Wave Propagation on the Surface of a Sphere

Don Funkhouser

Geophysics Department

Stanford University

June 7, 1972

I. Introduction

Many interesting phenomena can be described as the occurance of a wave confined to the surface of a sphere. Rayleigh waves produced by earthquakes are probably the most obvious example of this principle. Radio science people have encountered radio waves trapped in various atmospheric layers and noticed them being transmitted around the globe and back. Atmospheric pressure waves produced by explosions have been seen to propagate around the Earth several times, being trapped in the air-land interface. From a wave analysis point of view, even the so-called free-Earth oscillations could be thought of as many surface waves combining and interfering to form the longer observed periods.

The scientist's bidding in life is to devise models of observed natural phenomena which closely approximate the real event. So closely that, in fact, the real event's behavior can be completely formulated beforehand. With this ultimate goal in mind, the scientist must grope for a deeper and deeper understanding of the event in order to mimic it -- and thus knowledge expands. With this cosmic directive in mind, I have attempted to make some initial contributions toward modeling the phenomena desribed above. The following paper represents the sum of my efforts thus far. The objectives in starting this study were to come up with a method by which one could reallistically propagate a disturbance on a sphere numerically, do it cheaply enough computer-wise to make it feasible, and to design it with enough flexibility so as to model some real phenomena (i.e. have it include options for making the wave velocity space-variable). Not all of these objectives have been fully reallized, but at least no dead-ends have been encountered yet. The computational methods used here bear a striking similarity to those used by Jon Claerbout (1970)[1] , mainly due to his generous supply of guidance and inspiration in formulating the solution to this problem.

II. Mathematical Derivations

The logical place to start in attempting to model wave phenomena is to look at the basic wave equation -- the classical scaler wave equation :

$$(1) \quad \nabla^2 P = 1/c^2 \, \partial P/\partial t^2$$

P is the wave function and c is the wave velocity, which can be space-variable. Since we are dealing with a sphere, the next logical step would be to use spherical coordinates for our description. The expression for the Laplacian in spherical coordinates is :

$$(2) \quad \nabla^2 P = [1/r^2][\partial(r^2 \partial P/\partial r)/\partial r] + [1/r^2 \sin\theta][\partial(\sin\theta \partial P/\partial\theta)/\partial\theta] + [1/r^2 \sin^2\theta][\partial P/\partial\phi^2]$$

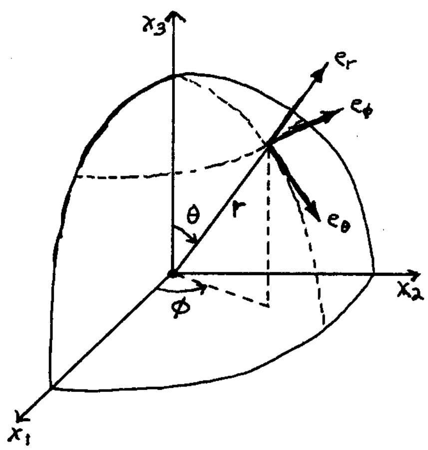
The typical definition of sperical coordinates is used, as is illustrated in Plate 1. We will define the center of the system to be at the center of the sphere, so that the "North Pole" is at $\theta = 0^\circ$ and the "South Pole" at $\theta = 180^\circ$. As we progress around the "Equator", ϕ will go from 0° to 360°. When talking about confining something to the surface of a sphere, we are essentially taking a surface described by sweeping out a constant radius. In that light, let us assume that r is constant and that this constant is equal to 1. This simplifies our expression in (2) to give :

$$(3) \quad \nabla^2 P = [1/\sin\theta][\partial(\sin\theta \, P/\partial\theta)/\partial\theta] + [1/\sin^2\theta][\partial^2 P/\partial\phi^2]$$

Putting this into (1) gives:

$$(4) \quad [1/\sin\theta][\partial(\sin\theta \, P/\partial\theta)/\partial\theta] + [1/\sin^2\theta][\partial^2 P/\partial\phi^2] - 1/c^2 \, \partial^2 P/\partial t^2 = 0$$

Now comes the problem of how to solve this equation in a meaningful and also simple way. If we consider a



$$X_1 = r \sin\theta \cos\phi$$
$$X_2 = r \sin\theta \sin\phi$$
$$X_3 = r \cos\theta$$

Spherical Coordinate Definitions

Plate 1

small strip around the equator, we can see that in a small angle approximation, this strip would be essentially flat. This gives us a hint to try and model this wave equation as a disturbance propagating in one direction, parallel to the "Equator". The way to do this is to assume an exponential dependence of both ϕ and t, to get a perturbation type solution to (4) of the form :

$$(5) \quad P = F(\theta,\phi) \exp [i(m\phi - wt)]$$

Let us now take all necessary derivatives of (5) in order to get a perturbation solution to (4), whilst making the definitions $S = \sin\theta$ and $C = \cos\theta$:

$$(6) \quad \partial P/\partial\theta = F_\theta \exp [i(m\phi - wt)]$$

$$(7) \quad \partial(S\,\partial P/\partial\theta)/\partial\theta = [CF_\theta + SF_{\theta\theta}] \exp [i(m\phi - wt)]$$

$$(8) \quad \partial P/\partial\phi = [F_\phi + imF] \exp [i(m\phi - wt)]$$

$$(9) \quad \partial^2 P/\partial\phi^2 = [F_{\phi\phi} + 2imF_\phi - m^2F] \exp [i(m\phi - wt)]$$

$$(10) \quad \partial^2 P/\partial t^2 = -w^2F \exp [i(m\phi - wt)]$$

If we now take (7), (9), and (10) and plug them into (4), whilst multiplying everything by $S^2 \exp [-i(m\phi - wt)]$, we arrive at:

$$(11) \quad S^2F_{\theta\theta} + CSF_\theta + F_{\phi\phi} + 2imF_\phi + (w^2S^2/c^2 - m^2)F = 0$$

This equation looks somewhat less formidable than (4), but we have to decide what to do about c, which we want to be space-variable. Let us first assume that c is a constant, say c_s. Since we would want the F term to vanish at the "Equator" ($\theta = 90^o$), this would imply taking $m^2 = w^2/c_s^2$. If our c term varies only a small bit from our c_s, we can put this approximation into our F term to get:

$$(12) \quad w^2[S^2/c(\theta,\phi)^2 - 1/c_s^2]\,F$$

If $c(\theta,\phi) = c_s$, equation (12) reduces to $-C^2m^2F$, which indeed vanishes at the "Equator". Let us now define :

$$(13) \quad v^2 = -[S^2c_s^2/c(\theta,\phi)^2 - 1]$$

so that we transform (11) to :

$$(14) \quad S^2F_{\theta\theta} + CSF_\theta + F_{\phi\phi} + 2imF_\phi - m^2v^2F = 0$$

This is still a formidable equation to work with (due mainly to the prescence of both an $F_{\theta\theta}$ term and an $F_{\phi\phi}$ term), so we need to look at Problem #2 on page 279 of <u>Waveform Analysis Class Notes</u>[2].for a very useful trick. Let us define a new operator such that :

$$(15) \quad T^2F = S^2F_{\theta\theta} + CSF_{\theta} \quad \text{or} \quad T^2 = S\partial[S\partial F/\partial\theta]/\partial\theta$$

Putting this expression into (14) gives :

$$(16) \quad F_{\phi\phi} + 2imF_{\phi} + [T^2 - m^2v^2]\ F = 0$$

If we now let F go outside we get :

$$(17) \quad [d_{\phi\phi} + 2imd_{\phi} + (T^2 - m^2v^2)\ ]\ F = 0$$

if we now let $x = d_{\phi}$ and $x^2 = d_{\phi\phi}$, we get something that looks like a quadratic equation :

$$(18) \quad [\ x^2 + 2imx + (T^2 - m^2v^2)]\ F = 0$$

The standard quadratic folmula looks like :

$$(19) \quad a + bx + cx^2 = 0$$

If x is small, a linear approximation to this equation would be to have $x = -a/b$. If we use this approximation for our quadratic, we would have the form :

$$(20) \quad a + bx + cx\ (-a/b) = 0$$

Or, in combining terms :

$$(21) \quad ab + (b^2 - ac)\ x = 0$$

If we now identify terms between (18) and (19), we see that $a = T^2 - v^2m^2$, $b = 2im$, $c = 1$. Putting these into the form of (21) gives us a good approximation to equation (17) :

$$(22) \quad [(T^2 - v^2m^2)(2im) + (-4m^2 - T^2 + v^2m^2)d_{\phi}]\ F = 0$$

Combining terms gives :

$$(23) \quad (T^2-v^2m^2)(2im)F + [m^2(v^2-4)-T^2]\ dF/d\phi = 0$$

Let us now simplify by defining :

$$(24) \quad A_1 = im \quad A_2 = -im^3v^2 \quad A_3 = m^2(v^2-4)/d\phi \quad A_4 = -1/d\phi$$

This gives us :

$$(25) \quad 2(A_1T^2 + A_2)\ F + (A_4T^2 + A_3)\ dF = 0$$

We now have a form that can almost be solved using a finite-difference formulation, such as the Crank-Nicolson method.[3] Let us define an F_j^n , such that the n superscript denotes the $\emptyset$ grid spacing and the j subscript denotes the θ grid spaces. We can formulate that $F = \frac{1}{2}(F_j^{n+1} + F_j^n)$ and $dF = (F_j^{n+1} - F_j^n)$ while still keeping a convergent system, since this calculation scheme would be centered at $F_j^{n+\frac{1}{2}}$. If we now substitute this difference formulation into (25) and collect the terms, we get :

$$(26)\quad [(A_2+A_3) + (A_1+A_4)T^2]\ F_j^{n+1} = [(A_3-A_2) + (A_4-A_1)T^2]\ F_j^n$$

