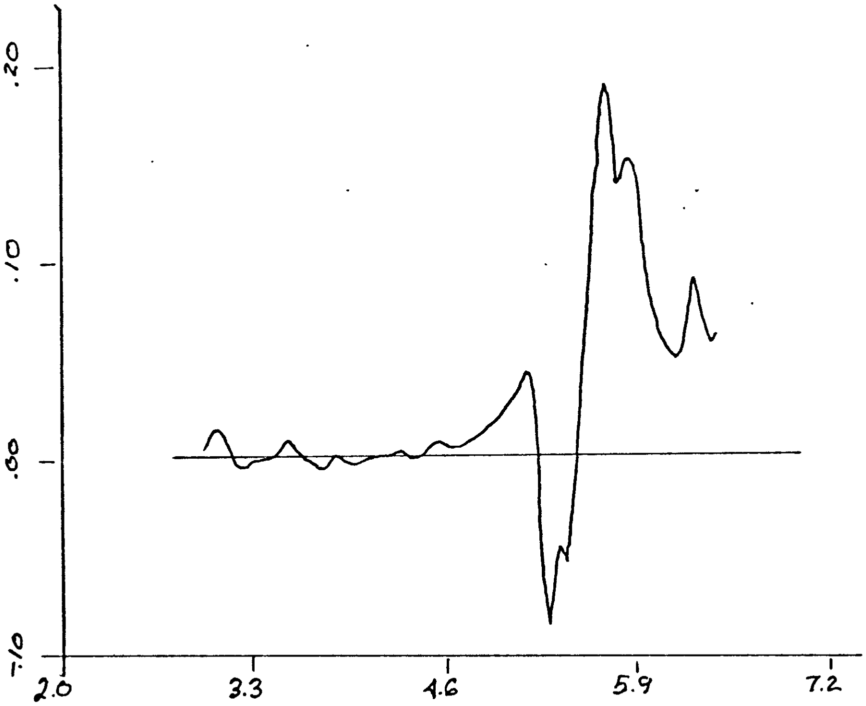


Fig. 19: z,5,I,c

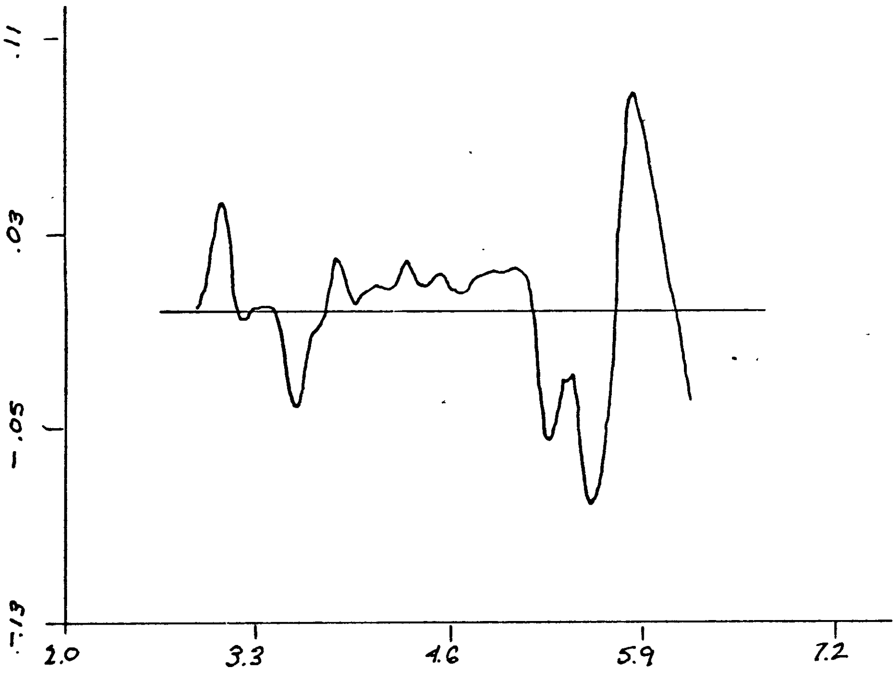


Fig. 20: r,5,I,C

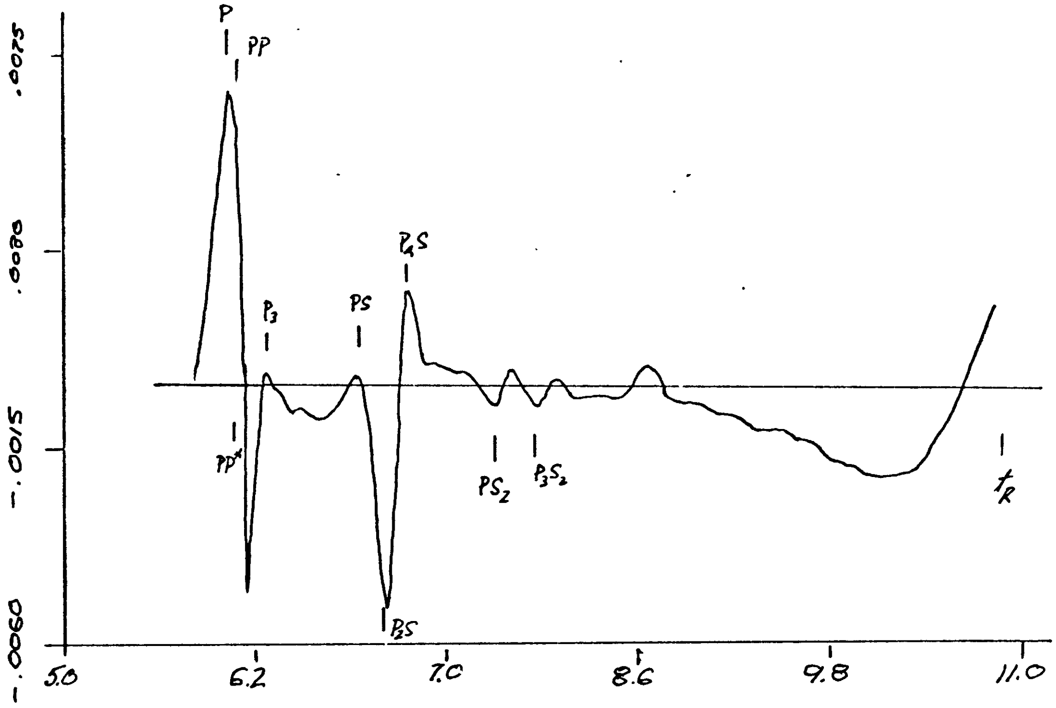


Fig. 21: z,10,I,P

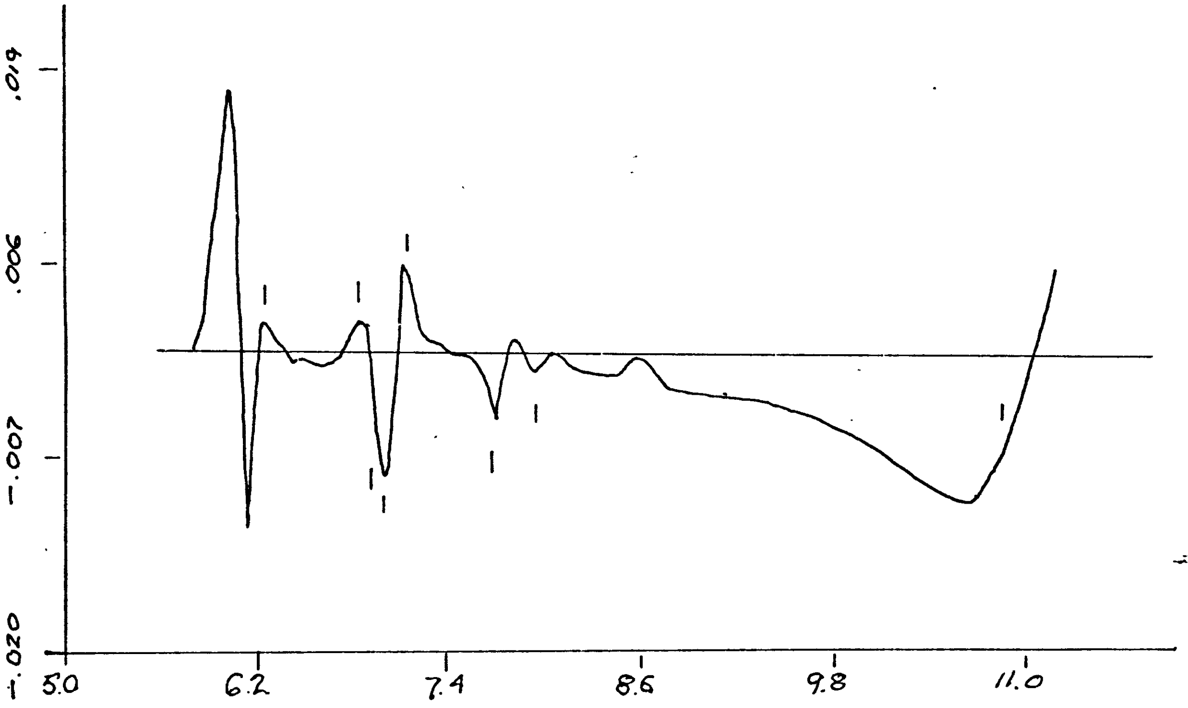


Fig. 22: r,10,I,P

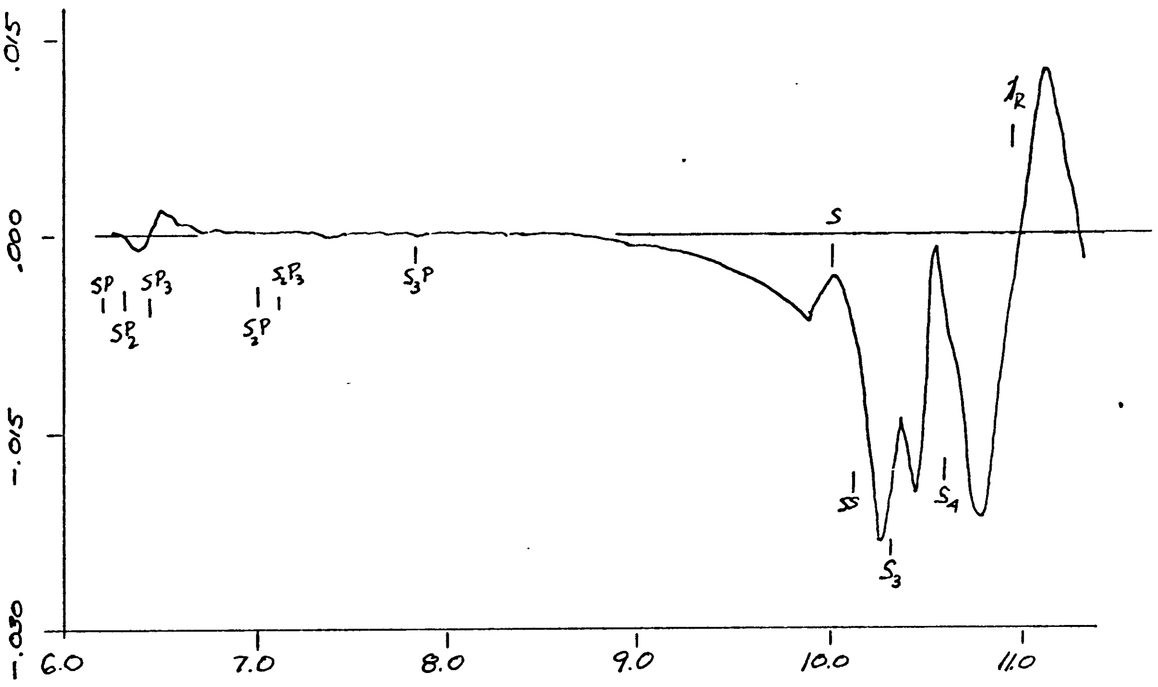


Fig. 23: z,10,I,S

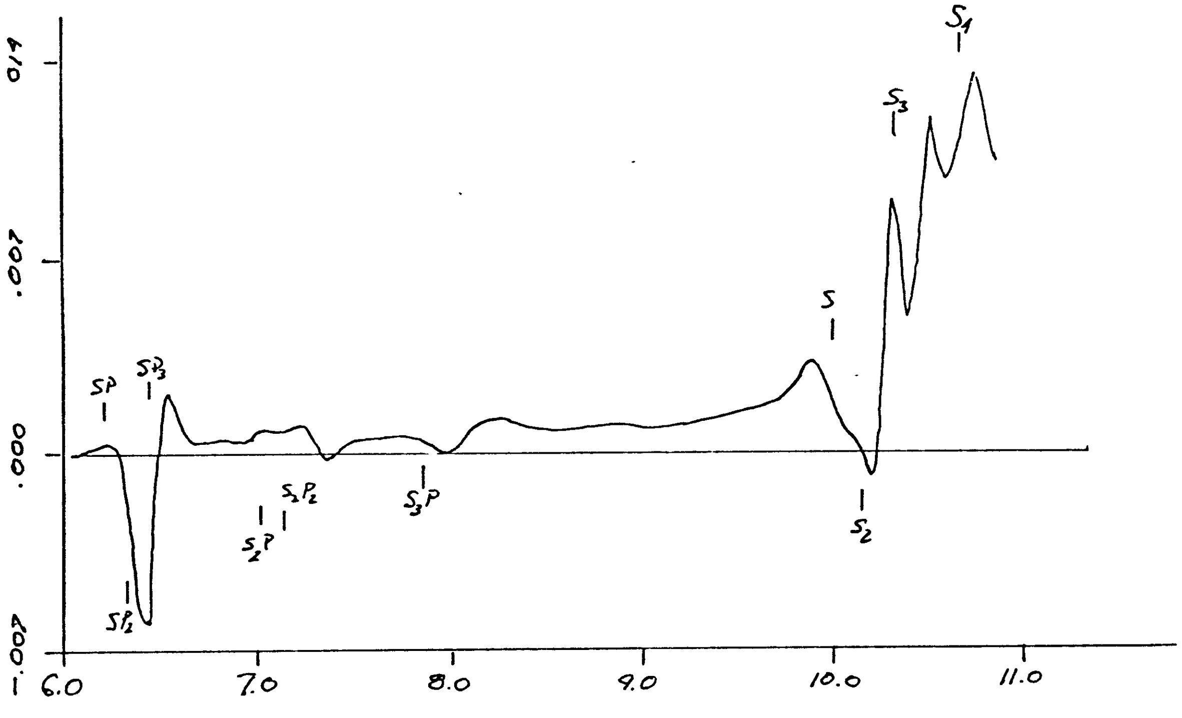


Fig. 24: r,10,I,S

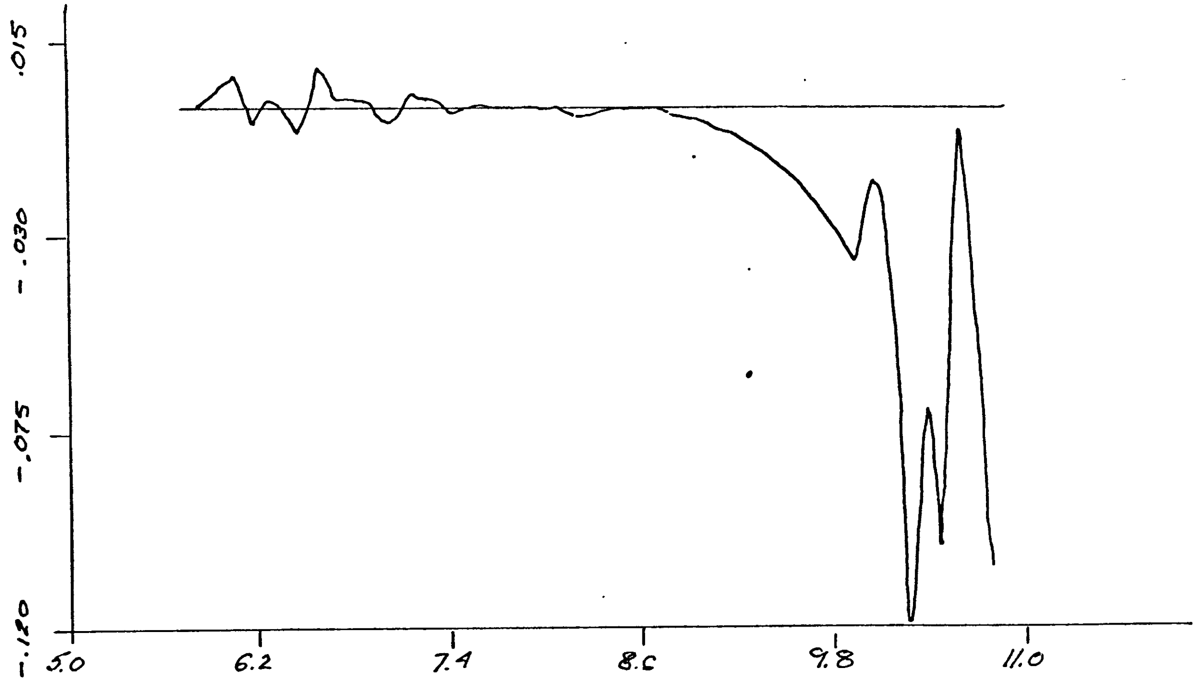


Fig. 25: z,10,I,C

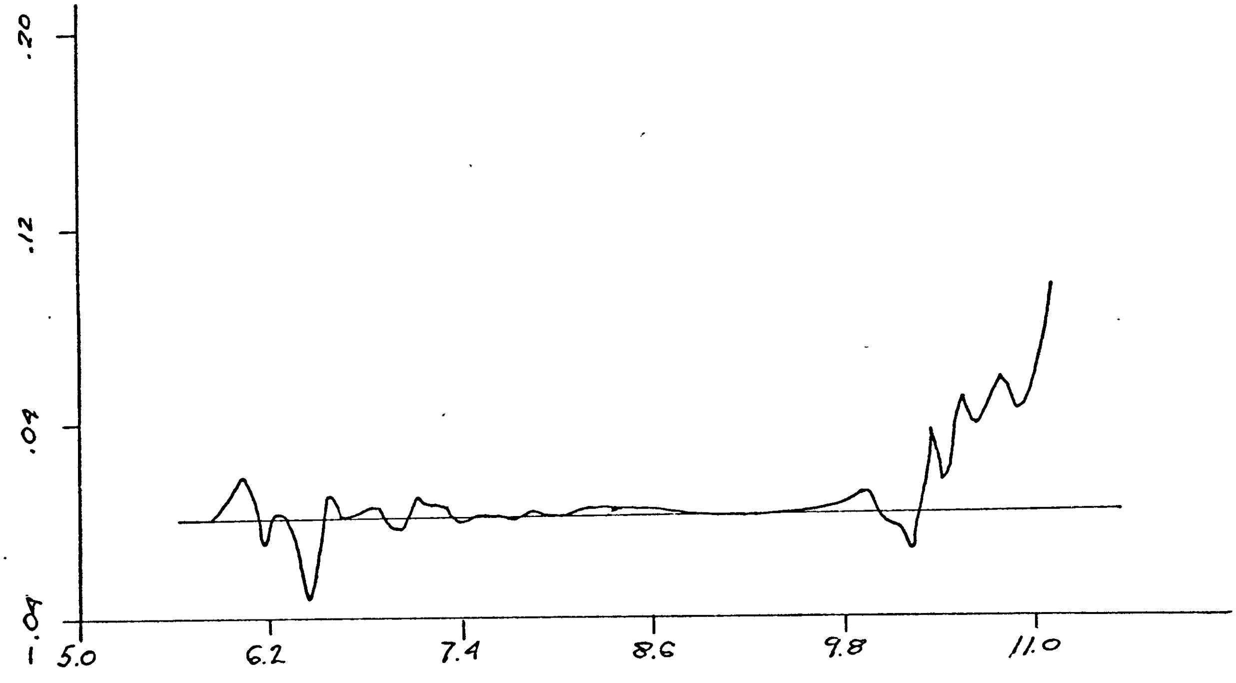


Fig. 26: r,10,I,C

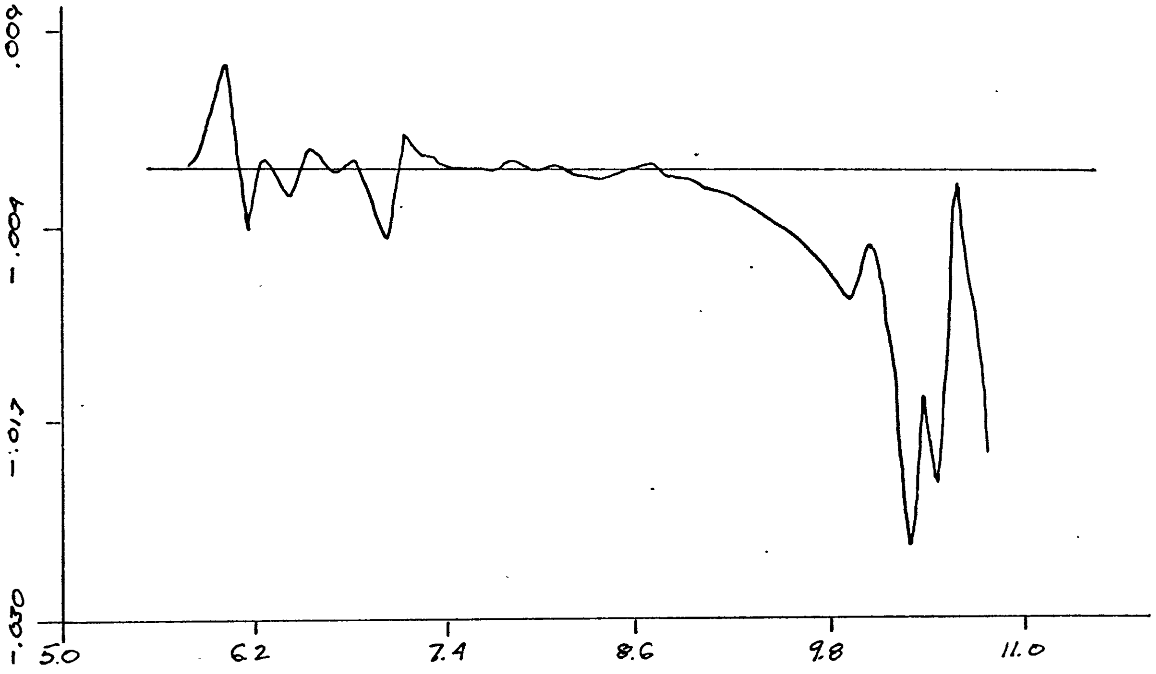


Fig. 27: z,10,I,C Shear source has been multiplied by 0.2 .

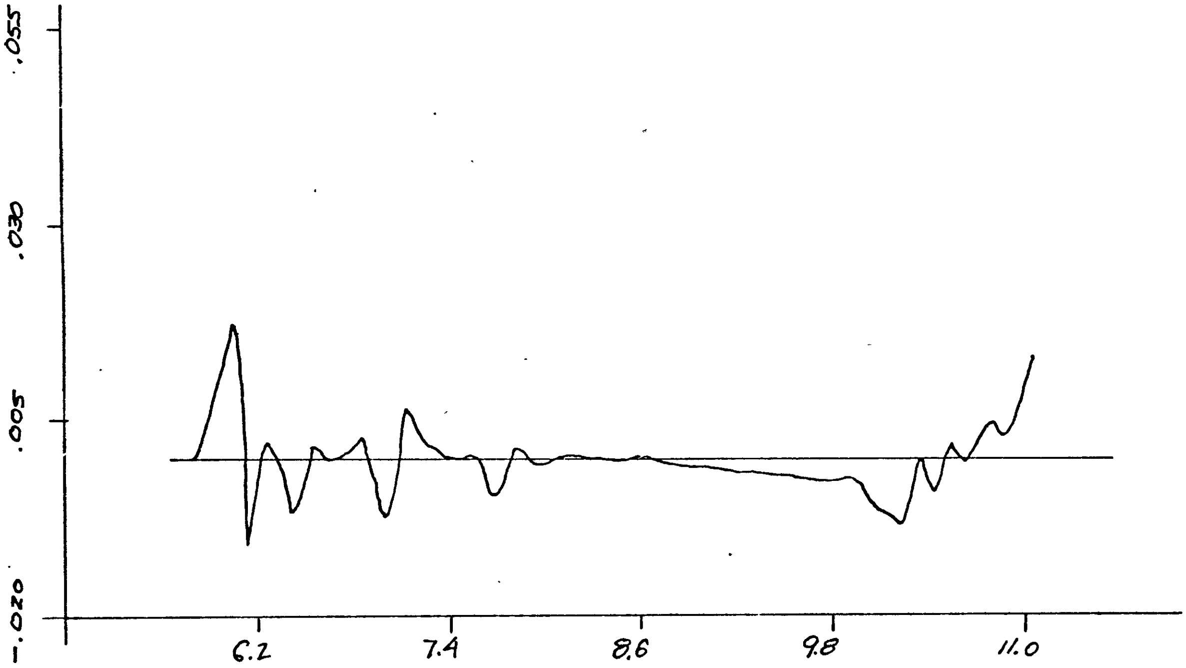


Fig. 28: r,10,I,C Shear source has been multiplied by 0.2 .

