NRL Report 7815

Acoustic Transmission Loss by Single-Profile Ray Tracing (Program RTRACE)

EVAN B. WRIGHT

Large Aperture Systems Branch
Acoustics Division

December 23, 1974

NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 7815	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* ACOUSTIC TRANSMISSION LOSS BY SINGLE-PROFILE RAY TRACING (PROGRAM RTRACE)		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL Problem 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR*(s)* Evan B. Wright		8. CONTRACT OR GRANT NUMBER*(s)* NRL Problem S01-36
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, D.C. 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Project RF 52-552-402
11. CONTROLLING OFFICE NAME AND ADDRESS Department of the Navy Office of Naval Research Arlington, Va., 22217		12. REPORT DATE 13. NUMBER OF PAGES 71
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Acoustics
Computer program
Propagation
Ray tracing
Sound-speed
Transmission loss
Underwater sound

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

A program called RTRACE for the calculation of underwater acoustic transmission loss by ray tracing and summation has been written for NRL's CDC-3800 computer. RTRACE uses a single-profile, flat-bottom model of the medium with an inverse quadratic function of depth to interpolate in the sound-speed profile. Random- or coherent-phase intensities may be calculated and results may be printed or plotted, along with detailed ray histories and a ray trajectory plot.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

This page intentionally blank.

CONTENTS

INTRODUCTION 1

THEORY 2

INPUT DATA STRUCTURE 6

TEST CASES AND DISCUSSION 16

ACKNOWLEDGMENTS 22

REFERENCES 22

APPENDIX A—Formulation of Ray Theory Solutions 23

APPENDIX B—Descriptions of Subroutines 25

ACOUSTIC TRANSMISSION LOSS BY SINGLE-PROFILE RAY TRACING (PROGRAM RTRACE)

INTRODUCTION

Long-range acoustic ray tracing calculations tend to be complicated and time consuming to make because of the necessity of introducing new depth vs sound-speed profiles into the computation as range from the source increases, and because the water depth must be taken as variable, in a piecewise-linear (or some other) approximation of the bathymetric profile. In such calculations, one wants to trace as few rays as will suffice to delineate the acoustic field (although the number of receiver ranges may be large or unlimited), and it is helpful to have advance knowledge of which rays will reach the receivers.

The situation is different if the ranges involved are short enough so that one sound-speed profile can be used throughout. If, also, the bottom can be considered flat (or if bottom interaction is negligible), the program can reduce the computation time by taking advantage of the periodic form of the ray trajectories, and a large number of rays can be traced in a small amount of computer time. RTRACE, the program described in this report, is for such flat-bottom, single-profile calculations. The program is useful not only for the exact solution in such situations but also for quick first approximations in cases where a more general computer program will be used later. Examples of more general programs employing range-dependent sound-speed profiles and variable bottom depths are the NRL programs TRIMAIN [1] and GRASS [2].

The version of RTRACE described here uses unmodified ray theory throughout. All intensity calculations are made from closed-form solutions for individual ray trajectories, and it will be seen later that results are sensitive to small variations in the sound-speed profile. The following summary gives the major characteristics and limitations of RTRACE:

1. The source language for RTRACE is Fortran-63. The program is compiled and run at NRL on a CDC-3800, using two memory banks. (The actual program storage used is 58,100 words.) Peripheral equipment required for the various output options includes a digital (CALCOMP 565) plotter (with associated software) and an extra line printer, or a secondary storage unit to save selected outputs for deferred printout.

2. The medium is represented by a single depth vs sound-speed profile; i.e., there are vertical but not horizontal gradients in the sound-speed field. The user may enter a profile with as many as 100 depth points. If the profile is entered in depth-temperature-salinity (DTS) form, the program will compute the sound speeds by Wilson's equation [3].

3. The program fits the sound speed in each layer of the profile to a curvilinear interpolating function

Note: Manuscript submitted August 21, 1974.

$$c = \frac{c_a^2}{1 - k_a(z - z_a)^2} .$$

This fit is continuous through the first derivative dc/dz. The fitting algorithm includes a provision for reducing the number of layers in the profile by discarding superfluous interfaces, and the user may control this process by specifying a fitting tolerance.

4. The bottom is represented as a flat, specularly reflecting boundary with a reflection coefficient varying with the grazing angle but not with range. The user may enter arrays of real-valued bottom loss and phase shift tabulated at 1° intervals of grazing angle.

5. The surface is represented as a flat, specularly reflecting boundary with a constant user-specified reflection coefficient and a constant phase shift of 180°.

6. The volume attenuation in the medium is assumed to follow a modified Marsh-Schulkin formula:

$$\alpha = 3.025(10^{-4})f^2 + \frac{44f^2}{4100 + f^2}$$

where α is in decibels per kilometer, and the user specifies the frequency f in kilohertz.

7. The user will select the rays to be traced by specifying an upper limit, a lower limit, and an increment of grazing angle at the source. If the user requests it, the program will automatically supply a more dense ray distribution for source angles near the horizontal. Any number of rays may be traced.

8. The user will specify M receiver depths and N equally spaced receiver ranges, where $M \leqslant 50$ and the product $M \times N$ must not exceed 2000. The transmission loss at the receivers may be computed by either coherent- or random-phase ray summation. The calculated transmission losses may be listed, printer plotted, or CALCOMP plotted. The user may also request a listing of the "eigenrays" from source to each receiver, a listing of the layer-by-layer history of each ray traced, and a CALCOMP trajectory plot of all rays traced.

9. The user may specify source and/or receiver directivity characteristics by supplying vertical beam patterns tabulated at 1° intervals.

More detailed descriptions of the input/output options, with a sample data deck setup, will be found in the third section of this report.

THEORY

Any two-dimensional ray-tracing algorithm begins with the scalar point function $c(r, z)$; i.e., with sound speed as a function of range and depth. The use of ray theory to

solve an acoustic propagation problem implies that the acoustic wavelength is small compared with the distance over which the sound speed may change significantly; i.e., ray theory is the high-frequency limit of a more general wave theory. The condition $\lambda \delta c/c \ll 1$, where $\delta c/c$ is the fractional change in sound speed over a distance of one wavelength λ in the sound-speed field, is sufficient to make ray solutions valid except near turning points [4,5]. As noted in the introduction, the use of a single-profile model neglects range dependence entirely.

Ray tracing without range dependence is then the process of solving the differential equation of the ray trajectory

$$\frac{dr}{dz} = \frac{c}{\sqrt{c_m^2 - c(z)^2}} \tag{1}$$

where c_m is the characteristic parameter of the particular ray. For certain functions $c(z)$, this equation can be integrated in closed form. To use data gathered at sea in the form of a sound-speed profile (a set of discrete depth vs sound-speed pairs), an interpolating function is needed to define $c(z)$ for all depths z. The form of the function used and the method of fitting the data points define the ray-tracing algorithm.

The algorithm used in RTRACE is based upon the interpolating formula

$$c(z) = \left[\frac{c_{ai}^2}{1 - k_{ai}(z - z_{ai})^2}\right]^{1/2} \qquad z_i \leqslant z \leqslant z_{i+1} \tag{2}$$

where $\{z_i\}$ is the set of depths of the sound-speed profile. The advantages of this formula have been discussed by previous authors [6,7]. The fitting parameters $\{c_{ai}, k_{ai}, z_{ai}\}$ are chosen so that the resulting piecewise function passes continuously through the profile points and has a continuous derivative dc/dz. Then the ray elements for each profile layer (i.e., the integrals of range, ray intensity, and travel time for each ray passing through a layer of the profile) can be evaluated in closed form. The solutions, in the formulation of Pedersen and Gordon [7], are outlined in Appendix A of this report and need not be discussed further here.

The algorithm used in RTRACE for fitting the interpolating function to the input profile is a modification of the "point-slope-point-slope" method [7]. In Fig. 1, z_i and z_{i+1} are depths at which the sound speeds c_i and c_{i+1} and the gradients y_{i+1} are known. A pair of bridging curves passing through points (z_i, c_i) and (z_{i+1}, c_{i+1}) with the given values of slope at these points is determined such that c and dc/dz are continuous at the depth z_0 where they meet. The sound speed at the intermediate layer boundary, $c_0 = c(z_0)$, is assigned arbitrarily, while z_0 is to be determined. The value c_0 is computed so as to be intermediate between c_i and c_{i+1}. The interpolated profile derived by this algorithm preserves maxima and minima on the input profile without the addition of either maxima or minima.

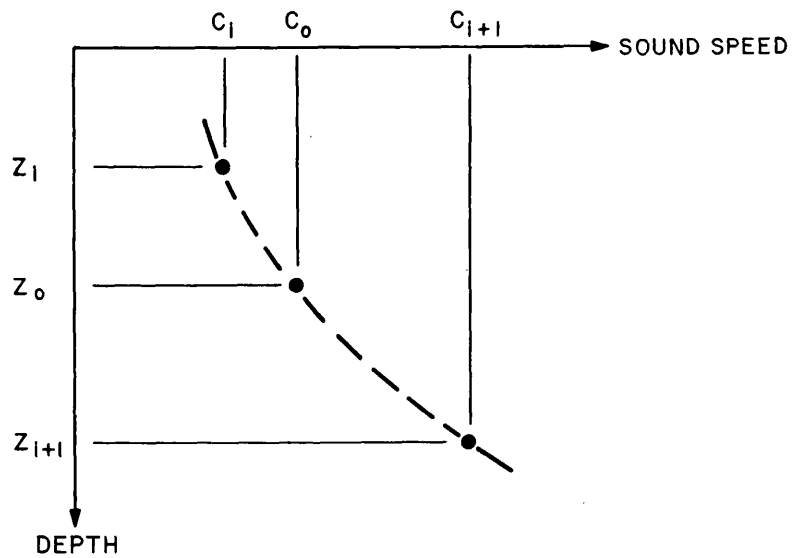


Fig. 1—A pair of interpolating curves between two points on the input profile

This process transforms one layer in the input profile into two layers in the fitted profile, each having its own values of the parameters c_a, k_a, and z_a. To avoid doubling the number of layers in the profile, RTRACE contains provisions for discarding input layers; i.e., if a pair of interpolating curves, bridging several input layers, can be found passing sufficiently close to the intermediate input points, then these input points can be discarded and replaced by the one derived intermediate point. The test for "sufficient closeness" depends on the depth and gradient, as well as on the error in sound speed, and can be controlled by specifying a single tolerance parameter.

The parameters calculated for the interpolating function will depend upon the slope y_i associated with each input point (z_i, c_i). These slope values are not part of the input information; the program must assign an estimate of the gradient to each input point computed so as to yield a smooth-fitted profile. In RTRACE this is accomplished by computing the mean gradients for the layers above and below each input point and assigning their geometric mean as the slope value at the point; i.e., by making

$$g_i = \frac{c_i - c_{i-1}}{z_i - z_{i-1}}$$

and

$$y_i = \sqrt{g_i g_{i+1}} \qquad \text{if } g_i g_{i+1} > 0$$

or

$$y_i = 0 \qquad \text{if } g_i g_{i+1} \leqslant 0.$$

If g_i and g_{i+1} do not have the same sign, the point (z_i, c_i) will be a relative maximum or minimum on the profile and will not be discarded from the fitted profile. If g_i and g_{i+1} do have the same sign, that sign will, of course, be assigned to y_i.

For the point at the surface, and similarly at the bottom,

$$y_i = \frac{g_2^2}{y_2}, \quad \text{if } y_2 \neq 0$$

or

$$y_i = 0 \qquad \text{if } y_2 = 0.$$

After this computation, the y_i array is smoothed to moderate any large jumps in slope that it may contain. The fitting parameters are then computed, and the actual ray tracing can begin.

For this model, the trajectory of a given ray in the range-depth plane is strictly periodic in range. This holds also for the travel time, phase, and range derivative increments, T, P, and B, which are the other quantities accumulated along the ray path for the purpose of calculating intensity at the receivers. When the ray has been traced through one period and back to the source depth, its entire history to any desired range can be obtained by repeated addition of the appropriate range, time, and range derivative increments.

Ray histories need to be known for those depths at which transmission loss is to be computed: i.e., the receiver depths specified in the input data. This information need be held in storage for only two rays at a time: the ray currently being traced and the last one traced. The tables are searched for cases in which these two "neighboring" rays, at a given receiver depth, are interposed by a receiver range (see Fig. 2). Then the values of T, B, and P at the receiver are found by linear interpolation. This gives an intensity contribution to be added in the appropriate cell of the $M \times N$ receiver intensity table.

Thus the nature of the table limits in this program can be seen: storage arrays of dimension $M \times N$ are needed to accumulate the receiver intensities as the rays are traced, one after another; but there is no limit to the number of rays that can be traced because information relating to any ray is automatically discarded as soon as two more rays have been traced. The program is inherently suited to short-range computation (a fact appropriate for the single-profile, flat-bottom model). Because the spacing between rays can be made arbitrarily small, the acoustic field can be computed in precise detail.

Certain computational difficulties can arise in using the closed-form solutions of Appendix A to trace rays from one interface to the next. These can be put (as in Pedersen and Gordon [7] into two clases:

1. A ray is horizontal at a layer interface.

2. $c_m = c_{ai}$ for some layer i.

These are cases where the expression for B_i becomes infinite or indeterminate. In some cases, the expressions can be evaluated in the limit; in others, the theoretical intensity becomes infinite. In RTRACE most of the trouble is avoided by the following methods:

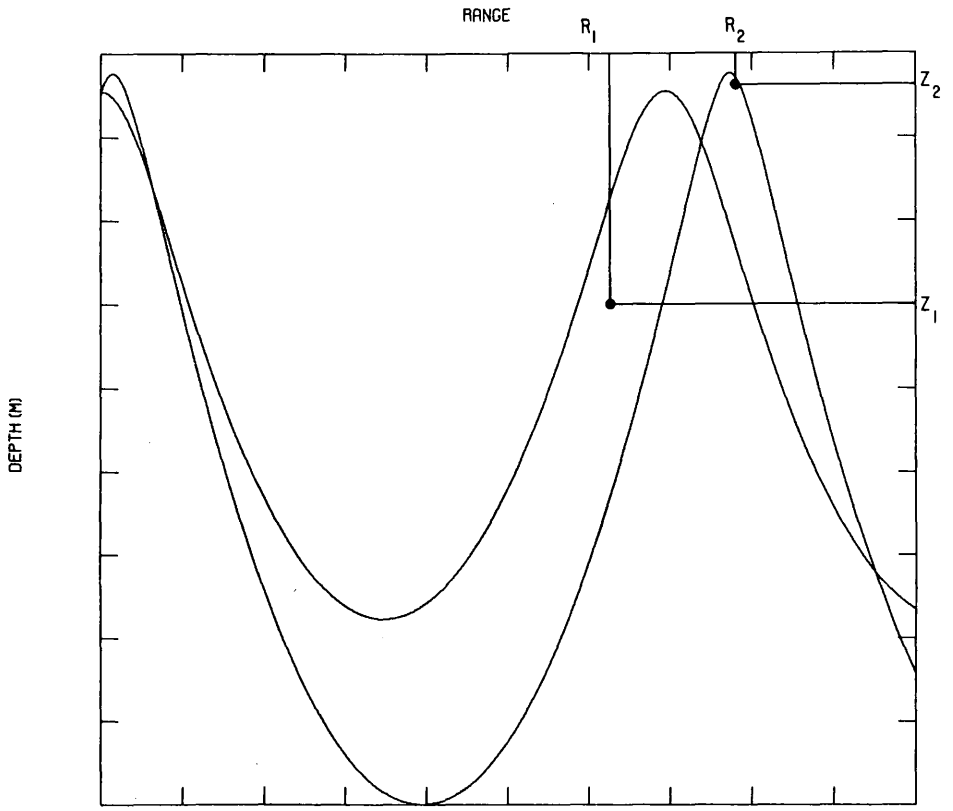


Fig. 2—Interpolation of ray parameters. The two neighboring rays are interposed by the receiver at point (R_1, z_1). The ray parameters of the eigenray between them are found by linear interpolation. The eigenray reaching point (R_2, z_2) is found by interpolation between the two "grazing ray" crossings.

1. A ray that is horizontal at the source is not traced. When the 0° ray is encountered in the source angle distribution, a ray of a slightly different angle is traced instead.

2. A ray that becomes horizontal at a depth where the sound-speed profile has a maximum is not traced. Instead, two rays, one of slightly smaller and one of slightly larger parameter c_m are traced, and trajectory elements are never interpolated between these two rays. Thus, a ray horizontal at a profile maximum is replaced by two neighboring rays, with a "shadow zone" between. The ray trajectory plot of Fig. 3 shows cases where the "bifurcating rays" are traced.

The remaining cases, where limiting expressions for B_i must be used, are programmed as in Eqs. (13) and (17) in Pedersen and Gordon [7].

INPUT DATA STRUCTURE

The input data deck for RTRACE may be set up for multiple runs; i.e., for any number of ray traces to be done consecutively. For each ray trace the card types shown

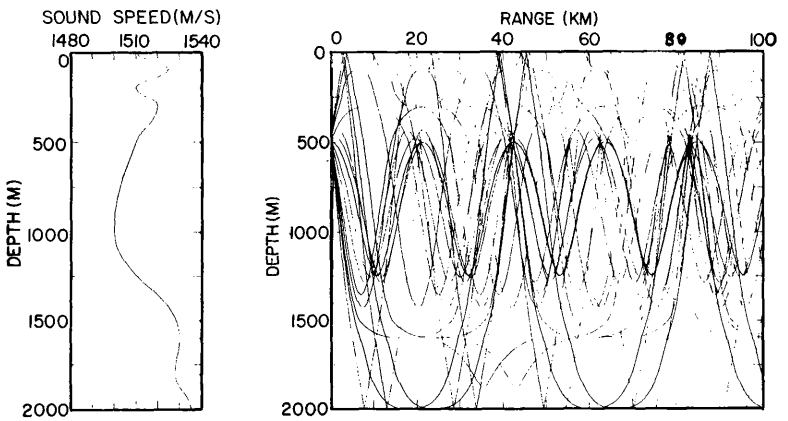


Fig. 3—This ray trajectory plot shows rays that cycle about the sound channel axis, and also the split or bifurcating rays corresponding to maxima on the profile. The contracted layer depths are marked on the profile plot.

in Table 1 are used; the variable names and input formats used in the Fortran source program are shown. (See Appendix B for a description of the subroutines and definitions of the most important symbols.)

The number of receiver depths must not exceed 50. The number of receiver ranges is computed by the Fortran expression $(RL\text{-}RF)/DR + 1$, and the product of the two numbers must not exceed 2000.

A ray-tracing "curved-earth" correction to depths and sound speeds [8] is ordinarily applied, but way be omitted as follows:

1. To omit the correction to source and receiver depths, enter the source depth with a minus sign.

2. To omit the correction to profile depths and sound speeds, enter the first sound speed with a minus sign (or in the case of DTS input, enter the first salinity with a minus sign).

If not omitted, the curved-earth correction replaces each depth z_i by $z_i(1 + z_i/2R)$ and each sound speed c_i by $c_i(1 + z_i/R)$, where R is the radius of the earth, taken as 6317.2213 km.

If a more dense ray distribution for angles near the horizontal is desired, attach a minus sign to DGAMD.

The first depth in the sound-speed profile must always be zero (the surface). If a nonzero number is entered in its place, it will be interpreted as a multiplicative factor for changing the standard fitting tolerances. The standard tolerances used in the program are

$$\epsilon_c = 0.1\ m/s,$$
$$\epsilon_{z_i} = 1.0\ m,$$
$$\epsilon_z = 0.0025.$$

Table 1
Card Types in Input Data Deck

Card Type	Format	Variables	Explanation
1. Title card	(10A8)	Title	Any 80-character title to be attached to printed and plotted outputs
2. First parameter card	(10F8.0)	GAMUD	Upgoing ray limit (degrees) } Ray angles to be traced
		GAMDD	Downgoing ray limit (degrees) } Ray angles to be traced
		DGAMD	Increment (degrees) } Ray angles to be traced
		RF	First range (km) } Receiver ranges
		DR	Increment (km) } Receiver ranges
		RL	Last range (km) } Receiver ranges
		FKHZ	Frequency (kHz)
		IATT	If not zero, use volume attenuation.
		SLDB	Surface loss (dB)
		SORLEV	Source signal level (dB)
3. Second parameter card	(1514, 2F8.0)	ISCP	1 for coherent phase, otherwise random
		IPER	1 for eigenray printout
		IPRAY	1 for ray history output on logical unit 1
		IPL	0 for no plot; 1 for CALCOMP; 2 for print plot; 3 for both plots
		IM	0 for depth vs sound-speed profile 1 for DTS input
		NSRM	Maximum number of surface reflections
		NBRM	Maximum number of bottom reflections

Table 1—Continued

Card Type	Format	Variables	Explanation
		IDR IR2 ILG	Range increment (km) Final range (km) Plot length (in.) } For ray and transmission loss plots
		JDY JY2 JHGT	Depth increment (m) Maximum depth (m) Plot length (in.) } For ray trajectory plots
		ITBP IRBP	Source directivity flag Receiver directivity flag
		DATD DARD	Downtilt of source beam pattern (degrees) Downtilt of receiver beam pattern (degrees)
4. Source/receiver depth card	(10F8.0)	SD,(RCD(I), I = 1, M)	The source and receiver depths in meters. Receiver depths (up to 50) are read until a blank or negative number is encountered (use a blank card if necessary).
5. Sound-speed profile card a. If IM = 0	(10F8.0)	(ZIL(I), VIL(I), I = 1, NIB)	Depths in meters; sound speed in m/sec. Data pairs are read until a blank or negative depth is encountered (use a blank card if necessary).
b. If IM = 1	(F5.0, 2F5.2, 2I, 2F6.2)	Z	Depth
		T	Temperature
		S	Salinity in parts per thousand.
		IR*	Depth unit flag: 0 for meters; 1 for yards; 2 for feet.
		ITR*	Temperature unit flag: 0 for Celsius; 1 for Fahrenheit.

*These variables appear on first 5b card only.