Since simplificaion is the name of the game, let us define :

$$(27)\quad A_5 = A_2 + A_3 \quad A_6 = A_1 + A_4 \quad A_7 = A_3 - A_2 \quad A_8 = A_4 - A_1$$

to give us :

$$(28)\quad [A_5 + A_6T^2]\ F_j^{n+1} = [A_7 + A_8T^2]\ F_j^n$$

All is well and good, but what do we do with T^2 ? Let us go back to the definition of T^2 :

$$(29)\quad T^2 = S\partial[S\partial F/\partial\theta]/\partial\theta$$

We wish to get a finite difference formulation for T^2, which is centered at $F_j^{n+\frac{1}{2}}$. To do this, we define :

$$(30)\quad \partial F/\partial\theta = (F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n)\ /\ d\theta$$

So we get :

$$(31)\quad T^2 = S_j\partial[S_j(F_{j+\frac{1}{2}}^n - F_{j-\frac{1}{2}}^n)/d\theta]/\partial\theta$$

In finite difference formulation this goes to :

$$(32)\quad T^2 = [S_j/(d\theta)^2][S_{j+\frac{1}{2}}(F_{j+1}^n - F_j^n) - S_{j-\frac{1}{2}}(F_j^n - F_{j-1}^n)]$$

Upon collecting terms, we get :

$$(33)\quad T^2 = [S_jS_{j+\frac{1}{2}}/(d\theta)^2]\ F_{j+1}^n - [S_j(S_{j+\frac{1}{2}}+S_{j-\frac{1}{2}})/(d\theta)^2]F_j^n + [S_jS_{j-\frac{1}{2}}/(d\theta)^2]\ F_{j-1}^n$$

Further simplification makes us define :

$$(34)\quad T_1 = S_jS_{j+\frac{1}{2}}/(d\theta)^2 \quad T_2 = S_j(S_{j+\frac{1}{2}}+S_{j-\frac{1}{2}})/(d\theta)^2 \quad T_3 = S_jS_{j-\frac{1}{2}}/(d\theta)^2$$

We finally get a nice expression for T^2 :

$$(35) \quad T^2 = T_1 F^n_{j+1} + T_2 F^n_j + T_3 F^n_{j-1}$$

We now have what we were looking for -- an expression for T^2 that we can put into (28). Doing so and collecting terms gives us :

$$(36) \quad (A_6 T_1) F^{n+1}_{j+1} + (A_5 + A_6 T_2) F^{n+1}_j + (A_6 T_3) F^{n+1}_{j-1} = (A_8 T_1) F^n_{j+1} + (A_7 + A_8 T_2) F^n_j + (A_8 T_3) F^n_{j-1}$$

Simplification demands that we define :

$$(37) \quad A_k = A_6 T_1 \quad B_k = A_5 + A_6 T_2 \quad C_k = A_6 T_3$$
$$D_k = A_8 T_1 \quad E_k = A_7 + A_8 T_2 \quad F_k = A_8 T_3$$

Finally, we get, in a nice, neat package :

$$(38) \quad A_k F^{n+1}_{j+1} + B_k F^{n+1}_j + C_k F^{n+1}_{j-1} = D_k F^n_{j+1} + E_k F^n_j + F_k F^n_{j-1}$$

What we now have is a numerical means by which we can propagate a disturbance F^{n+1} if we are given an initial disturbance F^n. Definitions for all terms used to arrive at equation (38) can be found in equations (13), (24), (27), (34), and (37).

III. Programming Notes

The programming of the resultant equation (38) is fairly straightforward, but getting the information in and out in a useable form proved to be the major hangup. Equation (38) specifies a matrix equation of the form [A][T] = [B][D] , where the coefficients in [A] and [B] are dependent on both θ and ϕ, and the vector [D] is specified initially. The equation is then solved for [T], which represents the next spatial step in the wave propagation. This is then put into (38) as the new [D] vector, the equation solved again, the wave propagated, etc., etc.. Since the equation we are trying to solve involves only 2nd order partial derivatives, the Crank-Nicolson formulation yields a system which has [A] and [B] as tri-diagonal matrices. Hence solving (38) amounts to nothing more than calculating the various coefficients and solving a simple tri-diagonal matrix system. Algorithms exist for very quick solutions to such systems, but have one inherent problem with their use -- boundary conditions. A pictorial diagram of the system specified by (38) might look like :

$$\begin{bmatrix} B_1 & A_1 & & & \\ C_2 & B_2 & A_2 & & \\ & C_3 & B_3 & A_3 & \\ & & & & A_{n-1} \\ & & & C_n & B_n \end{bmatrix} \begin{bmatrix} T_1 \\ T_2 \\ T_3 \\ \\ T_n \end{bmatrix} = \begin{bmatrix} E_1 & D_1 & & & \\ F_2 & E_2 & D_2 & & \\ & F_3 & E_3 & D_3 & \\ & & & & D_{n-1} \\ & & & F_n & E_n \end{bmatrix} \begin{bmatrix} D_1 \\ D_2 \\ D_3 \\ \\ D_n \end{bmatrix}$$

To specify (38) completely, one should have A_n acting on a T_{n+1} in the last step, but due to the limits of our vectors being of finite size, this is not possible. One way to handle the adverse effects of this problem is to specify the so-called zero-slope end conditions. In this method, any energy coming out to the boundaries is matched up with the boundary and sent back into the matrix. This effect of having energy bounce back into the array is a real one (as evidenced by some of the plots in the Results section), but one can effectively try to keep energy from getting to the boundaries by being clever. In the case of this problem, the effect is even less of a worry since energy going out to the boundaries is sent back to the "Equator", due to the curvature of the sphere. The main work-horse of the program is resident in the tridiagonal matrix subroutine CMVTR. All of the operations must be done in complex arithmetic, due to the presence of imaginary constants popping up in equation (24). Provisions have been made to make the wave velocity spatially variable, so that in essence all of the matrix coefficients must be calculated for each step. Since the solution to (38) is just for monochromatic waves, programming provisions were made to simulate multiple frequency propagations through means of superimposing individual run. The input waveform can either be calculated or read in. Capabilities for approximating a plane wave source, a tilted plane wave source, an offset plane wave source, or any combinations of these are controlled by the input parameter card. The main effort in the programming was to get output in a useable form -- that which lends itself to quick and easy transmittal of phase and amplitude information associated with the waveforms. This problem is non-trivial and hence four separate plotting programs were developed for this purpose :

(1) A simple line-printer produced plot which printed positive amplitudes with the symbols from "1" to "9" and with negative amplitudes as blanks.

(2) A double-print line-printer plot which was striving for the effect of a variable-density type of plot.

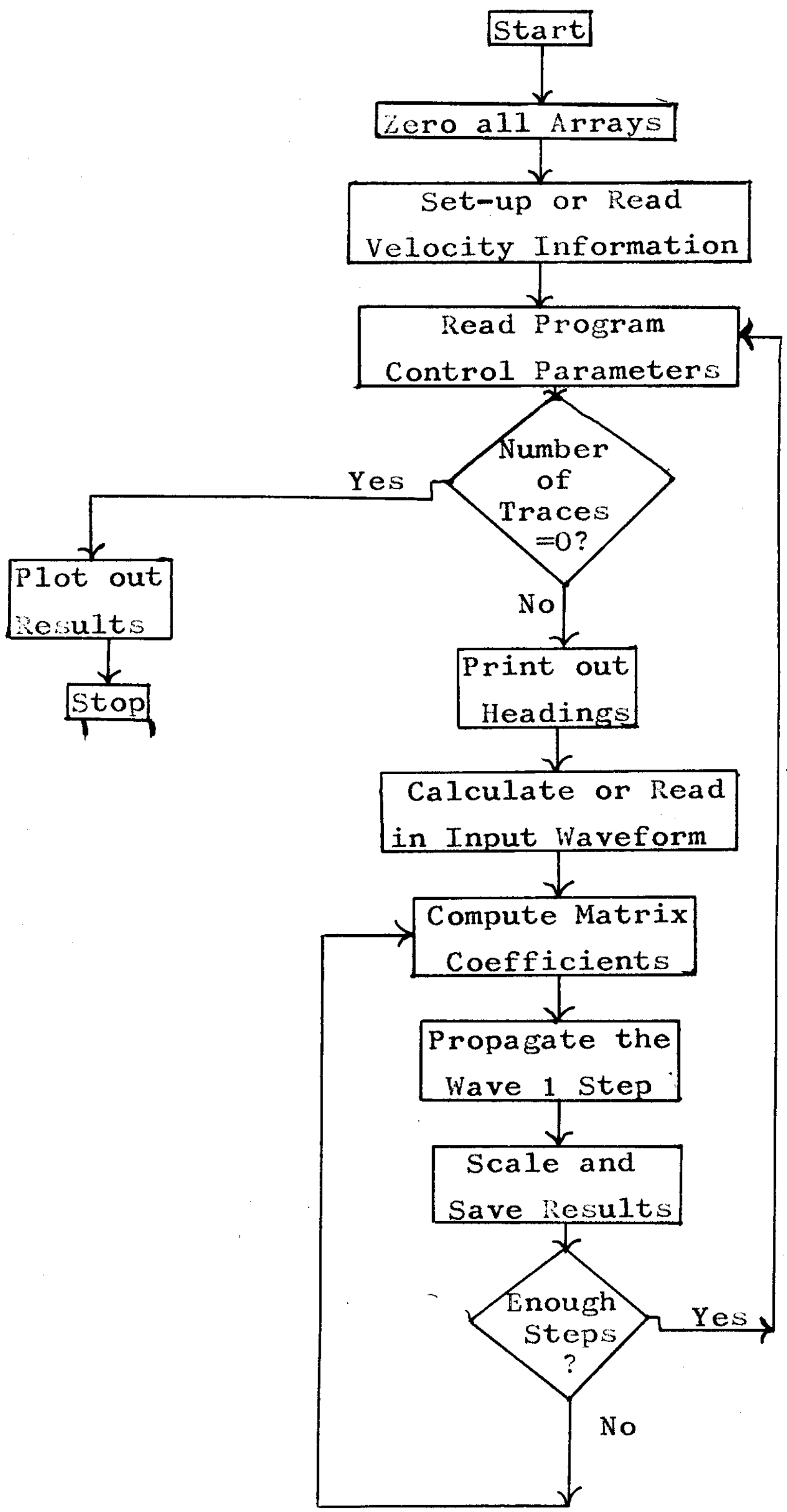
(3) A variable-density plot made on a Versatec plotter which plotted only the positive amplitudes and did a linear interpolation between adjacent steps.