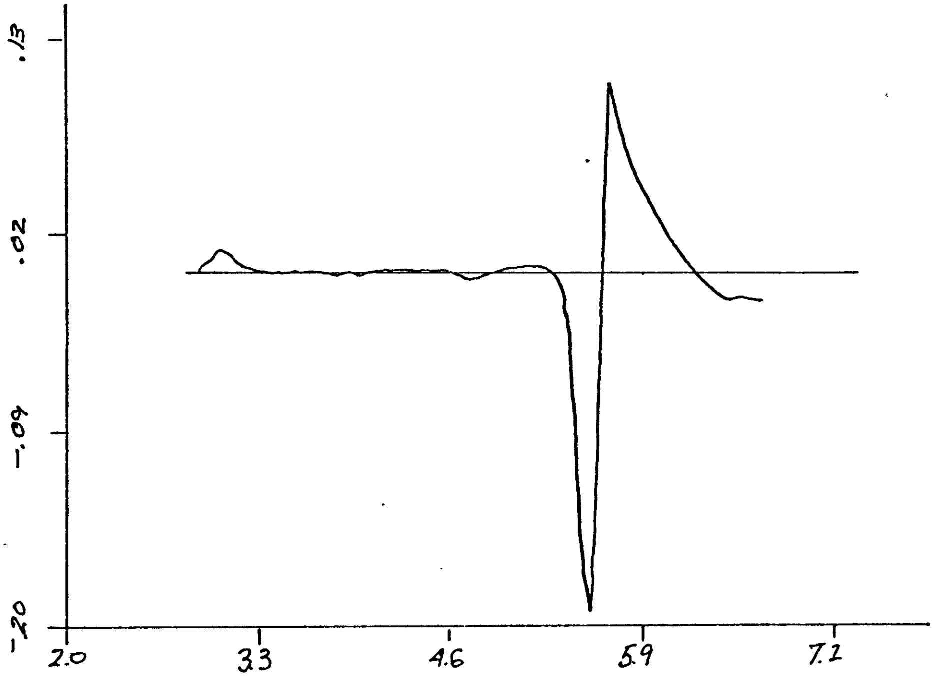


Fig. 29: z,5,I,P, h=0.05

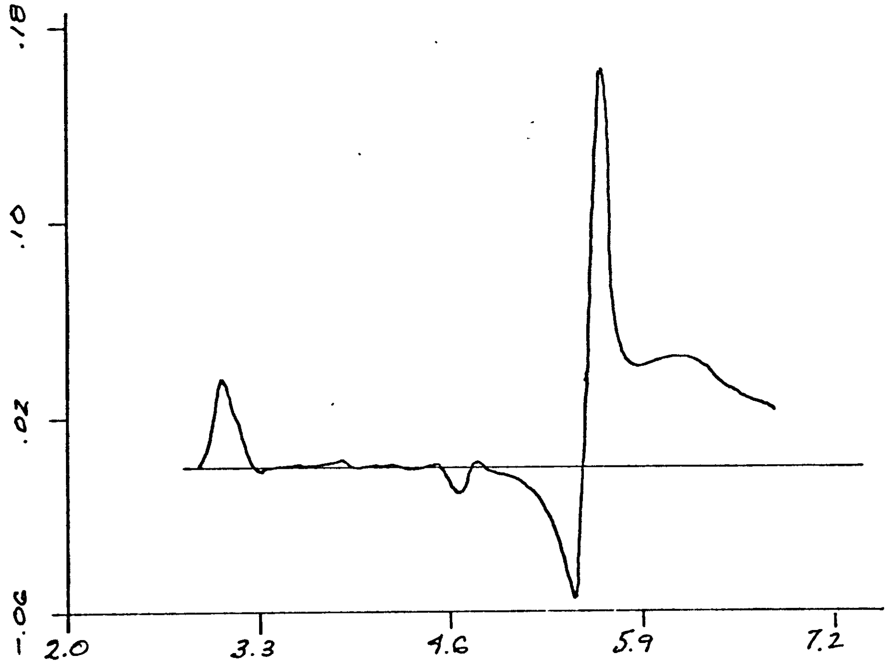


Fig. 30: r,5,I,P, h=0.05

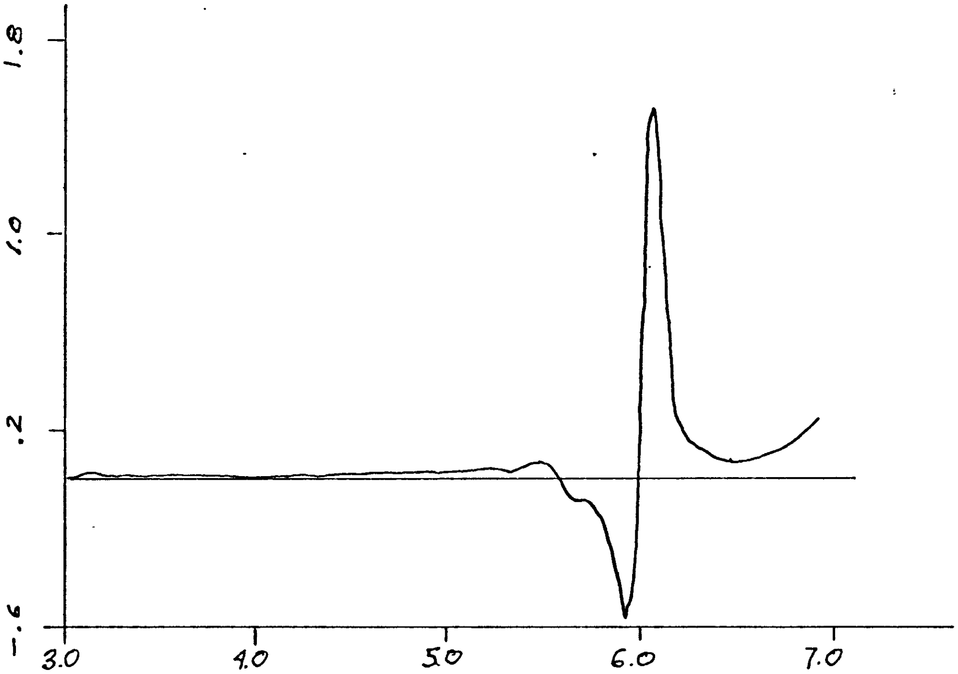


Fig. 31: z,5,I,S, h=0.05

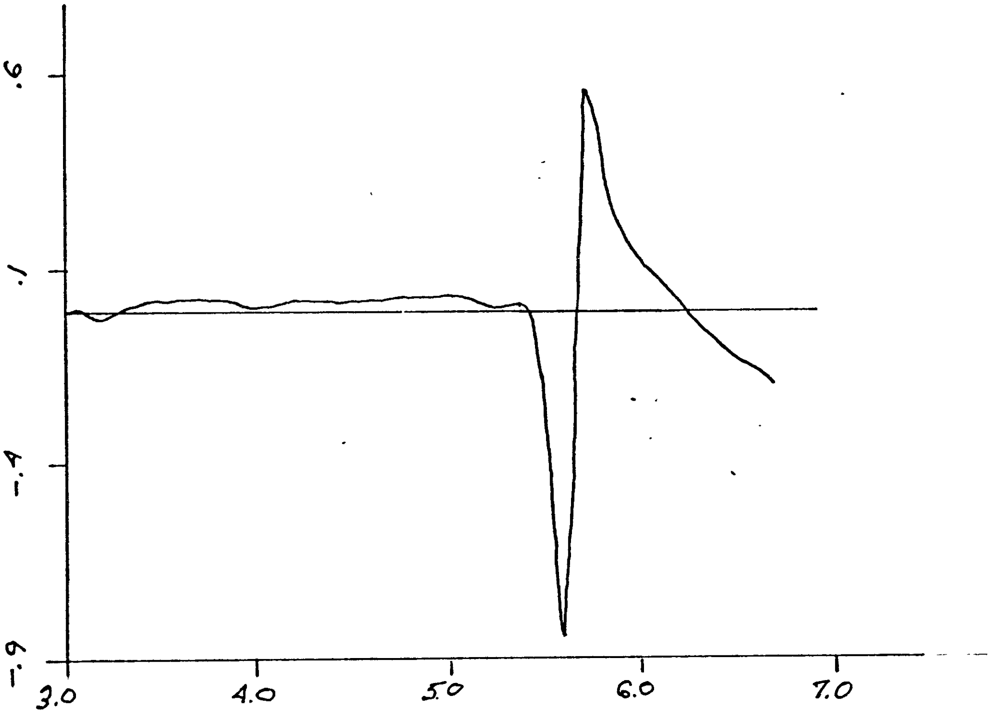


Fig. 32: r,5,I,S, h=0.05

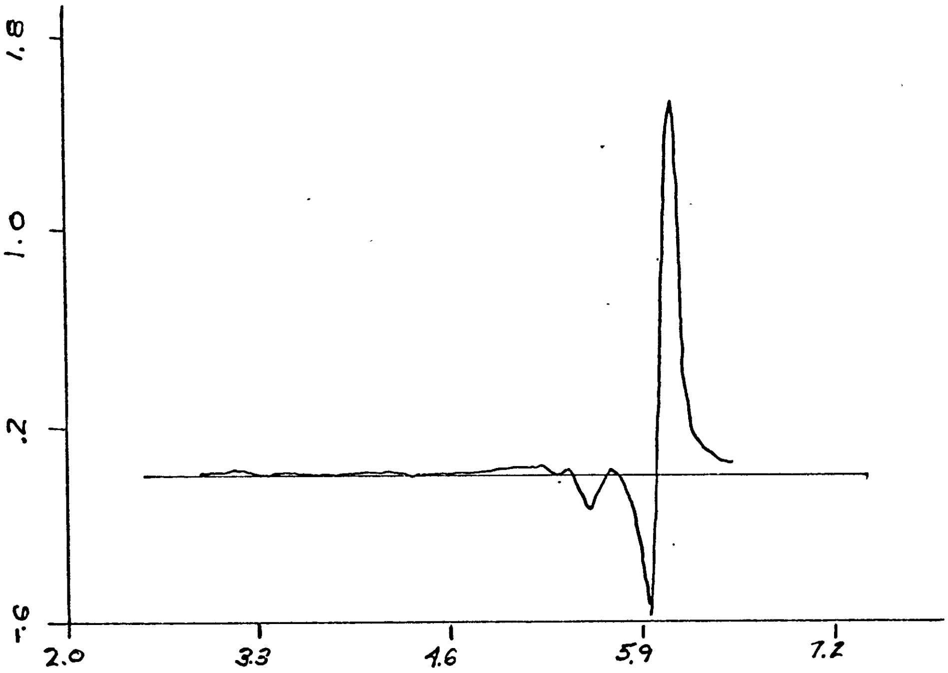


Fig. 33: z,5,I,C, h=0.05

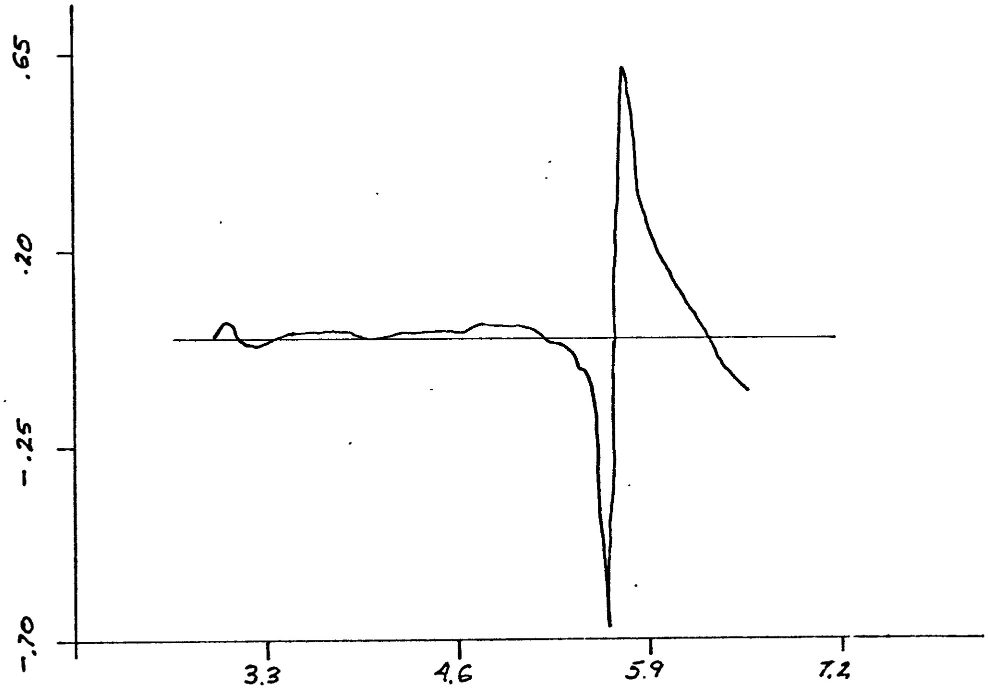


Fig. 34: r,5,I,C, h=0.05

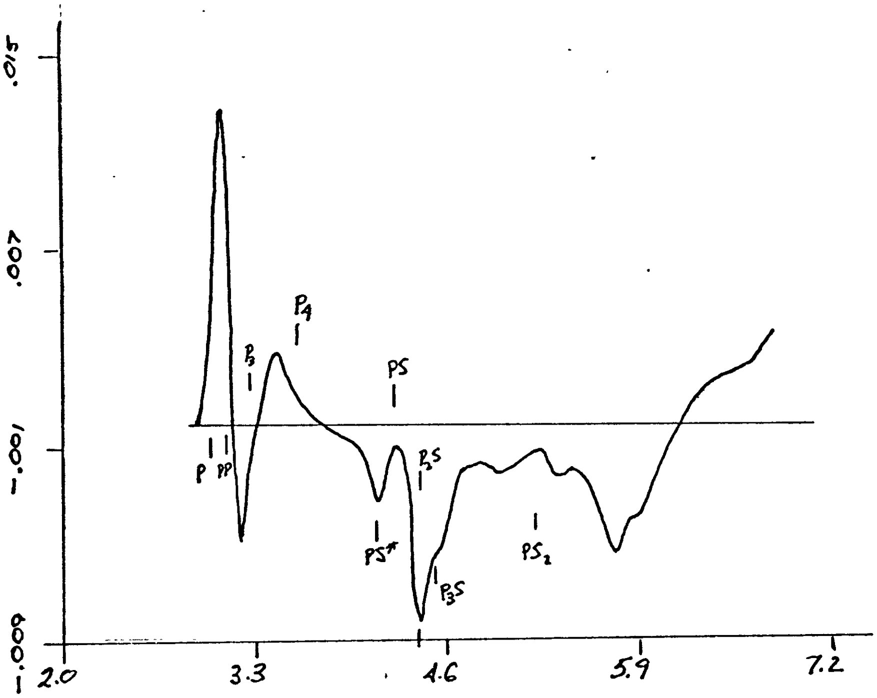


Fig. 35: z,5,II,P

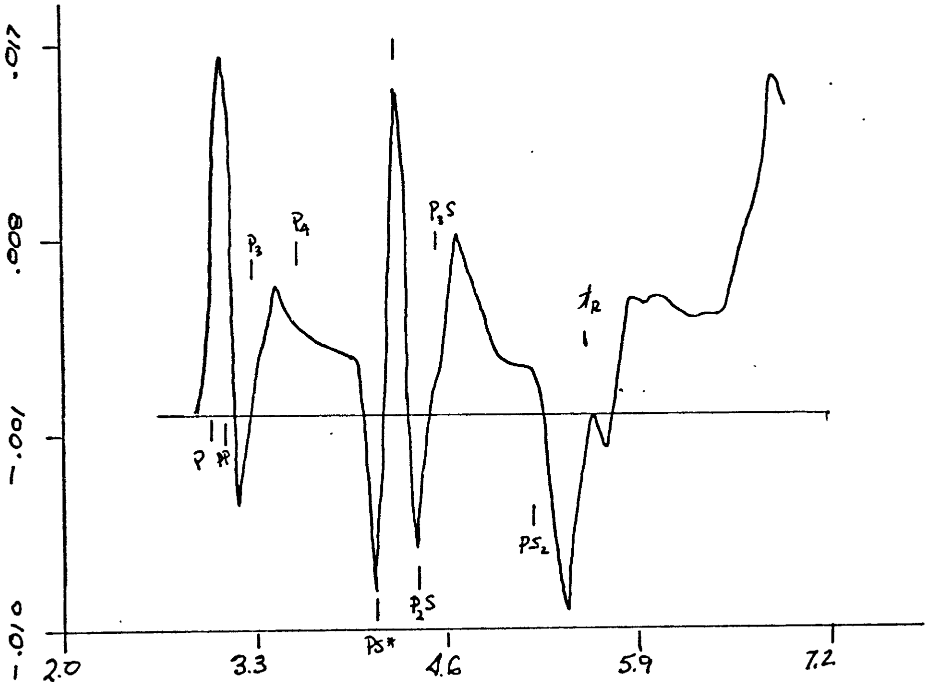


Fig. 36: r,5,II,P

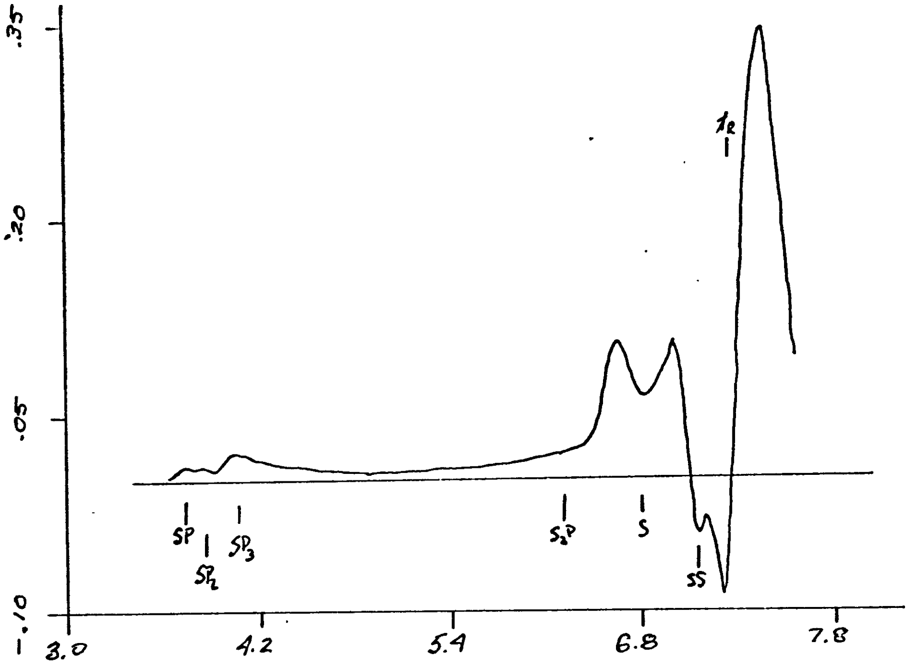


Fig. 37: z,5,II;S

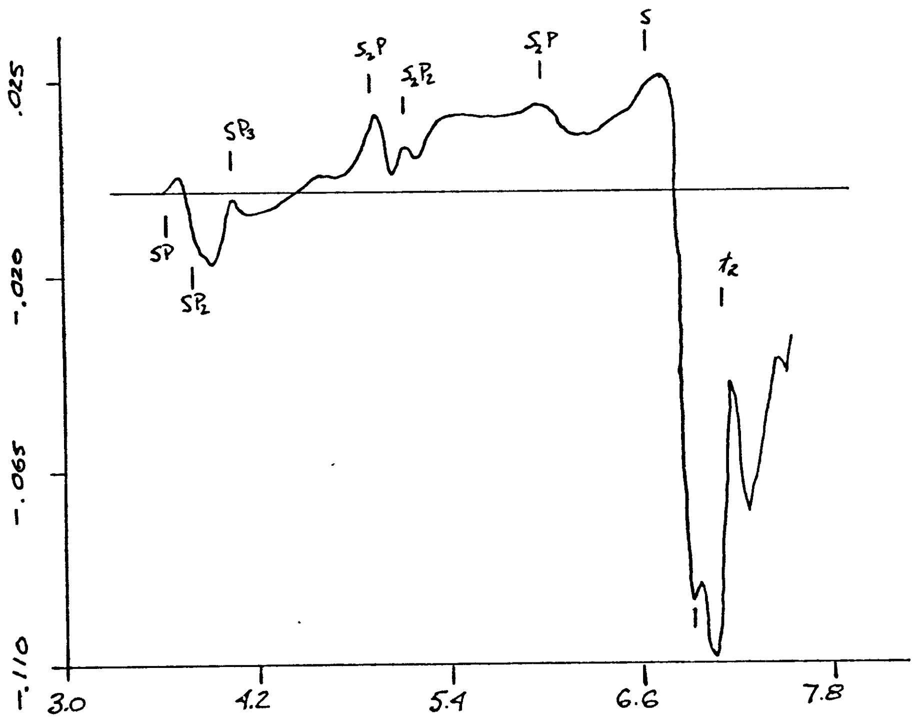


Fig. 38: r,5,II,S

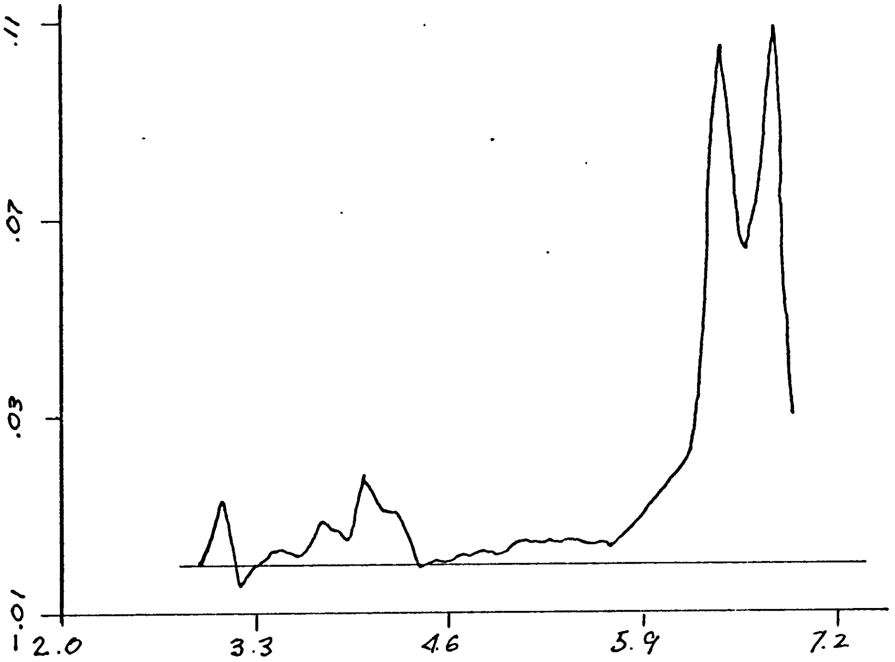


Fig. 39: z,5,II,C

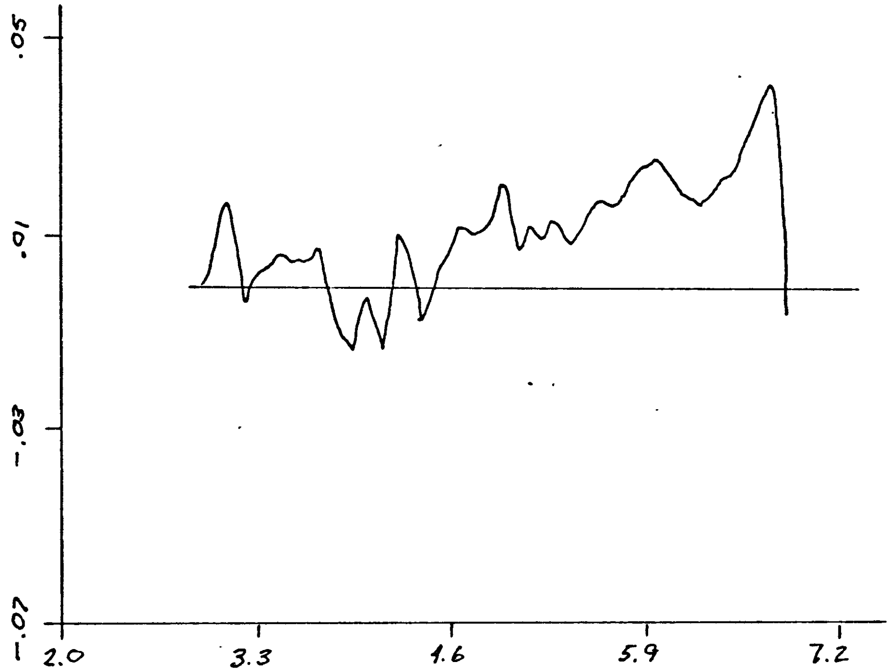


Fig. 40: r,5,II,C

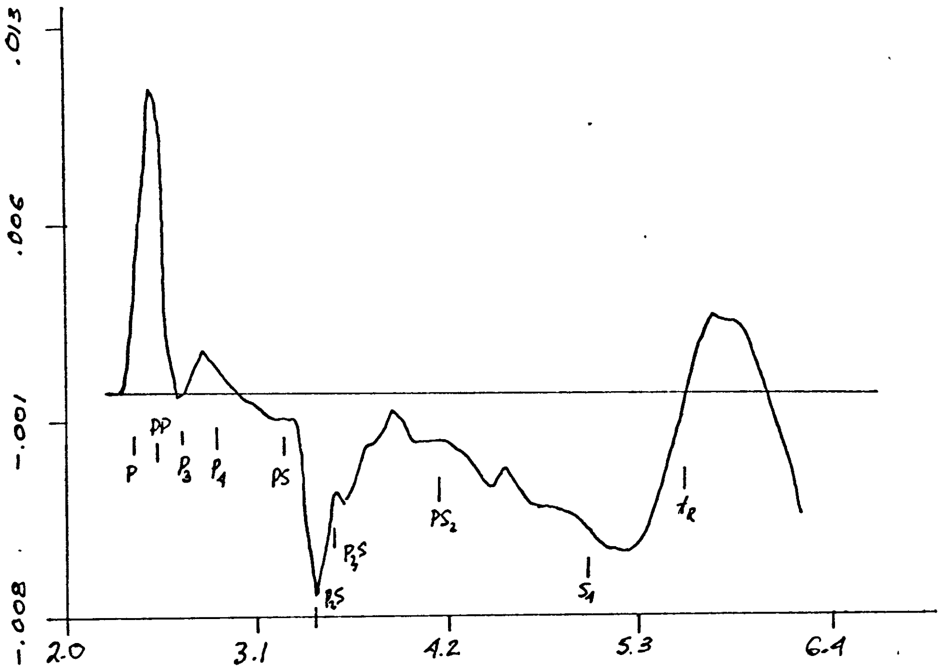


Fig. 41: z,5,III,P

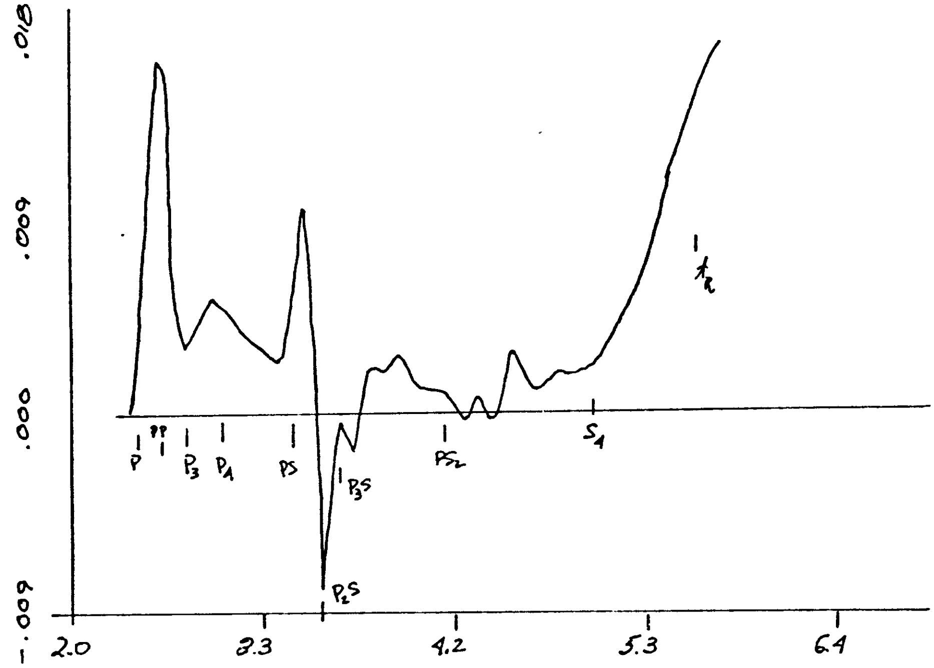


Fig. 42: r,5,III,P

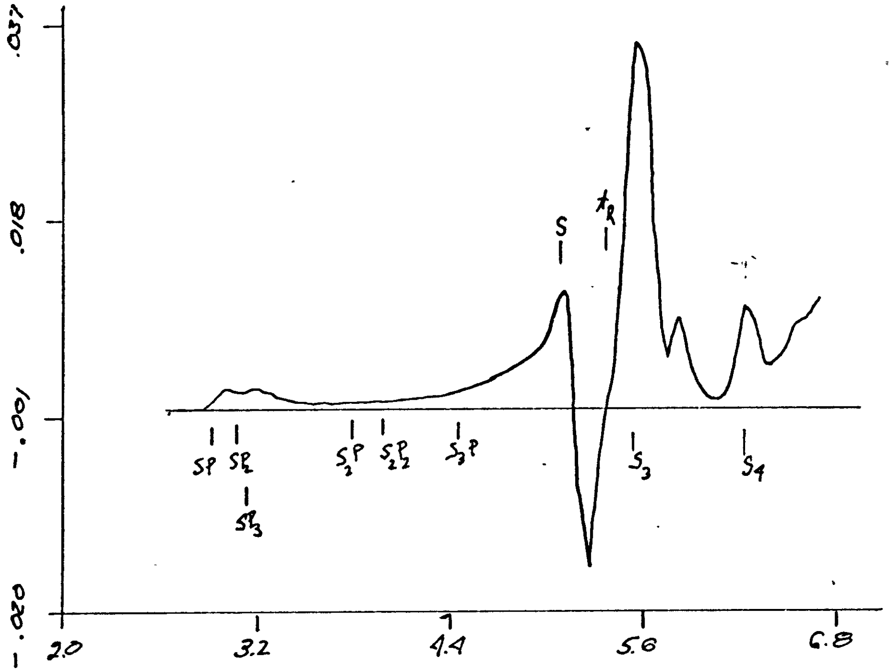


Fig. 43: z,5,III,S

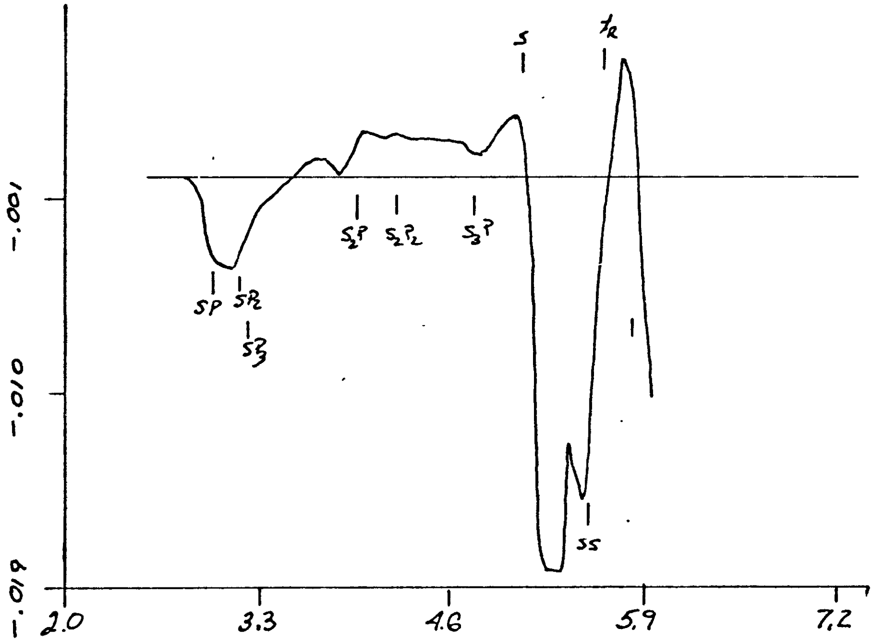


Fig. 44: r,5,III,S

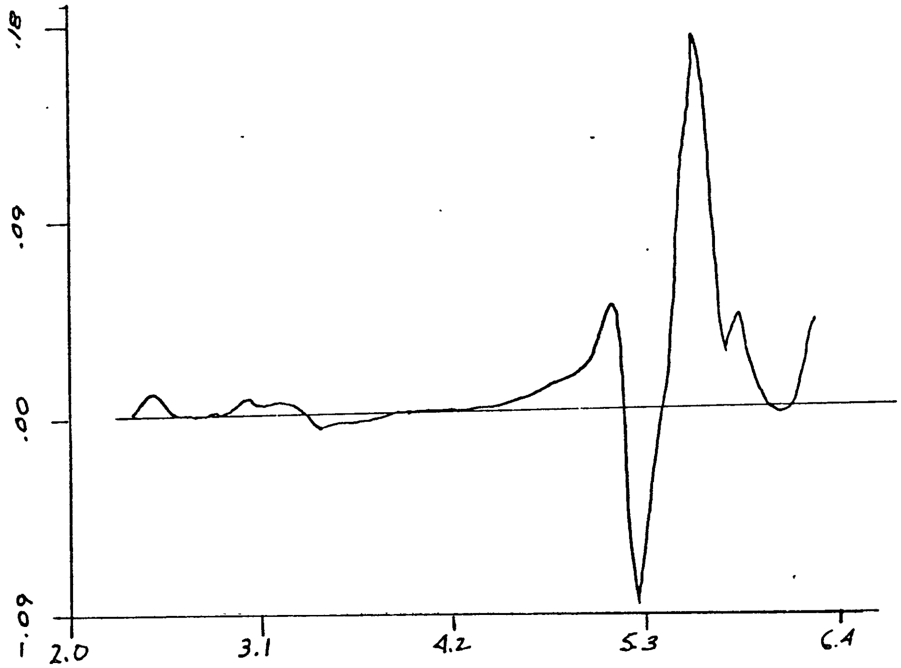


Fig. 45: z,5,III,C

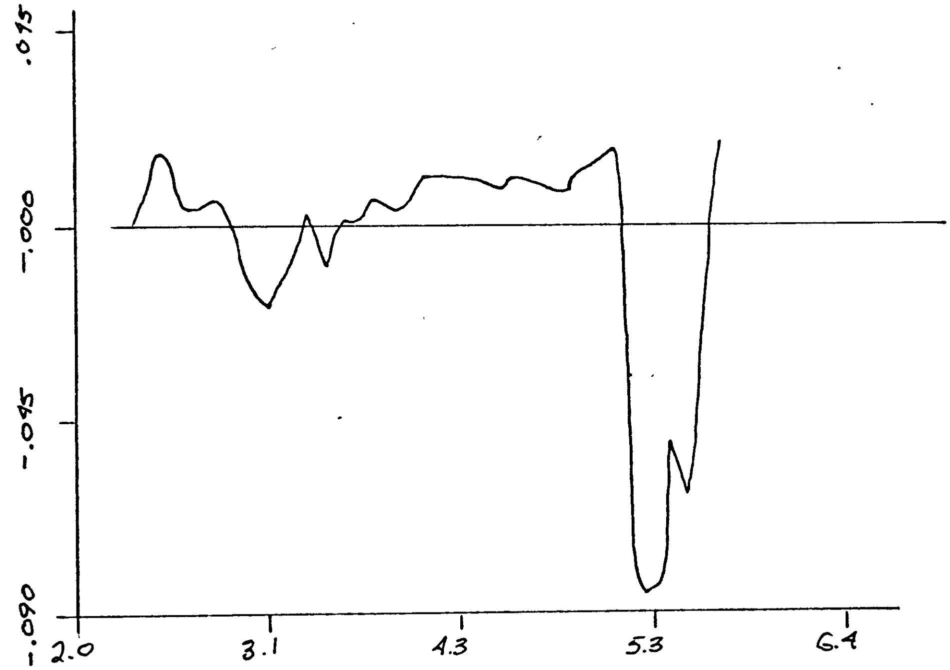


Fig. 46: r,5,III,C

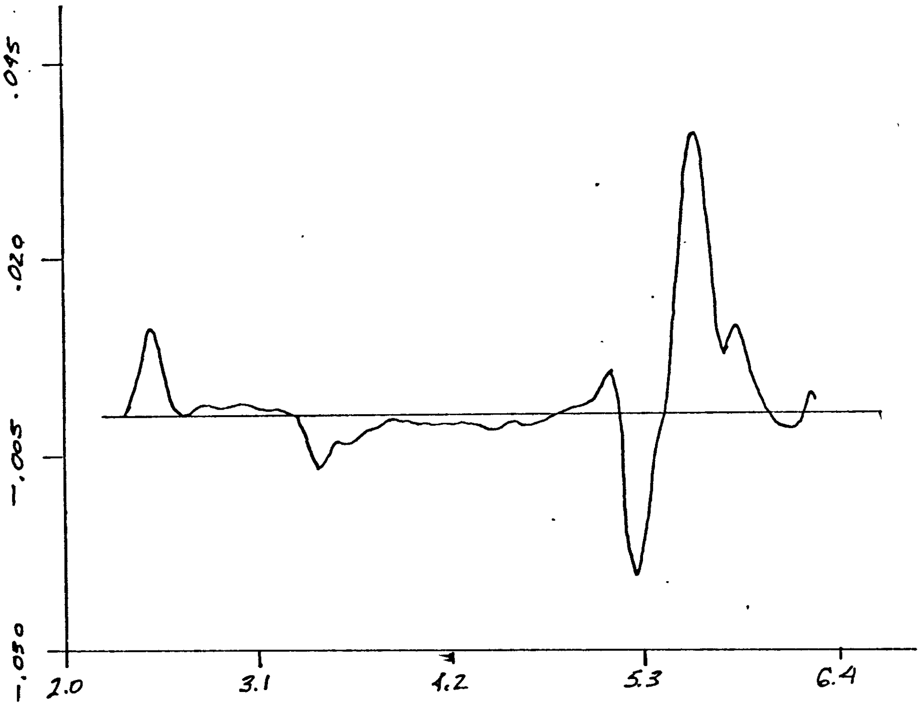


Fig. 47: z,5,III,C Shear source has been multiplied by 0.2 .

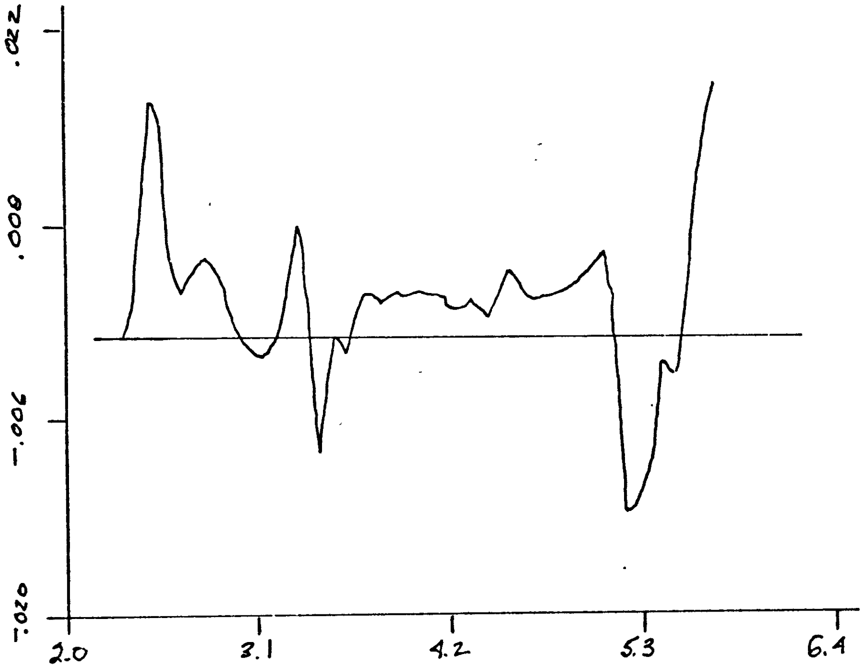


Fig. 48: r,5,III,C Shear source has been multiplied by 0.2 .

BIBLIOGRAPHY

1. Harkrider, D. G., 1964, Surface waves in multilayered elastic media. 1. Rayleigh and Love waves from buried sources in a multilayered elastic half-space, BSSA, 54, pp. 627-680.

2. Helmberger, D. V., 1965, Head waves from the Oceanic Mohorovicic Discontinuity, Ph.D. thesis, U.C.S.D.

3. Pekeris, C. L., Alterman, Z., Abramovici, F., and Jarosch, H., 1965, Propagation of a compressional pulse in a layered solid, Reviews of Geophysics, 3, p. 25.

4. Phinney, R. A., 1967, Calculation of Elastic Wave Propagation in a Layered Medium, DASA contract report DA-49-146-XZ-441.

APPENDIX I

The reflection function describes the attenuation of energy due to internal reflections. The reflection function $\mathcal{C}_\eta(p)$ is given by

$$\mathcal{E}_\eta(p) = R_{DPP}^{l_{DPP}^{(n)}} R_{DPS}^{l_{DPS}^{(n)}} R_{DSP}^{l_{DSP}^{(n)}} R_{DSS}^{l_{DSS}^{(n)}} R_{UPP}^{l_{UPP}^{(n)}} R_{UPS}^{l_{UPS}^{(n)}} R_{USP}^{l_{USP}^{(n)}} R_{USS}^{l_{USS}^{(n)}}$$