Table 1—Continued

Card Type	Format	Variables	Explanation
		LAT* LONG*	Latitude in degrees. Longitude in degrees. Repeat 5b cards to the bottom of the profile and terminate with a blank card.
6. †Bottom-loss table card (if NBRM > 0)	(20F4.2)	(BLT(J), J = 1, NBOT)	Bottom loss in decibels, tabulated by degrees.
7. †Bottom phase shift table card (If NBRM > 0 and ISCP = 1)	(20F4.2)	(EPBLT(J), J = 1, NBOT)	Bottom phase shift in radians, tabulated by degrees.
8. †Source beam pattern card (If ITBP ≠ 0)	(20F4.2)	(VBPT(J), J = 1, NB)	Source decibels, tabulated by degrees.
9. †Receiver beam pattern card (If IRBP ≠ 0)	(20F4.2)	(VBPR(J), J = 1, NB)	Receiver decibels, tabulated by degrees.

†Optional input.

An input point (z_i, c_i) on the profile will be discarded if the calculated values c and g of sound speed and gradient in the current layer satisfy the relation

$$|c - c_i| \leqslant \epsilon_c + g(\epsilon_z + z_i \epsilon_{z'})$$

where c and g are calculated at the depth z_i.

The bottom loss table is tabulated at 1° intervals, starting with grazing angle 0°. Input is terminated by a blank or zero. The last loss will be extended to all higher angles. The bottom phase shift table (in radians, to be added to ray phase ωt) is read in similarly. The program interpolates linearly in these tables to get the bottom loss and phase shift for a particular ray grazing angle.

Source and receiver beam patterns (in decibels, to be subtracted from source or receiver level) are read in similarly to the bottom loss table. If a pattern has been declared asymmetric (ITBP = 2 or IRBP = 2), two card sets (first upgoing, then downgoing) are used, each beginning with 0° and terminating with a blank. Linear interpolation is used in these tables also.

```
***********   DATA DECK FOR THE EXAMPLE OF SECTION 3   ***********



RTRACE EXAMPLE WITH BOTTOM AND SURFACE LOSS, RAYS SUMMED COHERENTLY
    15.      15.       .5       2.       2.     200.       .1       1.       1.
  1   0    0   2    0  99  99
  91.44   10.288   91.44
     0.   1490.9     10.   1491.0     20.   1489.6     30.   1480.3     50.   1474.9
    75.   1469.6    100.   1467.1    125.   1465.5    150.   1463.8    200.   1460.4
   250.   1458.1    300.   1457.0    400.   1455.8    500.   1456.2    600.   1457.1
   700.   1458.3    800.   1459.5    900.   1461.0   1000.   1462.4   1100.   1463.9
  1200.   1465.4   1300.   1466.9   1400.   1468.5   1500.   1470.1   1750.   1474.4
  1850.   1476.0     -1.
 .1   1.   2.   3.   4.   5.   6.   7.   8.   9.  10.
```

Fig. 4—Sample input data deck

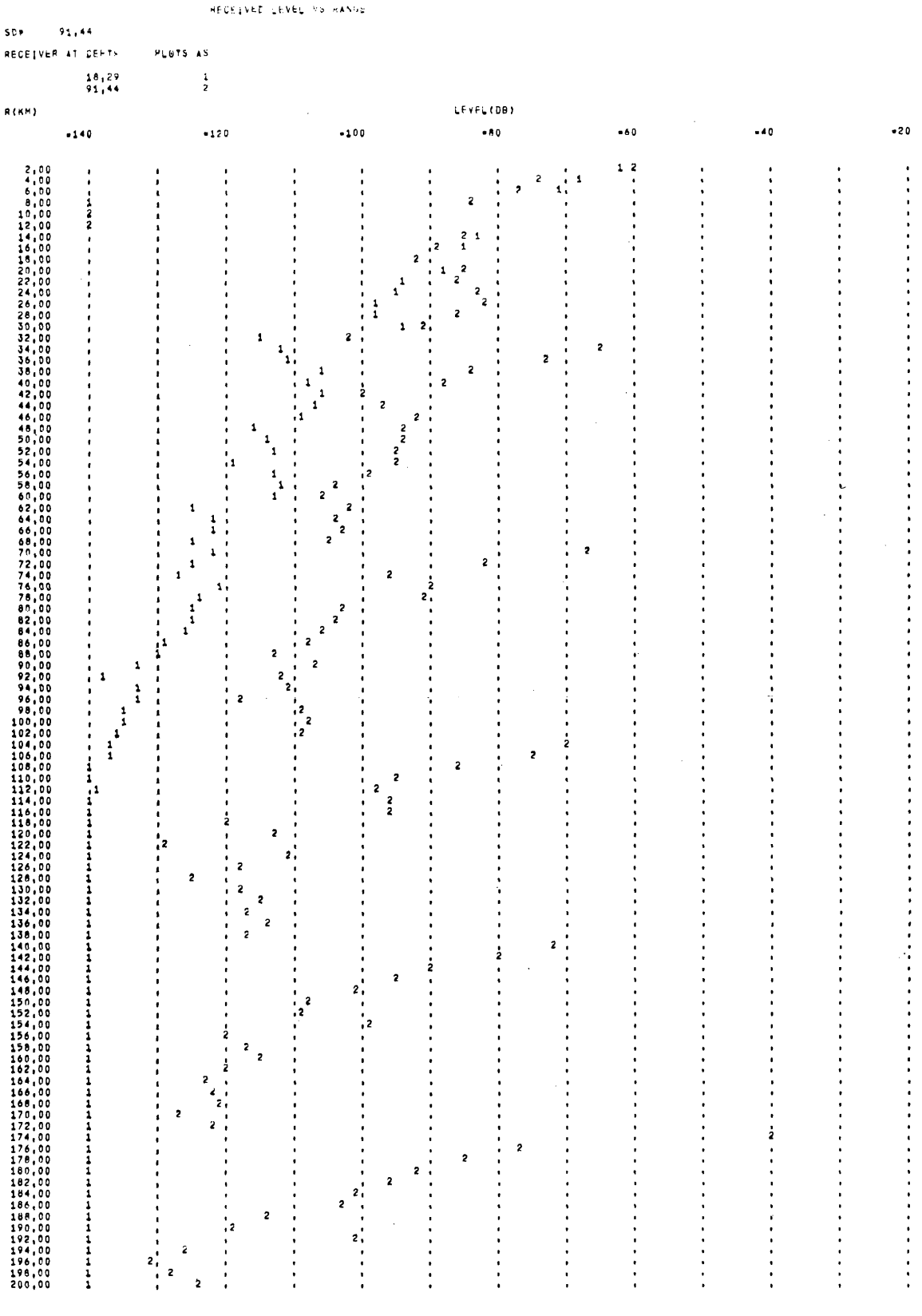
Figure 4 shows the input data deck for an example in which a bottom loss table and surface loss are entered and a coherent ray sum is specified. The following printout gives the computer output. The time required to load the program from a binary object deck and execute it for this example was 29 s.

```
CURVILINEAR GRADIENT RAY THEORY   RTSAGE EXAMPLE WITH BOTTOM AND SURFACE LOSS, RAYS SUMMED COHERENTLY


     ALL INPUT TO THIS PROGRAM IS IN METRIC UNITS,I.E., METERS, AND METERS/SEC, EXCEPT TEMPERATURE WHICH MAY BE INPUT IN EITHER DEGREES
     C OR DEGREES F   THE OUTPUT IS IN METRIC UNITS
PROFILE INPUT DATA

INPUT PROFILE        0,0 1490,90      10,0 1491,00      20,0 1489,60      30,0 1480,30      50,0 1474,90
INPUT PROFILE       70,0 1469,60     100,0 1457,10     125,0 1465,50     150,0 1463,80     200,0 1460,40
INPUT PROFILE      250,0 1458,10     300,0 1457,00     400,0 1455,80     500,0 1456,70     600,0 1457,10
INPUT PROFILE      700,0 1458,30     800,0 1459,50     900,0 1461,00    1000,0 1462,40    1100,0 1463,90
INPUT PROFILE     1200,0 1465,40    1300,0 1466,90    1400,0 1468,50    1500,0 1470,10    1750,0 1474,40
INPUT PROFILE     1850,0 1476,00

SD/RCD        91,44       18,29       91,44

PRINTER PLOT OF PROFILE, AN X MARKS THE VELOCITY,  AN I MARKS AN INPUT POINT EXTRAPOLATED TO THE NEAREST LINE.

DEPTH      1490                                                                                                                  1500
        0,00 ,
       18,50 ,
       37,00 ,
       55,50 ,
       74,00 ,
       92,50 ,
      111,00 ,
      129,50 ,
      148,00 ,
      166,50 ,
      185,00 ,
      203,50 ,
      222,00 ,
      240,50 ,
      259,00 ,
      277,50 ,
      296,00 ,
      314,50 ,
      333,00 ,
      351,50 ,
      370,00 ,
      388,50 ,
      407,00 ,
      425,50 ,
      444,00 ,
      462,50 ,
      481,00 ,
      499,50 ,
      518,00 ,
      536,50 ,
      555,00 ,
      573,50 ,
      592,00 ,
      610,50 ,
      629,00 ,
      647,50 ,
      666,00 ,
      684,50 ,
      703,00 ,
      721,50 ,
      740,00 ,
      758,50 ,
      777,00 ,
      795,50 ,
      814,00 ,
      832,50 ,
      851,00 ,
      869,50 ,
      888,00 ,
      906,50 ,
      925,00 ,
      943,50 ,
      962,00 ,
      980,50 ,
      999,00 ,
     1017,50 ,
     1036,00 ,
     1054,50 ,
     1073,00 ,
     1091,50 ,
     1110,00 ,
     1128,50 ,
     1147,00 ,
     1165,50 ,
     1184,00 ,
     1202,50 ,
     1221,00 ,
     1239,50 ,
     1258,00 ,
     1276,50 ,
     1295,00 ,
     1313,50 ,
     1332,00 ,
     1350,50 ,
     1369,00 ,
     1387,50 ,
     1406,00 ,
     1424,50 ,
     1443,00 ,
     1461,50 ,
     1480,00 ,
     1498,50 ,
     1517,00 ,
     1535,50 ,
     1554,00 ,
     1572,50 ,
     1591,00 ,
     1609,50 ,
     1628,00 ,
     1646,50 ,
     1665,00 ,
     1683,50 ,
     1702,00 ,
     1720,50 ,
     1739,00 ,
     1757,50 ,
     1776,00 ,
     1794,50 ,
     1813,00 ,
     1831,50 ,
     1850,00 ,
```

```
SOURCE        13        91,44      1467,7555

RCV        5        1        1        19,29      [illegible]

RCV       13        2        2        91,44      1467,7555


  (PROFILE)        NO,       DEPTH(M)       VELOCITY(M/SEC)              Z1              CAS            KACA       LRCVN       DC/DZ
INTERFACE NO,       1          0,000          1490,900000   0,0000000*000  2,2357828*006  1,0985894*006       0          0,0000
INTERFACE NO,       2          3,000          1490,949997   *5,0000000*000  2,2330410*006  *1,0985694*006       0          0,0200
INTERFACE NO,       3         10,000          1491,000000   0,0000000*000  2,2330810*006  *2,7601895*006       0          0,0000
INTERFACE NO,       4         17,451          1490,799507   2,7613172*000  2,2217661*006  *4,5340686*006       0         *0,1879
INTERFACE NO,       5         18,288          1490,118438   3,5983156*000  2,2217661*006  *4,5340686*006       1         *0,2448
INTERFACE NO,       6         20,000          1489,600000   1,7306370*000  2,2198388*006  *7,9422812*006       0         *0,3608
INTERFACE NO,       7         25,199          1484,928159   *7,4248305*000  2,1893434*006   7,6729172*006       0         *1,4313
INTERFACE NO,       8         30,000          1480,300000   *1,2909008*001  2,1817542*006   3,4593192*006       0         *0,5011
INTERFACE NO,       9         37,735          1477,592600   5,9987864*001  2,2011280*006  *1,0159089*006       0         *0,1997
INTERFACE NO,      10         50,000          1474,900000   *2,1298050*003  1,6167648*006   1,8711609*007       0         *0,2392
INTERFACE NO,      11         61,166          1472,242845   *3,6256626*001  2,1549370*006   1,4303620*006       0         *0,2367
INTERFACE NO,      12         75,000          1469,600000   *3,6054964*001  2,1520367*006   1,1279834*006       0         *0,1456
INTERFACE NO,      13         91,440          1467,755507   *1,9614964*001  2,1520367*006   1,1279834*006       2         *0,0789
INTERFACE NO,      14         94,309          1467,545705   *1,8851935*003  1,9822137*006   1,0831139*007       0         *0,0673
INTERFACE NO,      15        125,000          1465,500000   5,4408356*002  2,2016121*006  *1,9627306*007       0         *0,0660
INTERFACE NO,      16        178,849          1461,785952   *1,2831677*002  2,1234073*006   4,2368704*007       0         *0,0719
INTERFACE NO,      17        250,000          1458,100000   *6,2890226*001  2,1231424*006   4,0394532*007       0         *0,0318
INTERFACE NO,      18        270,723          1457,549689   *1,2378665*002  2,1206140*006   2,3576205*007       0         *0,0213
INTERFACE NO,      19        300,000          1457,000000   2,6254302*002  2,1290825*006  *1,4145276*007       0         *0,0162
INTERFACE NO,      20        334,682          1456,399629   *6,5317675*001  2,1193536*006   3,0174423*007       0         *0,0184
INTERFACE NO,      21        400,000          1455,800000   0,0000000*000  2,1193536*006   1,4337634*007       0          0,0000
INTERFACE NO,      22        596,281          1457,048391   *2,0063788*003  2,1605004*006  *4,5072495*008       0          0,0126
INTERFACE NO,      23        700,000          1458,300000   1,3135403*003  2,1038983*006   5,4274847*008       0          0,0120
INTERFACE NO,      24       1016,949          1462,581037   1,2951906*004  1,8879136*006   1,9257096*008       0          0,0150
INTERFACE NO,      25       1300,000          1466,900000   2,1463752*003  2,1041001*006   4,7818861*008       0          0,0155
INTERFACE NO,      26       1573,558          1471,428896   *2,0827553*003  2,2205109*006  *5,1544436*008       0          0,0176
INTERFACE NO,      27       1850,000          1476,000000                                                        0          0,0154


   CONTRACTED LAYERS      DEPTH      VELOCITY      LRCVNC
          1                0,00   1490,900000          0
          2               10,00   1491,000000          0
          3               18,29   1490,118438          1
          4               91,44   1467,755507          2
          5              400,00   1455,800000          0
          6             1850,00   1476,000000          0


 CONVERT CAS     2,154937*006   TE ANGLE     0,017109      0,9803
 CONVERT CAS     2,160500*006   TE ANGLE     0,053570      3,0693
 CONVERT CAS     2,178576*006   TE ANGLE    *0,105744     *6,0587
 CONVERT CAS     2,181754*006   TE ANGLE     0,112400      6,4401
 CONVERT CAS     2,189343*006   TE ANGLE     0,126845      7,2677
 CONVERT CAS     2,201128*006   TE ANGLE     0,146370      8,3864
 CONVERT CAS     2,201612*006   TE ANGLE     0,147114      8,4290
 CONVERT CAS     2,219839*006   TE ANGLE     0,172674      9,8935
 CONVERT CAS     2,220511*006   TE ANGLE     0,173540      9,9431
 CONVERT CAS     2,221766*006   TE ANGLE     0,175144     10,0350
 CONVERT CAS     2,222783*006   TE ANGLE     0,176432     10,1088
 CONVERT CAS     2,223081*006   TE ANGLE    *0,176808    *10,1304

                        GAMLD        GAMCD       DGAMD        RF          DR          RL       ITBP      DATD     IRBP      DARD
VBPL AND RANGES      15,00000     15,00000     0,50000    2000,000    2000,000  200000,000      *0     *0,000      *0     *0,000

                       ISCP    IABET     IPER     IPRAY    IUVRS     NSRM     NBRM      IPL
CONTROL INDICES         1        0        0        0        0       99       99        2

VOL ATTEN = 1,103*004 DB/KM AT 0,100 KHZ,  SL(DB) = 1,00  SOURCE LEVEL =  *0,0


                    GA        BRC        BL(DB)      ARG(RC)
BOTTOM LOSS          0      0,9772        0,10       *0,00
BOTTOM LOSS          1      0,7943        1,00       *0,00
BOTTOM LOSS          2      0,6310        2,00        0,00
BOTTOM LOSS          3      0,5012        3,00        0,00
BOTTOM LOSS          4      0,3981        4,00        0,00
BOTTOM LOSS          5      0,3162        5,00        0,00
BOTTOM LOSS          6      0,2512        6,00        0,00
BOTTOM LOSS          7      0,1995        7,00        0,00
BOTTOM LOSS          8      0,1585        8,00        0,00
BOTTOM LOSS          9      0,1259        9,00        0,00
BOTTOM LOSS         10      0,1000       10,00        0,00
BOTTOM LOSS         11      0,1000       10,00        0,00
BOTTOM LOSS         12      0,1000       10,00        0,00
BOTTOM LOSS         13      0,1000       10,00        0,00
BOTTOM LOSS         14      0,1000       10,00        0,00
BOTTOM LOSS         15      0,1000       10,00        0,00
BOTTOM LOSS         16      0,1000       10,00        0,00
BOTTOM LOSS         17      0,1000       10,00        0,00
BOTTOM LOSS         18      0,1000       10,00        0,00
BOTTOM LOSS         19      0,1000       10,00        0,00
BOTTOM LOSS         20      0,1000       10,00        0,00

CM SET LT REL MAX        3   *1,0574*001   1475,999843        0,105743

CM SET GT REL MAX        3   *1,0574*001   1476,000157        0,105745

CM SET LT REL MAX       12   *1,7681*001   1490,999734        0,176807

CM SET GT REL MAX       12   *1,7681*001   1491,000266        0,176809

CM SET LT REL MAX        3   *1,0574*001   1475,999843        0,105743

CM SET GT REL MAX        3   *1,0574*001   1476,000157        0,105745

CM SET LT REL MAX       12   *1,7681*001   1490,999734        0,176807

CM SET GT REL MAX       12   *1,7681*001   1491,000266        0,176809
```

COHERENT SUM OF RAYS

SOUND LEVEL, RCV AT	18,3	91,4
RKM = 2,000	-62,4	-60,3
RKM = 4,000	-68,3	-74,4
RKM = 6,000	-71,5	-77,0
RKM = 8,000	-200,0	-84,4
RKM = 10,000	-200,0	-200,0
RKM = 12,000	-200,0	-200,0
RKM = 14,000	-63,1	-84,7
RKM = 16,000	-85,1	-89,3
RKM = 18,000	-92,2	-91,7
RKM = 20,000	-87,6	-84,8
RKM = 22,000	-93,5	-86,3
RKM = 24,000	-94,7	-83,2
RKM = 26,000	-97,8	-81,5
RKM = 28,000	-97,5	-86,3
RKM = 30,000	-94,2	-91,4
RKM = 32,000	-114,7	-102,2
RKM = 34,000	-111,7	-64,8
RKM = 36,000	-110,9	-72,5
RKM = 38,000	-105,6	-84,2
RKM = 40,000	-108,4	-87,6
RKM = 42,000	-106,1	-100,1
RKM = 44,000	-107,3	-96,8
RKM = 46,000	-109,3	-92,5
RKM = 48,000	-115,6	-94,1
RKM = 50,000	-113,9	-94,2
RKM = 52,000	-113,3	-94,7
RKM = 54,000	-118,5	-95,1
RKM = 56,000	-113,0	-98,7
RKM = 58,000	-112,1	-104,2
RKM = 60,000	-113,1	-105,7
RKM = 62,000	-124,6	-102,5
RKM = 64,000	-121,9	-104,5
RKM = 66,000	-122,1	-103,1
RKM = 68,000	-124,9	-105,3
RKM = 70,000	-122,4	-66,5
RKM = 72,000	-124,6	-82,1
RKM = 74,000	-126,8	-95,6
RKM = 76,000	-120,8	-89,7
RKM = 78,000	-123,9	-91,0
RKM = 80,000	-125,5	-102,9
RKM = 82,000	-125,0	-103,8
RKM = 84,000	-126,5	-106,1
RKM = 86,000	-128,6	-108,5
RKM = 88,000	-130,0	-113,2
RKM = 90,000	-133,2	-107,5
RKM = 92,000	-138,1	-112,0
RKM = 94,000	-132,9	-110,9
RKM = 96,000	-132,9	-117,7
RKM = 98,000	-135,2	-108,9
RKM = 100,000	-135,1	-108,4
RKM = 102,000	-136,0	-109,1
RKM = 104,000	-136,6	-69,7
RKM = 106,000	-137,5	-74,7
RKM = 108,000	-139,6	-85,6
RKM = 110,000	-139,8	-94,9
RKM = 112,000	-139,1	-97,7
RKM = 114,000	-139,7	-96,1
RKM = 116,000	-144,2	-95,9
RKM = 118,000	-146,4	-120,4
RKM = 120,000	-147,4	-113,0
RKM = 122,000	-147,1	-129,1
RKM = 124,000	-148,3	-111,4
RKM = 126,000	-147,6	-117,6
RKM = 128,000	-147,4	-125,2
RKM = 130,000	-148,5	-117,7
RKM = 132,000	-147,9	-114,5
RKM = 134,000	-150,2	-116,8
RKM = 136,000	-151,6	-113,8
RKM = 138,000	-151,6	-117,2
RKM = 140,000	-152,9	-71,8
RKM = 142,000	-155,2	-80,4
RKM = 144,000	-158,4	-89,6
RKM = 146,000	-158,9	-94,7
RKM = 148,000	-160,3	-100,6
RKM = 150,000	-159,3	-107,6
RKM = 152,000	-160,5	-108,6
RKM = 154,000	-162,2	-99,0
RKM = 156,000	-161,8	-119,7
RKM = 158,000	-163,3	-116,9
RKM = 160,000	-164,8	-115,1
RKM = 162,000	-168,2	-119,6
RKM = 164,000	-167,2	-123,5
RKM = 166,000	-169,8	-122,1
RKM = 168,000	-171,3	-120,9
RKM = 170,000	-187,4	-127,1
RKM = 172,000	-186,6	-121,9
RKM = 174,000	-189,6	-40,1
RKM = 176,000	-200,0	-77,2
RKM = 178,000	-200,0	-84,5
RKM = 180,000	-200,0	-91,9
RKM = 182,000	-200,0	-96,0
RKM = 184,000	-200,0	-101,3
RKM = 186,000	-200,0	-103,3
RKM = 188,000	-200,0	-113,7
RKM = 190,000	-200,0	-118,5
RKM = 192,000	-200,0	-101,3
RKM = 194,000	-200,0	-126,5
RKM = 196,000	-200,0	-130,5
RKM = 198,000	-200,0	-128,3
RKM = 200,000	-200,0	-124,4


RECEIVED LEVEL VS RANGE
SD= 91.44
RECEIVER AT DEPTH PLOTS AS
18.29 1
91.44 2
R(KM)
LEVEL(DB)
-140
-120
-100
-80
-60
-40
-20

TEST CASES AND DISCUSSION

The way RTRACE works is shown by the sample run of the previous section. In this section two test cases are described, with a discussion of the results.