(4) A variable-density Versatec plot, with a bias level of 8 bits, so that both positive and negative amplitudes could be inferred. A linear interpolation function was used here as well, with only every other trace being displayed

Examples of each of these types of plotting routines are included in the Results section.

The main advantage in using plots (1) or (2) is that they would enable one to run under WATFIV, thus cutting down on the computer time needed, which is quite useful in the debugging stage of development. Plots (3) and (4) were by far superior, but were more expensive to produce, since execution would have to proceed under Fortran Level G. A typical printer-plot run for a monochromatic wave with a constant velocity took on the order of 10 seconds of CPU time. Similar runs with a Versatec plot took on the order of 20 seconds of CPU time. A listing of the final program, a typical set of input parameters (those used to produce Example #7), and listings of the various plot routines are included here for reference. All calculations were made on the Stanford Computation Center's IBM 360/67 Computer.

Summary Flowchart of Program Logic



```
C.....THIS IS A PROGRAM DESIGNED TO PROPAGATE A WAVE ON THE SURFACE OF A
C.....SPHERE USING A CRANK-NICOLSON FINITE DIFFERENCE FORMULATION OF THE
C.....SCALER WAVE EQUATION IN SPHERICAL COORDINATES.  INCLUDED ARE VAR-
C.....IOUS OPTIONS TO USE A SPACE-VARIABLE VELOCITY, DIFFERENT INPUT -
C.....WAVEFORMS, AND FOUR DIFFERENT DISPLAY PLOTS OF THE DATA.
C.....
C.....PROGRAMMED BY DON FUNKHOUSER, LAST REVISION = JUNE 1, 1972
C.....
C..........................................................................
      COMMON     T(70),D(70),F(70),E(70),AK(70),BK(70),CK(70),DK(70)
      COMMON EK(70),FK(70),ZAP(70,250),CM,AN,ZAPMX,DP,N,IPLT,NTM,NO
      COMPLEX T,D,F,E,AK      ,BK,CK,DK,EK,FK,CMPLX,CEXP,CZ,CZD,D1
      COMPLEX A1,A2,A3,A4,A5,A6,A7,A8,A9
      DIMENSION             IMAP(70,250)
      INTEGER*2 IMAP
      DO 115 I=1,70
          DO 116 J=1,250
               IMAP(I,J)=0
               ZAP(I,J)=0.
116       CONTINUE
115   CONTINUE
C.....SET UP THE VELOCITY INFORMATION
      DO 305 I=35,70
      DO 305 J=1,250
          IMAP(I,J)=1
305   CONTINUE
C.....READ IN THE PROGRAM CONTROL PARAMETERS, DEFINED AS FOLLOWS:
C.....N=NUMBER OF POINTS IN THE THETA DIRECTION  NIT=LIMIT TO THE
C.....NUMBER OF PHI ITERATIONS  AN=ANGLE SPREAD FROM THE CENTER  TP=
C.....TOTAL PHI TO CALCULATE FOR  DS=INCLINATION ANGLE FOR THE INPUT
C.....WAVEFORM  VELDF=VELOCITY DIFFERENTIAL  PTPW=POINTS PER WAVELENGTH
C.....NN=THETA OFFSET FOR THE INPUT WAVEFORM  IFLG=INPUT OPTION, IF = 1
C.....POINTS CAN BE ENTERED MANUALLY  IPLT=PLOT OPTION AS DESCRIBED IN
C.....THE SUBROUTINES.  IF N=0, THE PROGRAM TERMINATES AND THE RESULTS
C.....ARE PLOTTED.
73    READ (5,1) N,NIT,AN,TP,DS,VLDF,PTPW,NN,IFLG,IPLT
      IF (N.EQ.0) GO TO 191
1     FORMAT (2I5,5F10.5,3I5)
      RDCVT=0.017453
      DT=2.*AN/(N-1)
      DP=1.000*DT
      CM=TP/(PTPW*DP)
      WRITE (6,3) N,NIT,DT,TP,CM,AN
3     FORMAT ('1','# OF POINTS = ',I5,3X,'# OF ITERATIONS = ',I5,3X,'DEL
     *TA THETA = ',F8.4,3X,'TOTAL PHI = ',F8.4,3X,'W/C = ',F5.2,3X,'ANG.
     * LIM. = ',F5.2)
      DTR=DT*RDCVT
      DPR=DP*RDCVT
      THI=(90.-AN)*RDCVT-DTR
      VELRT=1.0+VLDF
      WRITE (6,9) DP,       PTPW,DS,NN,VELRT
9     FORMAT (1X,'DELTA PHI = ',F8.4,3X,                          3X,
     *'POINTS PER WAVELENGTH =  ',F5.1,3X,'INPUT ANGLE = ',F5.1,3X,
     *'OFFSET = ',I5,3X,'VEL. RATIO = ',F7.3)
      NIT1=TP/DP+1
      DTRS=DTR*DTR
      DTR2=DTR/2.
      WRITE (6,2)
2     FORMAT ('1')
```

```
      N1=N-1
      IF (IFLG.EQ.1) GO TO 19
C.....COMPUTE VALUES FOR THE INPUT WAVEFORM
      DO 15 I=1,N
         SINT=SIN(3.14159*(I+NN-1)/(N-1))
         T(I)=20.*SINT*SINT*CEXP(CMPLX(0.,(DS*6.28*I)/N))
         D(I)=0.
15    CONTINUE
C.....READ IN INPUT WAVEFORM, IF DESIRED
      GO TO 17
19    READ (5,14) (T(I),I=1,N)
14    FORMAT (20F4.1)
17    ZAPMX=0.
      NJ=N
      VELTM=1.0-1.0/(VELRT*VELRT)
      A4=-1./DPR
      A1=CM*(0.,1.)
      A9=CM*CM*CM*(0.,-1.)
      A6=A1+A4
      A8=A4-A1
      CZD  =CEXP(CMPLX(0.,CM*DPR))
      CZ=-CZD
      DO 20 J=1,NIT
         IF (J.GT.NIT1) GO TO 199
         THT=THI
         CZ=CZ*CZD
C.........COMPUTE THE MATRIX COEFFICIENTS
      DO 10 I=1,N
         THT=THT+DTR
         SINT=SIN(THT)
         COST=SINT*SINT*(1.0-IMAP(I,J)*VELTM)-1.0
         A3=CM*CM*(COST*COST-4.)/DPR
         A2=A9*COST*COST
         A5=A2+A3
         A7=A3-A2
         SINTP=SIN(THT+DTR2)
         SINTN=SIN(THT-DTR2)
         T1=SINT*SINTP/DTRS
         T2=-SINT*(SINTP+SINTN)/DTRS
         T3=SINT*SINTN/DTRS
         AK(I)=A6*T1
         BK(I)=A5+A6*T2
         CK(I)=A6*T3
         DK(I)=A8*T1
         EK(I)=A7+A8*T2
         FK(I)=A8*T3
10    CONTINUE
C.....SCALE THE RESULTS AND SAVE FOR THE NEXT PASS
         DO 30 I=1,N
                ZI=T(I)*CZ
                IF (ZI.GT.ZAPMX) ZAPMX=ZI
                 ZAP(I,J)=ZI+ZAP(I,J)
30       CONTINUE
C.........COMPUTE NEW VECTOR
         DO 25 I=2,N1
                D(I)=FK(I)   *T(I-1)+EK(I)*T(I)+DK(I)*T(I+1)
C.........SOLVE THE TRIDIAGONAL SYSTEM
25       CONTINUE
         CALL CMVRTK(N)
20    CONTINUE
```

```
199    WRITE (6,2)
       NTM=J
       GO TO 73
191    CONTINUE
       WRITE (6,89)
89     FORMAT (1X,'SUCCESSFUL END OF PROGRAM')
C.....PLOT OUT THE RESULTS
       CALL PLOTIT(1)
       STOP
       END
       SUBROUTINE CMVRTR(IDUM)
C.....THIS IS A SUBROUTINE TO SOLVE A COMPLEX, VARIABLE COEFFICIENT TRI-
C.....DIAGONAL MATRIX.
       COMMON    T(70),D(70),F(70),E(70),AK(70),BK(70),CK(70),DK(70)
       COMMON EK(70),FK(70),ZAP(70,250),CM,AN,ZAPMX,DP,N,IPLT,NTM,NO
       COMPLEX T,D,F,E,AK       ;BK,CK,DK,EK,FK,CMPLX,CEXP,CZ(70),D1,DEN
       N1=N-1
       E(1)=(1.,0.)
       F(1)=(0.,0.)
       DO 10 I=2,N1
          DEN=BK(I)+CK(I)*E(I-1)
          E(I)=-AK(I)/DEN
          F(I)=(D(I) -CK(I)*F(I-1))/DEN
10     CONTINUE
       T(N)=F(N1)/(1.0-E(N1))
       DO 20 J=1,N1
          I=N-J
          T(I)=E(I)*T(I+1)+F(I)
20     CONTINUE
       RETURN
       END
       SUBROUTINE PLOTIT(IDUM)
C.....THIS IS A SUBROUTINE TO PLOT RESULTS ON THE VERSATEC PLOTTER.
C.....PLOTTING OF EVERY OTHER TRACE IS DONE WITH A BIAS OF 8 BITS,
C.....SO THAT BOTH POSITIVE AND NEGATIVE AMPLITUDES CAN BE OBSERVED.
C.....SCALING IS DONE AUTOMATICALLY ON THE LARGEST DATA SAMPLE
       COMMON    T(70),D(70),F(70),E(70),AK(70),BK(70),CK(70),DK(70)
       COMMON EK(70),FK(70),ZAP(70,250),CM,AN,ZAPMX,DP,N,IPLT,NTM,NO
       COMPLEX T,D,F,E,AK,BK,CK,DK,EK,FK
       LOGICAL*1 LINE(70),MASK(2,16),DARK(70),LIGHT(70),IV
       DIMENSION RI(70),RV1(70),RV2(70)
       DATA MASK/Z00,Z01,Z00,Z03,Z00,Z07,Z00,Z0F,Z00,Z1F,Z00,Z3F,Z00,Z7F,
      *Z00,ZFF,Z01,ZFF,Z03,ZFF,Z07,ZFF,Z0F,ZFF,Z1F,ZFF,Z3F,ZFF,Z7F,ZFF,
      *ZFF,ZFF/
       DATA DARK,LIGHT/70*ZFF,70*Z00/
       N=NO
       DO 10 I=1,100
          CALL WRITER (LIGHT,70)
10     CONTINUE
       DO 20 I=1,10
          CALL WRITER(DARK,70)
20     CONTINUE
       DO 21 I=1,70
          LINE(I)=MASK(1,1)
           RV1(I)=0.
21     CONTINUE
       SCAL2=ZAPMX/10.
       DO 300 K=1,NTM
       DO 30 I=1,N,2
           RV2(I)=ZAP(I,K)/SCAL2
```

```
         RI(I)=(RV2(I)-RV1(I))/8.
30    CONTINUE
      DO 35 J=1,8
         DO 38 I=1,N,2
                II=RV1(I)+J*RI(I)+0.5
                II=II+8
                IF (II.GT.16) II=16
                IF (II.LT.1) II=1
                LINE(I)=MASK(1,II)
                LINE(I+1)=MASK(2,II)
38       CONTINUE
         CALL WRITER(LINE,70)
35    CONTINUE
      DO 39 I=1,N,2
         RV1(I)=RV2(I)
39    CONTINUE
300   CONTINUE
      WRITE (6,89)
89    FORMAT (1X,'SUCCESSFUL END OF PLOT')
47    DO 50 I=1,300
         CALL WRITER(LIGHT,70)
50    CONTINUE
      RETURN
      END
/*
//GO.SYSIN DD *
   69  250        80.       360.        0.      0.333       2.           1
   69  250        80.       360.        0.      0.333       8.           1
   69  250        80.       360.        0.      0.333       4.           1
   69  250        80.       360.        0.      0.333       6.           1
       250        80.       360.        0.      0.333       6.           1
/*
      SUBROUTINE PLOTIT(IDUM)
C.....THIS IS A SUBROUTINE TO PLOT RESULTS ON THE VERSATEC PLOTTER.
C.....ALL TRACES ARE PLOTTED WITH ONLY THE POSITIVE AMPLITUDES.

      DATA DARK,LIGHT/70*ZFF,70*Z00/
      DATA MASK/Z00,Z01,Z03,Z07,Z0F,Z1F,Z3F,Z7F,ZFF/
      COMMON    T(70),D(70),F(70),E(70),AK(70),BK(70),CK(70),DK(70)
      COMMON RV1(70),RV2(70),EK(70),FK(70),CM,N,IV(70)
      COMPLEX T,D,F,E,AK,BK,CK,DK,EK,FK
      DIMENSION RI(70)
      LOGICAL*1 LINE(70),MASK(9),DARK(70),LIGHT(70)
      IF (IDUM.EQ.1) GO TO 23
      IF (IDUM.EQ.2) GO TO 47
      DO 10 I=1,100
         CALL WRITER (LIGHT,70)
10    CONTINUE
      DO 20 I=1,10
         CALL WRITER(DARK,70)
20    CONTINUE
      RETURN
23    CONTINUE
      RETURN
      DO 21 I=1,70
         LINE(I)=MASK(1)
21    CONTINUE
227   FORMAT (1X,F2.0)
      DO 30 I=1,N
         RI(I)=(RV2(I)-RV1(I))/8.
```

```
30     CONTINUE
223    FORMAT (1X,F2.0)
       DO 35 J=1,8
          DO 38 I=1,N
                II=RV1(I)+J*RI(I)+.5
                IF (II.LT.0) II=0
                II=II+1
                IF (II.GT.9) II=9
                LINE(I)=MASK(II)
38        CONTINUE
          CALL WRITER(LINE,70)
229    FORMAT (1X,F2.0)
35     CONTINUE
       DO 39 I=1,N
          RV1(I)=RV2(I)
39     CONTINUE
300    CONTINUE
47     DO 50 I=1,300
          CALL WRITER(LIGHT,70)
50     CONTINUE
       RETURN
       END
       SUBROUTINE PLOTIT(IDUM)
C.....THIS IS A PROGRAM TO PLOT RESULTS ON THE LINE PRINTER.
C.....TWO OPTIONS EXIST: (1) IF IPLT=0, A DOUBLE-PRINT PLOT IS MADE
C.....WHICH TRIES (ALTHOUGH POORLY) TO MAKE A VARIABLE DENSITY PLOT
C.....(2) IF IPLT=1, A SINGLE-PRINT PLOT IS MADE USING THE CHARACTERS
C.....'1' THROUGH '9', WITH NEGATIVE AMPLITUDES BEING PLOTTED AS BLANK.
       COMMON    T(70),D(70),F(70),E(70),AK(70),BK(70),CK(70),DK(70)
       COMMON EK(70),FK(70),ZAP(70,250),CM,AN,ZAPMX,DP,N,IPLT,NTM,NO
       LOGICAL*1 CH(16),DH(16),EH(11),CV(70),IV(70)
       COMPLEX T,D,F,E,AK      ,BK,CK,DK,EK,FK,CMPLX,CEXP,CZ(70),D1
       DATA CH/10*' ','M',':','H','H','X','0'/
       DATA DH/' ','.','-','+','*','/','7','A','O','$','S','O','I','8',
      *         'O','#'/
       DATA EH/' ','1','2','3','4','5','6','7','8','9','*'/
10     AQ=90.
       AR=AQ-AN
       AS=AQ+AN
       N=NO
       WRITE (6,13) AR,AQ,AS
13     FORMAT (9X,F3.0,32X,F3.0,32X,F4.0)
20     IF (IPLT.EQ.1) GO TO 25
               SCAL2=ZAPMX/20.
       DO 201 J=1,NTM
            DO 202 I=1,NO
               ZI=ZAP(I,J)/SCAL2
               IF (ZI.LT.0.3) GO TO 31
               II=ZI+1.
               IF (II.GT.16) II=16
               IV(I)=CH(II)
          CV(I)=DH(II)
               GO TO 202
31             IV(I)=CH(1)
          CV(I)=CH(1)
202         CONTINUE
          RPHI=(J-1)*DP
          WRITE (6,8) RPHI,(IV(I),I=1,N)
8         FORMAT (' ',F6.2,3X,'(',69A1,')')
          WRITE (6,18)    (CV(I),I=1,N)
```