$l_{DPP}^{(n)}$ is the number of reflections of the P to P type off the lower boundary, $l_{UPP}^{(n)}$ is the number of P to P reflections off the free surface, etc. The reflection coefficients are computed from the conditions of continuity of stress and displacement at the interface, and are given by (from Helmberger, 1965):

$$R_{DPP} = (-\textcircled{1} + \textcircled{2} + \textcircled{3} - \textcircled{4} - \textcircled{5} + \textcircled{6}) / \mathcal{D}_1$$

$$R_{DPS} = 2p\eta_1 \left[(k_2 - p^2)(k_3 - p^2) - \eta_2\eta_2'(k_1 - p^2) \right] / \mathcal{D}_1$$

$$R_{DSP} = -2p\eta_1' \left[(k_2 - p^2)(k_3 - p^2) - \eta_2\eta_2'(k_1 - p^2) \right] / \mathcal{D}_1$$

$$R_{DSS} = (-\textcircled{1} + \textcircled{2} + \textcircled{3} - \textcircled{4} + \textcircled{5} - \textcircled{6}) / \mathcal{D}_1$$

$$\mathcal{D}_1 = \textcircled{1} + \textcircled{2} + \textcircled{3} + \textcircled{4} - \textcircled{5} - \textcircled{6}$$

$$\textcircled{1} = p^2(k_3 - p^2)^2 \qquad \textcircled{2} = \eta_1\eta_2\eta_1'\eta_2' p^2$$

$$\textcircled{3} = \eta_1\eta_1'(k_3 - p^2)^2 \qquad \textcircled{4} = \eta_2\eta_2'(k_1 - p^2)^2$$

$$\textcircled{5} = \eta_1\eta_2' k_1 k_2 \qquad \textcircled{6} = \eta_2\eta_1' k_1 k_2$$

$$k_1 = -\frac{1}{2}\left(\frac{\rho_1}{\mu_2-\mu_1}\right) \quad ; \quad k_2 = \frac{1}{2}\left(\frac{\rho_2}{\mu_2-\mu_1}\right) \quad ; \quad k_3 = k_1 + k_2$$

$$R_{UPP} = R_{USS} = [4\beta_1^4 p^2 \eta_1 \eta_1' - (1-2\beta_1^2 p^2)^2] / \mathcal{D}_2$$

$$R_{UPS} = 4p\eta_1 (1-2\beta_1^2 p^2) / \mathcal{D}_2$$

$$R_{USP} = -4p\eta_1' (1-2\beta_1^2 p^2) / \mathcal{D}_2$$

$$\mathcal{D}_2 = 4\beta_1^4 p^2 \eta_1 \eta_1' + (1-2\beta_1^2 p^2)^2$$

The receiver functions are included to take into account the effect of the free surface. For an impinging P (or SV) wave, we must add the upgoing P (or SV) wave to the reflected SV and P waves. The reflection coefficients are evaluated with the shear wave velocity and the rigidity set to zero for the medium above the free surface. The once-transformed displacements at the free surface due to an upgoing P wave are (derived from the stress free boundary condition--see Phinney, 1967):