The first test case is a theoretical one, described extensively by Pedersen and Gordon [7], with results that can be compared with those of RTRACE. One reason for including the test in this report is to exhibit a theoretical check on the ray trajectory programming of RTRACE. A second reason is that it shows the effect of varying the profile fitting tolerances in RTRACE and how "false caustics" arise on the range vs transmission loss curve.

A single-layer Epstein profile was used for this first test because closed-form solutions can be calculated for the Epstein profile [9]. The control profile from Pedersen and Gordon [7] was entered as data to RTRACE (tabulated at 3-m intervals to the channel axis and a precision of 10 decimal digits, as shows in Table 2, and a set of upgoing rays was traced to their first recrossing of the source depth. A ray diagram is shown in Fig. 5.

Figures 6a and 6b compare the Epstein profile computation with RTRACE results for the standard profile fitting tolerances. In Fig. 6a the solid line represents the range of the first recrossing as a function of ray parameter, according to the Epstein profile computation. The individual symbols show the ranges of rays traced by RTRACE.

The standard fitting tolerances of RTRACE are intended to be well below the experimental error in a real sound-speed profile; i.e., about 0.1 to 0.5 m/s. In this test case, using the standard tolerances caused RTRACE to discard 32 of the 36 layers in the input profile. Such a drastic reduction in the number of profile interfaces speeds up program execution considerably because the ray-tracing computation time is basically proportional to the number of layers traversed. Figures 6a and 6b show the price that is paid in computational error: the inaccuracy of the range and intensity computations arises not only from the interpolation error on the profile but also from the fact that reducing the number of layers tends to increase the discontinuities of the second derivative at the interface points on the profile. These discontinuities are, of course, inherent in the inverse-quadratic profile fit. They lead to slope discontinuities in the range curve and discontinuities (false caustics) in the intensity curve.

Figures 7a and 7b show that the situation can be helped by decreasing the fitting tolerances. For these figures, the RTRACE computation was repeated with the tolerances reduced to 1% of the standard values. This increased the number of fitted layers from 4 to 16. Convergence of the solutions to the Epstein curves can be seen, and the ray intensity discontinuities are greatly reduced.

Although it may appear that tightening the fitting tolerances is desirable for accurate theoretical results (such as calculating a very accurately defined set of eigenrays), a second test case shows that this may not be necessary for obtaining reasonably good predictive transmission loss curves. This second test compares some experimental transmission loss data with results from applying the RTRACE model.

In Figs. 8a and 8b, the individual symbols show transmission loss measured at 100 Hz for an 800-mil track in the North Atlantic. The source depth was 4200 m, and the

Table 2

INPUT EPSTEIN PROFILE FOR PO 67 TEST CASE

0.000	1626.363967	3.000	1625.641045	6.000	1624.889691	9.000	1624.111018
12.000	1623.306473	15.000	1622.477861	18.000	1621.627357	21.000	1620.757511
24.000	1619.871248	27.000	1618.971861	30.000	1618.062994	33.000	1617.148612
36.000	1616.232963	39.000	1615.320535	42.000	1614.415993	45.000	1613.524116
48.000	1612.649723	51.000	1611.797592	54.000	1610.972377	57.000	1610.178529
60.000	1609.420206	63.000	1608.701201	66.000	1608.024870	69.000	1607.394071
72.000	1606.811117	75.000	1606.277738	78.000	1605.795065	81.000	1605.363622
84.000	1604.983340	87.000	1604.653576	90.000	1604.373156	93.000	1604.140423
96.000	1603.953290	99.000	1603.809309	102.000	1603.705737	105.000	1603.639603
108.000	1603.607779	109.561	1603.603723				

FOUR-LAYER FITTED INTERFACES(STANDARD TOLERANCES)

	DEPTH	SOUNDSPEED	GRADIENT
1	0.000	1626.363967	-0.2364
2	33.666	1616.795560	-0.3312
3	69.000	1607.394071	-0.2021
4	80.432	1605.495541	-0.1301
	108.000	1603.609147	-0.0069
5	109.561	1603.603723	0.0000

SIXTEEN-LAYER FITTED INTERFACES(.01*STANDARD TOLERANCES)

	DEPTH	SOUNDSPEED	GRADIENT
1	0.000	1626.363967	-0.2364
2	7.827	1624.417428	-0.2610
3	15.000	1622.477861	-0.2798
4	22.668	1620.265916	-0.2970
5	30.000	1618.062994	-0.3039
6	37.457	1615.788774	-0.3061
7	45.000	1613.524116	-0.2944
8	52.184	1611.468242	-0.2780
9	60.000	1609.420206	-0.2461
10	66.827	1607.846669	-0.2149
11	75.000	1606.277738	-0.1691
12	81.298	1605.324600	-0.1336
13	90.000	1604.373156	-0.0852
14	94.541	1604.039342	-0.0619
15	102.000	1603.705737	-0.0276
16	104.190	1603.654728	-0.0190
	108.000	1603.608031	-0.0055
17	109.561	1603.603723	-0.0000

(THE SOURCE DEPTH IS NOT ONE OF THE FITTED LAYERS BUT IS ON CONTRACTED PROFILE)

0 53

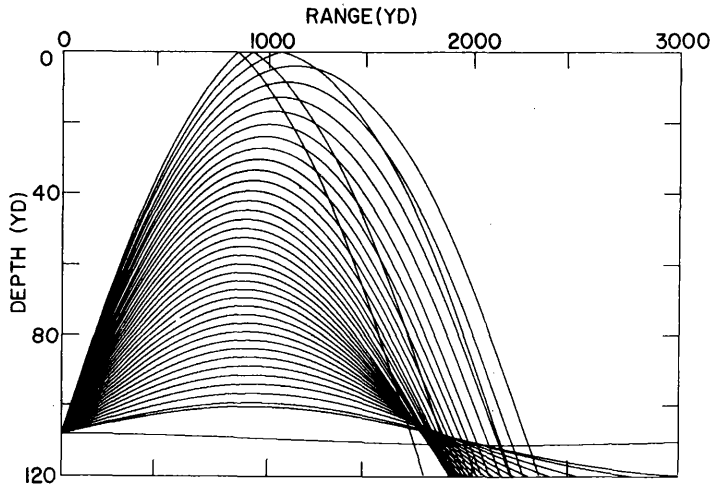


Fig. 5 — Ray trajectories for the Epstein profile test of Pedersen and Gordon [7]

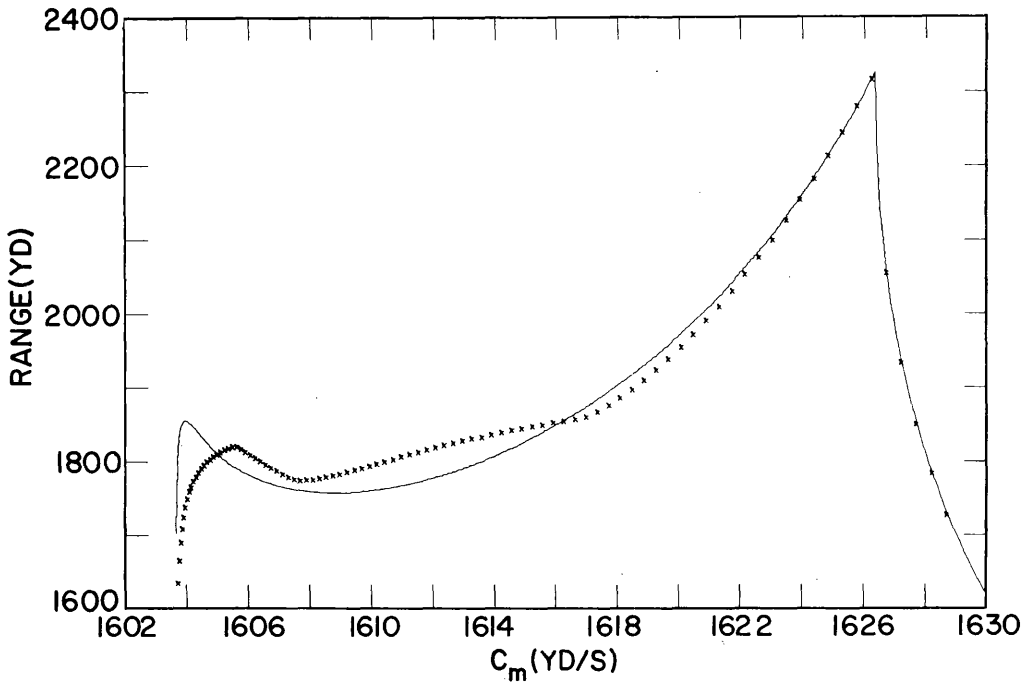
receiver depth was 244 m. The bathymetric profile, shown at the bottom of the graph, is not really flat; for this computation it was approximated as a 4500-m flat bottom.

The sound-speed profile entered into RTRACE for this test was derived from a temperature profile taken on station during the experiment and was extended to the bottom using archival data. The bottom-loss function used (fleet numerical weather central bottom class II) was determined from consulting bottom-loss classification charts for the region of the experiment.

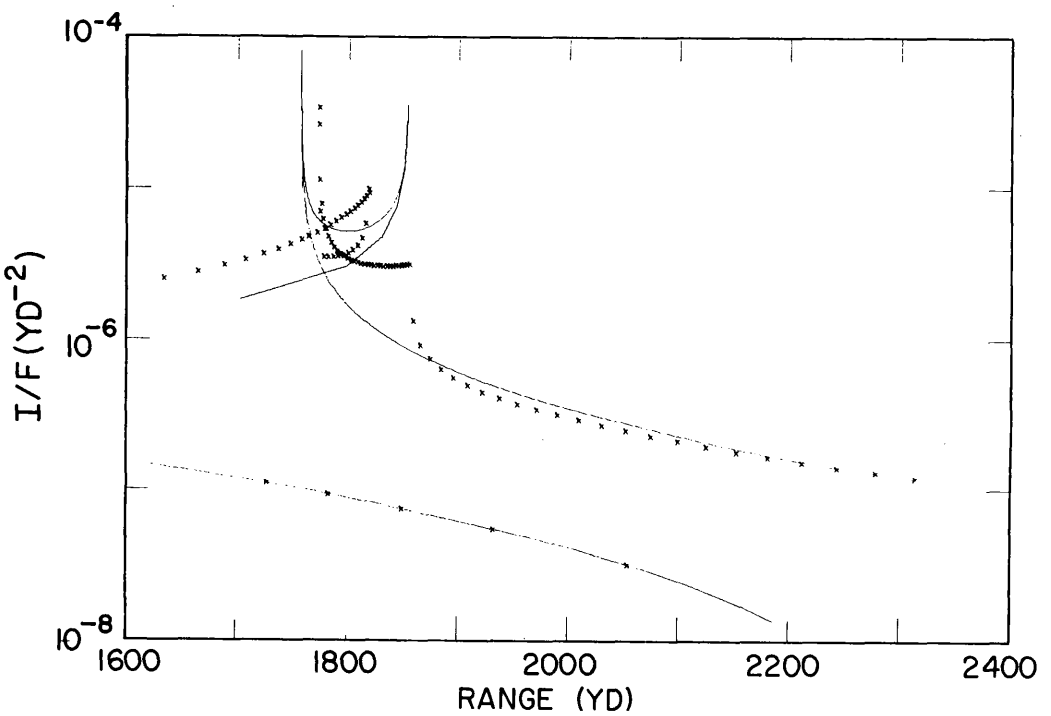
Figure 8a compares the experimental data with a random-sum transmission-loss calculation by RTRACE at 0.5-n. mi. intervals using the standard profile tolerances. Figure 8b shows the calculation repeated with the fitting tolerances reduced by a factor of 100; this increased the number of profile layers from 44 to 56. It can be seen that the pattern of convergence zones is essentially the same for both computed curves, with only the height of the caustic spikes differing. (As noted previously, Program RTRACE does not incorporate an intensity correction at caustics.) Because very high intensities at caustics are to be ignored anyway, either curve will serve as a mean approximation to the data.

This second test was intended to give an idea of how RTRACE is set up for a real predictive run, and fo the sort of results to expect. The program computed a reasonably correct transmission loss curve in this case, even though the "short-range, flat-bottom" criteria were not met, and its use was considerably cheaper and more convenient than would be the case for a more general model.

In conclusion, one can see that in addition to its use for the computation of accurate theoretical intensities and arrival structures (eigenrays) in short-range cases, RTRACE can be useful for getting approximate results for more complicated real-ocean problems.

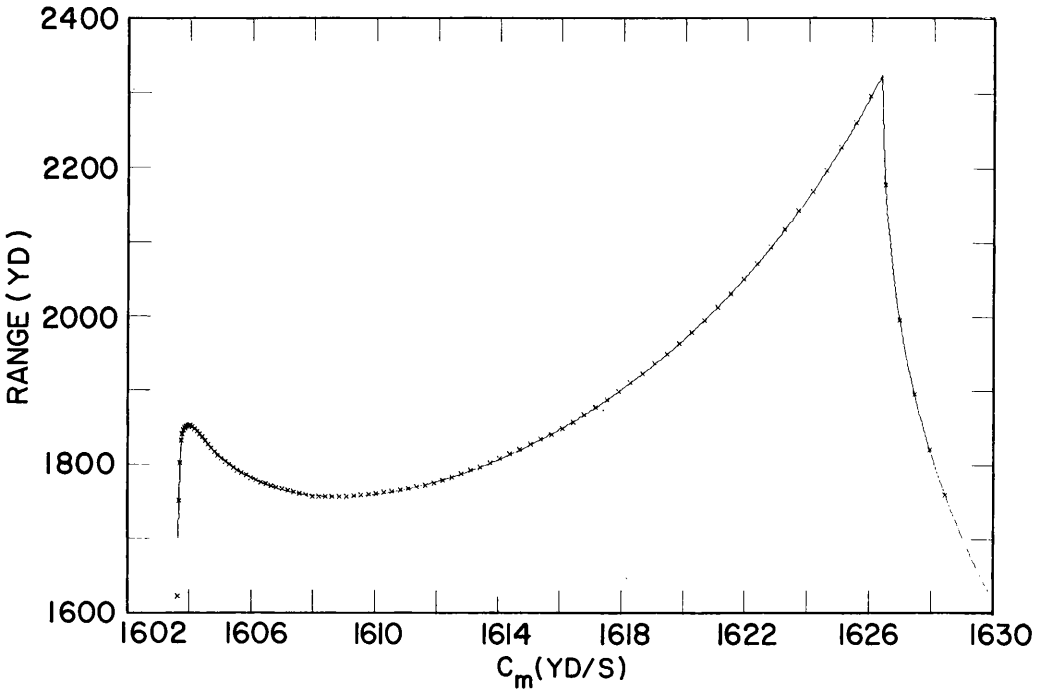


(a) Range vs ray parameter.

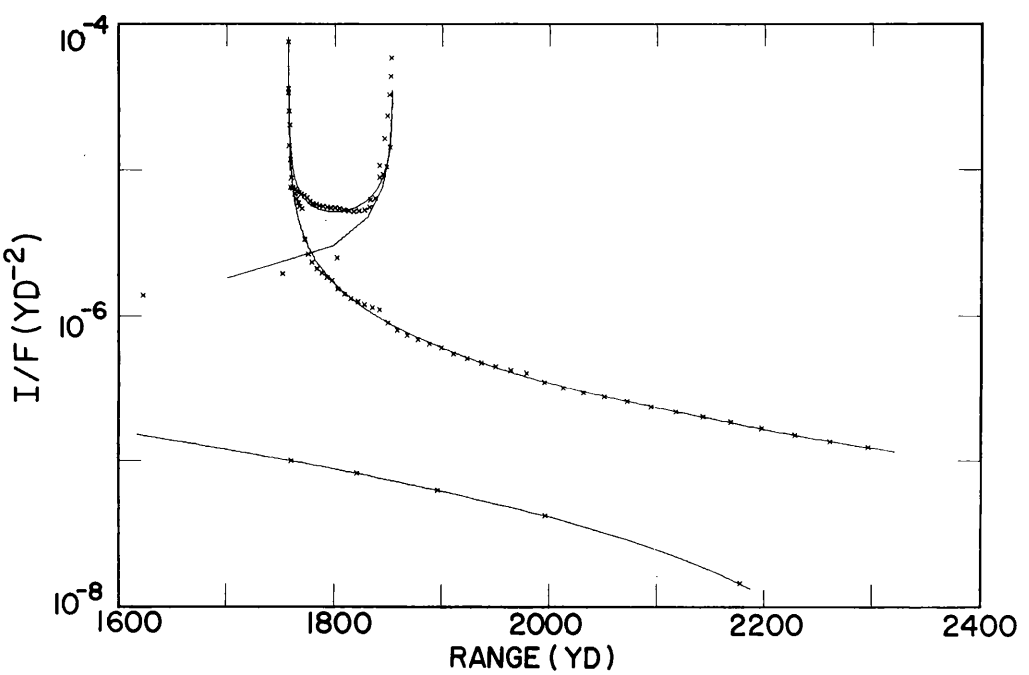


(b) Ray intensity vs range.

Fig. 6 — Comparison of Epstein profile computations with RTRACE results for the standard profile fitting tolerances: – = Epstein; x = RTRACE.

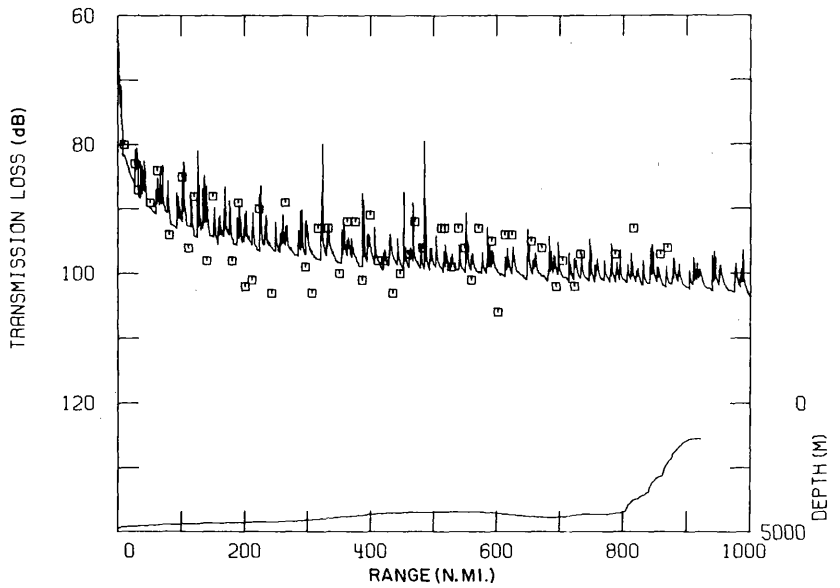


(a) Range vs ray parameter.

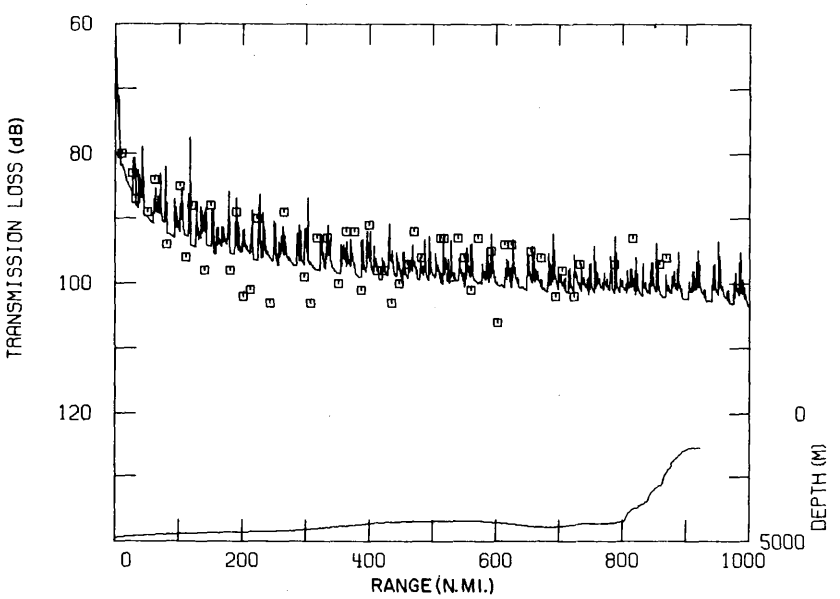


(b) Ray intensity vs range.

Fig. 7 — Comparison of Epstein profile computations with RTRACE results for tolerances reduced to 1% of standard values: - = Epstein; x = RTRACE.



(a)



(b)

Fig. 8 — Transmission loss vs range for the experimental test case: – = experimental data; ⊡ = RTRACE. (a) Stand and profile fitting tolerances (44 layer) were used. (b) Reduced fitting tolerances (56 layer) were used.