IV. Results and Interpretations

At first, results were non-existent. Scaling problems and various program bugs produced some really terrible first efforts. Fortunately though, after numerous program manipulations, reasonably good results were obtained. The following descriptions are about eight examples included in this report which are felt to be worthy of comment. All of the Versatec-produced plots have a true exaggation constant -- that is to say 1° in the horizontal direction takes as much space as 1° in the vertical direction. The θ direction runs across the page, so that the poles would lie off to the sides, with the Equator centered in the middle of the page. The ϕ direction steps down the page, the top being denoted by a solid horizontal bar. All plots cover from $\phi = 0^{\circ}$ to $\phi = 360^{\circ}$ and from $\theta = 10^{\circ}$ to $\theta = 170^{\circ}$, unless otherwise specified.** To get full use of the information presented in each plot, it has been found that a "Chinese filter" method adds a new dimension to the plot. Either get at least 4 feet from the plot or look at the plot from the side in the plane of the paper with eyes squinted. After a while, one will see hills and valleys forming, giving the impression of a 3-D representation of a standing wave in a ripple tank.

Example #1 :

This is a plot of the first case tried on the program. A plane wave was put in at the top (perpendicular to the Equator) and propagated around the globe, with a constant wave velocity assumed. Frequency was specified such that there were 6 grid points per wavelength.. This case was tried first, for formulation of a point source was hard to come up with. The reasoning for using a plane source is as follows : Suppose that one had an explosion at the North Pole. Energy would then radiate in all directions and eventually focus at the South Pole, since there would be no other place for the energy to go. If one looked at a small section (say 90°) of this wavefront between the poles, he would see what appears to be a plane wave at the

**Note : All plots are Mercator projection -- all traces represent blocks of equal angle content.

Equator. The energy pattern would appear to spread out until it reached the Equator, and then start to curve inwards on the other side of the Equator, thus giving a zero-curvature wave on the Equator. So, it was decided to start a plane wave at this midpoint and see if it would come to a focus. This is indeed what happened. The energy focused 90° later and approximated another point source, which expanded into another plane wave in 90°, and then to another focus in 180°. The wave at 180° from the start is not exactly a plane wave, due to the small angle approximations used in the derivations and also due to the reflecting of energy off of the boundaries, as described in the Programming Notes section. This was a fairly good result, for it behaved just as one would expect. The plot even exhibits a phase shift when going through a focus, as one would expect. In real life, this example represents an earthquake with a focus on the Equator.