(A1) $$\bar{U}_z^P(r,0,s) = \frac{1}{2\pi^2(\lambda+2\mu)}\,\mathrm{Im} \int_0^{i\infty} \frac{p}{\eta_1} K_0(spr) \{-\eta_1 + \eta_1' R_{UPP} - p R_{UPS}\} e^{-\eta_1 h}$$

(A2) $$\bar{U}_r^P(r,0,s) = \frac{1}{2\pi^2(\lambda+2\mu)}\,\mathrm{Im} \int_0^{i\infty} \frac{p}{\eta_1} K_1(spr) \{p + p R_{UPP} + \eta_1' R_{UPS}\} e^{-\eta_1 h}$$

and due to an upgoing SV wave are

(A3) $$\bar{u}_z^S(r,0,s) = \frac{1}{2\pi^2\mu}\, \mathrm{Im} \int_0^{i\infty} \frac{p}{\eta_1'} K_0'(spr) \left\{ -p - p\, R_{uss} + \eta_1 R_{usp} \right\} e^{-\eta_1' h}$$

(A4) $$\bar{u}_r^S(r,0,s) = \frac{1}{2\pi^2\mu}\, \mathrm{Im} \int_0^{i\infty} \frac{p}{\eta_1'} K_1(spr) \left\{ -\eta_1' + \eta_1' R_{uss} + p\, R_{usp} \right\} e^{-\eta_1' h}$$

The functions in the brackets are our receiver functions. One need only to substitute the reflection coefficients given above and grind out the algebra. When this is done, the results are:

$$\mathcal{R}_z^P(p) = \frac{-2\eta_1 (1-2p^2\beta_1^2)}{D}$$

$$\mathcal{R}_r^P(p) = \frac{-4\beta_1^2 p \eta_1 \eta_1'}{D}$$

$$\mathcal{R}_z^S(p) = \frac{-4\beta_1^2 p \eta_1 \eta_1'}{D}$$

$$\mathcal{R}_r^S(p) = \frac{-2\eta_1' (1-2p^2\beta_1^2)}{D}$$

where $$D = (1-2p^2\beta_1^2)^2 + 4\beta_1^4 \eta_1 \eta_1' p^2$$

APPENDIX II

COMPUTER PROGRAM

Included here is a printout of the program used to evaluate equations (9) and (10). This program is an adaptation of one written by D. V. Helmberger. A brief description of how it works may be helpful. The program begins by defining the rays. A ray is described by the number of traverses across the layer that are made as a P wave, the number as an S wave, the reflection coefficients needed to be multiplied to form $\mathcal{C}_n(p)$, and the mode at the receiver. For each ray the following procedure is followed. The program then computes the exact position p_0 at which $\frac{dp}{d\tau} \to \infty$, i.e., where the contour leaves the real axis. There is a singularity in the integrand at this point and special precautions must be taken. A series of values of real p are then generated. The spacing is weighted so that computation points are closely spaced near reflection, refraction, and surface wave arrival times, and sparsely spaced elsewhere. The imaginary parts of p corresponding to the real values just generated are computed. The functions of p are then evaluated for these values of complex p. The higher order terms are convolved with their appropriate functions of t and added to the lowest order term. The result is then convolved with $1/\sqrt{t}$. This is repeated for all the ray paths and the results are summed. An auxilliary program differentiates the output and convolves it with a source and receiver function to give the final answer.

```
C        MAIN
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/ORSTF/C(100),S(100),D(100),TH(100),X
      COMMON/CONFIX/DEL,NN,NDP,TMX,XDIM,YDIM,DP,KO
      COMMON/STUFF/CC(100),SS(100),DD(100),TTH(100),XX,RCSQ(100),
     1               RSSQ(100)
      COMMON/THY/T(1000),PP(1000),RP(600)
      DIMENSION R(10)
      COMMON/SYTH/XD11,YD11,XD22,YD22,XD33,YD33
      COMMON/PLOTC/CON,NNF,NPT,NSYN
      COMMON/FIXP/DDN(100),ARN(100),FLAT
      COMMON/STOR/P(1000),TD(1000)
      COMMON/THZ/TT(1000),PPZ(1000),PPR(1000)
      COMMON/TINP/DELTM,DLTM,NDA,MTD,DLTP,NDB,JO,NDIRT
      COMMON/LPRINT/PRNT,PRNTS,KST,KEND,PRNTC,NDC,DET
      COMMON/CTRSTF/NCASE,NPRAY,NYT,NDUM,VRL2,VRL3
      DIMENSION XL(2),YL1(4),YL2(4),YL3(4)
      DATA XL/'TIME SEC'/
      DATA YL2/'  THEORETICAL UZ'/
      DATA YL3/'  THEORETICAL UR'/
      LOGICAL PRNT,PRNTS,PRNTC,FLAT
      FLAT=.TRUE.
      PRNTS=.FALSE.
      PRNT=.FALSE.
      PRNTC=.FALSE.
      READ (5,100) NMOD
      DO 200 J=1,NMOD
200   READ (5,300) C(J),S(J),D(J),TH(J)
100   FORMAT (I10)
300   FORMAT (4F10.0)
600   READ(5,500) X,IGO,KPEK
500   FORMAT (F10.0,2I10)
      READ (5,501)  DP,TMX,NN
501   FORMAT (2F10.0,I10)
      IF (KPEK)  700,800,800
800   S(3)=1.1*S(2)
```

```
      C(2)=S(2)*SQRT(3.)
      D(3)=D(2)/0.605
      C(3)=S(2)*SQRT(3.63)
 700  CONTINUE
      XD22=10.
      YD22=3.
      YDIM=YD22
      XDIM=XD22
      CALL NEWPLT('M5207','5852','WHITE ','BLACK')
      NNF=1
      NPT=0
      JO=3
      CALL CURAY (JO)
      DO 30 M=1,1
      AMU=D(2)*(C(2)**2)/3.
      CON=1./(6.*3.14159*AMU)
      CON=1.
      XX=X
      MTD=2
      DELTM=.0025
      DMAX=1.E-4
      DLTM=.1
      NDP=30
      DET=1.E-5
      DTIM=DP
      DEL=DP
      DLTP=DP
      NYT=3
      VRL2=.92*S(2)
      VRL3=.92*S(3)
      NPRAY=40
      K=NN/2
      NCASE=1
      CALL SETUP  (1,0,01,01,0,0,0)
      DO 20 J=1,NN
 20   PP(J)=PPZ(J)
```

```
      CALL STEP (NN,1,2,0,0,DP)
      DO 21 J=1,NN
   21 PP(J)=PPR(J)
      CALL STEP (NN,1,2,0,0,DP)
      IF (IGO) 30,30,600
   30 CONTINUE
      CALL ENDPLT
         END
      SUBROUTINE RECVR (A1,B1,P,KR,DZ,DR,KSSP,RPP,RPS,RSP,RSS)
C        TO COMPUTE  SURFACE RECEIVER AND REFLECTION FUNCTIONS
      IMPLICIT COMPLEX*16 (A-H,O-Z)
      REAL*8 A1,B1,BSQ
      BSQ        = B1**2
      PSQ        = P*P
      ET         =CDSQRT((1./A1**2)-PSQ)
      EP         =CDSQRT((1./BSQ  )-PSQ)
      D1         = 1.-2.*BSQ*PSQ
      EE         = ET*EP*P*BSQ*4.
      D2 = P*BSQ*EE
      SDF=ET
      IF (KSSP .EQ. 0)  SDF=EP
      DENA = D1**2 + D2
      DEN  = SDF*DENA
      R1         = 2.*D1/DEN
      R2         = EE/DEN*(-1.)
      IF (KR   .EQ. 0) GO TO 100
C        P AT RCVR
      DZ         = R1*ET
      DR         =-R2
      GO TO 200
  100 CONTINUE
C        S AT RCVR
      DZ         =-R2
      DR         = R1*EP
  200 CONTINUE
      RPP = (-D1**2 + D2)/DENA
```

```
      R3 = 4.*P*D1/DENA
      RPS =  R3*ET
      RSP = -R3*EP
      RSS = RPP
      RETURN                                                            00000230
      END                                                               00000240
      SUBROUTINE PLN1(PO,TO,K,N,TC,NRY,V2)                              00000010
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/TINP/DELTM,DLTM,NDA,MTD,DLTP,NDB,JO,NDIRT
      COMMON/SPE/DELP(800),DD1,DD2,DD3,DD4,NO
      COMMON/ORSTF/C(100),S(100),D(100),TH(100),X                       00000040
      COMMON/TRESL/PZ1(500),PR1(500)                                    00000050
      COMMON/MAGIC/PQ(1200),DDPT(1200),TTT(1200)
      COMMON/EXACT/PHIZ(1000),PHIR(1000),TT(1000),NEND,NM
      COMMON / LPRINT/ PRNT,PRNTS                                       00000080
      LOGICAL PRNT,PRNTS                                                00000090
      COMPLEX*16  PQ,DDPT,FNZ,FNR,FNZ1,FNR1,Q
      P         = 1./V2                                                 00000110
      DO 80 I=2,NO                                                      00000120
      J         = I-1                                                   00000130
      P         = P+DELP(J)                                             00000140
      Q         = P+0.*(0.,1.)                                          00000150
      CALL HELP(K,N,P,TTP,DTP,NRY)                                      00000160
      IF(PRNT) PRINT 100,Q,P,DTP,TTP                                    00000170
      TT(I)     = TTP                                                   00000180
      DDPT(I)   = DTP                                                   00000190
      CALL PSICO(Q,FNZ,FNR,FNZ1,FNR1,I,NRY)                             00000200
      IF(PRNT) PRINT 100,FNZ,FNR,FNZ1,FNR1                              00000210
      PHIZ(I)   = FNZ * (0.,-1.)
      PHIR(I)   = FNR * (0.,-1.)
      PZ1(I)    = FNZ1 * (0.,-1.)
      PR1(I)    = FNR1 * (0.,-1.)
      IF ((TO-TTP) .LT. DLTP) GO TO5                                    00000260
 80     CONTINUE                                                        00000270
 100   FORMAT(8E15.4)                                                   00000280
      WRITE(6,100) PO,TO                                                00000290
```

```
5     NO       = J+1                                                    00000300
       I       = NO                                                     00000310
4     IF(TO-TTP.LT.DLTP) GO TO 3                                        00000320
      PP       = PO - P                                                 00000330
      P        = P + PP/2.0                                             00000340
      I        = I+1                                                    00000350
      NO       = I                                                      00000360
      Q        = P                                                      00000370
      CALL HELP(K,N,P,TTP,DTP,NRY)                                      00000380
      TT(I)    = TTP                                                    00000390
      DDPT(I)  = DTP                                                    00000400
      CALL PSICO(Q,FNZ,FNR,FNZ1,FNR1,I,NRY)                             00000410
      IF(PRNT) PRINT 100,FNZ,FNR,FNZ1,FNR1                              00000420
      PHIZ(I)  = FNZ * (0.,-1.)
      PHIR(I)  = FNR * (0.,-1.)
      PZ1(I)   = FNZ1 * (0.,-1.)
      PR1(I)   = FNR1 * (0.,-1.)
      GO TO 4                                                           00000470
3     TT(1)    = TC                                                     00000480
      PHI7(1)  = 0.                                                     00000490
      PHIR(1)  = 0.                                                     00000500
      PR1(1)   = 0.                                                     00000510
      PZ1(1)   = 0.                                                     00000520
      RETURN                                                            00000530
      END                                                               00000540
      SUBROUTINE PLN2(PO,TO,K,MO,M,NRY)                                 00000010
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/FIXP/DDN(100),ARN(100)                                     00000020
      COMMON/TINP/DELTM,DLTM,NDA,MTD,DLTP,NDB,JO,NDIRT
      COMMON/MAGIC/PP(1200),DDPT(1200),TT(1200)
      COMMON/EXACT/PHI7(1000),PHIR(1000),TTT(1000),NEND,NM
      COMMON/TRESL/PZ1(500),PR1(500)                                    00000060
      COMMON/ORSTF/C(100),S(100),D(100),TH(100),X                       00000070
      DIMENSION FF(50)     ,GG(50)                                      00000080
      COMMON / LPRINT/ PRNT,PRNTS                                       00000090
      LOGICAL PRNT,PRNTS                                                00000100
```

```
      COMPLEX*16   PP,BT,DDPT,RP,RPP,GC,P
      COMPLEX*16  FNZ,FNR,FNZ1,FNR1
      DO 5 I=MO,M                                                       00000130
      TTT(I)    = TT(I)                                                 00000140
      P         = PP(I)                                                 00000150
      CALL PSICO(P,FNZ,FNR,FNZ1,FNR1,I,NRY)                             00000160
      PHIZ(I)   = FNZ *(0.,-1.)
      PHIR(I)   = FNR *(0.,-1.)
      PR1(I)    = FNR1 * (0.,-1.)
      PZ1(I)    = FNR1 * (0.,-1.)
      IF(PRNT) PRINT 100,FNZ,FNR,FNZ1,FNR1                              00000210
5        CONTINUE                                                       00000220
100     FORMAT(8E15.4)                                                  00000230
3     NO        = MO-2                                                  00000240
      DP        = DLTP                                                  00000250
      P         = PO*(1.,0.)+0.*(0.,1.)                                 00000260
      I         = MO-1                                                  00000270
      Q         = PO                                                    00000280
      DDPT(I)   = SF2(Q,K,NRY,DP)                                       00000290
      TTT(I)    = TO                                                    00000300
      WRITE(6,100) P,DDPT(I)                                            00000310
      CALL PSICO(P,FNZ,FNR,FNZ1,FNR1,I,NRY)                             00000320
      IF(PRNT) PRINT 100,FNZ,FNR,FNZ1,FNR1                              00000330
      PREZ      = FNZ
      PIMZ      = FNZ * (0.,-1.)
      PRER      = FNR
      PIMR      = FNR * (0.,-1.)
      IF (PRNT) PRINT 100,PREZ,PIMZ,PRER,PIMR                           00000380
      F1        = 0.                                                    00000390
      G1        = 0.                                                    00000400
      SUM       = 0.                                                    00000410
      TUM       = 0.                                                    00000420
      IF(MO.LE.3) GO TO 46                                              00000430
      TNN       = TO-DP                                                 00000440
      CALL  INTERP(TTT,PHIZ,M,TNN,Y)                                    00000450
      F1        = Y                                                     00000460
```

```
      FF(1)     = Y
      CALL INTERP(TTT,PHIR,M,TNN,Y)
      G1        = Y
      GG(1)     = Y
      SUM       = SUM+2.*PIMZ*DSQRT(TO-TTT(NO))
      TUM       = TUM+2.*PIMR*DSQRT(TO-TTT(NO))
      IF (PRNT) PRINT 4, SUM
      IF (PRNT) PRINT 7, TUM
7     FORMAT (5X'TUM = 'G18.6)
4     FORMAT (5X'SUM = 'G18.6)
      DELL      = (TTT(NO)-TNN )/5.
      TT(1)     = TNN
      FF(6)     = PHIZ(NO)
      GG(6)     = PHIR(NO)
      PHI7(I)   = PHIZ(NO)
      PHIR(I)   = PHIR(NO)
      DO 41 J=2,6
      TT(J)     = TT(J-1)  +DELL
      CALL INTERP(TTT,PHIZ,M,TT(J),Y)
      FF(J)     = Y
      CALL INTERP(TTT,PHIR,M,TT(J),Y)
      GG(J)     = Y
      TUM       = TUM+(GG(J-1)+GG(J))/2.*DELL
      SUM       = SUM+(FF(J-1)+FF(J))/2.*DELL
      IF (PRNT) PRINT 4, SUM
      IF (PRNT) PRINT 7, TUM
41    CONTINUE
46    TPP       = TTT(MO)
      TT(1)     = TTT(MO)
      IF(TTT(MO)-TO.GT.DP) GO TO 43
      SUM       = SUM+2.*PREZ*DSQRT(TTT(MO)-TO)
      TUM       = TUM+2.*PRER*DSQRT(TTT(MO)-TO)
      IF (PRNT) PRINT 4, SUM
      IF (PRNT) PRINT 7, TUM
      CALL INTERP(TTT,PHIZ,M,TPP,Y)
      FF(1)     = Y
```

```
      CALL INTERP(TTT,PHIR,M,TPP,Y)
      GG(1)    = Y
      DELL     = (TO+DP-TTT(MO))/5.
      DO 42 J=2,6
      TT(J)    = TT(J-1)  +DELL
      CALL INTERP(TTT,PHIZ,M,TT(J),Y)
      FF(J)    = Y
      CALL INTERP(TTT,PHIR,M,TT(J),Y)
      GG(J)    = Y
      TUM      = TUM+(GG(J-1)+GG(J))/2.*DELL
      SUM      = SUM+(FF(J-1)+FF(J))/2.*DELL
      IF (PRNT) PRINT 4, SUM
      IF (PRNT) PRINT 7, TUM
 42    CONTINUE
      F3       = FF(6)
      G3       = GG(6)
      PHIZ(I)  = (3.*SUM/DP-F1-F3)/4.
      PHIR(I)  = (3.*TUM/DP-G1-G3)/4.
      GO TO 44
 43    TTT(MO) = TO+DP
      PHIZ(MO) = PREZ/(DP**.5)
      PHIR(MO) = PRER/(DP**.5)
      F3       = PHIZ(MO)
      G3       = PHIR(MO)
      SUM      = SUM+2.*PREZ*DSQRT(DP)
      TUM      = TUM+2.*PRER*DSQRT(DP)
      PHIZ(I)  = (3.*SUM/DP-F1-F3)/4.
      PHIR(I)  = (3.*TUM/DP-G1-G3)/4.
      IF (PRNT) PRINT 4, SUM
      IF (PRNT) PRINT 7,TUM
 44     CONTINUE
      PREZ1    = FNZ1
      PIMZ1    = FNZ1 * (0.,-1.)
      PRER1    = FNR1
      PIMR1    = FNR1 * (0.,-1.)
      IF(PRNT) PRINT 100, PREZ1,PIMZ1,PRER1,PIMR1
```

```
      IF(MO.LE.3) GO TO 47                                              00001190
      DO 81 J=2,NO                                                      00001200
      AREA     = (PR1(J)+PR1(J-1))*(TTT(J)-TTT(J-1))                    00001210
      PHIR(J)  = PHIR(J)+AREA/2.                                        00001220
      AREA     = (PZ1(J)+PZ1(J-1))*(TTT(J)-TTT(J-1))                    00001230
      PHIZ(J)  = PHIZ(J)+AREA/2.                                        00001240
   81   CONTINUE                                                        00001250
      PHIZ(I)  = PHIZ(I)+2.*PIMZ1*DSQRT(TTT(I)-TTT(NO))
      PHIR(I)  = PHIR(I)+2.*PIMR1*DSQRT(TTT(I)-TTT(NO))                   000  27
   47   CONTINUE                                                        00001280
      PHIZ(MO) = PHIZ(MO)+2.*PREZ1*DSQRT(TTT(MO)-TTT(I))                  00   29
      PHIR(MO) = PHIR(MO)+2.*PRER1*DSQRT(TTT(MO)-TTT(I))                 0000130
      JM       = MO+1                                                   00001310
      DO 82 J=JM,M                                                      00001320
      AREA     = (PR1(J)+PR1(J-1))*(TTT(J)-TTT(J-1))                    00001330
      PHIR(J)  = PHIR(J)+AREA/2.                                        00001340
      AREA     = (PZ1(J)+PZ1(J-1))*(TTT(J)-TTT(J-1))                    00001350
      PHIZ(J)  = PHIZ(J)+AREA/2.                                        00001360
   82  CONTINUE                                                         00001370
      WRITE (6,111)  TTT(I),PHIZ(I),PHIR(I)
  111 FORMAT  (2X,'CRITICAL TIME',G20.8,5X,'PHIZ',G20.8,'PHIR',G20.8)
      END                                                               00001380
      SUBROUTINE CONTOR(TMX,M,KN,N,MO)                                  00000010
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/MAGIC/PP(1200),DDPT(1200),TT(1200)
      COMMON/SPE/DELP(800),DD1,DD2,DD3,DD4,NO
      COMMON/PATHC/PO,TO,K                                              00000040
      COMMON/TINP/DELTM,DLTM,NDA,MTD,DLTP,NDB,JO,NDIRT
      COMMON/TFIX/TN1,TN2,TN3,TN4,JN1,JN2,JN3,JN4                       00000060
      COMMON/CTRSTF/NCASE,NPRAY,NYT,NDUM,VRL2,VRL3
      DIMENSION DER(400)                                                00000080
      LOGICAL PRNT,PRNTS,PRNTC                                          00000090
      COMMON/LPRINT/PRNT,PRNTS,KST,KEND,PRNTC,NDC,DET
      COMPLEX*16  PP,DDPT,P,CT,DEV
      JN       = 10                                                     00000120
    1 Q        = PO                                                     00000130
```

```
      I         = MO-1
      PIL       = 1.E-6
      PP(I)     = PO
      TT(I)     = TO
      KM        = 100
      IF(DELP(1).LE.DELTM) DELP(1)=DELTM
33     I        = KM-1
      L         = 0
      DO 10 J=1,JN
      L         = L+1
      I         = I+1
9     Q         = Q+DELP(J)
      PI        = PO*.2
      DL        = PI*.45
      CALL TIME2(Q,PI,DL,P,DEV,CT,KN,N,PIL)
      RTIME     = CT
      TIMEI     = CT*(0.,-1.)
      DR        = DEV
      DDPT(I)   = DEV
      TT(I)     = RTIME
      PP(I)     = P
      IF(TT(I)-TO.LE.DTIM) GO TO 32
      IF(DR.LE.0.) GO TO 30
      IF(DABS(TIMEI).GT.1.E-3) GO TO 30