ACKNOWLEDGMENTS

Program RTRACE was originally written and tested by H. P. Bucker and E. L. Wright, formerly of NRL. The author of this report has made some corrections and additions, and prepared this documentation.

REFERENCES

1. B. G. Roberts, "Horizontal-Gradient Acoustical Ray Trace Program—TRIMAIN," NRL Report 7827, December 16, 1974.
2. J. J. Cornyn, "GRASS: A Digital Computer Ray Tracing and Transmission Loss Prediction System, 2 vols., NRL Reports 7621 and 7642, 1973.
3. W. D. Wilson, "Equation for the Speed of Sound in Sea Water," *J. Acoust. Soc. Amer.* **32**, 1357 (1960).
4. C. B. Officer, *Introduction to the Theory of Sound Transmission*, McGraw-Hill, 1958, New York, 40.
5. I. Tolstoy and C. S. Clay, *Ocean Acoustics*, McGraw-Hill, New York, 1966, p. 52.
6. K. R. Stewart, "Ray Acoustical Model of the Ocean Using a Depth/Sound-speed Profile With a Continuous First Derivative," *J. Acoust. Soc. Amer.* **38**, 339 (1965).
7. M. A. Pedersen and D. F. Gordon, "Comparison of Curvilinear and Linear Profile Approximation in the Calculation of Underwater Sound Intensities by Ray Theory," *J. Acoust. Soc. Amer.* **41**, 419 (1967).
8. A. G. D. Watson, "The Effect of the Earth's Sphericity on the Propagation of Sound in the Sea," Admirality Research Laboratory Report ARL/N 29/L 1958.
9. M. A. Pedersen and De W. White, "Ray Theory of the General Epstein Profile, *J. Acoust. Soc. Amer.* **44**, 765 (1968).

Appendix A
FORMULATION OF RAY THEORY SOLUTIONS*

1. Characteristic parameter of ray m:

$$c_m = \text{sound speed at which the ray is horizontal (Snell's law is } c = c_m \cos\theta.)$$

2. Parameters of profile layer i:

$$c_i = \text{sound speed at upper interface,}$$

$$z_{1i} = \text{depth of upper interface,}$$

$$z_{2i} = \text{depth of lower interface,}$$

$$c_{ai}^2,\ k_{ai},\ \text{and } z_{ai} = \text{parameters of the interpolating function in layer } i \text{ (see Eq. (2).)}$$

3. Auxiliary quantities defined for convenience in computation:

$$A_i = \pm c_m \left| \frac{K_{ai}}{c_{ai}^2} \right|^{1/2}, \quad \text{(negative sign used if } c_{ai}^2 < 0)$$

$$D_i = \frac{c_{ai}^2}{c_m^2 - c_{ai}^2}$$

$$E_i = \frac{c_m^2 + c_{ai}^2}{c_{ai}^2},$$

$$R_{1i} = A_i(z_{1i} - z_{ai}),$$

$$R_{2i} = A_i(z_{2i} - z_{ai}),$$

$$B_{1i} = \cot\theta_i(z_{1i} - z_{ai}),$$

$$B_{2i} = \cot\theta_{i+1}(z_{2i} - z_{ai}),$$

$$T_{1i} = \tan\theta_i(z_{1i} - z_{ai}),$$

$$T_{2i} = \tan\theta_{i+1}(z_{2i} - z_{ai}), \text{ where } \theta_i = \cos^{-1}\frac{c_i}{c_m}.$$

*M. A. Pedersen and D. F. Gordon, "Comparison of Curvilinear and Linear Profile Approximation in the Calculation of Underwater Sound Intensities by Ray Theory," J. Acoust. Soc. Amer. 41, 419 (1967).

4. Solution for range in layer i:

$$R_i = (A_i)^{-1} \cos^{-1} [D_i(|R_{1i}R_{2i}| + \tan\theta_i \tan\theta_{i+1})] \qquad \text{If } c_{ai}^2 > 0 \text{ and } k_{ai} > 0;$$

$$R_i = \left| (A_i)^{-1} \ln \frac{|R_{1i}| + \tan\theta_i}{|R_{2i}| + \tan\theta_{i+1}} \right| \text{, for all other cases.}$$

If the ray forms an apex in the layer, these become

$$R_i = (A_i)^{-1} \cos^{-1} [(D_i)^{1/2} |R_{2i}|]$$

and

$$R_i = \left| (A_i)^{-1} \ln \frac{|D_i|^{-1/2}}{|R_{2i}| + \tan\theta_{i+1}} \right| \text{, respectively.}$$

If the ray forms a nadir in the layer, these become

$$R_i = (A_i)^{-1} \cos^{-1} [(D_i)^{1/2} |R_{1i}|]$$

and

$$R_i = \left| (A_i)^{-1} \ln \frac{(|R_{1i}| + \tan\theta_i)}{|D_i|^{-1/2}} \right| \text{, respectively.}$$

5. Solution for travel time in layer i:

$$T_i = (2c_m)^{-1}(E_i R_i - T_{1i} + T_{2i})$$

6. Solution for range derivative in layer i:

$$c_m \frac{dR_i}{dc_m} \equiv B_i$$
$$= D_i(B_{1i} - B_{2i}) - R_i$$

7. Ray intensity at receiver (point h) from a unit source (point s):

$$\frac{I}{F} = \frac{c_s}{c_h} \frac{\cot\theta_s \cot\theta_h}{R|B|}$$

where $R = \sum_i R_i$ and $B \sum_i B_i$ are the trajectory element sums for the layers traversed by the ray.

Appendix B
DESCRIPTION OF SUBROUTINES

In this appendix the function of the subroutines is described and the more important symbols are identified. A complete listing of all routines is also given.

RTRACE

The main program reads the title card and two parameter cards and initializes the CALCOMP plotter (if used). It also reads the source and receiver depth card so that TRACER can be called with the receiver array dimensions determined by the number of receiver depths used.

TRACER

TRACER is the principal subroutine for ray tracing and computing transmission loss. It reads the balance of the input data, converts it where necessary, and prints it out. Then, after the profile interpolation and print-plotting is done, TRACER begins selecting upgoing rays for tracing. For each angle (ray parameter) selected, SETRIN is called to compute the layer increments of range, travel time, phase, and intensity. A ray is traced by using repeated addition to fill in the arrays RT2, TIME 2, PHASE 2, and SLM 2, which record the values of the four trajectory elements at crossings of the receiver depths. The arrays NSRL, NBRL, NTOL, and NTUL are used to keep the turnaround history of the ray. The arrays RT1, TIME1, PHASE 1, and SLM1 hold the trajectory elements for the ray traced just before the current one. For each receiver depth, eigenrays (arrivals at receivers) are determined by linear interpolation rangewise, wherever the current ray and the previous ray are found to bracket a receiver range. (The interpolation is always between rays having the same turnaround history and never between the 0^0_+ and 0^0_- rays, or between any other "shadow zone" pairs.) In addition, single-ray interpolation is used to find eigenrays where a ray "grazes" a receiver depth.

Each eigenray yields a contribution to the receiver intensities stored in arrays CELL 1 and CELL 2. When the upgoing rays are done, the downgoing rays are traced similarly, to get more contributions.

When all rays have been traced, the intensities are converted to levels in decibels for all depths, one range at a time, and printed or plotted out. Then control returns to the main program to read in the next set of data.

The following is a list of the important symbols used in TRACER.

A0, A1, A2	coefficients in Wilson's equation
ATKYD	volume attenuation (dB/km)

ATTF	volume attenuation near the source (dB/s)
B	accumulation of $c_m \ \partial R/\partial \ c_m$ for ray being traced
BIN	array of layer increments of $c_m \ \partial R/\partial c_m$ for ray
BLT	array of bottom losses in decibels, tabulated by degrees
C2	latitude angle in Wilson's equation
CAS	array of c_a^2; square of layer parameter c_a
CBOT	sound speed at bottom of profile
CELL, CELL 2	arrays for accumulation of receiver intensity, indexed by receiver depth and range
CM	ray parameter (m/s); the sound speed at which the current ray will become horizontal
CMOLD	parameter of the previous ray
CMS	square of c_m
DARD	downtilt of receiver beam pattern (degrees)
DATD	downtilt of source beam pattern (degrees)
DBL	array of transmission losses for plot
DEL	a constant barely greater than 1.0 in the CDC-3800
DGAMD	input value of ray distribution increment (degrees)
DGAMR	input value of ray distribution increment (rad)
DGAMRM	for saving initial value of DGAMR
DR	range increment between receivers (km)
DTR	degrees per radian, 57.2957795
DUML	input buffer for beam patterns
EPBLT	array of bottom phase shifts in radians, tabulated by degree of grazing angle
F	direct ray amplitude
FACIN2	for each interface i, the quantity $$\frac{c_s}{c_i}\left[\left(\frac{c_m^2}{c_i^2}-1\right)\left(\frac{c_m^2}{c_s^2}-1\right)\right]^{-1/2}$$
FBL	bottom reflection coefficient for the current ray
FCSBL	source intensity with directivity included
FKHZ	transmitter frequency (kHz)
FSL	surface reflection coefficient (a constant)
GAMB	receiver grazing angle for forward bracketing ray (rad)

GAMDD	downgoing ray limit (degrees)
GAMDR	downgoing ray limit (rad)
GAMR	receiver grazing angle for current ray (rad)
GAMR1	receiver grazing angle for previous ray (rad)
GAMS	receiver grazing angle for rear bracketing ray (rad)
GAMUD	upgoing ray limit (degrees)
GAMUR	upgoing ray limit (rad)
GAZ	upgoing layer grazing angle (degrees)
GMIN	minimum equivalent angle on profile
GRAD	array of sound-speed profile gradients (fitted layers)
GRZ	angle used in Wilson's equation
I1L	number of receiver crossings for previous ray
I2	receiver depth crossing index
I2L	number of receiver crossings for current ray
IABOT	flag for bottom loss
IATT	flag for volume attenuation
IDBP	dummy
IDR	range increment for numbering plot axis (km)
IGRM	flag for shadow zone noninterpolable rays
ILAT, ILATS	for conversion of latitude in DTS input
ILG	range axis length for plots (in.)
ILON, ILONS	for conversion of longitude in DTS input
IM	flag for DTS input
IN	profile layer index
INTEMP	for saving IN when using contracted layer index
INV1	array of ray crossings for previous ray
INV2	array of ray crossings for current ray
IPER	flag for eigenray printout
IPL	flag for transmission-loss plots
IPRAY	flag for ray trace printout
IR	depth unit flag for DTS input
IR2	range upper limit for numbering plot axis
IRBP	flag for use of receiver beam pattern
ISCP	flag for coherent ray sum

ISLD	flag for relative maximum at source depth
ITBP	flag for source beam pattern
ITR	temperature unit flag for DTS input
IUVRS	flag for horizontal concentration of rays
JDY	depth increment for numbering y axis of ray plot (m)
JHGT	height of ray trajectory plot (in.)
JY2	depth upper limit for numbering ray plot axis
KA	array of profile fitting parameter K_a
LAT	input latitude for DTS input (degrees, minutes)
LCONL	array of contracted layer flags
LN	index to profile layers
LNB	number of profile layers including receivers
LNRC	number of receiver depths
LNSD	number of layers having equivalent angles
LONG	input longitude for DTS input (degrees, minutes)
LRCVN	array of layer depths, flagged for receiver numbers
LRCVNC	array of contracted layer depths, flagged for receiver numbers
M0, M1, M2, M3	factors in Wilson's equation
MAXCEL	maximum number of receiver ranges (equal to 2000/NDEPTH)
MAXM	number of receiver ranges
MAXNRC	maximum number of receiver depths (equal to NDEPTH)
MINV2	maximum number of ray crossings of a receiver depth (equal to 2000/NDEPTH)
NBR	running sum of bottom reflections for current ray
NBRL	array of bottom reflection sums, indexed by receiver depth and crossing number
NBRM	maximum allowable number of bottom reflections
NCROSS	maximum allowable number of receiver crossings
NDEPTH	number of receiver depths
NDR	index of downgoing ray
NECUR	flag for curved-earth profile correction
NIB	number of depths in profile (=LNB+1 after receiver in insertion)
NIBC	number of depths in contracted profile
NIS	index of source depth in profile
NISC	index of source depth in contracted profile

NOCL	maximum number of contracted layer depths
NORL	maximum number of profile depths
NRANGE	number of receiver ranges
NRCDR	flag for direct ray depth
NRECUR	flag for curved-earth correction of receiver and source depths
NSR	running sum of surface reflections for current ray
NSRL	array for surface reflection sums, indexed by receiver depth and crossing number
NSRM	maximum allowable number of surface reflections
NTO	running sum of turnovers (apexes) of current ray
NTOL	array of turnover sums, indexed by receiver depth and crossing number
NTU	running sum of turnunders (nadirs) of current ray
NTUL	array of turnover sums, indexed by receiver depth and crossing number
NUR	index of upgoing ray
OMEGA	frequency of source (rad/s)
P	accumulated phase for current ray being traced
PHASE 1	array of accumulated phase for last ray traced, indexed by receiver depth and crossing number
PHASE 2	array of accumulated phase for current ray, indexed by receiver depth and crossing number
PHB	phase of forward ray bracketing receiver
PHS	phase of backward ray bracketing receiver
PI	constant, 3.14159265
PIN	array of layer increments of phase for ray
R	accumulated range in meters for current ray
RADEARTH	mean radius of earth = 6,371,221.3 m
RB	range of forward ray bracketing receiver
RCD	array of receiver depths (m)
RF	first receiver range (km)
RIN	array of layer increments of range for ray
RKY	range converted to kilometers
RL	final receiver range (km)
RLP	final receiver range + 70 km (rays are traced to this range)
RS	range of backward ray bracketing receiver

RT1	array of accumulated ranges for last ray traced, indexed by receiver depth and crossing number
RT2	array of accumulated ranges for current ray, indexed by receiver depth and crossing number
S	salinity in parts per thousand (also used for temporary storage)
SD	source depth (m)
SDL	array of equivalent angles for profile parameters C_{ai}
SIG	reduced density σ_t, used to compute seawater pressure
SIG2	square of SIG
SL	initial intensity at first receiver range
SLDB	surface reflection loss (dB)
SLM1	array of intensity for last ray traced, indexed by receiver depth and crossing number
SLM2	array of intensity for current ray, indexed by receiver depth and crossing number
SLMIN	minimum intensity for a ray (equivalent to -200 dB)
SLVB	intensity of forward ray bracketing a receiver
SLVS	intensity of backward ray bracketing a receiver
SORLEV	source level (dB)
T	accumulated travel time for current ray (s)
TIMB	temperature travel time of forward ray bracketing a receiver
TIME 1	array of travel times for last ray traced, indexed by receiver depth and crossing number
TIME 2	array of travel times for current ray, indexed by receiver depth and crossing number
TIMS	travel time of backward ray bracketing a receiver
TIN	array of layer increments of travel time for a ray
VBPR	array of receiver directivity (transmission factor)
VBPT	array of source directivity (transmission factor)
VELF	sound speed at profile interface (ft/s)
VELM	sound speed at profile interface (ft/s)
VELY	sound speed at profile interface (yd/s)
VIL	array of sound speeds at interfaces (m/s)
VILC	sound speeds of contracted profile (m/s)
VSOR	sound speed at source depth (m/s)
VSOR2	square of VSOR

Z1	array of reduced depths for sound speed profile
Z2	array of reduced depths for sound speed profile
Z BOT	bottom depth (m)
ZIL	array of profile interface depths (m)
ZILC	array of contracted depths (m)
ZTO	turnover (apex) depth for current ray (m)
ZTU	turnunder (nadir) depth for current ray (m)

PROFILER

This subroutine computes the fitting parameters c_{ai}, k_{ai}, z_{ai} for the layers in the interpolated sound-speed profile.

PUSH

This subroutine replaces the input profile (arrays ZIL, VIL) with the interface points of the interpolated profile; the receiver depths are included. It also computes the gradient values at these points (array GRAD). Note that the array KA, which formerly represented the fitting parameter k_{ai}, is recomputed as $(k_{ai}/c_{ai}^2)^{1/2}$, which is a convenient quantity to use in the calculations that follow.

CONTRACT

This subroutine stores the contracted profile in the arrays ZILC and VILC. (The contracted profile is that subset of the interface points of the interpolated profile which contains the source and receiver depths and the depths of all profile maxima and minima, including the surface and bottom depths.)

PROF PLOT

This subroutine print-plots the fitted profile on the standard output unit, with the input profile points shown to indicate approximately the closeness of fit. Note that if the depth of the profile exceeds 2000, the portions below 1000 will be print-plotted to a reduced vertical scale. If a ray trajectory CALCOMP plot has been specified (JHGT $\neq$ 0), the profile is also CALCOMP-plotted to the same vertical scale as the ray plot.

SETRIN

This subroutine computes for one ray per call the ray trajectory increments (arrays RIN, TIN, BIN, PIN) for one cycle through the contracted profile. It also computes the depth at which a ray will turn over or turn under, and it interpolates in the source beam pattern and bottom-loss tables to find the decibel loss levels for the given ray angle.

PLOTTER

Subroutine PLOTTER produces, if specified on the second option card, a CALCOMP plot of transmission loss as a function of range, for as many as 20 receiver depths. The vertical plotting parameters are standard (50 to 130 dB at 10 dB/in.), whereas the horizontal (range) parameters are the same as for the ray trajectory plot. Therefore, to get a transmission loss CALCOMP plot, the horizontal plot parameters must be specified on the second option card, even if the ray trajectory plot is not wanted.

PRNT PLTR

This routine makes a printed plot of transmission loss as a function of range for as many as 20 receiver depths. (The use of this printed plot for more than four or five receivers is not recommended. A single receiver gives maximum legibility because a bar graph is produced in this case.)

RAY PLT

This routine makes a CALCOMP plot of all rays traced. Because the number and distribution of rays for a legible plot, and for a good intensity computation, may be incompatible, it may be necessary to do these in separate runs.

RAY PLT uses a cubic function of range to interpolate between points where the ray crosses contracted layer depths. The slope of the ray trajectory is mapped correctly at the end points of each segment, except where this would cause the cubic to have unnatural wiggles. Because the plot is most accurate near the contracted layer depths, it may be desirable to add receiver depths to the input data to cover the water column depth better, if an especially accurate trajectory plot is desired. This is another reason why a separate run for plotting might be used.

The range and depth limits on the ray plot need not be the same as those used for the ray trace itself.

Figure B1 shows a sample ray trajectory plot with the associated profile.

AXES

This CALCOMP plotting routine draws a set of rectilinear axes with title and labels.

The complete computer program listing is given in the following pages.