<u>Example #2</u> :

This plot was another plane source input, but it was tilted to try and model an earthquake happening somewhere in the Nothern Hemisphere, and having the Rayleigh waves focus in the Southern Hemisphere. Velocity was again assumed to be constant and frequency was such that six grid points per wavelength were calculated. This plot also behaves as one would want it to. The energy starts moving out and comes to a focus on the right side of the Equator. This focus then generates energy which propagates to focus on the left side of the Equator.

<u>Example #3</u> :

This plot was another plane source input, but it was offset from the center by about 20°. Constant velocity and 6 points per wavelength were again assumed. In this case, an earthquake occurring on the Equator which propagates a Rayleigh wave at an angle to the Equator was attempted to be modeled. One would expect energy to propagate up into the hemisphere it was aimed at, but to be bent back down to the Equator by the time it is ready to focus, due to the sphere's curvature. As the energy gets farther away from the Equator, the curvature of the sphere makes the wave travel faster, due to decreasing the actual distance involved in traversing a one-degree block. As the wave travels faster out farther, it is brought back quicker and eventually ends up at the Equator, right on schedule. Various boundary reflections and angle approximations produced the inconsistencies in the lower half of the plot.

Example #4 :

In this example, a point source was attempted to be modeled. A constant velocity and 6 points per wavelength were again assumed. For this model, a sinc function designed to fit inside 11 data points was constructed. The waves expanded and converged to a focus 180^o from the start as one would expect. It was found that if one used a narrower width for the point source model that the energy would expand out to the boundaries much quicker, thus getting into many complicated boundary reflection problems.

Example #5 :

In this example, two monochromatic waves were superimposed in attempts to highlight the phase shift occurring through the focus. Constant velocity was assumed and three waves were passed around the globe.-- one with 6.3 points, 8.43 points, and one with 5 points per wavelength. In this plot θ goes form $\theta = 10^o$ to $\theta = 170^o$. The interference pattern between the waves is quite evident, as expressed in the occurrence of the large amplitude peak followed by a weak doublet. The phase shift through the focus is somewhat better displayed in this picture than it is in Example #1. Notice the fact that the first big wavefront consists of three distinct large amplitude packets. As the wave proceeds through the focus, the bigger wavefront now consists of only two large amplitude packets.-- evidence of a 90^o phase shift.

Example #6:

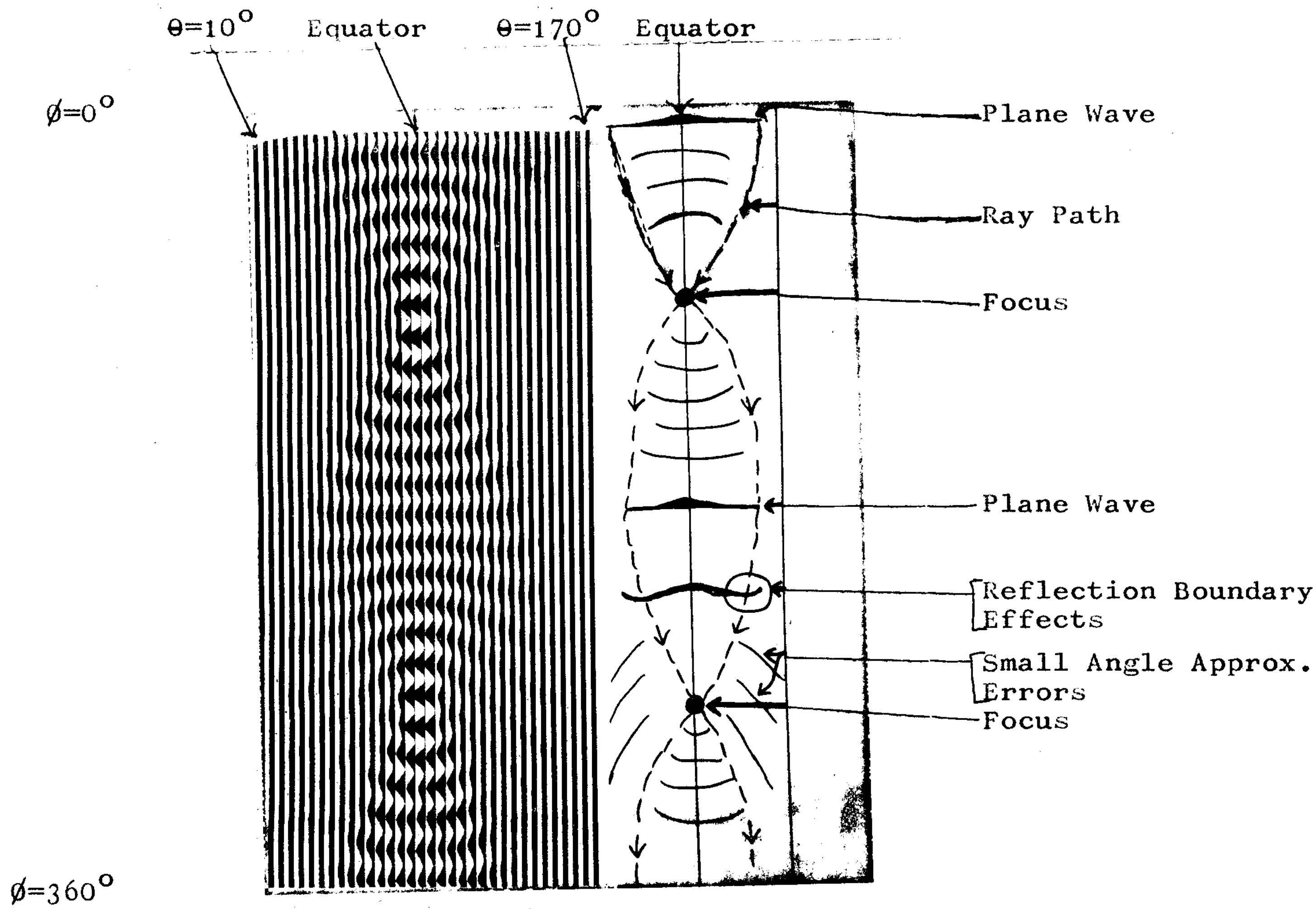
This picture represents the only example in which velocity changes were incorporated. The right side of the grid space (i.e. the Nothern Hemisphere) has been set up with a higher velocity than the left side. The velocity differential represents a 4:3 ratio and the superposition of three waves were used for this calculations. Alth ough this represents one of the simplest model containing velocity information, there are many complicated situations developing. As the plane wave starts propagating, the right side starts moving out quicker, as evidenced by more curvature in the first wavefront noted. As both sides approach the theoretical focus, the waves exhibit a phase shift across the Equator, due to the difference in velocities. Then the real complications start to happen. What appears to be a reflected wave comes off of the high-low interface at a small angle, propagates out to assume a plane wave shape, and then focusses back down to the Equator, all of the time staying in the high-velocity hemisphere. The other hemisphere exhibits much more complicated structure. Once past the focus, two different wavefronts can be identified, mainly due to some apparent phase shift between them. The outer wavefront is most probably the bigger part of the high-velocity hemisphere's wave, which was transmitted through the interface at the time of focussing, with an angle difference due to a Snell's Law phenomena. This wave then expands into a plane wave form and proceeds to converge on a very complicated focus. The inner wave is perhaps a reflected wave generated by the low-velocity wavefront impinging on the interface at focussing at such an angle that it could not enter the other side. A low angle of reflection keeps the energy close to the Equator, but it still expands out to plane wave form, and then into a poorly-defined focus. The improving of the resolution by adding more compute steps and thus more wavelengths would help delineate the structure of this model much better, but computation costs forbid such an effort presently. General events are occurring as the theory would predict (such as the energy staying out of the high-velocity zone as much as possible), but more information is needed to thoroughly investigate this problem.

Example #7 :

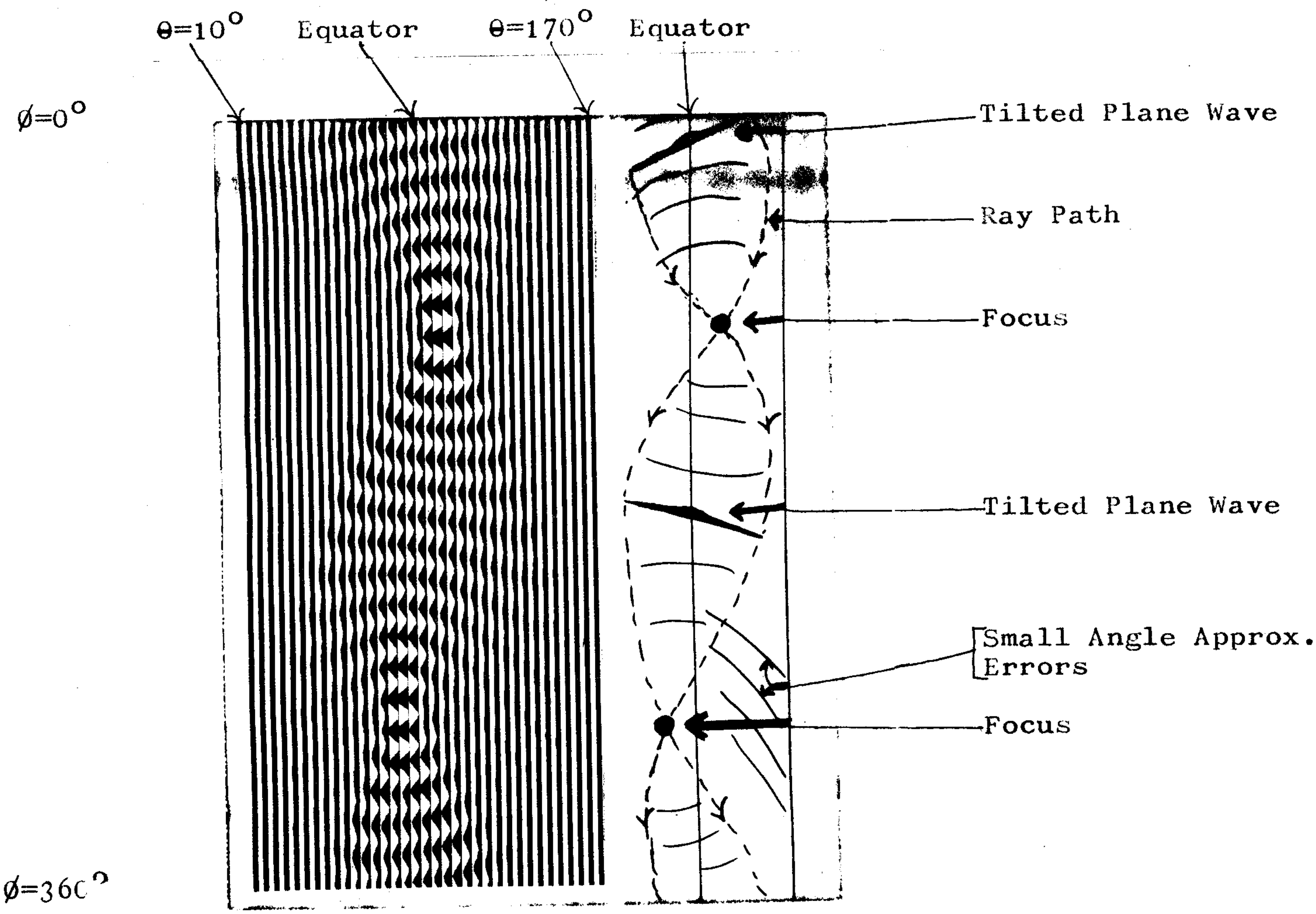
This is just an example of the other type of Versatec plot that was used. This case represents a sim ple plane wave propagating as in Example #1 Vertical exageration exists of 1.67 to 1 (shorter than it should be).