      IF(RTIME-TO.LT..001) GO TO 30
      JJ        = J+1
      DELP(JJ) =     ((PO-PP(I))/2.)
10      CONTINUE
30     IF(I.LE.KM) DELP(1)=5.*DELP(1)
      IF(I.LE.KM) GO TO 33
      I         = I-1
32     JJ       = I-KM+1
      DO 31 J=1,JJ
      LL        = MO+J-1
      NN        = I-J+1
      TT(LL)    = TT(NN)
```

```
      PP(LL)    = PP(NN)
      DDPT(LL)  = DDPT(NN)
 31    CONTINUE
      I         = LL
      DELP(JN)  =     (PP(LL)-PP(LL-1))
      J         = JN
      MM        = 1
      TM        = TMX+TO
      PI        = PP(LL)*(0.,-1.)
      Q         = PP(LL)
      IF(NCASE.EQ.1)RG=DABS(1./VRL2-Q)
      IF(NCASE.EQ.2)RG=DABS(1./VRL3-Q)
      CALL DELPS(NPRAY,RG,1,NYT)
      J1        = LL
      J2        = J1+NO
      IJ        = 0
      DO 21 J=J1,J2
      IJ        = IJ+1
 21    DER(J)   = DELP(IJ)
      JF        = J2
      IF(NCASE.EQ.1) GO TO 23
      RG        =DABS(1./VRL2-1./VRL3)
      CALL DELPS(NPRAY,RG,1,NYT)
      J1        = J2+1
      J2        = J1+NO
      IJ        = 0
      DO 24 J=J1,J2
      IJ        = IJ+1
 24    DER(J)   = DELP(IJ)
      JF        = J2
 23    IF(PRNTC)  WRITE(6,100) (DER(J),J=J1,J2)
 100    FORMAT(6E12.4)
      DO 25 J=LL,JF
      Q         = Q+DER(J)
      I         = I+1
      DELPR     =     (PP(I-1)-PP(I-2))
```

```
      DELPI     =        (PP(I-1)-PP(I-2))    *(0.,-1.)
      PI        = (DELPI*DER(J)     )/DELPR +PI
      DL        = PI*.5
      CALL TIME2(Q,PI,DL,P,DEV,CT,KN,N,PIL)
      PP(I)     = P
      RTIME     = CT
      DDPT(I)   = DEV
      TT(I)     = RTIME
      IF(TT(I).GT.TM) GO TO 13
 25     CONTINUE
 12     Q       = Q+MM*DER(JF)
      I         = I+1
      DELPR     =       (PP(I-1)-PP(I-2))
      DELPI     =        (PP(I-1)-PP(I-2)) *(0.,-1.)
      PI        = (DELPI*MM*DER(JF))/DELPR +PI
      DL        = PI*.5
      CALL TIME2(Q,PI,DL,P,DEV,CT,KN,N,PIL)
      PP(I)     = P
      RTIME     = CT
      DDPT(I)   = DEV
      TT(I)     = RTIME
      IF(TT(I).GT.TM) GO TO 13
 14    IF(TT(I)-TT(I-1).LE.DLTM) MM=MTD*MM
      GO TO 12
 13     CONTINUE
      M         = I
      END
      SUBROUTINE DELPS (NNN,RG,NN,N)
      IMPLICIT REAL*8  (A-H,O-Z)
      DIMENSION PP(200)
      COMMON/SPE/DELP(800),DD1,DD2,DD3,DD4,NO
      RG        = RG-1.E-08
      PI        = 3.141593
      AN        = PI/(NNN*2.)
      J         = NN
      A         = AN
```

```
      DELP(J)   = RG*(DSIN(A)**N)                                            00000090
      TO        = DELP(J)                                                    00000100
      A         = A+AN                                                       00000110
       K        = 1                                                          00000120
      PP(1)     = DELP(1)                                                    00000130
 1      J       = J+1                                                        00000140
      K         = K+1                                                        00000150
      PP(K)     = RG*DSIN(A)**N
      DELP(J)   = PP(K)-PP(K-1)                                              00000170
      DELP(J)   =DABS (DELP(J))                                              00000180
      TO        = TO+DELP(J)                                                 00000190
      A         = A+AN                                                       00000200
      IF(TO.LT.RG) GO TO 1                                                   00000210
 2      NO      = J-1                                                        00000220
      END                                                                    00000230
      REAL FUNCTION SF2*8 (P,K,NRY,DP)
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/STUFF/C(100),S(100),D(100),TH(100),X,RCSQ(100),RSSQ(100)        00000020
      COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)         00000030
      PSQ       = P ** 2                                                     00000040
      J         = 2                                                          00000050
      ESQ       =DABS (RCSQ(J)-PSQ)                                          00000060
      E         =DSQRT(ESQ)                                                  00000 7
      TE        = (TH(4)+TH(2)*LTP(NRY))*RCSQ(2)/(ESQ*E)                     00000080
      ESQ       =DABS (RSSQ(J)-PSQ)                                          000 0 9
      E         =DSQRT(ESQ)                                                  00000 00
      TE        = (TH(3) + TH(2)*LTS(NRY))*RSSQ(2)/(ESQ*E) + TE              00000110
      TE        = TE*2.                                                      00000120
      SR        = 1.                                                         00000130
      SF2       = SR/DSQRT(TE)                                               00000140
      END                                                                    00000150
      SUBROUTINE HELP(K,N,P,TTP,DTP,NRY)                                     00000010
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/STUFF/C(100),S(100),D(100),TH(100),X,RCSQ(100),RSSQ(100)        00000020
      COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)         00000030
      COMMON/LPRINT/PRNT,PRNTS,KST,KEND,PRNTC,NDC,DET
```

```
      LOGICAL PRNT,PRNTS,PRNTC,FLAT                                     00000050
  100 FORMAT(10X,'SUB. HELP,P,E,TOTEM,BLTEM,DTP,TTP',/,6(E16.6))        00000060
      PSQ        = P**2                                                 00000070
      J          = 2                                                    00000080
      E          =DSQRT(DABS(RCSQ(J)-PSQ))                              00  0 9
      TOTEM      = E*(TH(4)+TH(2)*LTP(NRY))                             00000100
      BLTEM      = -(TH(4)+TH(2)*LTP(NRY))/E                            00000110
      IF(PRNTC) WRITE(6,100) P,E,TOTEM,BLTEM,DTP,TTP                    00000120
      E          =DSQRT(DABS(RSSQ(J)-PSQ))                              0000  T
      TOTEM      = TOTEM+TH(2)*LTS(NRY)*E +TH(3)*E                      00000140
      BLTEM      = BLTEM-TH(2)*LTS(NRY)/E -TH(3)/E                      00000150
      BL         = X + P*BLTEM                                          00000160
      TO         = P*X + TOTEM                                          00000170
      DTP        = 1./BL                                                00000180
      TTP        = TO                                                   00000190
      IF(PRNTC) WRITE(6,100) P,E,TOTEM,BLTEM,DTP,TTP                    00000200
      RETURN                                                            00000210
      END                                                               00000220
      SUBROUTINE TIME2(PR,PI,DL,Q,DPT,T,KN,N,PIL)                       00000010
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)    00000020
      COMMON/PATHC/PO,TO,K                                              00000030
      DIMENSION E(100)                                                  00000040
      COMPLEX*16 P,E,BL,T,PC,DPT,Q                                      00000060
      DIMENSION Y1(50),Y4(50),X1(50),X4(50)                             00000070
      COMMON/STUFF/C(100),S(100),D(100),TH(100),R                       00000080
      LOGICAL PRNT,PRNTS,PRNTC                                          00000090
      COMMON/LPRINT/PRNT,PRNTS,KST,KEND,PRNTC,NDC,DET
      NRY        = N                                                    00000110
      X1M        = 1.E+4                                                00000120
      X4M        = 0.0                                                  00000130
      NNN        = 0                                                    00000140
      I          = 0                                                    00000150
    6  P         = PR*(1.,0.)+PI*(0.,1.)                                00000160
      T          = P*R                                                  00000170
      J          = 2                                                    00000180
```

```
      BL        = 1./(C(J)**2)-P*P                                      00000190
      E(J)      = CDSQRT(BL)                                            00000200
      T         = T+E(J)*(TH(4)+TH(2)*LTP(NRY))                         00000210
      J         = 3                                                     00000220
      BL        = 1./(S(2)**2)-P*P                                      00000230
      E(J)      = CDSQRT(BL)                                            00000240
      T         = T+E(J)*(TH(3)+TH(2)*LTS(NRY))                         00000250
      IF(PRNTC) WRITE(6,110) P,E(2),T                                   00000260
      TI        = T*(0.,-1.)
      IF (DABS(TI) .LE. DET) GO TO 2
      IF(I.GT.15) GO TO 2                                               00000290
      I         = I+1                                                   00000300
      X1(I)     = 100.                                                  00000310
      X4(I)     = 0.0                                                   00000320
      IF(TI.GT.0.) Y1(I)=TI                                             00000330
      IF(TI.GT.0.) X1(I)=PI                                             00000340
      IF(TI.LT.0.) Y4(I)=TI                                             00000350
      IF(TI.LT.0.) X4(I)=PI                                             00000360
      IF(I.EQ.1) GO TO 43                                               00000370
      IF(NNN.GT.1) GO TO 44                                             00000380
      IF(TI*TL.LF.0.) GO TO 44                                          00000390
43     IF(TI.GT.0.) PI=PI-DL                                            00000400
      IF(TI.LE.0.) PI= PI+DL                                            00000410
      IF(PI.LE.1.E-5) PI=PK/2.                                          00000420
      NNN       = 1                                                     00000430
      PK        = PI                                                    00000440
      TL        = TI                                                    00000450
      GO TO 6                                                           00000460
44     DO 52 J=1,I                                                      00000470
      IF(X1(J).GT.X1M) GO TO 53                                         00000480
      X1M       = X1(J)                                                 00000490
      NJ        = J                                                     00000500
53    IF(X4(J).LE.X4M) GO TO 54                                         00000510
      X4M       = X4(J)                                                 00000520
      MJ        = J                                                     00000530
54     CONTINUE                                                         00000540
```

```
52      CONTINUE
     Y1M       = Y1(NJ)
     Y4M       = Y4(MJ)
     DPI       = (X1M-X4M)/(Y1M-Y4M)
     DPM       = Y1M*DPI
     PI        = X1M-DPM
     NNN       = 2
     IF(PI.LE.PIL) PI=PIL
     GO TO 6
2       CONTINUE
     BL        = 0.
     BL        = BL-(TH(4)+TH(2)*LTP(NRY))/E(2)
     BL        = BL-TH(2)*LTS(NRY)/E(3)-TH(3)/E(3)
     BL        = R+P*BL
     Q         = P
     DPT       = 1./BL
     IF(PRNTC) WRITE(6,110) P,E(2),T,DPT
110  FORMAT (1H0,4X'P =  '2G18.6/5X'E(1) ='2G17.6/5X'T =  '2G18.6/
    +5X'DPT ='2G18.6)
     END
     SUBROUTINE HIGH(NDP,TMX,K,KI,N)
     IMPLICIT REAL*8  (A-H,O-Z)
     COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)
     COMMON/PLACE/THIC,H,KSSP
     COMMON/EXACT/PHIZ(1000),PHIR(1000),TD(1000),NEND,NM
     COMMON/MAGIC/PP(1200),DDPT(1200),TT(1200)
     COMMON/SPE/DELP(800),DD1,DD2,DD3,DD4,NO
      COMMON/PATHC/PO,TO,KK
     COMMON/TINP/DELTM,DLTM,NDA,MTD,DLTP,NDB,JO,NDIRT
     COMMON/ORSTF/CC(100),SS(100),DD(100),TTH(100),XX
     COMMON / LPRINT/ PRNT,PRNTS
     LOGICAL PRNT,PRNTS
     COMPLEX*16  PP,DDPT
     JN1       = 12
     JN2       = 10
     JN3       = 8
```

```
      JN4       = 100                                                   00000160
      TN1       = .8                                                    00000170
      TN2       = .2                                                    00000180
      TN3       = .1                                                    00000190
      TN4       = .001                                                  00000200
      V2        = CC(3)                                                 00000210
      NRY       = N                                                     00000220
      KNRY      = NRY*KSSP                                              00000230
      DEL       = 1./SS(2)                                              00000240
      IF(LTP(NRY).GT.0) GO TO 82                                        00000250
      IF(KSSP.EQ.0) GO TO 81                                            00000260
 82   DEL       = 1./CC(2)                                              00000270
 81    P        = -1.E-9                                                00000280
      DET       = 1.E+12                                                00000290
       CALL FIND2(P,KK,DEL,DET,PO,TO,N)                                 00000300
      NNN       = NDP                                                   00000310
      NK        = 2                                                     00000320
      IF(NRY.EQ.1) V2= CC(2)                                            00000330
      IF(KNRY.EQ.1) V2=.9*CC(2)                                         00000340
      P         = 1./V2                                                 00000350
      RG        =DABS(PO-P)
      CALL HELP(K,N,P,TTP,DTP,N)                                        00000370
      TC        = TTP                                                   00000380
      TG        = TO-TTP                                                00000390
      IF(PO.LE.1./V2) GO TO 6                                           00000400
      IF(TG.GT.TN1) GO TO 6                                             00000410
      JN        = JN1                                                   00000420
      IF(TG.GT.TN2) GO TO18                                             00000430
      JN        = JN2                                                   00000440
      IF(TG.GT.TN3) GO TO18                                             00000450
      JN        = JN3                                                   00000460
 18   QZ        = RG/(JN+1)                                             00000470
      DO 15 J = 1,JN                                                    00000480
      DELP(J)   = QZ                                                    00000490
 15     CONTINUE                                                        00000500
      NO        = JN                                                    00000510
```

```
      IF(TG.LT.TN4) GO TO 2                                             00000520
      GO TO 19                                                          00000530
 6     CALL DELPS(NNN,RG,1,NK)                                          00000540
      IF (.NOT.PRNT) GO TO 19                                           00000550
      PRINT 7, V2, XM, PO, RG, TC, TO, (DELP(J),J=1,NO)                 00000560
 7    FORMAT (1H0,4X'V2 = 'G13.6,5X'XM = 'G13.6,5X'PO = 'G13.6/5X'RG = '00000570
     +G13.6,5X'TC = 'G13.6,5X'TO = 'G13.6/5X'DELP'/(G15.6))             00000580
 19     IF(PO.LE.1./V2) GO TO 2                                         00000590
      CALL PLN1(PO,TO,K,N,TC,N,V2)                                      00000600
 2     MO       = NO+2                                                  00000610
      IF(TG.LT.TN4) MO=2                                                00000620
      IF(PO.LT.1./V2) MO=2                                              00000630
      CALL CONTOR(TMX,M,KN,N,MO)                                        00000640
      IF (.NOT.PRNT) GO TO 620                                          00000650
      WRITE (6,5)                                                       00000660
 5    FORMAT (1H0,13X'PP'27X'DDPT'24X'TT')                              00000670
      JJ        = MO                                                    00000680
      WRITE(6,200) (PP(J),DDPT(J),TT(J),J=JJ,M)                         00000690
 200   FORMAT(5E15.4)                                                   00000700
 620  CALL PLN2(PO,TC,K,MO,M,N)                                         00000710
      NEND      = M                                                     00000720
      NM        = NO                                                    00000730
      IF(PO.LT.1./V2) NM=0                                              00000740
      IF(TG.LT.TN4)NM= 0                                                00000750
      LK          =-19                                                  00000760
      WRITE (6,98)                                                      00000770
      KJMP      = 5
      LK        = 1-KJMP
97    LK          =LK+KJMP
      IF (LK .GT. NEND)  GO TO 99                                       00000790
      WRITE (6,100) TD(LK),PHIZ(LK),PHIR(LK),LK                         00000800
      GO TO 97                                                          00000810
99    CONTINUE                                                          00000820
98    FORMAT    (12X,'SUB. HIGH     TD,PHIZ,PHIR',2X)                   00000830
100   FORMAT (3E18.6,I10)