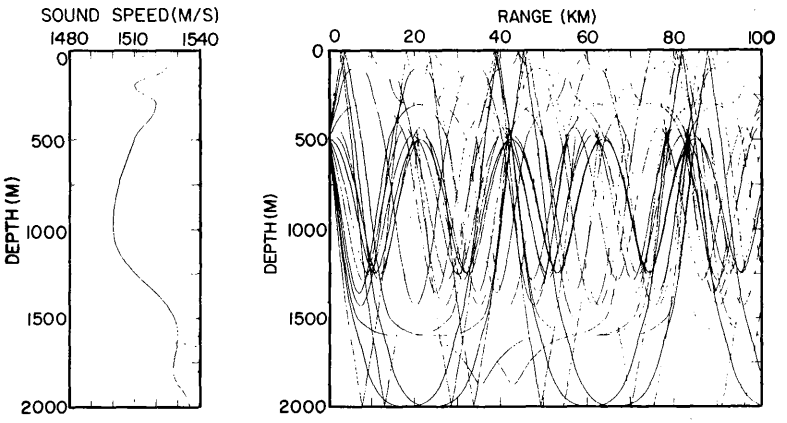


Fig. B1 — This ray trajectory plot shows rays that cycle about the sound channel axis, and also the split or bifurcating rays corresponding to maxima on the profile. The contracted layer depths are marked on the profile plot.

```
      PROGRAM RTRACE
      COMMON/GRAPH/IDR,IR2,ILG,JDY,JY2,JHGT,ZBOT
      DATA(IFIRST=1)
      COMMON/TITLE/TITLE(10)
      COMMON/TWO CARDS/GAMUD,GAMDD,DGAMD,RF,DR,RL,F KHZ,IATT,SLDB,SORLEV
     1   ,ISCP,IPER,IPRAY,IPL,IM,NSRM,NBRM,ITBP,IRBP,DATD,DARD
      DIMENSION PEN CODES(254)
      COMMON/TRAJCT/RT1(2000),RT2(2000),TIME1(2000),TIME2(2000),
     1   PHASE1(2000),PHASE2(2000),SLM1(2000),SLM2(2000)
      COMMON/HIST/NSRL(2000),NBRL(2000),NTOL(2000),NTUL(2000)
      COMMON/RAY SUM/CELL(2000),CELL2(2000)
      COMMON /MRIN/ VIL(100),ZIL(100),ZI(100),Z2(100),KA(100),CAS(100),
     1   CONL(100),RCVN(100),RCVNC(60),VIL C(60),ZIL C(60),FACIN2(100),
     2   RIN(100),TIN(100),BIN(100),PIN(100),VBPT(361),BLT(91)
     3   ,SD,RCD(50),FCSBL,CM,CBOT,FBL,VSOR,VSOR2,OMEGA,IDBP,IABOT,GAMR,
     4   NIS,ZTO,ZTU,NISC,NIBC,NIB,LNB,LNRC,NL,NUR,NDR,GAZ(98),GRAD(100)
C*****   RTRACE IS A PROGRAM WRITTEN IN FORTRAN-63 FOR THE CDC-3800 COMPUTER
C    AT THE NAVAL RESEARCH LABORATORY,  IT COMPUTES UNDERWATER ACOUSTIC TRANS-
C    MISSION LOSS BY SINGLE-PROFILE,FLAT-BOTTOM RAY TRACING AND SUMMATION.
C    THE INPUT SOUNDSPEED PROFILE IS FITTED TO A STRATIFIED INVERSE QUADRATIC
C    FUNCTION OF DEPTH(CF JASA 41,419-1967). COHERENT-PHASE OR RANDOM-PHASE
C    TRANSMISSION LOSS MAY BE PRINT-PLOTTED OR CALCOMP-PLOTTED, CALCOMP PLOTS
C    OF THE FITTED PROFILE AND RAY TRAJECTORIES MAY ALSO BE OBTAINED.
C    THIS PROGRAM WAS WRITTEN BY H P BUCKER AND E L WRIGHT, FORMERLY OF NRL.
C    FOR FURTHER INFORMATION SEE N R L REPORT      ,E B WRIGHT CODE 8167.
    2 READ 900,TITLE
      IF(EOF,60)1,3
    1 IF(IFIRST.EQ.0)CALL STOPPLOT
      STOP
    3 READ 901,GAMUD,GAMDD,DGAMD,RF,DR,RL,F KHZ,IATT,SLDB,SORLEV
      READ 902,ISCP,IPER,IPRAY,IPL,IM,NSRM,NBRM,IDR,IR2,ILG,JDY,JY2,JHGT
     1   ,ITBP,IRBP,DATD,DARD
      IF(JHGT.EQ.0.AND.IPL.NE.1.AND.IPL.NE.3)GO TO 5
      IF(IFIRST.EQ.1)CALL PLOTS(PENCODES,254,17)
      IFIRST=0
      CALL SYMBOL(0.,1.,.105,TITLE,90.,80)
      CALL PLOT(3.,0.,-3)
    5 PRINT 903,TITLE
      PRINT 14
   12 NI = 0
   06 NTEMP=NI+9
      READ 9014,(RCD(J,1),J=NI,NTEMP)
      DO 07 J = 1,10
      LNRC = NI +J
      IF (LNRC.LE.2) GO TO 07
      IF (RCD(LNRC).LT.0.001) GO TO 08
   07 CONTINUE
      NI = NI +10
      GO TO 06
   08 LNRC = LNRC -2
      IF(RCD(2).LT..001)RCD(2)=.1
      NDEPTH=LNRC
      NRANGE=2000/NDEPTH
      NCROSS=2000/NDEPTH
      CALL TRACER(NDEPTH,NRANGE,NCROSS,RT1,RT2,TIME1,TIME2,
     1   PHASE1,PHASE2,SLM1,SLM2,NSRL,NBRL,NTOL,NTUL,CELL,CELL2)
```

```
      GO TO 2
   14 FORMAT (1H0,5X, *ALL INPUT TO THIS PROGRAM IS IN METRIC UNITS,I.E.
     1, METERS, AND METERS/SEC, EXCEPT TEMPERATURE WHICH MAY BE INPUT IN
     2 EITHER DEGREES*,/,6X,
     *                    *C OR DEGREES F   THE OUTPUT IS IN METRIC UNITS
     3*)
  900 FORMAT(10A8)
  901 FORMAT(10F8.0)
  902 FORMAT(15I4,2F8.0)
  903 FORMAT(*1CURVILINEAR GRADIENT RAY THEORY  *,10A8//)
 9014 FORMAT (10F8.2 )
      END
```

```
      SUBROUTINE TRACER(NDEPTH,NRANGE,NCROSS,RT1,RT2,TIME1,TIME2,
     1   PHASE1,PHASE2,SLM1,SLM2,NSRL,NBRL,NTOL,NTUL,CELL,CELL2)
      COMMON/GRAPH/IDR,IR2,ILG,JDY,JY2,JHGT,ZBOT
      DIMENSION RT1(NDEPTH,NCROSS),RT2(NDEPTH,NCROSS),TIME1(NDEPTH,
     1   NCROSS),TIME2(NDEPTH,NCROSS),PHASE1(NDEPTH,NCROSS),
     2   PHASE2(NDEPTH,NCROSS),SLM1(NDEPTH,NCROSS),SLM2(NDEPTH,NCROSS),
     3   NSRL(NDEPTH,NCROSS),NBRL(NDEPTH,NCROSS),NTOL(NDEPTH,NCROSS),
     4   NTUL(NDEPTH,NCROSS),CELL(NDEPTH,NRANGE),CELL2(NDEPTH,NRANGE)
      DIMENSION SDL(100),DBL(50),INV1(100),INV2(100),DUML(180),
     1   EPBLT(91),VBPR(361)
      COMMON /MRIN/ VIL(100),ZIL(100),Z1(100),Z2(100),KA(100),CAS(100),
     1   LCONL(100),LRCVN(100),LRCVNC(60),VILC(60),ZILC(60),FACIN2(100),
     2   RIN(100),TIN(100),BIN(100),PIN(100),VBPT(361),BLT(91)
     3   ,SD,RCD(50),FCSBL,CM,CBOT,FBL,VSOR,VSOR2,OMEGA,ITBP,IABOT,GAMR,
     4   NIS,ZTO,ZTU,NISC,NIBC,NIB,LNB,LNRC,NL,NUR,NDR,GAZ(98),GRAD(100)
      REAL LAT, M0,M1,M2,M3, LONG
      COMMON/TITLE/TITLE(10)
      COMMON/TWO CARDS/GAMUD,GAMDD,DGAMD,RF,DR,RL,F KHZ,IATT,SLDB,SORLEV
     1   ,ISCP,IPER,IPRAY,IPL,IM,NSRM,NBRM,IDBP,IRBP,DATD,DARD
      DATA (PI=3.14159265),(DTR=57.2957795),(MINV2=100),(NI=0),
     1   (SLMIN=1.E-20),(NOCL=60),(NORL=100),(ISLD=0)
      DATA (DEL=20014000000000001B)
      DATA (RADEARTH=6371221.3)
 9001 FORMAT (/ 8H SD/RCD  , 10F11.2 /)
 9005 FORMAT (1H )
 9007 FORMAT ( 20H PROFILE INPUT DATA    /)
 9008 FORMAT (/24X,5HGAMUD,8X,5HGAMDD,7X,5HDGAMD,7X,2HRF,10X,2HDR,8X,
     1   2HRL,7X,4HITBP,5X,4HDATD,3X,4HIRBP,5X,4HDARD )
 9009 FORMAT (17H VBPL AND RANGES  , 2F12.5, F12.5, 3F12.3,2(I6,F10.3)/)
 9010 FORMAT(/12H VOL ATTEN = E10.3,9H DB/KM AT,F6.3,15H KHZ,  SL(DB) =,
     1   F5.2,16H  SOURCE LEVEL =, F6.1 /)
 9011 FORMAT (/19X, 2HGA,7X,3HBRC,9X,6HBL(DB),6X,7HARG(RC)  )
 9012 FORMAT (/20H SOUND LEVEL, RCV AT,10F11.1 )
 9013 FORMAT (/24H VERTICAL BEAM PATTERNS  /11H DEPRESSION,5X,
     *8HVBPT(DB),10X,8HVBPR(DB) )
 9016 FORMAT (//44H BOTTOM ABSORBS ALL INCIDENT SOUND ENERGY      //)
 9017 FORMAT ( 22H1COHERENT SUM OF RAYS  /)
 9018 FORMAT ( 20H1RANDOM SUM OF RAYS  /)
 9019 FORMAT ( 17H CONTROL INDICES    , 11I8//)
 9020 FORMAT(6H VBPT  ,10F11.6)
 9021 FORMAT(6H VBPR  ,10F11.6)
 9022 FORMAT(1H0/1H0,3X,9H(PROFILE),7X,3HNO,,6X,8HDEPTH(M),6X,15HVELOCIT
     1Y(M/SEC),14X,2HZ1,13X,3HCAS,11X,4HKACA,6X,5HLRCVN,6X,5HDC/DZ )
 9023 FORMAT ( 9H VBP(DB)  , 10F10.2 / 9X, 10F10.2 )
 9025 FORMAT(   3H0TU  , 4I5,F13.6,F11.4,F10.6,10X,F10.3)
 9024 FORMAT(   3H0TO  , 4I5,F13.6,F11.4,F10.6,F10.3)
 9026 FORMAT (  3H0SR  , 4I5,F13.6,F11.4,F10.6)
 9027 FORMAT (  3H0BR  , 4I5,F13.6,F11.4,F10.6,30X,F10.3)
 9028 FORMAT (  19H0CM SET LT REL MAX   , I8,E14.4,F14.6,F16.6 )
 9029 FORMAT (  19H0CM SET GT REL MAX   , I8,E14.4,F14.6,F16.6 )
 9030 FORMAT (/22X,4HISCP,3X,5HIABOT,4X,4HIPER,4X,5HIPRAY,3X,5HIUVRS,
     1   4X,4HNSRM,4X,4HNBRM ,5X,3HIPL)
 9031 FORMAT(/// 10X, 20H LIST OF EIGEN RAYS          )
 9032 FORMAT (/3X,3HNUR,3X,3HNDR,3X,3HNTO,3X,3HNSR,3X,3HNTU,3X,3HNBR,5X,
     1   5HRANGE,6X,5HDEPTH,6X,5HGZA-S,6X,5HGZA-R,7X,4HTIME,7X,
```

```
     2   12H S LEVEL(DB)     )
 9035 FORMAT (5X,3HNUR,2X,3HNDR,3X,2HLN,3X,2HIN,11X,2HCM,6X,5HRANGE,6X,
     14HGAMR,7X,3HZTO,7X,3HZTU,2X,8HZILC(IN),2X,8HVIL(NIB),11X,2HS2,12X,
     21HB,12X,1HP)
 9036 FORMAT (1H1,60X,*RAY HISTORIES* ,//)
 9050 FORMAT (10H TEST VBT  , 4I8,F12.2,E14.4 )
C*****   THIS IS THE MAIN RAY-TRACING AND SUMMATION ROUTINE FOR RTRACE.
C   THE SHORT CALLING PROGRAM IS USED TO SET UP THE VARIABLE DIMENSIONS FOR THE
C   RECEIVER ARRAYS.
      MAXNRC=NDEPTH
      MAXCEL=NRANGE
      ITBP=IDBP
      MINV2=NCROSS
 9100 DO 9102 I = 1, NOCL
 9102 LRCVNC(I) = 0
      DO 9104 I = 1, NORL
 9104 LRCVN(I) = 0
C*****   CHECK TO SEE IF DEPTH-TEMPERATURE-SALINITY INPUT HAS BEEN SPECIFIED.
      IF(IM.GT.0)GO TO 1000
      PRINT 9007
C*****   READ IN A SOUNDSPEED PROFILE, TERMINATED BY A NON-POSITIVE DEPTH.
      NI=1                                                          EBW12/70
      NTEMP=5                                                       EBW12/70
    2 READ 9909,(ZIL(J),VIL(J),J=NI,NTEMP)                          EBW12/70
 9909 FORMAT(5(F8.0,F8.4))                                          EBW12/70
      IF(ZIL(NTEMP).LE.0.)GO TO 4                                   EBW12/70
      NI=NI+5                                                       EBW12/70
      NTEMP=NI+4                                                    EBW12/70
      GO TO 2                                                       EBW12/70
    4 DO 5 J=1,5                                                    EBW12/70
      NIB=NTEMP-J                                                   EBW12/70
      IF(ZIL(NIB).GT.0.)GO TO 5555                                  EBW12/70
    5 CONTINUE                                                      EBW12/70
 5555 NECUR=0
      IF(VIL.LT.0.)NECUR=1
      VIL=ABSF(VIL)
      PRINT 9908,(ZIL(J),VIL(J),J=1,NIB)
 9908 FORMAT(* INPUT PROFILE*5(F10.1,F8.2))                         EBW12/70
    6 LNSD = NIB
      IF(NECUR.EQ.0)GO TO 12
C*****   APPLY THE CURVED-EARTH RAYTRACING CORRECTION TO THE SOUNDSPEED PROFILE.
      DO 10 J=1,NIB
      ZIL(J)=ZIL(J)*(1.+.5*ZIL(J)/RADEARTH)
   10 VIL(J)=VIL(J)*(1.+ZIL(J)/RADEARTH)
   12 IF(RCD.GE.0)GO TO 802
C*****   APPLY THE CURVED-EARTH CORRECTION TO SOURCE AND RECEIVER DEPTHS.
      RCD=ABSF(RCD)
      LNP=LNRC+1
      DO 801 I=1,LNP
801   RCD(I)=RCD(I)*(1.+.5*RCD(I)/RADEARTH)
802   SD=RCD(1)
      ZBOT=ZIL(NIB)
      DO 09 NRC = 1,LNRC
   09 RCD(NRC) = RCD(NRC+1)
      PRINT 9001, SD, (RCD(J), J = 1,LNRC)
      IF(LNRC.GT.MAXNRC) PRINT 11, LNRC,MAXNRC
```

```
11    FORMAT(*0 NO. OF RECEIVER DEPTHS*,I5,* GT*,I5)
      LNRC=MINO(LNRC,MAXNRC)
C
C*****   CALL THE PROFILE-FITTING AND INTERPOLATING ROUTINE,
C   THIS ALSO CAUSES THE INTERPOLATED PROFILE TO BE PRINT-PLOTTED, THE SOURCE
C   AND RECEIVER DEPTHS TO BE INSERTED INTO THE PROFILE, AND THE CONTRACTED
C   LAYERS TO BE STORED IN THE ARRAYS ZILC AND VILC.
      CALL PROFILER
C*****   COMPUTE THE LIST SDL OF EQUIVALENT ANGLES FOR ALL LAYERS,  THE VALUE
C   SDL(LNSD) IS FLAGGED NEGATIVE FOR RELATIVE MAXIMA ON THE PROFILE.
      LNSD=0
      DO 15 I=1,LNB
      IF(CAS(I).LT.VSOR2) GO TO 15
      LNSD=LNSD+1
      SDL(LNSD)=ATAN(SQRT(CAS(I)/VSOR2-1.))
      IF(I.EQ.1) GO TO 15
      IF(VIL(I).GT.VIL(I-1).AND.VIL(I).GE.VIL(I+1)) SDL(LNSD)=-SDL(LNSD)
15    CONTINUE
      I=1
      IF (CBOT.LE.VSOR) GO TO 16
      LNSD=LNSD+1
      SDL(LNSD)=-ACOS (VSOR/CBOT)
16    GMIN=10.
      JM=0
      DO 17 J=I,LNSD
      IF(ABS(SDL(J)).GE.GMIN) GO TO 17
      GMIN=ABS(SDL(J))
      JM=J
17    CONTINUE
      IF(I.EQ.1) GO TO 19
      IF(ABS(GMIN-ABS(SDL(I-1))).GT.1.E-7) GO TO 19
      IF(SDL(JM).GT.0.) GO TO 18
      SDL(I+1)=SDL(JM)
18    SDL(JM)=SDL(LNSD)
      LNSD=LNSD-1
      GO TO 20
19    GMIN=SDL(I)
      SDL(I)=SDL(JM)
      SDL(JM)=GMIN
      I=I+1
20    IF(I.LE.LNSD) GO TO 16
      PRINT 24, NIS, ZIL(NIS), VSOR
   24 FORMAT (/ 8H SOURCE  , I8, F12.2,F14.4 /)
      NRC = 1
      DO 26 IN = 1, NIB
      K = LRCVN(IN)
      IF (K.EQ.0) GO TO 26
      PRINT 25, IN,NRC,K,RCD(NRC),VIL(IN)
   25 FORMAT (/ 5H RCV  , 3I8,.F12.2,F14.4 )
      NRC = NRC +1
   26 CONTINUE
C*****   PRINTOUT OF THE FITTED PROFILE INTERFACES,INCLUDING LAYER PARAMETERS
C   AND GRADIENTS AT THE INTERFACES,
      PRINT 9022
      DO 28 IN=1,LNB
      PRINT 27,IN,ZIL(IN),VIL(IN),Z1(IN),CAS(IN),KA(IN),LRCVN(IN),
```

```
     1   GRAD(IN)
27    FORMAT(15H INTERFACE NO, ,I7,F14.3,F20.6,7X,3E15.7,I6,F14.4)
   28 CONTINUE
      PRINT 29,NIB,ZIL(NIB),VIL(NIB),LRCVN(NIB),GRAD(NIB)
29    FORMAT(15H INTERFACE NO, ,I7,F14.3,F20.6,52X,I6,F14.4)
C*****   PRINTOUT OF THE CONTRACTED PROFILE,WHICH CONSISTS OF THE INTERFACES
C    AT WHICH SOURCE AND RECEIVERS ARE LOCATED, OR AT WHICH THE SOUNDSPEED HAS
C    A LOCAL EXTREMUM.  THIS INCLUDES THE SURFACE AND BOTTOM.
      PRINT 30
   30 FORMAT(//4X,*CONTRACTED LAYERS      DEPTH      VELOCITY      LRCVNC*)
      DO 31 IN=1,NIBC
   31 PRINT 32,IN,ZILC(IN),VILC(IN),LRCVNC(IN)
   32 FORMAT(I11,10X,F10.2,F13.6,I11)
      PRINT 900$ $ PRINT 900$
      DO 33 I=1,LNSD
      S2=DTR*SDL(I)
      S3=(VSOR/COS(SDL(I)))**2
   33 PRINT 34,S3,SDL(I),S2
   34 FORMAT(*  CONVERT CAS *,E15.6,*  TO ANGLE *,F11.6,F11.4)
C*****   IF THE SOURCE ANGLE INCREMENT DGAMD IS NEGATIVE, THIS SIGNIFIES THAT
C    THE RAY DISTRIBUTION IS TO BE CONCENTRATED AROUND THE HORIZONTAL.
C    THIS IS ADVISABLE TO GET BETTER INTERPOLATION OF RANGE AND RAY INTENSITY.
      IUVRS=0
      IF(DGAMD.LT.0.)IUVRS=1
      DGAMD=ABSF(DGAMD)
C*****   NBRM IS THE MAXIMUM NUMBER OF BOTTOM REFLECTIONS.  IF IT IS ZERO, THE
C    BOTTOM WILL ABSORB ALL RAYS STRIKING IT, SO NO BOTTOM LOSS TABLE IS READ IN.
      IABOT=0
      IF(NBRM.EQ.0)IABOT=1
      RF=1000.*RF
      DR=1000.*DR
      RL=1000.*RL
C*****   RAYS WILL BE TRACED TO RL*70 KM TO ENSURE INTERPOLATING ALL ARRIVALS.
      RLP=RL*70000.0
      PRINT 9008
      PRINT 9009, GAMUD,GAMDD,DGAMD,RF,DR,RL,ITBP,DATD,IRBP,DARD
      PRINT 903
      PRINT 9019, ISCP,IABOT,IPER,IPRAY,IUVRS,NSRM,NBRM,IPL
      OMEGA = 2000.0*PI*FKHZ
C*****   IF VOLUME ATTENUATION HAS BEEN SPECIFIED,COMPUTE IT IN DB/KM.
      ATKYD = 0.
      IF(F KHZ.LT.,001.OR,IATT.EQ.0.)GO TO 37
      S2 = FKHZ**2
      ATKYD=0.0003025*S2  +44.*S2/(4100.+S2)
   37 PRINT 910,ATKYD, FKHZ, SLDB, SORLEV
      ATTF = 0.001*VSOR*ATKYD
      FSL = 10.0**(-0.1*SLDB)
      MAXM = (RL *RF +.1)/DR +1.
      IF (MAXM.LE.MAXCEL ) GO TO 39
      PRINT 38, MAXM
   38 FORMAT (/28H TOO MANY RANGE CELLS REQ,       , I6/)
      MAXM = MAXCEL
   39 GAMUR = GAMUD/DTR
      GAMDR = GAMDD/DTR
      DGAMR = DGAMD/DTR
      DGAMRH = DGAMR
```

```
      PRINT 9005
C*****   IF LAYER-BY-LAYER RAYTRACE PRINTOUT IS SPECIFIED,
C   WRITE A HEADER ON LUN 1,  THIS UNIT SHOULD BE EQUIPPED TO PR,
      IF (IPRAY.GT.0) WRITE (1,9036)
      IF (IPRAY.GT.0) WRITE (1,9035)
C*****   CHECK FOR A PARTIALLY REFLECTING BOTTOM,
      IF (IABOT.NE.1 ) GO TO 59
      PRINT 9016
      GO TO 69
C*****   SET THE BOTTOM LOSS AND PHASE SHIFT TABLES TO ZERO, READ NEW VALUES,
   59 DO 60 J = 1,91
      BLT(J) = 0.
   60 EPBLT(J) = 0.
      M = 0
61    NTEMP=M+19
      READ 62,(BLT(J+1),J=M,NTEMP)
      M = M +20
      IF (M.LT.100.AND.BLT(M).GT.0.001 ) GO TO 61
C*****   IF COHERENT PHASE SUMMATION IS NOT SPECIFIED, PHASE-SHIFT TABLE SKIPPED
      IF (ISCP.NE.1) GO TO 63
      N = 0
   67 NTEMP=N+19
      READ 62,(EPBLT(J+1),J=N,NTEMP)
      N = N +20
      IF (N.LT.100.AND.ABS(EPBLT(N)).GT.0.001 ) GO TO 67
   62 FORMAT ( 20F4.2 )
   63 DO 64 J = 2,91
      IF ( BLT(J).LT.0.001 ) BLT(J) = BLT(J-1)
      S2 = EPBLT(J-1) -EPBLT(J-2)
      IF (ABS(EPBLT(J)).GT.0.001) GO TO 64
      EPBLT(J) = EPBLT(J-1) +S2
      IF (EPBLT(J).GT.0.) EPBLT(J) = 0.
   64 CONTINUE
C*****   PRINT OUT THE BOTTOM LOSS AND PHASE-SHIFT TABLES,
      PRINT 9011
      DO 68 J = 1, 91
      K = J -1
      S2 = BLT(J)
      BLT(J) = 10.0**(-0.1*S2)
   65 IF (K.GT.M) GO TO 68
      S3 = DTR*EPBLT(J)
      PRINT 66, K, BLT(J), S2, S3
   66 FORMAT ( 13H BOTTOM LOSS  , I8, F12.4, 2F12.2 )
   68 CONTINUE
   69 CONTINUE
      NRCDR = 0
      IF (LRCVN(NIS).EQ.0.OR.VIL(NIS-1).GT.VSOR+0.0001.OR.
     1   VIL(NIS+1).GT.VSOR+0.0001) GO TO 85
      NRCDR = NISC
      PRINT 82, NRCDR
   82 FORMAT (/12H DIRECT RAY  , I8/)
   85 CONTINUE
C*****   PREPARE TO CHECK TO SEE IF SOURCE AND RECEIVER BEAM PATTERNS ARE USED,
      DO 88 I = 1, 361
      VBPT(I) = 1.
   88 VBPR(I) = 1.
```

```
      IF (ITBP.EQ.0) GO TO 100
      IT = DATD
      N = 0
      PRINT 9005
   90 NTEMP=N+19
      READ 62,(DUML(J+1),J=N,NTEMP)
      PRINT 9023,(DUML(J+1),J=N,NTEMP)
      N = N + 20
      IF (N.LT.180.AND.DUML(N).GT.0.001) GO TO 90
      PRINT 9005
   93 DO 95 I = 1,180
      J=IT+I
      K=IT-I+2
      IF (J.LT.1) J = J +360
      IF (K.LT.1) K = K +360
      IF(I.GT.N.OR.I.GT.10.AND.DUML(I).LT..001) GO TO 94
      S2 = 10.0**(-0.1*DUML(I))
   94 VBPT(J) = S2
      VBPT(K) = S2
      IF(I.GT.N) GO TO 95
      PRINT 9050, I,J,K,IT,DUML(I),S2
   95 CONTINUE
      J2 = 181
C*****   IF AN ASYMMETERIC SOURCE PATTERN IS SPECIFIED, READ IN LOWER HALF.
      IF (ITBP.EQ.1) GO TO 99
      J2 = 361
      N = 0
   96 NTEMP=N+19
      READ 62,(DUML(J+1),J=N,NTEMP)
      PRINT 9023,(DUML(J+1),J=N,NTEMP)
      N = N +20
      IF (N.LT.180.AND.DUML(N).GT.0.001) GO TO 96
      DO 98 I = 1,180
      K = IT -I +1
      IF (K.LT.1) K = K +360
      IF(I.GT.N.OR.I.GT.10.AND.DUML(I).LT..001) GO TO 97
      S2 = 10.0**(-0.1*DUML(I))
   97 VBPT(K) = S2
   98 CONTINUE
   99 VBPT(361) = VBPT(1)
      PRINT 9005
      IF(ITBP.NE.1) PRINT 9020,(VBPT(J),J=1,J2)
      PRINT 9005
C*****   CHECK FOR USE OF A RECEIVER BEAM PATTERN.
  100 IF (IRBP.EQ.0) GO TO 140
      N = 0
  110 NTEMP=N+19
      READ 62,(DUML(J+1),J=N,NTEMP)
      PRINT 9023,(DUML(J+1),J=N,NTEMP)