Example #8 :

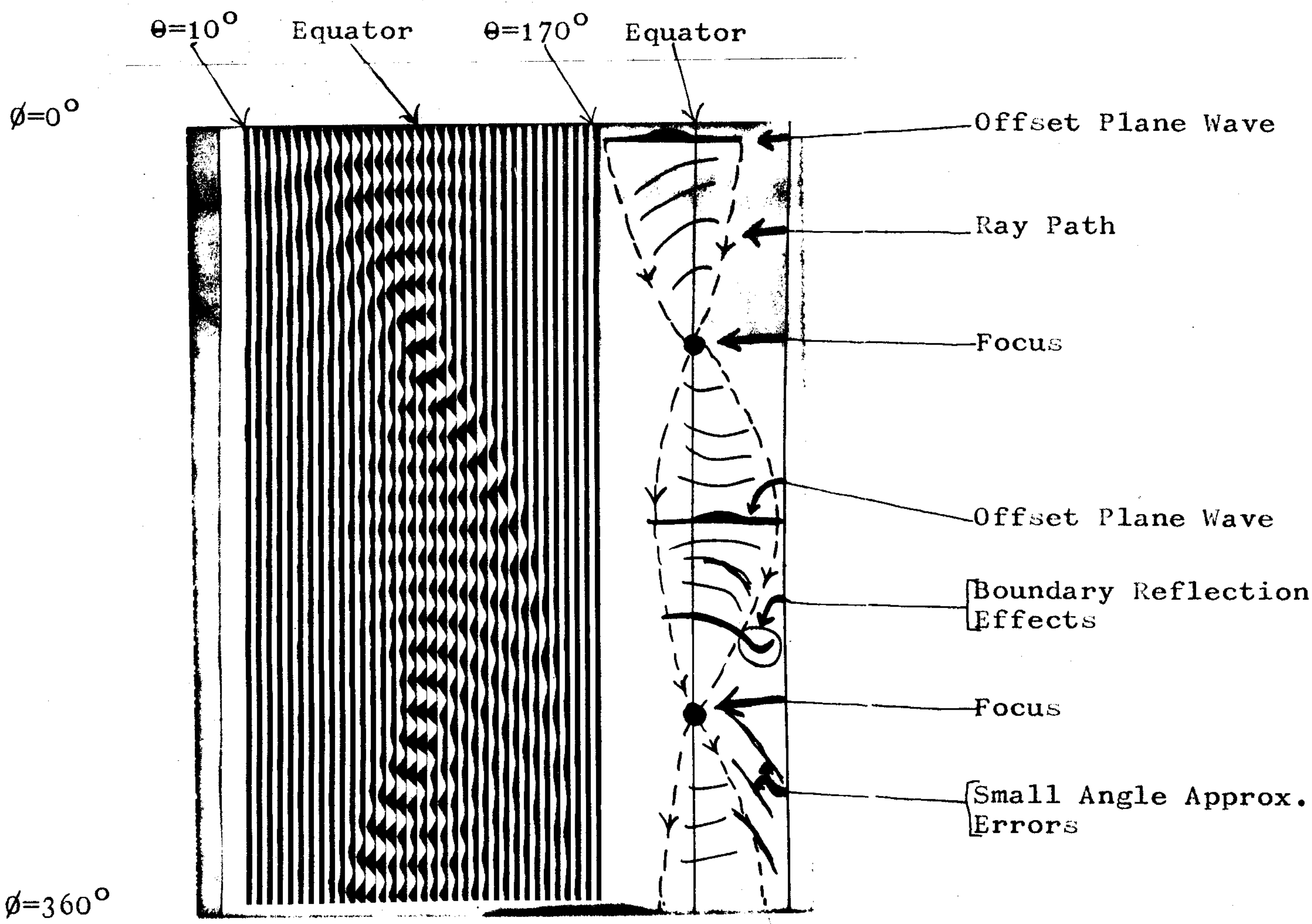
This is just a compilation of various printer plots of similar cases as the above, for purposes of illustrating this type of output.


θ=10°
Equator
θ=170°
Equator
ϕ=0°
Plane Wave
Ray Path
Focus
Plane Wave
Reflection Boundary Effects
Small Angle Approx. Errors
Focus
ϕ=360°

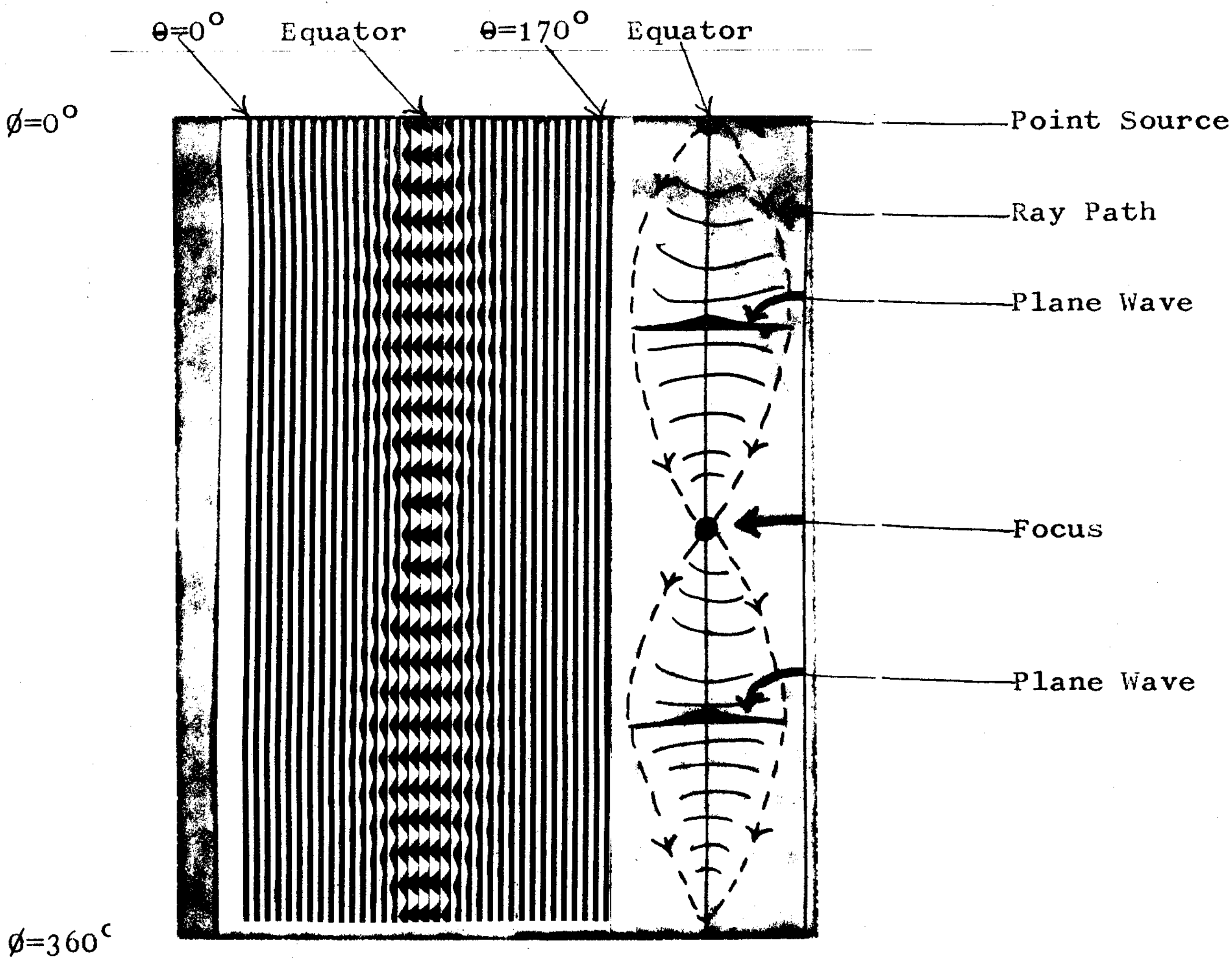
Example #1


θ=10°
Equator
θ=170°
Equator
ϕ=0°
ϕ=360°
Tilted Plane Wave
Ray Path
Focus
Tilted Plane Wave
Small Angle Approx.
Errors
Focus