      END                                                               00000850
```

```
        SUBROUTINE RAYDEF                                                        00000010
        COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)           00000020
        COMMON/PLACE/THIC,H,KSSP                                                 00000030
        COMMON /LPRINT/ PRNT,PRNTS                                               00000040
        LOGICAL PRNT,PRNTS                                                       00000050
        REAL*8 THIC,H
        WRITE (6,800)                                                            00000060
        READ (5,400) THIC,H,KSSP                                                 00000070
  1200    FORMAT(4I1)                                                            00000080
        WRITE (6,700) THIC,H
 700    FORMAT (/10X,'LAYER THICKNESS ',F6.3/10X,'SOURCE DEPTH',F6.3)
        IF (KSSP .EQ. 0) WRITE (6,1300)
        IF (KSSP .EQ. 1) WRITE (6,1400)
 1300    FORMAT (15X,'SHEAR SOURCE')
 1400    FORMAT (15X,'COMPRESSIONAL SOURCE')
         READ(5,1200) (KUD(J),LTS(J),LTP(J),KRSP(J),J=19,64)                     00000090
         LTP(1)  = 0                                                             00000100
         LTP(2)  = 1                                                             00000110
         LTP(3)  = 0                                                             00000120
         LTP(4)  = 2                                                             00000130
         LTP(5)  = 1                                                             00000140
         LTP(6)  = 0                                                             00000150
         LTP(7)  = 3                                                             00000160
         LTP(8)  = 2                                                             00000170
         LTP(9)  = 1                                                             00000180
        LTP(10)  = 0                                                             00000190
        LTP(11)  = 1                                                             00000200
        LTP(12)  = 2                                                             00000210
        LTP  (13)= 1                                                             00000220
        LTP  (14)= 2                                                             00000230
        LTP  (15)= 1                                                             00000240
        LTP  (16)= 4                                                             00000250
        LTP  (17)= 0                                                             00000260
        LTP  (18)= 5                                                             00000270
         LTS(1)  = 0                                                             00000280
         LTS(2)  = 0                                                             00000290
```

```
  LTS(3)   = 1                                                          00000300
  LTS(4)   = 0                                                          00000310
  LTS(5)   = 1                                                          00000320
  LTS(6)   = 2                                                          00000330
  LTS(7)   = 0                                                          00000340
  LTS(8)   = 1                                                          00000350
  LTS(9)   = 2                                                          00000360
  LTS(10)  = 3                                                          00000370
  LTS(11)  = 2                                                          00000380
  LTS(12)  = 1                                                          00000390
 LTS   (13)= 1                                                          00000400
 LTS   (14)= 1                                                          00000410
 LTS   (15)= 2                                                          00000420
 LTS   (16)= 0                                                          00000430
 LTS   (17)= 4                                                          00000440
 LTS   (18)= 0                                                          00000450
  KUD(1)   = 1                                                          00000460
  KUD(2)   = 0                                                          00000470
  KUD(3)   = 0                                                          00000480
  KUD(4)   = 1                                                          00000490
  KUD(5)   = 1                                                          00000500
  KUD(6)   = 1                                                          00000510
  KUD(7)   = 0                                                          00000520
  KUD(8)   = 0                                                          00000530
  KUD(9)   = 0                                                          00000540
 KUD(10)   = 0                                                          00000550
 KUD(11)   = 0                                                          00000560
 KUD(12)   = 0                                                          00000570
 KUD   (13)= 1                                                          00000580
 KUD   (14)= 0                                                          00000590
 KUD   (15)= 0                                                          00000600
 KUD   (16)= 1                                                          00000610
 KUD   (17)= 1                                                          00000620
 KUD   (18)= 0                                                          00000630
 KRSP(1)   = 1                                                          00000640
 KRSP(2)   = 1                                                          00000650
```

```
      KRSP(3)   = 0
      KRSP(4)   = 1
      KRSP(5)   = 0
      KRSP(6)   = 0
      KRSP(7)   = 1
      KRSP(8)   = 0
      KRSP(9)   = 0
      KRSP(10) = 0
      KRSP(11) = 0 — WRONG, [illegible]BLE [illegible]
      KRSP(12) = 0 — WRONG }
      KPSP (13)= 1
      KRSP (14)= 1
      KRSP (15)= 0
      KRSP (16)= 1
      KRSP (17)= 0
      KRSP (18)= 1
      KR        = 18
      DO 200 NR=1,KR
      DO 100 K=1,8
  100 LREF(NR,K)      = 0
  200 CONTINUE
      IF (KSSP .EQ. 0) GO TO 300
C        P SOURCE
      LREF(2,1)= 1
      LREF(3,2)= 1
      LREF(4,5)= 1
      LREF(4,1)= 1
      LREF(5,5)= 1
      LREF(5,2)= 1
      LREF(6,6)= 1
      LREF(6,4)= 1
      LREF(7,1)= 2
      LREF(7,5)= 1
      LREF(8,1)= 1
      LREF(8,5)= 1
      LREF(8,2)= 1
```

```
      LREF(9,4)= 1                                                      00001020
      LREF(9,1)= 1                                                      00001030
      LREF(9,6)= 1                                                      00001040
      LREF(10,2)      = 1                                               00001050
      LREF(10,8)      = 1                                               00001060
      LREF(10,4)      = 1                                               00001070
      LREF(11,2)      = 1                                               00001080
      LREF(11,8)      = 1                                               00001090
      LREF(11,3)      = 1                                               00001100
      LREF(12,2)      = 1                                               00001110
      LREF(12,7)      = 1                                               00001120
      LREF(12,1)      = 1                                               00001130
      LREF(13,3)      = 1                                               00001140
      LREF(13,6)      = 1                                               00001150
      LREF(14,1)      = 1                                               00001160
      LREF(14,3)      = 1                                               000   X
      LREF(14,6)      = 1                                               00001180
      LREF(15,2)      = 2                                               00001190
      LREF(15,7)      = 1                                               00001200
      LREF(16,1)      = 2                                               00001210
      LREF(16,5)      = 2                                               00001220
      LREF(17,4)      = 2                                               00001230
      LREF(17,6)      = 1                                               00001240
      LREF(17,8)      = 1                                               00001250
      LREF(18,1)      = 3                                               00001260
      LREF(18,5)      = 2                                               00001270
      READ(5,1000)  IB,IE                                               00001280
      READ(5,1100) ((LREF(II,JJ),JJ=1,8),II=IB,IE)                      00001290
      GO TO 900                                                         00001300
  300 CONTINUE                                                          00001310
C         S SOURCE                                                      00001320
      LREF(2,3)= 1                                                      00001330
      LREF(3,4)= 1                                                      00001340
      LREF(4,7)= 1                                                      00001350
      LREF(4,1)= 1                                                      00001360
      LREF(5,7)= 1                                                      00001370
```

```
      LREF(5,2)= 1
      LREF(6,8)= 1
      LREF(6,4)= 1
      LREF(7,3)= 1
      LREF(7,1)= 1
      LREF(7,5)= 1
      LREF(8,3)= 1
      LREF(8,5)= 1
      LREF(8,2)= 1
      LREF(9,3)= 1
      LREF(9,6)= 1
      LREF(9,4)= 1
      LREF(10,4)      = 2
      LREF(10,8)      = 1
      LREF(11,4)      = 1
      LREF(11,8)      = 1
      LREF(11,3)      = 1
      LREF(12,4)      = 1
      LREF(12,7)      = 1
      LREF(12,1)      = 1
      LREF(13,3)      = 1
      LREF(13,8)      = 1
      LREF(14,3)      = 2
      LREF(14,6)      = 1
      LREF(15,2)      = 1
      LREF(15,4)      = 1
      LREF(15,7)      = 1
      LREF(16,1)      = 2
      LREF(16,5)      = 1
      LREF(16,7)      = 1
      LREF(17,4)      = 2
      LREF(17,8)      = 2
      LREF(18,1)      = 2
      LREF(18,3)      = 1
      LREF(18,5)      = 2
      READ(5,1000)   IB,IE
```

```
      READ(5,1100) ((LREF(II,JJ),JJ=1,8),II=IB,IE)
C
 900  CONTINUE
 1000 FORMAT(2I5)
 1100  FORMAT(8I1)
      IF (PRNT) WRITE(6,500) (KUD(J),J=1,KR),(KRSP(J),J=1,KR),
     1                       (LTP(J),J=1,KR),( LTS(J),J=1,KR)
      IF (PRNT) WRITE (6,600) ((LREF(J,L),L=1,8),J=1,6)
      IF (PRNT) WRITE (6,600) ((LREF(J,L),L=1,8),J=7,12)
 400  FORMAT (2F10.0,I10)
 500  FORMAT (' RAYDEF',4(12I2,2X))
 600  FORMAT (' R2',6(8I2,2X))
 800  FORMAT (10X,'WHOOPIE, WE MADE IT TO RAYDEF')
         RETURN
         END
      COMPLEX FUNCTION CR*16 (P,C)
      IMPLICIT REAL*8  (A-H,O-Z)
      COMPLEX*16  P,CZ
      CZ=1./C**2-P*P
      U=CZ
      X=CZ*(0.,-1.)
      R=DSQRT(X*X + U*U)
      W1= DABS(R+U)/2.
      W2= DABS(R-U)/2.
      R1=DSQRT(W1)
      R2=DSQRT(W2)
      CR=R1-R2*(0.,1.)
      END
      REAL FUNCTION CONVS*8 (FA,FP,DEL,NF,N)
      IMPLICIT REAL*8  (A-H,O-Z)
      DIMENSION  FP(1000),FA(1000)
C     COMPUTES CONVOLUTION OF FP AND FA T=DEL*2N
C       NF MUST BE ODD
      NN       = N
      DN       = DEL
      IF(NN.LT.1) GO TO 2
```

```
      NDO        = MINO(NN,(NF-1)/2)
      IP         = 2
      NP         = 2*NN
      EVEN       = FP(IP)*FA(NP)
      ODD        = 0
      IF(NDO.LT.2) GO TO 11
      DO 10 I=2,NDO
      IP         = IP+1
      NP         = NP-1
      ODD        = ODD+FP(IP)*FA(NP)
      IP         = IP+1
      NP         = NP-1
      EVEN       = EVEN+FP(IP)*FA(NP)
10        CONTINUE
11      CONTINUE
      ENDS       = FP(1)*FA(2*NN+1)+FP(IP+1)*FA(NP-1)
      CONVS      = DN*(ENDS+4.*EVEN+2.*ODD)/3.
      RETURN
2       CONVS    = 0.
      END
      SUBROUTINE FA(DEL,J2)
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/SRF/F(1000)
      DO 5 J=2,J2
      F(J)       = 1./(((J-1)*DEL)**.5)
5       CONTINUE
      F(1)       = (11.-4.*2.**.5)/((2.*DEL)**.5)
      END
      SUBROUTINE STEP(NFA,NFAD,NDECK,NPLOT,NPRT,DP)
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/SYTH/XD11,YD11,XD22,YD22,XD33,YD33
      COMMON/SRF/F(1000)
      COMMON/THY/T(1000),PP(1000),RP(600)
      COMMON/STOR/P(1000),TD(1000)
      DIMENSION XL(2),YL1(4),YL2(4),YL3(4)
      DATA XL/'TIME SEC'/
```

```
      DATA YL3/' STEP  RESPONSE '/
      L         = 0
      CALL FA(DP,NFA)
      NFF       = NFA/2
      DO 20 N=1,NFF,NFAD
      L         = L+1
      P(L)      = CONVS(PP,F,DP,NFA-1,N-1)
      TD(L)     = 2.*DP*(N-1)
 20    CONTINUE
      IF(NDECK.LT.1 ) GO TO 1
      NN        = L
      WRITE(7,200) NN,NN,NN
      WRITE(7,100) TD(1),DP,DP
      WRITE(7,100) (P(J),J=1,L)
 1     IF(NPRT.LT.1) GO TO 2
      WRITE(6,300) (TD(J),P(J),J=1,L)
 2     IF(NPLOT.LT.1 ) GO TO 3
      CALL PICTUR(XD22,YD22,XL,-8,YL3,-16,
     2    TD,P,L,0.,0)
 3    CONTINUE
 100   FORMAT(5E15.6)
 200  FORMAT(3I10)
 300   FORMAT(4E15.6)
       END
      SUBROUTINE SETUP(K,MM,NS,NO,MO,MPLOT,MPUNCH)
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/CONFIX/DEL,NN,NDP,TMX,XDIM,YDIM,DP,KO
      COMMON/FOURCT/MF,NMF,KMF,KNMF
      COMMON/THZ/TT(1000),PPZ(1000),PPR(1000)
      COMMON/EXACT/PHIZ(1000),PHIR(1000),TD(1000),NEND,NM
      COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)
      COMMON/PLACE/THIC,H,KSSP
      COMMON/LPRINT/PRNT,PRNTS
      LOGICAL PRNT,PRNTS
      DIMENSION XL(2),YL1(4),YL2(4),YL3(4)
      DATA YL2/' THEORETICAL PO '/
```

```
      DATA XL/'TIME SEC'/                                               00000120
      IF(MO.EQ.0) CALL RAYDEF                                           00000130
      NF       = 1                                                      00000140
      IF(MO.GT.1) GO TO 11                                              00000150
      I        = K                                                      00000160
      TT(1)    = TS(I)                                                  00000170
      PPZ(1)   = 0.                                                     00000180
      PPR(1)   = 0.                                                     00000190
      DO 10 J=2,NN                                                      00000200
      TT(J)    = TT(J-1) +DEL                                           00000210
      PPZ(J)   = 0.                                                     00000220
      PPR(J)   = 0.                                                     00000230
 10    CONTINUE                                                         00000240
 11     CONTINUE                                                        00000250
      DO 32 N=NS,NO                                                     00000260
      WRITE (6,103)  N
103   FORMAT (5X,'RAY NUMBER',I3)
      CALL HIGH(NDP,TMX,K,KI,N)                                         00000270
      CALL ADJUST(NN,NFIX)                                              00000280
      N2       = 1                                                      00000290
      M        = NM+1                                                   00000300
      IF(NFIX.LT.1) GO TO 37                                            00000310
      N2       = NFIX+1                                                 00000320
      N1       = NFIX-1                                                 00000330
      IF(N1.LE.2) GO TO 42                                              00000340
      DO 31 J=1,N1                                                      00000350
      CALL INTERP(TD,PHIZ,NEND,TT(J),Y)                                 00000360
      PPZ(J)   = PPZ(J)+Y*NF                                            00000370
      CALL INTERP(TD,PHIR,NEND,TT(J),Y)                                 00000380
      PPR(J)   = PPR(J)+Y*NF                                            00000390
 31     CONTINUE                                                        00000400
 42    CONTINUF                                                         00000410
      PPZ(NFIX)= PPZ(NFIX)+NF*PHIZ(M)                                   00000420
      PPR(NFIX)= PPR(NFIX)+NF*PHIR(M)                                   00000430
 37      CONTINUF                                                       00000440
       DO 33 J=N2,NN                                                    00000450
```

```
      CALL INTERP(TD,PHIZ,NEND,TT(J),Y)                                 00000460
      PPZ(J)   = PPZ(J)+Y*NF                                            00000470
      CALL INTERP(TD,PHIR,NEND,TT(J),Y)                                 00000480
      PPR(J)   = PPR(J)+Y*NF                                            00000490
33      CONTINUE                                                        00000500
      L=0
32     CONTINUE                                                         00000510
7     IF(.NOT.PRNT) GO TO 12                                            00000520
      PRINT 13,(TD(J),PHIZ(J),PHIR(J),J=1,NEND)                         00000530
      PRINT 14,(TT(J),PPZ(J),PPR(J),J=1,NN)                             00000540
13    FORMAT (1H0,15X,'TD',15X,'PH',/,3(E18.6))                         00000550
14    FORMAT (1H0,15X,'TT',15X,'PP',/,3(E18.6))                         00000560