      N = N +20
      IF(N.LT.180.AND.DUML(N).GT..001) GO TO 110
      IT = DARD
  113 DO 115 I = 1,180
      J=IT+I-1
      K = IT -I +1
      IF (J.LT.1) J = J +360
```

```
      IF (K.LT.1) K = K +360
      IF(I.GT.N.OR.I.GT.10.AND.DUML(I).LT.,001) GO TO 114
      S2 = 10.0**(-0.1*DUML(I))
  114 VBPR(J) = S2
      VBPR(K) = S2
  115 CONTINUE
      J2 = 181
C*****   IF ASYMMETRIC RECEIVER PATTERN HAS BEEN SPECIFIED, READ IN LOWER HALF.
      IF (IRBP.EQ.1) GO TO 119
      J2 = 361
      N = 0
  116 NTEMP=N+19
      READ 62,(DUML(J+1),J=N,NTEMP)
      PRINT 9023,(DUML(J+1),J=N,NTEMP)
      N = N +20
      IF(N.LT.180.AND.DUML(N).GT.,001) GO TO 116
      DO 118 I = 1,180
      K = IT -I +1
      IF (K.LT.1) K = K +360
      IF(I.GT.N.OR.I.GT.10.AND.DUML(I).LT.,001) GO TO 117
      S2 = 10.0**(-0.1*DUML(I))
  117 VBPR(K) = S2
  118 CONTINUE
  119 VBPR(361) = VBPR(1)
      PRINT 9021,(VBPR(J),J=1,J2)
C*****   IF THE INTERPOLATION AND PRINTOUT OF EIGENRAYS HAS BEEN SPECIFIED,
C   PRINT OUT A HEADER HERE.
140   IF(IPER.EQ.0) GO TO 141
      PRINT 9031
      PRINT 9032
C
C*****   CLEAR THE RECEIVER INTENSITY ARRAYS BEFORE START OF RAYTRACE.
141   DO 142 IN=1,LNRC
      INV2(IN) = 0
      DO 142 J = 1,MAXM
      CELL(IN,J) = 0.
      CELL2(IN,J) = 0.
  142 CONTINUE
C*****   START SELECTING RAYS WITH THE FIRST UPGOING ONE, DOWNGOING INDEX=0.
      NUR = 1
      NDR = 0
C*****   IGRM IS A FLAG TO INDICATE THAT INTERPOLATION AT A RECEIVER BETWEEN
C   THE CURRENT RAY AND THE LAST ONE TRACED IS NOT VALID(SHADOW ZONE).
      IGRM=0
      CMOLD=VSOR
C*****   IF GAMUR IS NON-POSITIVE THERE ARE NO UPGOING RAYS SPECIFIED.
      IF (GAMUR) 175,175,143
C*****   THE ZERO-DEGREE RAY IS NOT TRACED.  START WITH .000001 RADIANS.
  143 NSD = 1
      GAMR = 1.E-6
C*****   IF GAMDR IS NEGATIVE THE RAYS ARE ALL ABOVE THE HORIZONTAL, BEGINNING
C   AT -GAMDR.
      IF (GAMDR) 144,145,145
  144 GAMR = -GAMDR
      GO TO 202
C*****   COMPUTATION OF RAY PARAMETER FOR GAMR=.000001 RADIAN.
```

```
145   CM=VSOR*DEL
      GO TO 203
C*****   CHECK FOR A RELATIVE PROFILE MAXIMUM AT THE SOURCE DEPTH.
  150 IF (ISLD.EQ.1) GO TO 153
C*****   CHECK TO SEE IF THE RAY DISTRIBUTION IS TO BE MADE MORE DENSE NEAR 0.
      IF (IUVRS.EQ.0) GO TO 151
      DGAMR = (0.01 *5.8*GAMR)*DGAMRH *1.E-6
      IF(DGAMR.GT..5*GAMR) DGAMR=.5*GAMR
      IF (GAMR+DGAMR.GT.GAMUR*.00001.AND.NUR.GE.1) DGAMR=GAMUR-GAMR
      IF (GAMR+DGAMR.GT.GAMDR*.00001.AND.NDR.GE.1) DGAMR=GAMDR-GAMR
  151 GAMR = GAMR +DGAMR
  152 S2 = ABS(SDL(NSD))
C*****   WHEN THE RAY DISTRIBUTION REACHES A VALUE CORRESPONDING TO A RELATIVE
C   MAXIMUM ON THE PROFILE, A RAY JUST BELOW THAT VALUE, AND ONE JUST ABOVE IT
C   WILL BE TRACED.  THIS IS THE BIFURCATING RAY, WITH SHADOW ZONE BETWEEN.
C   THE RAY DISTRIBUTION MAY BE CHANGED FOR MORE UNIFORM COVERAGE IN THIS AREA.
      IF (ABS(S2 -GAMR).GT.0.6*DGAMR) GO TO 157
      IF (SDL(NSD).GT.0.) GO TO 156
      ISLD = 1
      GAMR = S2 -1.E-6
      CM = VSOR/COS(GAMR)
      PRINT 9028, NSD,SDL(NSD),CM,GAMR
      GO TO 154
  153 ISLD = 0
      GAMR = ABS(SDL(NSD)) +1.E-6
      CM = VSOR/COS(GAMR)
      PRINT 9029,NSD,SDL(NSD),CM,GAMR
      IGRM=1
  154 IF (NDR.EQ.0) GO TO 155
      NDR = NDR +1
      GO TO 253
  155 NUR = NUR +1
      GO TO 203
156   GAMR=S2
  157 IF (S2.GT.GAMR.OR.NSD.GE.LNSD) GO TO 158
      NSD = NSD +1
      GO TO 152
  158 IF (NDR.EQ.0) GO TO 160
      NDR = NDR +1
      IF (GAMR -GAMDR -DGAMR +0.00001)252,400,400
  160 NUR = NUR +1
      IF (GAMR -GAMUR -DGAMR +0.00001)202,175,175
C*****   START SELECTING DOWNGOING RAYS, THE UPGOING RAY INDEX=0.
  175 NDR = 1
      NUR = 0
      CMOLD=VSOR
      IF (GAMDR) 400,400,176
  176 NSD = 1
      GAMR = 1.E-6
      IF (GAMUR) 177,178,178
  177 GAMR = -GAMUR
      GO TO 252
178   CM=VSOR*DEL
      GO TO 253
C*****   COMPUTE THE RAY PARAMETER CM AND START RAYTRACE AT SOURCE DEPTH.
C*****   CM IS THE PROFILE VALUE AT WHICH THIS RAY TURNS AROUND.
```

```
  202 CM = VSOR/COS(GAMR)
      IF(CM.LE.CMOLD) CM=DEL*CMOLD
  203 CMS = CM**2
      CMOLD=CM
      IN = NISC
C*****   SETRIN COMPUTES THE RAY TRAJECTORY INCREMENTS FOR ONE CYCLE THROUGH
C    THE CONTRACTED PROFILE.
      CALL SET RIN(CMS)
      NSR = 0
      NBR = 0
      NTO = 0
      NTU = 0
      R = 0.
      B = 0.
      P = 0.
      T = 0.
C*****   THIS CALL INITIALIZES THE RAY PLOT FOR AN UPGOING RAY.
      CALL RAY PLT(0.,SD,VSOR,CM,NDR)
C*****   ACCUMULATION OF TRAJECTORY INCREMENTS FOR A LAYER ENTERED AT BOTTOM.
  210 LN = IN -1
      R = R +RIN(LN)
      B = B +BIN(LN)
      P = P +PIN(LN)
      T = T +TIN(LN)
      IN = LN
C*****   IF CM.LT.VILC(LN) THE RAY WILL NOT REACH INTERFACE--I E TURNS OVER.
      IF (CM-VILC(LN)) 201,204,204
  201 NTO = NTO +1
      CALL RAY PLT(R,ZTO,CM,CM,NDR)
      IF (IPRAY.EQ.0.OR.R.LT.RF.OR.R.GT.RL) GO TO 260
      WRITE (1,9024) NUR,NDR,LN,IN,CM,R,GAMR,ZTO
      GO TO 260
  204 CALL RAY PLT(R,ZILC(LN),VILC(LN),CM,NDR)
C*****   CHECK FOR A LAYER PRINTOUT, IF IPRAY=1.
      IF (IPRAY.EQ.0.OR.R.LT.RF.OR.R.GT.RL) GO TO 206
      S2 = FCSBL*FACIN2(IN)/(R*B)
      WRITE (1,205) NUR,NDR,LN,IN,R,GAMR,ZILC(IN),S2,B,P
  205 FORMAT (  3HOUP   , 4I5,13X,F11.4,F10.6,20X,F10.3,10X,3E13.4)
C*****   IF THIS IS NOT A RECEIVER DEPTH SKIP OVER THE PARAMETER TABULATION.
  206 IF (LRCVNC(IN).EQ.0 ) GO TO 212
        INTEMP=IN
      IN=LRCVNC(IN)
      I2 = INV2(IN) +1
      IF (I2.GT.MINV2) GO TO 300
      RT2(IN,I2) =R
      PHASE2(IN,I2) = P
      TIME2(IN,I2) = T
      NSRL(IN,I2) = NSR
      NBRL(IN,I2) = NBR
      NTOL(IN,I2) = NTO
      NTUL(IN,I2) = NTU
      S2=ABS(FCSBL*FACIN2(INTEMP)/(R*B))
      IF(R.EQ.0.) S2=1.
      IF (S2.LT.SLMIN) GO TO 300
      IF (IRBP.EQ.0) GO TO 211
      S=360.-GAZ(INTEMP)
```

```
      N = S
      S2 = S2*(VBPR(N+1) +(S-N)*(VBPR(N+2) -VBPR(N-1)))
  211 SLM2(IN,I2) = S2
      INV2(IN) = I2
      IN=INTEMP
  212 IF (IN.GT.1 ) GO TO 215
C*****   RAY HAS BEEN TRACED TO A SURFACE HIT, CHECK TO SEE IF IT SHOULD BE
C   CONTINUED (REFLECTION WITH LOSS) OR TERMINATED HERE.
      IF (NSR.EQ.NSRM) GO TO 300
      FCSBL = FCSBL*FSL
      NSR = NSR +1
      P = P +PI
      IF (IPRAY.EQ.0.OR.R.LT.RF.OR.R.GT.RL) GO TO 265
      WRITE (1,9026) NUR,NDR,LN,IN,CM,R,GAMR
      GO TO 265
  215 IF ( R-RLP) 210,300,300
  252 CM = VSOR/COS(GAMR)
      IF(CM.LE.CMOLD) CM=DEL*CMOLD
  253 CMS = CM**2
C*****   INITIALIZATION FOR A DOWNGOING RAY.
      IN = NISC
      CMOLD=CM
      CALL SET RIN (CMS)
      NSR = 0
      NBR = 0
      NTO = 0
      NTU = 0
      R = 0.
      B = 0.
      P = 0.
      T = 0.
C*****   THIS CALL INITIALIZES THE RAY PLOT FOR A DOWNGOING RAY.
      CALL RAY PLT(0.,SD,VSOR,CM,NDR)
C*****   ACCUMULATION OF TRAJECTORY INCREMENTS FOR A LAYER ENTERED AT TOP.
  260 LN = IN
      R = R +RIN(LN)
      B = B +BIN(LN)
      P = P +PIN(LN)
      T = T +TIN(LN)
      IN = LN +1
C*****   IF CM.LT.VILC(IN) THE RAY MUST TURN UNDER BEFORE REACHING INTERFACE.
      IF (CM-VILC(IN)) 251,254,254
  251 NTU = NTU +1
      CALL RAY PLT(R,ZTU,CM,CM,NDR)
      IF (IPRAY.EQ.0.OR.R.LT.RF.OR.R.GT.RL) GO TO 210
      WRITE (1,9025) NUR,NDR,LN,IN,CM,R,GAMR,ZTU
      GO TO 210
  254 CALL RAY PLT(R,ZILC(IN),VILC(IN),CM,NDR)
      IF (IPRAY.EQ.0.OR.R.LT.RF.OR.R.GT.RL) GO TO 256
      S2 = FCSBL*FACIN2(IN)/(R*B)
      WRITE (1,255) NUR,NDR,LN,IN,R,GAMR,ZILC(IN),S2,B,P
  255 FORMAT (  3H0DW  , 4I5,13X,F11.4,F10.6,20X,F10.3,10X,3E13.4)
  256 IF (LRCVNC(IN).EQ.0) GO TO 262
        INTEMP=IN
      IN=LRCVNC(IN)
      I2 = INV2(IN) +1
```

```
      IF (I2.GT.MINV2) GO TO 300
      RT2(IN,I2) =R
      PHASE2(IN,I2) =P
      TIME2(IN,I2) = T
      NSRL(IN,I2) = NSR
      NBRL(IN,I2) = NBR
      NTOL(IN,I2) = NTO
      NTUL(IN,I2) = NTU
      S2=ABS(FCSBL*FACIN2(INTEMP)/(R*B))
      IF(R.EQ.0.) S2=1.
      IF (S2.LT.SLMIN) GO TO 300
      IF (IRBP.EQ.0) GO TO 261
      S=GAZ(INTEMP)
      N = S
      S2 = S2*(VBPR(N+1) +(S-N)*(VBPR(N+2) -VBPR(N+1)))
  261 SLM2(IN,I2) = S2
      INV2(IN) = I2
      IN=INTEMP
  262 IF (IN.LT.NIBC) GO TO 265
C*****   RAY HAS BEEN TRACED TO A BOTTOM HIT.  CHECK TO SEE IF IT SHOULD BE
C   TRACED ANY FURTHER (REFLECTION WITH LOSS) OR TERMINATED HERE.
      IF (IABOT.EQ.1.OR.NBR.EQ.NBRM) GO TO 300
      FCSBL = FCSBL*FBL
      NBR = NBR +1
      IF (IPRAY.EQ.0.OR.R.LT.RF.OR.R.GT.RL) GO TO 264
C*****   RAYTRACE PRINTOUT ON LUN 1 FOR BOTTOM REFLECTION.
      WRITE(1, 9027) NUR,NDR,LN,IN,CM,R,GAMR,VIL(NIB)
  264 IF (ISCP.NE.1) GO TO 215
      S = DTR*ACOS(VILC(NIBC)/CM)
      N = S
      P = P +EPBLT(N+1) +(S-N)*(EPBLT(N+2) -EPBLT(N+1))
      GO TO 215
  265 IF (R -RLP) 260,300,300
  300 IN = 0
  305 IN = IN +1
      IF (LRCVNC(IN).EQ.0 ) GO TO 350
        INTEMP=IN
      IN=LRCVNC(IN)
      I2L = INV2(IN)
C
C*****   PREPARE TO INTERPOLATE AT A RECEIVER, IF IT IS NOT CONTRAINDICATED.
      IF ( NUR.EQ.1.OR.NDR.EQ.1.OR.I2L.LE.0.OR.IGRM.EQ.1) GO TO 340
      I1L = INV1(IN)
      IF ( I1L.LE.0 ) GO TO 370
      I2 = 0
  310 I2 = I2 +1
C*****   SINCE THE PARAMETER ARRAYS ARE INDEXED BY DEPTH(IN)AND BY CROSSING
C   NUMBER(I2), INTERPOLATION IS AUTOMATICALLY BETWEEN TWO RAYS HAVING THE SAME
C   TURNAROUND HISTORY.  NON-INTERPOLABLE CASES HAVE ALREADY BEEN ELIMINATED.
      R2 = RT2(IN,I2)
      R1 = RT1(IN,I2)
      IF (R1.LT.RF.AND.R2.LT.RF) GO TO 339
      IF (R2.GT.R1) GO TO 316
      RS = R2
      RB = R1
      SLVS = SLM2(IN,I2)
```

```
      SLVB = SLM1(IN,I2)
      PHS = PHASE2(IN,I2)
      PHB = PHASE1(IN,I2)
      TIMS = TIME2(IN,I2)
      TIMB = TIME1(IN,I2)
      GAMS = GAMR
      GAMB = GAMR1
      GO TO 320
  316 RS =R1
      RB =R2
      SLVS = SLM1(IN,I2)
      SLVB = SLM2(IN,I2)
      TIMS = TIME1(IN,I2)
      TIMB = TIME2(IN,I2)
      PHS = PHASE1(IN,I2)
      PHB = PHASE2(IN,I2)
      GAMS = GAMR1
      GAMB = GAMR
C*****   NOW FIND THE EIGENRAY PARAMETERS BY LINEAR INTERPOLATION,
  320 RBS = RB -RS
      S2  = (SLVB -SLVS)/RBS
      S3 = (PHB -PHS)/RBS
      S4 = (TIMB -TIMS)/RBS
      IF (RS.GT.RF) GO TO 330
      M = 1
      R3 = RF -RS
      GO TO 332
324   FORMAT(6I6,F10.0,F11.1,2F11.5,F14.6,F12.1)
  330 M = (RS -RF)/DR +2.
      R3 = RF +(M-1)*DR -RS
  332 IF (R3.GT.RBS.OR.M.GT.MAXM) GO TO 339
      P4 = TIMS +R3*S4
      SL = 10.0**(-0.1*P4*ATTF)*(SLVS +R3*S2)
C*****   IF THE EIGENRAY PRINTOUT IS SPECIFIED, THE INTERPOLATED SOURCE ANGLE
C   AND ANGLE AT THE RECEIVER OF THE EIGENRAY ARE COMPUTED AND PRINTED,
      IF (IPER.EQ.0) GO TO 335
      P5 = 10.0*ALOG10(SL) +SORLEV
      P3 = GAMS +R3*(GAMB -GAMS)/RBS
      S5=DTR*ACOS(VILC(INTEMP)*COS(P3)/VSOR)
      P3 = DTR*P3
      P2 = RS +R3
      PRINT 324, NUR,NDR,NTOL(IN,I2),NSRL(IN,I2),NTUL(IN,I2),
     1   NBRL(IN,I2),P2,ZILC(INTEMP),P3,S5,P4,P5
      IF (IPER.EQ.1) GO TO 338
C*****   IF COHERENT PHASE SUMMATION IS SPECIFIED, SINE AND COSINE COMPONENTS
C   ARE SUMMED IN SEPARATE ARRAYS,
  335 IF (ISCP.NE.1) GO TO 336
      P5=SQRT(SL)
      S5 = -PHS -R3*S3
      CELL(IN,M)=CELL(IN,M)+P5*COS(S5)
      CELL2(IN,M)=CELL2(IN,M)+P5*SIN(S5)
      GO TO 338
C*****   SUM RANDOM-PHASE CONTRIBUTIONS IN A SINGLE ARRAY,
  336 CELL(IN,M) = CELL(IN,M) +SL
  338 M = M +1
  334 R3 = R3 +DR
```

```
      GO TO 332
  339 IF (I2.LT.I2L.AND.I2.LT.I1L ) GO TO 310
C*****   THE CURRENT RAY NOW BECOMES THE OLD ONE, BEFORE TRACING THE NEXT.
  340 INV1(IN) = INV2(IN)
      INV2(IN) = 0
      DO 342 I2 = 1,I2L
      RT1(IN,I2) = RT2(IN,I2)
      SLM1(IN,I2) = SLM2(IN,I2)
      PHASE1(IN,I2) = PHASE2(IN,I2)
      TIME1(IN,I2) = TIME2(IN,I2)
  342 CONTINUE
      IN=INTEMP
  350 IF (IN.LT.NIBC) GO TO 305
      GAMR1 = GAMR
      IGRM=0
      GO TO 150
  370 I2 = 1
  372 IF (I2+1.GT.I2L) GO TO 340
C*****   SINGLE-RAY INTERPOLATION IS FOR THE CASE WHEN THE CURRENT RAY CROSSES
C   THE RECEIVER DEPTH, BUT THE PREVIOUS ONE DID NOT.
      RS = RT2(IN,I2)
      RB = RT2(IN,I2+1)
      RBS = RB -RS
      IF (RB.LT.RF) GO TO 390
      SLVS = SLM2(IN,I2)
      SLVB = SLM2(IN,I2+1)
      S2 = (SLVB -SLVS)/RBS
      PHS = PHASE2(IN,I2)
      PHB = PHASE2(IN,I2+1)
      S3 = (PHB -PHS)/RBS
      TIMS = TIME2(IN,I2)
      TIMB = TIME2(IN,I2+1)
      S4 = (TIMB -TIMS)/RBS
      IF (RS.GT.RF) GO TO 380
      M = 1
      R3 = RF -RS
      GO TO 383
  380 M = (RS -RF)/DR +2.
      R3 = RF +(M-1)*DR -RS
  383 IF (R3.GT.RBS.OR.M.GT.MAXM) GO TO 390
      P4 = TIMS +R3*S4
      SL = 10.0**(-0.1*P4*ATTF)*(SLVS +R3*S2)
C*****   IF IPER=1, PRINT A LINE OF THE EIGENRAY PRINTOUT.
      IF (IPER.EQ.0) GO TO 385
      P5 = 10.0*ALOG10(SL) +SORLEV
      P3 = DTR*GAMR
      P2 = RS +R3
      S5=0.
      PRINT 324, NUR,NDR,NTOL(IN,I2),NSRL(IN,I2),NTUL(IN,I2),
     1   NBRL(IN,I2),P2,ZILC(INTEMP),P3,S5,P4,P5
      IF(IPER.EQ.1) GO TO 388
  385 IF (ISCP.NE.1) GO TO 386
C*****   ACCUMULATE A CONTRIBUTION TO COHERENT-PHASE INTENSITY.
      P5=SQRT(SL)
      S5 = -PHS -R3*S3
      CELL(IN,M)=CELL(IN,M)+P5*COS(S5)
```

```
      CELL2(IN,M)=CELL2(IN,M)+P5*SIN(S5)
      GO TO 388
C*****   ACCUMULATE A CONTRIBUTION TO RANDOM-PHASE INTENSITY.
  386 CELL(IN,M) = CELL(IN,M) +SL
  388 M = M +1
      R3 = R3 +DR
      GO TO 383
  390 I2 = I2 +2
      GO TO 372
400   IF(IPER.EQ.1.AND.IPL.EQ.0) GO TO 500
      R = RF
C*****   PRINT OUT A HEADER FOR COHERENT OR RANDOM TRANSMISSION LOSS.
      IF (ISCP.EQ.1 ) GO TO 402
      PRINT 9018
      GO TO 404
  402 PRINT 9017
C*****   PRINT OUT THE RECEIVER DEPTHS.
  404 PRINT 9012, (RCD(K), K = 1, LNRC )
      PRINT 9005
      M = 1
      VOLAT = 0.
  410 RKY = 0.001*R
      NRC = 1
      DO 429 IN = 1, NIBC
      J2 = LRCVNC(IN)
      IF (J2.EQ.0) GO TO 429
C*****   SELECT COHERENT OR RANDOM INTENSITY FOR OUTPUT.
      IF (ISCP.NE.1) GO TO 420
      S2 = CELL(J2,M)
      S3 = CELL2(J2,M)
C*****   IF A DIRECT RAY HAS BEEN FOUND,ADD A TERM FOR ITS COHERENT INTENSITY.
      IF (IN.NE.NRCDR) GO TO 415
      S4 = -OMEGA*R/VSOR
      F=1.
      IF(ITBP.NE.0) F=F*VBPT(1)
      IF(IRBP.NE.0) F=F*VBPR(1)
      F=SQRT(F)
      S2=S2+F*COS(S4)/R
      S3=S3+F*SIN(S4)/R
  415 S2 = S2**2 +S3**2
      GO TO 424
  420 S2 = CELL(J2,M)
C*****   IF A DIRECT RAY HAS BEEN FOUND,ADD A TERM FOR ITS RANDOM INTENSITY.
      IF (IN-NRCDR) 424,422,424
422   F=1.
      IF(ITBP.NE.0) F=F*VBPT(1)
      IF(IRBP.NE.0) F=F*VBPR(1)
      S2=S2+F/R**2
  424 DBL(NRC) = -200.
      IF (S2) 428,428,425
  425 DBL(NRC) = 10.0*ALOG10(S2) +SORLEV
  428 NRC = NRC +1
  429 CONTINUE
      PRINT 430, RKY,(DBL(J),J =1,LNRC)
  430 FORMAT (6H RKM = , F9.3, 5X, 10F11.1 )
      CELL2(1,M)=RKY
```

```
      DO 431 J=1,LNRC
  431 CELL(J,M)=DBL(J)
      IF (R.0.1.GT.RL) GO TO 500
      M = M +1
      R = R +DR
      IF(M.LE.MAXM) GO TO 410
  500 CONTINUE
C*****   IF THE RAY TRAJECTORY PLOT HAS BEEN MADE, THE PEN MUST BE SPACED OVER.
      IF(JHGT.NE.0.)CALL PLOT(ILG+3.,0.,-3)
      IF(IPL.EQ.0) GO TO 599
C*****   CALL TO PLOTTER PRODUCES A CALCOMP OF TRANSMISSION LOSS VS RANGE.
      IF(IABS(IPL-2).EQ.1) CALL PLOTTER(SD,RCD,LNRC,RF,DR,RL,MAXM,TITLE,
     1 CELL,NDEPTH,NRANGE)
C*****   THE FIRST CALL INITIALIZES THE PRINT-PLOT OF TRANSMISSION LOSS VS RANGE
      IF(IPL.GE.2) CALL PRNTPLTR(SD,RCD,LNRC)
      DO 515 I=1,MAXM
      RKY=CELL2(1,I)
      DO 510 J=1,LNRC
  510 DBL(J)=CELL(J,I)
C*****   SUBSEQUENT CALLS PRINT ONE LINE(RANGE INCREMENT) EACH.
      IF(IPL.GE.2) CALL PRPLOT1(RKY,DBL)
515   CONTINUE
C
C*****   END OF RAYTRACE AND TRANSMISSION LOSS OUTPUT.  RETURN FOR NEXT CASE.
  599 RETURN
C
C
C*****   DEPTH-TEMPERATURE-SALINITY INPUT, LATITUDE AND LONGITUDE ON FIRST CARD.
C   THE FIRST CARD MAY ALSO CONTAIN FLAGS FOR DEPTH AND TEMPERATURE UNITS.
C   Z MAY BE IN YARDS(IR=1),FEET(IR=2),OTHERWISE IT IS ASSUMED TO BE IN METERS.
C   T MAY BE IN FAHRENHEIT DEGREES(ITR=1),OTHERWISE IT IS ASSUMED CENTIGRADE.
C   S IS ALWAYS IN PARTS PER THOUSAND.
 1000 IN=1
      READ 1015, Z,T,S,IR,ITR,LAT,LONG
      ZIL=Z
      Z=0.
C*****   IF THE FIRST SALINITY VALUE IS NEGATIVE, CURVED-EARTH CORRECTION USED.
      NECUR=0
      IF(S.LT.0.)NECUR=1
      S=ABSF(S)
      GO TO 1042
 1005 IN = IN +1
C*****   SUBSEQUENT INPUT CARDS HAVE ONLY DEPTH,TEMPERATURE,SALINITY.
 1012 READ 1015, Z,T,S
 1015 FORMAT (F3.0,2F5.2,2I2,2F6.2 )
C*****   A BLANK CARD SIGNIFIES THE END OF THE PROFILE INPUT.