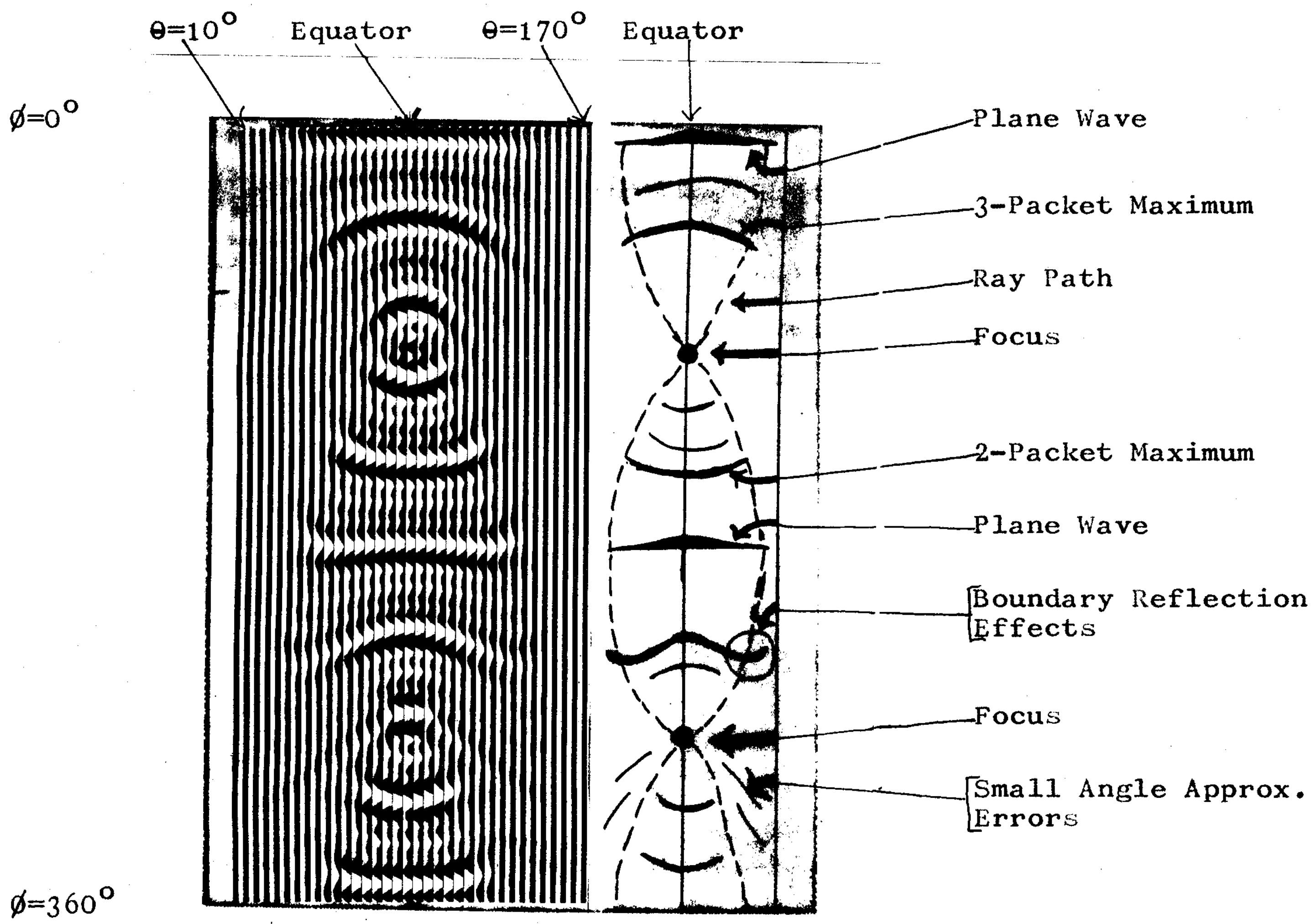
Example #2


θ=10°
Equator
θ=170°
Equator
ϕ=0°
ϕ=360°
Offset Plane Wave
Ray Path
Focus
Offset Plane Wave
Boundary Reflection Effects
Focus
Small Angle Approx. Errors

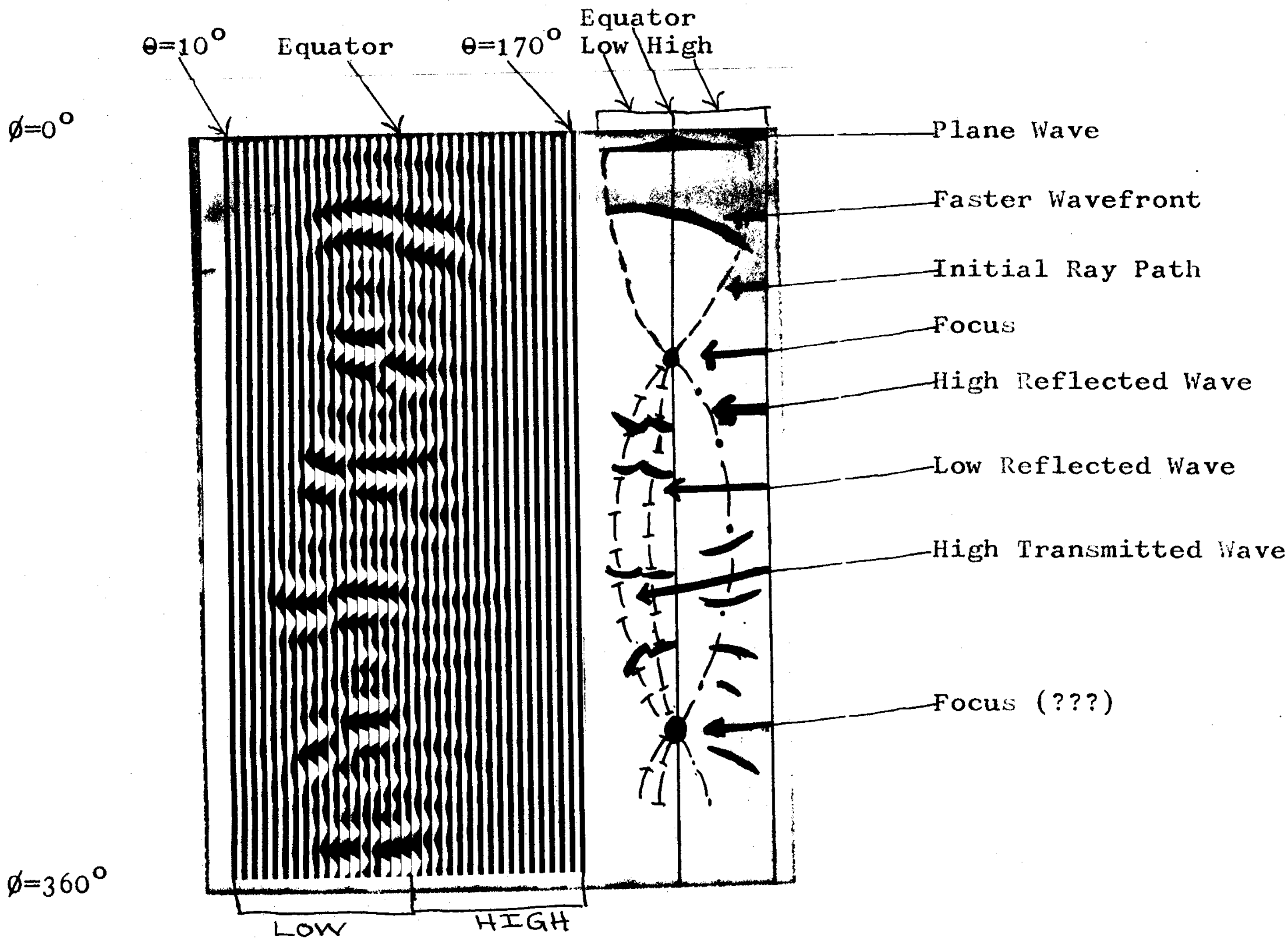
Example #3


θ=0°
Equator
θ=170°
Equator
ϕ=0°
Point Source
Ray Path
Plane Wave
Focus
Plane Wave
ϕ=360°