12    IF(MPLOT.LT.1) GO TO 1                                            00000570
      CALL PICTUR(XDIM,YDIM,XL,-8,YL2,-16,                              00000580
     2 TT,PPZ,NN,0.,0)                                                  00000590
      CALL PICTUR(XDIM,YDIM,XL,-8,YL2,-16,                              00000600
     2 TT,PPR,NN,0.,0)                                                  00000610
1     CONTINUE                                                          00000620
      IF(MPUNCH.LT.1) GO TO 2                                           00000630
      LN       = NN                                                     00000640
      NK       = LN                                                     00000650
      DEL      = DP                                                     00000660
      WRITE(7,100) TT(1),DP,DEL                                         00000670
      WRITE(7,200) NN,LN,NK                                             00000680
      WRITE(7,100) (PPZ(J),J=1,LN)                                      00000690
      WRITE(7,100) (PPR(J),J=1,LN)                                      00000700
2      CONTINUE                                                         00000710
200    FORMAT(3I10)                                                     00000720
100   FORMAT (5E15.6)                                                   00000730
      RETURN                                                            00000740
      END                                                               00000750
      SUBROUTINE ADJUST(NN,NFIX)                                        00000010
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/EXACT/PHIZ(1000),PHIR(1000),TD(1000),NEND,NM
      COMMON/THZ/ T(1000),PPZ(1000),PPR(1000)
      M        = NM+1                                                   00000040
```

```
      TR        = TD(M)
      I         = 0
 80    I        = I+1
      IF(I.GT.NN) GO TO 70
      IF(T(I).GT.TR) GO TO 81
      GO TO 80
 81    DNE      = TR-T(I-1)
      DPL       = T(I)-TR
      IF (DABS(DNE) .GT. DABS(DPL))  GOT O 83
      DELTA     = -DNE
      NFIX      = I-1
      GO TO 85
 83     DELTA   = DPL
      NFIX      = I
 85    DO 84 J=1,NEND
      TD(J)     = TD(J)+DELTA
 84     CONTINUE
      RETURN
 70     NFIX    = 0
      END
      REAL FUNCTION TS*8 (K)
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/PLACE/THIC,H,KSSP
      COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)
      COMMON/ORSTF/CC(100),SS(100),DD(100),TTH(100),XX
      COMMON/STUFF/C(100),S(100),D(100),TH(100),X,RCSQ(100),RSSQ(100)
      DIMENSION T(200)
      COMMON/LPRINT/PRNT,PRNTS,KST,KEND,PRNTC,NDC,DET
      LOGICAL PRNT,PRNTS
      DO 10 N=1,2
      DEL       = 1./SS(2)
      IF(KSSP.EQ.0) GO TO 81
      IF(LTS(N).GT.0) GO TO 81
      DEL       = 1./CC(2)
 81     P       = -1.E-9
      DET       = 1.E+12
```

```
      CALL FIND?(P,M,DEL,DET,PO,TO,N)                                   00000160
      P         = 1./CC(N+1)                                            00000170
      TTP       = TO                                                    00000180
      IF(PO.LE.P) GO TO 6                                               00000190
      CALL HELP(K,N,P,TTP,DTP,N)                                        00000200
 6    T(N)      = DMIN1(TO,TTP)                                         00000210
 10     CONTINUE                                                        00000220
      TS        = DMIN1(T(1),T(2))                                      000 02T
       N        = 2                                                     00000240
      IF (PRNT)WRITE (6,1) (T(J),J=1,N)                                 00000250
 1    FORMAT (5X,'T 1,2,3',3(E18.6))                                    00000260
 100   FORMAT(I10,2E18.6)                                               00000270
      RETURN                                                            00000280
      END                                                               00000290
      SUBROUTINE PSICO (P,FNZ,FNR,FNZ1,FNR1,I,NRY)                      00000010
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)    00000020
      COMMON/ORSTF/C(100),S(100),D(100),TH(100),X                       00000030
      COMMON/MAGIC/PP(1200),DDPT(1200),TT(1200)
      COMMON/PLOTC/CCN,NNF,NPT                                          00000050
      COMMON/PLACE/THIC,H,KSSP                                          00000060
      COMMON /LPRINT/ PRNT,PRNTS                                        00000070
      COMPLEX*16  P,RF,DZ,DR,G1,G2,G3,FNZ,FNR,FNZ1,FNR1,RC ,DDPT ,PP
     1      ,RPP,RPS,RSP,RSS                                            00000090
      LOGICAL  PRNT,PRNTS                                               00000100
      DIMENSION RF(16)                                                  00000110
      ROD       = D(2)/D(3)                                             00000120
      CALL REFFT (P,C(2),S(2),C(3),S(3),ROD,RPP,RPS,RSP,RSS)            00000130
      RF(1)     = RPP                                                   00000140
      RF(2)     = RPS                                                   00000150
      RF(3)     = RSP                                                   00000160
      RF(4)     = RSS                                                   00000170
      CALL RECVR (C(2),S(2),P,KRSP(NRY),DZ,DR,KSSP,RF(5),RF(6),RF(7),
     1           RF(8))
      RC        = (1.,0.)                                               00000240
      DO 100 J=1,8                                                      00000250
```

```
      IF (LREF(NRY,J) .EQ. 0) GOTO 200
      RC         = RC*RF(J)**LREF(NRY,J)
  200 CONTINUE
  100 CONTINUE
      IF(KSSP.EQ.1) GO TO 3
      IF(NRY.EQ.1) KRSP(NRY)=0
3      CONTINUE
      G1         = DDPT(I)*RC*CON
      G2         = G1*CDSQRT(P/(2.*X))
      FNZ        = DZ*G2
      FNR        = DR*G2
      G3         = G1/(-8.*CDSQRT(2.*P*X**3))
      FNZ1       = G3*DZ
      FNR1       = G3*DR*(-3.)
      IF (I .EQ. 20) GO TO 600
      IF (I .EQ. 40) GO TO 600
      IF (I .EQ. 60) GO TO 600
      GO TO 700
  600 CONTINUE
      IF (PRNT) WRITE (6,300) P,C(3),C(2),S(3),S(2),KUD(NRY),KRSP(NRY),
     1                        LTS(NRY),LTP(NRY)
      IF (PRNT) WRITE (6,400) (RF(J),LREF(NRY,J), J=1,4)
      IF (PRNT) WRITE (6,400) (RF(J),LREF(NRY,J), J=5,8)
      IF (PRNT) WRITE (6,500) G1,G2,G3,RC
  300 FORMAT (1X,'RECVR',2(E15.4),4(F10.3),2X,4(I2))
  400 FORMAT (1X,'RECVR',4(2(E13.4),I2))
  500 FORMAT (1X,'RECVR',8(E13.4))
  700 CONTINUE
      RETURN
      END
      SUBROUTINE CURAY(JO)
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON /SENSE/ DRCSQ(100) ,DRSSQ(100)
      COMMON/STUFF/C(100),S(100),D(100),TH(100),X,RCSQ(100),RSSQ(100)
      COMMON/ORSTF/CC(100),SS(100),DD(100),TTH(100),XX
      COMMON/FIXP/DDN(100),ARN(100),FLAT
```

```
      LOGICAL FLAT
      DIMENSION DEPTH(100)
       DIMENSION DT(100)
       X        = XX
      PRINT 2, X
2     FORMAT (1H1,10X'CURAY'/11X'RANGE'F10.0/16X'THICKNESS'9X'DEPTH'5X'P
     +-VELOCITY'5X'S-VELOCITY'8X'DENSITY')
      DEPTH(1) = TTH(1)/2.0
      DO 10 J = 2,JO
10     DEPTH(J)= DEPTH(J-1)+(TTH(J)+TTH(J-1))/2.
      DO 5 J = 1,JO
      Q        = 6371.0 / (6371.0-DEPTH(J))
      IF(FLAT) Q     = 1.
      ARN(J)   = 1./Q
      C(J)     = CC(J)*Q
       S(J)    = SS(J)*Q
      D(J)     = DD(J)*Q
       TH(J)   = TTH(J)*Q
      DRCSQ(J) = 1.0 / C(J) **2
      DRSSQ(J) = 1.0 / S(J)**2
      RCSQ(J)  = DRCSQ(J)
      RSSQ(J)  = DRSSQ(J)
5       CONTINUE
      DT(1)     = TTH(1)
      DO 20 J=2,JO
      DT(J)     = DT(J-1) +TTH(J)
20      CONTINUE
      DO 25 J=1,JO
      IF(FLAT) DT(J) = 0.0
      DDN(J)    = (6371.-DT(J))/6371.
25      CONTINUE
      PRINT 1, (J,TH(J),DEPTH(J),C(J),S(J),D(J),ARN(J),DDN(J),J=1,JO)
1     FORMAT (I5,5X,7G15.4)
      RETURN
      END
      SUBROUTINE FIND2 (Q,K,DEL,DET,PQ,TQ,NRY)
```

```
      IMPLICIT REAL*8  (A-H,O-Z)
      DIMENSION E(100)
      COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)      00000020
      COMMON/PLACE/THIC,H,KSSP                                              00000030
      COMMON/STUFF/C(100),S(100),D(100),TH(100),X                           00000040
      COMMON /SENSE/ RCSQ(100),RSSQ(100)                                    00000050
      COMMON / LPRINT/ PRNT,PRNTS                                           00000060
      LOGICAL PRNT,PRNTS                                                    00000070
      TH(1)     = (1.-KUD(NRY))*(THIC-H) + KUD(NRY)*H                      00000090
      TH(3)     = 0.                                                        00000100
      TH(4)     = 0.                                                        00000110
      IF (KSSP .EQ. 1)  TH(4)=TH(1)                                         00000120
      IF (KSSP .EQ. 0)  TH(3)=TH(1)                                         00000130
      KF        = 1                                                         00000140
      KOUNT     = 0                                                         00000150
      TDE       = DEL                                                       00000160
 8    P         = Q                                                         00000170
      KOUNT     = KOUNT + 1                                                 00000180
 5      P       = P+DEL                                                     00000190
      PSQ       = P ** 2                                                    00000200
      E(2)      = DSQRT(DABS(RCSQ(2)-PSQ))                                  00000210
      E(3)      = DSQRT(DABS(RSSQ(2)-PSQ))                                  00000220
      BLTEM     = -TH(2)*LTP(NRY)/E(2) -TH(2)*LTS(NRY)/E(3)                 00000230
      BLTEM     = BLTEM-TH(4)/E(2)                                          00000240
      BLTEM     = BLTEM-TH(3)/E(3)                                          00000250
      BL        = X + BLTEM*P                                               00000260
      IF(DABS (DEL).LE.1.E-18) GO TO 1                                      00000270
 6    IF (DABS(BL).LE.X/DET) GO TO 1                                        00000280
 2     IF(BL)3,1,4                                                          00000290
 3     DEL      =-DABS (DEL*.5)                                             00   TO
      GO TO 5                                                               00000310
 4     DEL      =DABS(DEL*.5)                                               00 0 3S
       GO TO 5                                                              00000330
 1    IF(DABS(BL).LT.1.E-8) GO TO 7                                         00000340
      IF (KOUNT.GE.5) GO TO 7                                               00000350
      Q         = Q/10.0                                                    00000360
```

```
      DEL        = TDE                                                  00000370
      GO TO 8                                                           00000380
 7    PO         = P                                                    00000390
      TOTEM      = E(2)*TH(4)+E(2)*LTP(NRY)*TH(2)+E(3)*LTS(NRY)*TH(2)   00000400
     1      +E(3)*TH(3)                                                 00000410
      TO         = P*X + TOTEM                                          00000420
      PQ         = PO                                                   00000430
      TQ         = TO                                                   00000440
      IF (DABS(BL).LT.1.0E-6) RETURN                                    00000450
      IF (.NOT.PRNT) RETURN                                             00000460
      WRITE (6,17) PO,TO,BL                                             00000470
 17   FORMAT (1H0,4X,'PO   ',E18.6,10X,'TO   ',E18.6,10X,'BL   ',E18.6)   0000048C
      RETURN                                                            00000490
      END                                                               00000500
      REAL FUNCTION PTIM*8  (P,K,NRY)
      IMPLICIT REAL*8  (A-H,O-Z)
      COMMON/RAYPAR/KUD(100),KRSP(100),LTS(100),LTP(100),LREF(100,8)    00000020
      COMMON/PLACE/THIC,H,KSSP                                          00000030
      COMMON/STUFF/C(100),S(100),D(100),TH(100),X,RCSQ(100),RSSQ(100)   00000040
      PSQ        = P ** 2                                               00000050
      J          = 2                                                    00000060
      F          = DSQRT(DABS(RCSQ(J)-PSQ))                               000000
      PTIM       = (TH(4) + LTP(NRY)*TH(2)) *E                          00000080
      E          =DSQRT(DABS(RSSQ(J)-PSQ))                               0000009
      PTIM       = PTIM + (TH(3) + LTS(NRY)*TH(2))*E                    00000100
      PTIM       = P*X + PTIM                                           00000110
      END                                                               00000120
      SUBROUTINE INTERP(XP,YP,N,X,Y)                                    00000010
      IMPLICIT REAL*8  (A-H,O-Z)
      DIMENSION  XP(1000),YP(1000)
 1    IF (X .GT.XP(N))GO TO 6                                           00000040
      IF (X .LT. XP(1)) GO TO 6                                         00000050
 2    DO 10 I=1,N                                                       00000060
      IF (XP(I) -X) 10,102,3                                            00000070
 10   CONTINUE                                                          00000080
 3    K          = I-1                                                  00000090
```

```
      DIF1      = XP(I) -XP(K)                                          00000100
      DIF2      = XP(I) -X                                              00000110
      RATIO     = DIF2/DIF1                                             00000120
      DIFY      =DABS (YP(I) - YP(K))                                   00000 30
      DR        = DIFY*RATIO                                            00000140
      IF (YP(I) .GT. YP(K)) GO TO 4                                     00000150
 5    Y         = YP(I) + DR                                            00000160
      RETURN                                                            00000170
 4    Y         = YP(I) - DR                                            00000180
      RETURN                                                            00000190
 102  Y         = YP(I)                                                 00000200
      RETURN                                                            00000210
 6    Y         = 0.                                                    00000220
      RETURN                                                            00000230
      END                                                               00000240
      SUBROUTINE REFFT(P,V1,S1,V2,S2,D ,RPP,RPS,RSP,RSS)                00000010
      IMPLICIT COMPLEX*16 (A-H,O-Z)
      REAL*8     K1,K2,K3,K4,D,V1,V2,S1,S2
      K4        = S2**2/(S1**2*D)                                       00000060
      B1        = .5/(1-K4)                                             00000070
      B2        = .5*K4/(K4-1)                                          00000080
      K1        = B1/S1**2                                              00000090
      K2        = B2/S2**2                                              00000100
      K3        = K1+K2                                                 00000110
      E1        = CR(P,V1)                                              00000120
      E2        = CR(P,V2)                                              00000130
      E1P       = CR(P,S1)                                              00000140
      E2P       = CR(P,S2)                                              00000150
      C1        = (P**2)*(K3-P**2)**2                                   00000160
      C2        = P**2*E1*E1P*E2P                                       00000170
      C3        = (E1*E1P)*(K2-P**2)**2                                 00000180
      C4        = E2P*(K1-P*P)**2                                       00000190
      C5        = K1*K2*E1*E2P                                          00000200
      C6        = K1*K2*E1P                                             00000210
      AP        = C1+C3-C5                                              00000220
      BP        = C2+C4-C6                                              00000230
```

```
A         = -C1+C3-C5                                              00000240
B         = -C2+C4-C6                                              00000250
BT        = AP+BP*E2                                               00000260
RPP       = (A-B*E2)/BT                                            00000270
APS       = 2.*P*E1 *(K2-P*P)*(K3-P*P)                             00000280
BPS       = 2.*P*E1*(K1-P*P)*E2P                                   00000290
RPS       = (APS-BPS*E2)/BT                                        00000300
A         = -C1 +C3 +C5                                            00000310
B         = -C2 +C4 +C6                                            00000320
 RSS      = (A-B*E2)/BT                                            00000330
ASP       = 2.*P*E1P*(K2-P*P)*(K3-P*P)                             00000340
BSP       = 2.*P*E1P*(K1-P*P)*E2P                                  00000350
RSP       = -(ASP-BSP*E2)/BT                                       00000360
END                                                                00000370
```