      IF (Z.EQ.0.0.AND.T.EQ.0.0.AND.S.EQ.0.0) 1260,1090
 1042 I = IR
      IT = ITR
      ILAT = LAT
      S2 = LAT -ILAT
      ILATS = 100.01*S2
      LAT = ILAT +S2/0.6
      ILON = LONG
      S2 = LONG -ILON
      ILONS = 100.01*S2
```

```
      LONG = ILON *S2/0.6
      PRINT 1045, ILAT,ILATS,ILON,ILONS,I,IT
 1045 FORMAT (/23H DEPTH-TEMP-SAL INPUT     ,3HLAT,I6,4H DEG,I6,4H MIN,4X
     1   ,4HLONG,I6,4H DEG,I6,4H MIN,5X,5HFLAGS,2I5 /)
      PRINT 1060
 1060 FORMAT (//7H   Z FT, 8X, 5H Z YD, 8X, 4HZ M , 7X, 4HTEMP,
     1   3X 5HTEMPF 4X 3HSAL 9X
     2   7HVEL F/S,6X,7HVEL Y/S,6X,7HVEL M/S,9X,8HPRESSURE ,2X,*DENSITY*
     3,/)
      C2 = COS(2.0*LAT/DTR)
      GRZ = 0.980616 -2.5928E-3*C2 +6.9E-6*C2*C2
 1090 IF (I.EQ.1) Z = 0.9144*Z
      IF (I.EQ.2) Z = 0.3048*Z
      IF (IT.EQ.1 ) T = (T-32.)*5./9.
      IF (S) 1092,1092,1094
 1092 S = SAVS
      GO TO 1100
 1094 SAVS = S
C*****   WILSON EQUATION AS IN TOLSTOY-CLAY, OCEAN ACOUSTICS(MCGRAW-HILL 1966)P6
 1100 S2 =S*S
      S3 =S*S2
      T2 =T*T
      T3 =T*T2
      T4 =T*T3
      SIG = -9.3445863E-2 +8.1487658E-1*S +4.8249614E-4*S2
     *    +6.7678614E-6*S3
      SIG2 = SIG*SIG
      A0 = (3.1186330E-6 +4.5317157E-3*T -5.4593903E-4*T2
     *    -1.9824828E-6*T3 -1.4385354E-10*T4 )/(67.26 +T )
      A1 = 1.0E-3 -4.7867E-6*T +9.8185E-8*T2 -1.0843E-9*T3
      A2 = 1.803E-8*T -8.164E-10*T2 +1.667E-11*T3
      TM3SP1 = A0 +A1*SIG +A2*SIG2 +1.0
      T10 = -2.2072E-7 +3.673E-8*T -6.63E-10*T2 +4.0E-12*T3
      T11 = 1.725E-8 -3.28E-10*T +4.0E-12*T2
      T12 = -4.5E-11 +1.0E-12*T
      M1 = T10 +T11*SIG +T12*SIG2
      T20 = -6.68E-14 -1.24064E-12*T +2.14E-14*T2
      T21 = -4.248E-13 +1.206E-14*T -2.0E-16*T2
      T22 = 1.8E-15 -6.0E-17*T
      M2 = T20 +T21*SIG +T22*SIG2
      M3 = 1.5E-17*T
      IF (Z) 1120,1110,1120
 1110 P = 10.1325
      GO TO 1150
 1120 SG = (GRZ +1.101E-7*(Z -ZM1 ))*(Z -ZM1 )
      P = PM1 +RM1*SG
      M0 = 1.0 -4.886E-6*P/(1.0 +1.83E-5*P )
      R = TM3SP1/(M0 +M1*P +M2*P*P +M3*P*P*P )
      P = PM1 +0.5*(R +RM1)*SG
 1150 M0 = 1.0 -4.886E-6*P/(1.0 +1.83E-5*P )
      R = TM3SP1/(M0 +M1*P +M2*P*P +M3*P*P*P )
      RM1 =R
      PM1 =P
      ZM1 =Z
      P = P*0.101971
 1200 P2 = P*P
```

```
      P3 = P*P2
      P4 = P2*P2
      VP = 1.6027E-1*P +1.0268E-5*P2 +3.5216E-9*P3 -3.3603E-12*P4
      VS = 1.39799*(S-35.) +1.6920E-3*(S-35.)**2
      VT = 4.5721*T -4.4532E-2*T2 -2.6045E-4*T3 +7.9851E-6*T4
      VSTP = (S -35.)*(-1.1244E-2*T +7.7711E-7*T2 +7.7016E-5*P
     1  -1.2943E-7*P2 +3.1580E-8*P*T +1.5790E-9*P*T2 )
     2  +P*(-1.8607E-4*T +7.4812E-6*T2 +4.5283E-8*T3 )
     3  +P2*(-2.5294E-7*T +1.8563E-9*T2 ) -1.9646E-10*P3*T
      VELM = 1449.14 +VP +VS +VT +VSTP
 1225 VELY = VELM/0.9144
      VIL(IN)=VELM
      VELF = VELY*3.0
      ZY = Z/0.9144
      IF(IN.GT.1)ZIL(IN)=Z
      ZF =  ZY*3.0
      TF = 1.8*T +32.
C*****   PRINT OUT THE CALCULATED PROFILE AND JUMP BACK TO THE MAINSTREAM.
      PRINT 1250,ZF,ZY,Z,T,TF,S, VELF,VELY,VELM, P,R
 1250 FORMAT (3(F9.2,3X),1X,2F7.2,F8.1,4X,3(F11.2,2X), 3X, F12.4 ,F9.5)
      GO TO 1005
 1260 NIB = IN -1
      NI = 0
      GO TO 06
9999  STOP
      END
```

```
      SUBROUTINE SETRIN(CMS)                                            SETR0000
C        CURRENT 7/2/70  STATEMENT 177 AND SECTION AT 30 FIXED          SETR0001
      COMMON /MRIN/ VIL(100),ZIL(100),Z1(100),Z2(100),KA(100),CAS(100), SETR0002
     1   LCONL(100),LRCVN(100),LRCVNC(60),VILC(60),ZILC(60),FACIN2(100),
     2   RIN(100),TIN(100),BIN(100),PIN(100),VBPT(361),BLT(91)
     3   ,SD,RCD(50),FCSBL,CM,CBOT,FBL,VSOR,VSOR2,OMEGA,ITBP,IABOT,GAMR,SETR0005
     4   NIS,ZTO,ZTU,NISC,NIBC,NIB,LNB,LNRC,NL,NUR,NDR,GAZ(98),GRAD(100)SETR0006
      REAL KA                                                           SETR0007
      DATA (DTR=57.2957795)                                             SETR0008
      DATA (WA=.16666666667),(WB=.66666666667),(C1=.33333333333)         SETR0009
      DIMENSION TINT(9),DR(9),DT(9),DB(9),VS(9)                         SETR0010
      ACOS(X)=ACOSF(SIGN(AMIN1(ABS(X),1.),X))                           SETR0011
      NIB=LNB+1                                                         SETR0012
      TGS=ABS(TANF(GAMR))                                               SETR0013
      ID=-2                                                             SETR0014
      FACIN2(NIS)=1./TGS**2                                             SETR0015
9     IN=NIS                                                            SETR0016
      VOLD=VSOR                                                         SETR0017
      TG1=TGS                                                           SETR0018
10    LN=IN+ID/2                                                        SETR0019
      IN=IN+ID+1                                                        SETR0020
      V=VIL(IN)                                                         SETR0021
      D=1.E200                                                          SETR0022
      IF(CM.LT.V) GO TO 11                                              SETR0023
      TG2=SQRT((CM-V)*(CM+V))/V                                         SETR0024
      IF(ABS(VOLD-V).LT.1.E-7*V) GO TO 30                               SETR0025
      IF(TG1.GE..75*TG2.AND.TG2.GE..75*TG1) GO TO 30                    SETR0026
      CT2=1./TG2                                                        SETR0027
      GO TO 12                                                          SETR0028
11    TG2=CT2=0.                                                        SETR0029
12    IF(ABS(CMS-CAS(LN)).GT.1.E-10*CMS) D=CAS(LN)/(CMS-CAS(LN))        SETR0030
      E=(CMS+CAS(LN))/CAS(LN)                                           SETR0031
      A=CM*KA(LN)                                                       SETR0032
      IF(ID.GT.-1) GO TO 125                                            SETR0033
      DZ1=Z2(LN)                                                        SETR0034
      DZ2=Z1(LN)                                                        SETR0035
      GO TO 126                                                         SETR0036
125   DZ1=Z1(LN)                                                        SETR0037
      DZ2=Z2(LN)                                                        SETR0038
126   R1=A*DZ1                                                          SETR0039
      R2=A*DZ2                                                          SETR0040
      B1=DZ1/TG1                                                        SETR0041
      B2=DZ2*CT2                                                        SETR0042
      T1=DZ1*TG1                                                        SETR0043
      T2=DZ2*TG2                                                        SETR0044
      IF(CM.GE.V) GO TO 15                                              SETR0045
      R=R1                                                              SETR0046
      IF(KA(LN).LT.0.) GO TO 13                                         SETR0047
      RIN(LN)=ACOS(SQRT(D)*ABS(R))/A                                    SETR0048
      GO TO 17                                                          SETR0049
13    RIN(LN)=ABS(ALOG((ABS(R)+TG1)*SQRT(ABS(D)))/A)                    SETR0050
      GO TO 17                                                          SETR0051
15    IF(KA(LN).LT.0.) GO TO 16                                         SETR0052
      RIN(LN)=ACOS(D*(ABS(R1*R2)+TG1*TG2))/A                            SETR0053
      GO TO 17                                                          SETR0054
```

```
16      RIN(LN)=ABS(ALOG((ABS(R1)+TG1)/(ABS(R2)+TG2))/A)                SETR0055
17      TIN(LN)=(E*RIN(LN)-(ID+1)*(T1-T2))/(2.*CM)                      SETR0056
        IF(CM.LT.V) GO TO 173                                           SETR0057
        BIN(LN)=RIN(LN)+ABS(D*(B2-B1))                                  SETR0058
        GO TO 177                                                       SETR0059
173     BIN(LN)=RIN(LN)-ABS(D*B1)                                       SETR0060
177     IF(D.GT.1.E-50) BIN(LN)=RIN(LN)+.5*CMS*ABS(B1*(1./(TG1*VOLD)**2 SETR0061
       1   1./(TG2*V)**2))                                              SETR0062
18      PIN(LN)=TIN(LN)*OMEGA                                           SETR0063
        IF(CM.LT.V) GO TO 20                                            SETR0064
C       FACIN2(IN)=VSOR/(TGS*TG1*V)                                     SETR0065
        FACIN2(IN)=VSOR/(TGS*TG2*V)                                     EBW 3 73
        VOLD=V                                                          SETR0066
        TG1=TG2                                                         SETR0067
        IF(IN.EQ.1) GO TO 19                                            SETR0068
        IF(IN.EQ.NIB) GO TO 900                                         SETR0069
        GO TO 10                                                        SETR0070
19      ID=0                                                            SETR0071
        GO TO 9                                                         SETR0072
20      IF(ID.LT.=1) GO TO 21                                           SETR0073
C       ZTU=ZIL(LN)+.5*RIN(LN)*TG1                                      SETR0074
        ZTU=ZIL(LN)-Z1(LN)+1./(KA(LN)*SQRT(ABS(D)*CMS))                 EBW 3 73
        GO TO 22                                                        SETR0075
C   21  ZTO=ZIL(LN+1)-.5*RIN(LN)*TG1                                    SETR0076
   21   ZTO=ZIL(LN)-Z1(LN)-1./(KA(LN)*SQRT(ABS(D)*CMS))                 EBW 3 73
22      IF(LN.EQ.1) GO TO 19                                            SETR0077
        IF(LN.EQ.NIB) GO TO 900                                         SETR0078
        LN=LN+ID+1                                                      SETR0079
        RIN(LN)=0.                                                      SETR0080
        BIN(LN)=0.                                                      SETR0081
        TIN(LN)=0.                                                      SETR0082
        PIN(LN)=0.                                                      SETR0083
        GO TO 22                                                        SETR0084
C          MODIFIED 6/29/70 FOR BETTER NUMERICAL INTEGRATION            SETR0085
30      IL=IN-ID-1                                                      SETR0086
        F=0.                                                            SETR0087
        DO 35 I=1,7                                                     SETR0088
        F=F+.125                                                        SETR0089
        ZINT=ZIL(IL)+F*(ZIL(IN)-ZIL(IL))                                SETR0090
        B=(KA(LN)*(ZINT+Z1(LN)-ZIL(LN)))**2                             SETR0091
        B=B*CAS(LN)                                                     SETR0092
        IF(KA(LN).LT.0.) B=-B                                           SETR0093
        VS(I)=CAS(LN)/(1.-B)                                            SETR0094
        TINT(I)=SQRT((CMS-VS(I))/VS(I))                                 SETR0095
        DR(I)=1./TINT(I)                                                SETR0096
        DT(I)=DR(I)/VS(I)                                               SETR0097
35      DB(I)=DT(I)/TINT(I)**2                                          SETR0098
        D=ABS(ZIL(IN)-ZIL(IL))                                          SETR0099
        DRA=D*(WA/TG1+WB*DR(4)+WA/TG2)                                  SETR0100
        DRB=.5*(DRA+D*(WB*(DR(2)+DR(6))-C1*DR(4)))                      SETR0101
        DRC=.5*(DRB+D*(C1*(DR(1)+DR(3)+DR(5)+DR(7))-WA*(DR(2)+DR(6))))  SETR0102
        ERA=(DRA-DRB)/15.                                               SETR0103
        ERB=(DRB-DRC)/15.                                               SETR0104
        DRAB=DRB-ERA                                                    SETR0105
        DRBC=DRC-ERB                                                    SETR0106
        ERAB=(DRAB-DRBC)/63.                                            SETR0107
```

```
      RIN(LN)=DRBC-ERAB                                                 SETR0108
      DTA=D*CM*(WA/(VIL(IL)**2*TG1)+WB*DT(4)+WA/(VIL(IN)**2*TG2))       SETR0109
      DTB=.5*(DTA+D*CM*(WB*(DT(2)+DT(6))+C1*DT(4)))                     SETR0110
      DTC=.5*(DTB+D*CM*(C1*(DT(1)+DT(3)+DT(5)+DT(7))-WA*(DT(2)+DT(6))))  SETR0111
      ETA=(DTA-DTB)/15.                                                 SETR0112
      ETB=(DTB-DTC)/15.                                                 SETR0113
      DTAB=DTB-ETA                                                      SETR0114
      DTBC=DTC-ETB                                                      SETR0115
      ETAB=(DTAB-DTBC)/63.                                              SETR0116
      TIN(LN)=DTBC-ETAB                                                 SETR0117
      DBA=D*CMS*(WA/(VIL(IL)**2*TG1**3)+WB*DB(4)+WA/(VIL(IN)**2*TG2**3))SETR0118
      DBB=.5*(DBA+D*CMS*(WB*(DB(2)+DB(6))+C1*DB(4)))                    SETR0119
      DBC=.5*(DBB+D*CMS*(C1*(DB(1)+DB(3)+DB(5)+DB(7))-WA*(DB(2)+DB(6))))SETR0120
      EBA=(DBA-DBB)/15.                                                 SETR0121
      EBB=(DBB-DBC)/15.                                                 SETR0122
      DBAB=DBB-EBA                                                      SETR0123
      DBBC=DBC-EBB                                                      SETR0124
      EBAB=(DBAB-DBBC)/63.                                              SETR0125
      BIN(LN)=DBBC-EBAB                                                 SETR0126
      GO TO 18                                                          SETR0127
  900 LN = 0                                                            SETR0128
      LNC = 0                                                           SETR0129
  910 LN = LN +1                                                        SETR0130
      LNC = LNC +1                                                      SETR0131
      RIN(LNC) = RIN(LN)                                                SETR0132
      BIN(LNC) = BIN(LN)                                                SETR0133
      PIN(LNC) = PIN(LN)                                                SETR0134
      TIN(LNC) = TIN(LN)                                                SETR0135
      FACIN2(LNC) = FACIN2(LN)                                          SETR0136
  920 IF (LN.EQ.LNB) GO TO 930                                          SETR0137
      IF (LCONL(LN+1).EQ.0) GO TO 910                                   SETR0138
      LN = LN +1                                                        SETR0139
      RIN(LNC) = RIN(LNC) +RIN(LN)                                      SETR0140
      BIN(LNC) = BIN(LNC) +BIN(LN)                                      SETR0141
      TIN(LNC) = TIN(LNC) +TIN(LN)                                      SETR0142
      PIN(LNC) = PIN(LNC) +PIN(LN)                                      SETR0143
      GO TO 920                                                         SETR0144
  930 FACIN2(LNC+1) = FACIN2(LN+1)                                      SETR0145
      LNC=LNC+1                                                         SETR0146
      DO 940 IN = 1,LNC                                                 SETR0147
      GAZ(IN) = 0.                                                      SETR0148
      IF (CM.GT.VILC(IN)) GAZ(IN) = DTR*ACOS(VILC(IN)/CM)               SETR0149
  940 CONTINUE                                                          SETR0150
      FCSBL = 1.                                                        SETR0151
      IF (ITBP.EQ.0) GO TO 960                                          SETR0152
      S = GAZ(NISC)                                                     SETR0153
      IF (NUR.GT.0) S = 360. -S                                         SETR0154
      N = S                                                             SETR0155