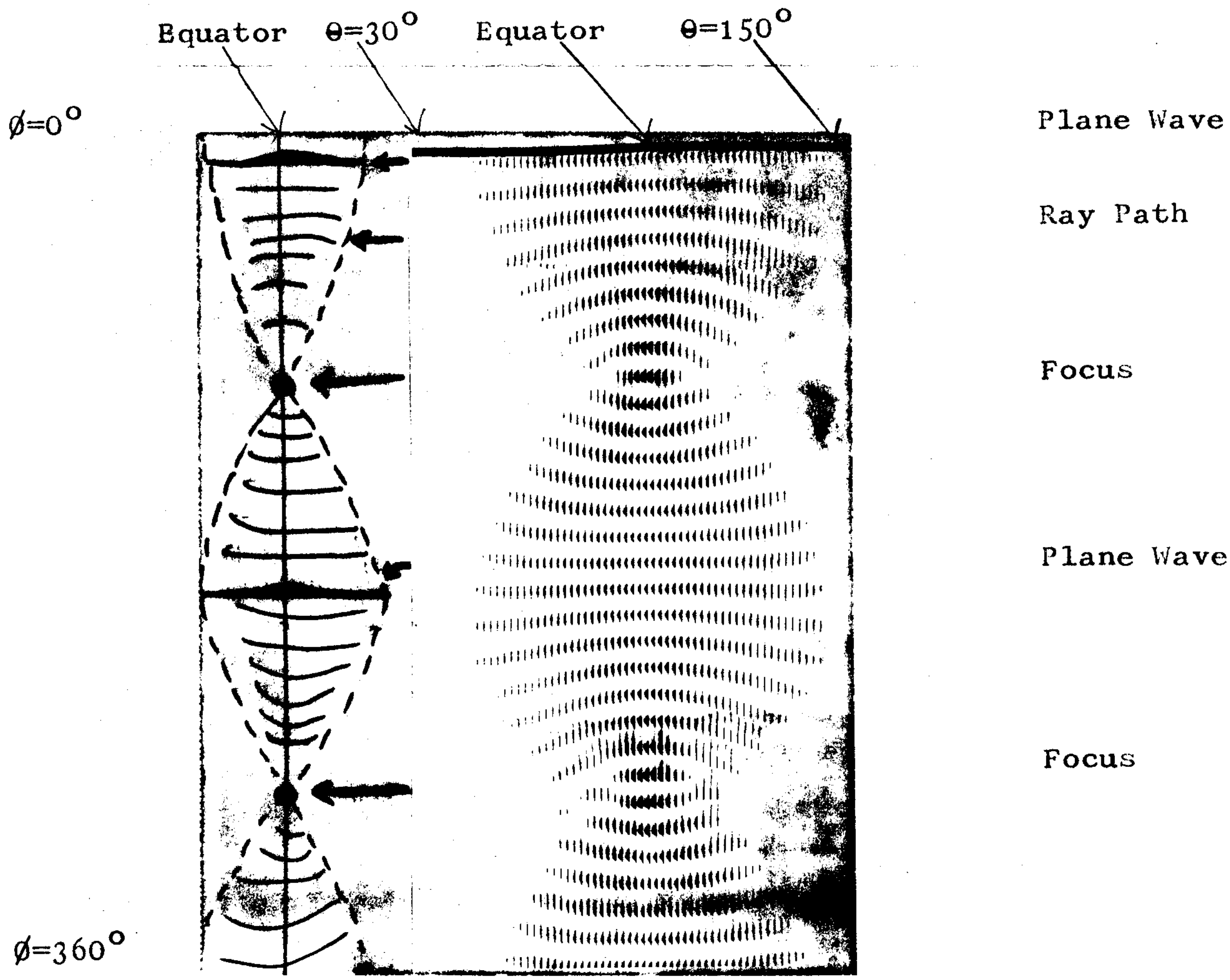
Example #4


θ=10°
Equator
θ=170°
Equator
ϕ=0°
ϕ=360°
Plane Wave
3-Packet Maximum
Ray Path
Focus
2-Packet Maximum
Plane Wave
Boundary Reflection Effects
Focus
Small Angle Approx. Errors

Example #5


θ=10°
Equator
θ=170°
Equator
Low
High
ø=0°
ø=360°
LOW
HIGH
Plane Wave
Faster Wavefront
Initial Ray Path
Focus
High Reflected Wave
Low Reflected Wave
High Transmitted Wave
Focus (???)

Example #6


Equator
θ=30°
Equator
θ=150°
ϕ=0°
ϕ=360°
Plane Wave
Ray Path
Focus
Plane Wave
Focus

Example #7

V. Conclusion

Results for the calculations thus far are quite satisfying, but much more time can be easily spent investigating the subtleties of this type of formulation. Improving the resolution of the results through the use of more calculation steps and through the use of a better plotting system would be the next step in developing a variable-velocity model that can be understood more fully. Once this method is making sense on simple models, investigation of global structures would present a fine opportunity for some very interesting work. The simulation of real Rayleigh waves and free-Earth oscillations is very much in the future, but it is not outside the realms of possibilities.

Today the sphere, tomorrow, ----

THE WORLD !!!

VI. Footnotes and Bibliography

1. Claerbout, Jon, "Coarse grid calculations of waves in inhomogeneous media with application to delineation of complicated seismic structures", Geophysics, vol. 35, pp.407--418

   Claerbout, Jon, "Toward a unified theory of reflector mapping", Geophysics, vol. 36, no. 3, 1971, pp.467-481

2. Claerbout, Jon, Waveform Analysis Class Notes, Geophysics Department, Stanford University, Stanford, California, 1970, pp. 1-298

3. Mitchell, A. R., Computational Methods in Partial Differential Equations, John Wiley & Sons, Camden, pp. 1-253

Appendix B

Description of Plot Tapes

Many problems can arise in attempting to read a tape on one computer which was written on another. In order to avoid as many of these as possible in sending plot tapes to our sponsors, we should agree to some sort of standard tape and data format. The format should be simple for us to generate and likewise, simple and conventional enough for the machine at the other end.

The following is a list of specifications for a tape format which should be widely compatible.

9 Track, 800 BPI

16 Bit, Two's Complement Integers

No Labels

No Headers

Standard Inter-record Gaps (IBM)

Standard End of File Marks (IBM)

Data Scaled Assuming a Plotter Clip Level at ± 1500

Trace 1	I R G	Trace 2	I R G	...	Trace N	I R G	E O F

Since we have neither headers nor labels, the plotting parameters should be sent, physically attached to the tape, with a concise description of what the plot is supposed to look like. No doubt this will help the person at the other end decide whether to do it over or mail it on.

Some possible plotter parameters are listed below which a person, familiar with seismic plotting, would be able to interpret.

TRSP	=	Trace Spacing (Traces/Inch)
TSC	=	Timing Scale (Inches/Second)
SAMPRT	=	Sampling Rate (milliseconds)
NFRAME	=	No. of frames to plot. Often this parameter refers to the no. of seismic profiles to be plotted.
NOTR	=	No. of traces per frame (or profile)
NSAMP	=	No. of 16 bit samples per trace

In addition, most plotters have a variety of display modes such as wiggly trace, wiggly trace-variable area, variable area, variable density, etc.

In writing a tape with FORTRAN we use the type INTEGER * 2 for the output array in order to output 16 bit two's complement. In connection with this we also make use of the A2 format (the null format). The END FILE n statement is used to mark the end of your data set.

The following example is meant to illustrate writing a one frame plot to a tape. It also illustrates the correct job control language of IBM 360-370 series computers.

```
        //S1 EXEC FORTHCG
        //FORT.SYSIN DD *
NOTE 1          INTEGER*2 IHW(70)
                COMPLEX CYL,CEXP
                CYL(X,T)=CEXP((0.,-1.)*W*SQRT(X*X+T*T))/SQRT(X*X+T*T)
                W=2.*3.14159/8.
NOTE 2          NOTR=70
                NSAMP=70
                DO 200 I=1,NOTR
                X=I-NOTR/2
                DO 100 J=1,NSAMP
                T=J
NOTE 3  100     IHW(J)=1500.*CYL(X,T)
NOTE 4  200     WRITE(8,800) ( IHW(J),J=1,NSAMP )
NOTE 5  800     FORMAT(4(255A2))
                END FILE 8
                REWIND 8
                STOP
                END
        /*
NOTE 6  //GO.FT08F001 DD DISP=(NEW,PASS),UNIT=TAPE9,VOL=SER=U1576,
        //  LABEL=(1,BLP,,OUT),DCB=(RECFM=U,BLKSIZE=2040)
        //*    THE FOLLOWING STEP READS THE PLOT TAPE AND
        //*    GENERATES A VERSATEK PLOT
        //S2   EXEC PGM=VVPLOT
        //STEPLIB DD DSN=S091.VVPLOT,UNIT=2314,VOL=SER=SYS23,DISP=OLD
        //FT06F001 DD SYSOUT=A
        //FT66F001 DD SYSOUT=A
        //DISPLAY DD UNIT=004
        //FT08F001 DD DISP=(OLD,KEEP),UNIT=TAPE9,VOL=SER=U1576,
        //  LABEL=(1,BLP,,IN),DCB=(RECFM=U,BLKSIZE=2040)
        //FT05F001 DD *
         &VVPLT VAP=50,AMPL=10.,NOTR=70,NSAMP=70,NFRAME=1,SAMPRT=4,
                VXCALC=1.,VXOUT=1.,&END
        /*
```

NOTES :

(1) SAMPLES WRITTEN ON PLOT TAPE ARE INTEGER*2 ,16-BIT, TWO'S COMPLEMENT INTEGERS.

(2) IN THIS EXAMPLE EACH TRACE IS 70 SAMPLES IN LENGTH (NSAMP) AND EACH FRAME IS 70 TRACES (NOTR)

(3) DATA WRITTEN TO PLOT TAPE ASSUME A PLOTTER CLIP LEVEL OF 1500 AND ARE SCALED ACCORDINGLY.

(4) FOR FORTRAN WRITES TO TAPE (FTN UNIT 8 =FT08F001) USE FORMAT CONTROL SUCH THAT ONE PHYSICAL RECORD = ONE LOGICAL RECORD (TRACE). NO EXTRANEOUS CONTROL WORDS ARE WRITTEN IN THIS MANNER AND AVOIDS PROBLEMS IN GOING FROM ONE SYSTEM TO ANOTHER.

(5) THE 'A2' FORMAT TRANSMITS ONE INTERNAL HALFWORD WITH NO CONVERSION DURING I/O , I.E. BITS IN MACHINE = BITS ON TAPE. DUPLICATION OF FORMAT FIELD MUST BE AT LEAST AS LARGE AS THE LENGTH OF THE TRACE WRITTEN. FORTRAN LIMITS DUP. FIELD TO A MAXIMUM OF 255, THEREFORE WE MUST DUP. THE DUP. TO WRITE LONG TRACES. IN THIS EXAMPLE WE WOULD BE ABLE TO WRITE 4*255=1020 HALFWORDS (SAMPLES).

(6) 'DDNAME' (FT08F001) MUST CORRESPOND TO FORTRAN LOGICAL UNIT 8. IN LABEL FIELD THE FIRST PARAMETER IS FILE ON TAPE TO BE WRITTEN. IN DCB FIELD THE BLOCK SIZE (BLKSIZE) MUST BE EQUAL TO OR EXECEED THE NUMBER OF BYTES PER RECORD WRITTEN.