      FCSBL = VBPT(N+1) +(S-N)*(VBPT(N+2) -VBPT(N+1))                   SETR0156
  960 N = GAZ(NIBC)                                                     SETR0157
      FBL = BLT(N+1) +(GAZ(NIBC) -N)*(BLT(N+2) -BLT(N+1))               SETR0158
      RETURN                                                            SETR0159
      END                                                               SETR0160
```

```
      SUBROUTINE PROFILER                                               PROF0000
      COMMON /MRIN/ VIL(100),ZIL(100),ZI(100),ZA(100),KA(100),CAS(100), PROF0001
     1   LCONL(100),LRCVN(100),LRCVNC(60),VILC(60),ZILC(60),FACIN2(100),
     2   RIN(100),TIN(100),BIN(100),PIN(100),VBPT(361),BLT(91)
     3   ,SD,RCD(50),FCSBL,CM,CBOT,FBL,VSOR,VSOR2,OMEGA,ITBP,IABOT,GAMR,PROF0004
     4   NIS,ZTO,ZTU,NISC,NIBC,NIB,LNB,LNRC,NL,CI(100),GRAD(100)        PROF0005
      REAL KA                                                           PROF0006
      IPLOT=1                                                           PROF0007
      IF(ZIL(1).LT.0.) IPLOT=0                                          PROF0008
      ZIL(1)=ABS(ZIL(1))                                                PROF0009
C*****   THE STANDARD FITTING TOLERANCES TIGHTENED BY E B W 5/73
      EPSV=.1  $  EPSZ=1.  $  FEZ=.0025
      IF(ZIL(1).EQ.0.) GO TO 5                                          PROF0012
      EPSV=EPSV*ZIL(1)                                                  PROF0013
      EPSZ=EPSZ*ZIL(1)                                                  PROF0014
      FEZ=FEZ*ZIL(1)                                                    PROF0015
5     ZIL(1)=0.                                                         PROF0016
      DO 10 I=3,NIB                                                     PROF0017
      G1=(VIL(I-2)-VIL(I-1))/(ZIL(I-2)-ZIL(I-1))                        PROF0018
      G2=(VIL(I-1)-VIL(I))/(ZIL(I-1)-ZIL(I))                            PROF0019
10    GRAD(I-1)=SIGN(SQRT(AMAX1(0.,G1*G2)),G1)                          PROF0020
      IF(GRAD(2).NE.0.) GO TO 15                                        PROF0021
      GRAD(1)=0.                                                        PROF0022
      GO TO 17                                                          PROF0023
15    GRAD(1)=((VIL(2)-VIL(1))/(ZIL(2)-ZIL(1)))**2/GRAD(2)              PROF0024
17    IF(GRAD(NIB-1).NE.0.) GO TO 19                                    PROF0025
      GRAD(NIB)=0.                                                      PROF0026
      GO TO 20                                                          PROF0027
19    GRAD(NIB)=((VIL(NIB)-VIL(NIB-1))/(ZIL(NIB)-ZIL(NIB-1)))**2/       PROF0028
     1   GRAD(NIB-1)                                                    PROF0029
20    DO 30 I=2,NIB                                                     PROF0030
      GB=(VIL(I)-VIL(I-1))/(ZIL(I)-ZIL(I-1))                            PROF0031
      TV=4.*GB**2                                                       PROF0032
      IF(GRAD(I)*GRAD(I-1).LT.TV) GO TO 30                              PROF0033
      IF(ABS(GRAD(I)).LT.ABS(GRAD(I-1))) GO TO 25                       PROF0034
      GRAD(I)=.9*TV/GRAD(I-1)                                           PROF0035
      GO TO 30                                                          PROF0036
25    GRAD(I-1)=.9*TV/GRAD(I)                                           PROF0037
30    CONTINUE                                                          PROF0038
      NL=0                                                              PROF0039
      CI(1)=VIL(1)                                                      PROF0040
      ZI(1)=ZIL(1)                                                      PROF0041
      I=1                                                               PROF0042
      J=2                                                               PROF0043
40    V1=1./VIL(I)**2                                                   PROF0044
      V3=1./VIL(J)**2                                                   PROF0045
      V2=(V1+V3)/2.                                                     PROF0046
      Z1=ZIL(I)                                                         PROF0047
      Z3=ZIL(J)                                                         PROF0048
      G1=-2.*GRAD(I)*V1/VIL(I)                                          PROF0049
      G3=-2.*GRAD(J)*V3/VIL(J)                                          PROF0050
      Z2=(Z1+Z3)/2.                                                     PROF0051
      DGB=(G3-G1)/2.                                                    PROF0052
      IF(DGB.EQ.0.) GO TO 45                                            PROF0053
      F=(Z1+Z3+(V1-V3)/DGB)/2.                                          PROF0054
```

```
      D=F*F-Z1*Z3-(Z1*(V2-V3)+Z3*(V1-V2))/DGB                          PR0F0055
42    D=SQRT(D)                                                        PR0F0056
      Z2=F+D                                                           PR0F0057
      IF(Z2.GT.Z3) Z2=F-D                                              PR0F0058
45    G2=2.*(V1-V2)/(Z1-Z2)-G1                                         PR0F0059
      DD=(G1-G2)/(Z1-Z2)                                               PR0F0060
      ZAA=Z1                                                           PR0F0061
      IF(DD.NE.0.) ZAA=Z1-G1/DD                                        PR0F0062
      CASA=1./(V1-.5*DD*(Z1-ZAA)**2)                                   PR0F0063
      AKA=-.5*DD*CASA                                                  PR0F0064
      DD=(G2-G3)/(Z2-Z3)                                               PR0F0065
      ZAB=Z3                                                           PR0F0066
      IF(DD.NE.0.) ZAB=Z3-G3/DD                                        PR0F0067
      CASB=1./(V3-.5*DD*(Z3-ZAB)**2)                                   PR0F0068
      AKB=-.5*DD*CASB                                                  PR0F0069
      IF(I+1.LT.J.AND.G2*(G1+G3).LT.0.) GO TO 70                       PR0F0070
      K=I+1                                                            PR0F0071
50    IF(K.GE.J) GO TO 75                                              PR0F0072
      ZK=ZIL(K)                                                        PR0F0073
      IF(ZK.LT.Z2) GO TO 55                                            PR0F0074
      VK=SQRT(CASB/(1.-AKB*(ZK-ZAB)**2))                               PR0F0075
      GK=-AKB*(ZK-ZAB)/CASB                                            PR0F0076
      GO TO 60                                                         PR0F0077
55    VK=SQRT(CASA/(1.-AKA*(ZK-ZAA)**2))                               PR0F0078
      GK=-AKA*(ZK-ZAA)/CASA                                            PR0F0079
60    GK=ABS(VK**3*GK/2.)                                              PR0F0080
      IF(ABS(VIL(K)-VK).GT.EPSV+GK*(EPSZ+FEZ*ZIL(K) )) GO TO 70        PR0F0081
      K=K+1                                                            PR0F0082
      GO TO 50                                                         PR0F0083
70    I=J-1                                                            PR0F0084
      NL=NL+2                                                          PR0F0085
      IF(I.EQ.NIB) GO TO 100                                           PR0F0086
      GO TO 40                                                         PR0F0087
75    CI(NL+2)=1./SQRT(V2)                                             PR0F0088
      CI(NL+3)=VIL(J)                                                  PR0F0089
      ZI(NL+2)=Z2                                                      PR0F0090
      ZI(NL+3)=Z3                                                      PR0F0091
      ZA(NL+1)=ZAA                                                     PR0F0092
      ZA(NL+2)=ZAB                                                     PR0F0093
      CAS(NL+1)=CASA                                                   PR0F0094
      CAS(NL+2)=CASB                                                   PR0F0095
      KA(NL+1)=AKA                                                     PR0F0096
      KA(NL+2)=AKB                                                     PR0F0097
      J=J+1                                                            PR0F0098
      IF(GRAD(J-1).EQ.0.) GO TO 70                                     PR0F0099
      IF(J-1.EQ.NIB) GO TO 70                                          PR0F0100
      GO TO 40                                                         PR0F0101
100   IF(IPLOT.EQ.1) CALL PROFPLOT                                     PR0F0102
      CALL PUSH                                                        PR0F0103
      CALL CONTRACT                                                    PR0F0104
      RETURN                                                           PR0F0105
      END                                                              PR0F0106
```

```
      SUBROUTINE PROFPLOT                                               PROF0000
      COMMON/GRAPH/IDR,IR2,ILG,JDY,JY2,JHGT,ZBOT
      COMMON /MRIN/ VIL(100),ZIL(100),ZI(100),ZA(100),KA(100),CAS(100), PROF0001
     1   LCONL(100),LRCVN(100),LRCVNC(60),VILC(60),ZILC(60),FACIN2(100),PROF0002
     2   RIN(100),TIN(100),BIN(100),PIN(100),VBPT(361),BLT(91)
     3   ,SO(51)     ,FCSBL,CM,CROT,FBL,VSOR,VSOR2,OMEGA,ITBP,IABOT,GAMR,PROF0004
     4   NIS,ZTO,ZTU,NISC,NIBC,NIB,LNB,LNBC,NL,CI(100),GRAD(100)          PROF0005
      DIMENSION LINE(121),IDOT(20),ICS(10) $ REAL KA $ ZMAX=ZI(NL+1)    PROF0006
C*****   THIS ROUTINE PRINT-PLOTS THE INTERPOLATED SOUNDSPEED PROFILE,ALONG
C    WITH THE INPUT POINTS.  IF A RAY-TRAJECTORY PLOT HAS BEEN SPECIFIED,A
C    CALCOMP PLOT OF THE FITTED PROFILE IS ALSO PRODUCED.
      CMIN=CMAX=CI(1) $ DO 10 I=1,NL $ C=CI(I+1) $ IF(C.GE.CMAX) CMAX=C PROF0007
      IF(C.LE.CMIN) CMIN=C                                              PROF0008
10    CONTINUE $ CMIN=10.*ANINT(CMIN/10.) $ CMAX=10.*ANINT(CMAX/10.+1.) PROF0009
      SC=(CMAX-CMIN)/120. $ IF(ZMAX.GT.2000.) GO TO 13                  PROF0010
      DZL=DZH=ZMAX/100. $ GO TO 15                                      PROF0011
13    DZL=20. $ DZH=(ZMAX-1000.)/50.                                    PROF0012
15    PRINT 900,CMIN,CMAX                                               PROF0013
      IF(JHGT.EQ.0)GO TO 17
      XSCALE=.05
      YSCALE=FLOAT(JHGT)/JY2
      KC1=(CMIN+CMAX+10.)/20.
      KC1=KC1*10-30
      CALL PLOT(0.,.5,-3)
      KC2=KC1+60
      KLG=3
      KDC=10
      CALL AXES(KC1,KDC,KC2,KLG,JY2,-JDY,0,JHGT,0,
     1   18HSOUNDSPEED(M/SEC)$,9HDEPTH(M)$,-4)
      NPEN=3
   17 DO 18 I=1,121                                                     PROF0016
18    LINE(I)=1H $ C=CMIN $ NDOT=0                                      PROF0017
20    IC=(C-CMIN)/SC+1.5 $ NDOT=NDOT+1 $ IDOT(NDOT)=IC $ C=C+10.        PROF0018
      IF(C.LE.CMAX+1.) GO TO 20 $ Z=0. $ IL=IN=1                        PROF0019
25    ZN=Z+DZL $ NIC=1 $ C=SQRT(CAS(IL)/(1.-KA(IL)*(Z-ZA(IL))**2))      PROF0020
      DO 26 I=1,NDOT $ J=IDOT(I)                                        PROF0021
26    LINE(J)=1H. $ IC=(C-CMIN)/SC+1.5 $ ICS(NIC)=IC                    PROF0022
      IF(IC.GE.1.AND.IC.LE.121) LINE(IC)=1HX                            PROF0023
29    IF(ZN.LT.ZIL(IN)) GO TO 35 $ IF(ZIL(IN)-Z.GT.ZN-ZIL(IN)) GO TO 35 PROF0024
30    IF(ZIL(IN).GE.ZI(IL)) GO TO 31 $ IL=IL-1 $ IF(IL.GT.0) GO TO 30   PROF0025
31    IF(ZIL(IN).LE.ZI(IL+1)) GO TO 32 $ IL=IL+1 $ IF(IL.LE.NL) GO TO 31 PROF0026
      IL=NL                                                             PROF0027
32    V=SQRT(CAS(IL)/(1.-KA(IL)*(ZIL(IN)-ZA(IL))**2)) $ DV=VIL(IN)-V    PROF0028
      IC=(C+DV-CMIN)/SC+1.5 $ NIC=NIC+1 $ ICS(NIC)=IC                   PROF0029
      IF(IC.GE.1.AND.IC.LE.121) LINE(IC)=1HI $ IN=IN+1                  PROF0030
      C=V
      IF(IN.LE.NIB)GO TO 29
   35 IF(JHGT.EQ.0)GO TO 36
      X=XSCALE*(C-KC1)
      Y=YSCALE*(JY2-Z)
      IF(X.GT.-.5.AND.X.LT.KLG+.5.AND.Y.GE.0.)CALL PLOT(X,Y,NPEN)
      NPEN=2
   36 PRINT 901,Z,LINE
      DO 37 I=1,NIC $ IC=ICS(I) $ IF(IC.GE.1.AND.IC.LE.121) LINE(IC)=1H PROF0034
37    CONTINUE $ Z=ZN                                                   PROF0035
```

```
39    IF(Z.LT.ZI(IL+1)) GO TO 45 $ IL=IL+1 $ GO TO 39                    PROF0036
45    IF(Z.GT.1000.) DZL=DZH $ IF(Z.LE.ZMAX) GO TO 25 $ PRINT 902        PROF0037
      IF(JHGT.NE.0)CALL PLOT(KLG,3.,.5,-3)
900   FORMAT(*0PRINTER PLOT OF PROFILE. AN X MARKS THE VELOCITY.  AN I MPROF0014
     1ARKS AN INPUT POINT EXTRAPOLATED TO THE NEAREST LINE.*/*0DEPTH*,  PROF0015
     2   F9.0,111X,F10.0)                                               PROF0016
901   FORMAT(F14.2,1X,121A1)                                             PROF0033
902   FORMAT(1H0)                                                        PROF0038
      RETURN $ END                                                       PROF0039
```

```
      SUBROUTINE PUSH                                                   PUSH0000
      COMMON /MRIN/ VIL(100),ZIL(100),ZI(100),ZA(100),KA(100),CAS(100), PUSH0001
     1   LCONL(100),LRCVN(100),LRCVNC(60),VILC(60),ZILC(60),FACIN2(100),PUSH0002
     2   RIN(100),TIN(100),BIN(100),PIN(100),VBPT(361),BLT(91)
     3   ,SD(51)     ,FCSBL,CM,CBOT,FBL,VSOR,VSOR2,OMEGA,ITBP,IABOT,GAMR,PUSH0004
     4   NIS,ZTO,ZTU,NISC,NIBC,NIB,LNB,LNRC,NL,CI(100),GRAD(100)         PUSH0005
      REAL KA $ N=LNRC+1 $ DO 5 I=1,100                                 PUSH0006
5     LRCVN(I)=0 $ DO 100 I=1,N $ DR=SD(I) $ M=NL+1 $ DO 10 J=1,M $ JZ=JPUSH0007
      IF(ABS(ZI(J)-DR).LT.,001) GO TO 40                                PUSH0008
      IF(ZI(J).LT.DR.AND.DR.LT.ZI(J+1)-.001) GO TO 20                   PUSH0009
10    CONTINUE $ GO TO 100                                              PUSH0010
20    NL=NL+1 $ DO 25 K=JZ,NL $ J=NL+JZ-K $ ZI(J+1)=ZI(J)               PUSH0011
      CI(J+1)=CI(J) $ CAS(J+1)=CAS(J) $ KA(J+1)=KA(J)                   PUSH0012
25    ZA(J+1)=ZA(J) $ ZI(JZ+1)=DR                                       PUSH0013
      CI(JZ+1)=SQRT(CAS(JZ+1)/(1.-KA(JZ+1)*(ZI(JZ+1)-ZA(JZ+1))**2))    PUSH0014
      IF(NIS.GT.JZ) NIS=NIS+1 $ JZ=JZ+1                                 PUSH0015
40    IF(I.NE.1) GO TO 45 $ VSOR=CI(JZ) $ NIS=JZ $ GO TO 100            PUSH0016
45    LRCVN(JZ)=I-1                                                     PUSH0017
100   CONTINUE $ GRAD(1)=CI(1)**3*KA(1)*(ZI(1)-ZA(1))/CAS(1)            PUSH0018
      DO 200 I=1,NL $ VIL(I)=CI(I) $ ZIL(I)=ZI(I)                       PUSH0019
      GRAD(I+1)=CI(I+1)**3*KA(I)*(ZI(I+1)-ZA(I))/CAS(I) $ B=KA(I)/CAS(I)PUSH0020
      KA(I)=SQRT(ABS(B)) $ IF(B.LT.0.) KA(I)=-KA(I) $ ZI(I)=ZI(I)-ZA(I) PUSH0021
200   ZA(I)=ZI(I+1)-ZA(I) $ VIL(NL+1)=CI(NL+1) $ ZIL(NL+1)=ZI(NL+1)     PUSH0022
      NIB=NL+1 $ LNB=NL $ VSOR2=VSOR**2 $ CBOT=VIL(NIB) $ RETURN $ END  PUSH0023
```

```
      SUBROUTINE CONTRACT                                               CONT0000
      COMMON /MRIN/ VIL(100),ZIL(100),ZI(100),ZA(100),KA(100),CAS(100), CONT0001
     1   LCONL(100),LRCVN(100),LRCVNC(60),VILC(60),ZILC(60),FACIN2(100),
     2   RIN(100),TIN(100),BIN(100),PIN(100),VBPT(361),BLT(91)
     3   ,SD,RCD(50),FCSBL,CM,CBOT,FBL,VSOR,VSOR2,OMEGA,ITBP,IABOT,GAMR,CONT0004
     4   NIS,ZTO,ZTU,NISC,NIBC,NIB,LNB,LNRC,NL,CI(100),GRAD(100)        CONT0005
      IN=INC=1                                                          CONT0006
      DO 5 I=1,60                                                       CONT0007
5     LRCVNC(I)=0                                                       CONT0008
      GO TO 20                                                          CONT0009
10    IF(IN.NE.NIS) GO TO 13                                            CONT0010
      NISC=INC                                                          CONT0011
      GO TO 20                                                          CONT0012
13    IF(LRCVN(IN).NE.0) GO TO 20                                       CONT0013
      LCONL(IN)=1                                                       CONT0014
      IF(VIL(IN).GT.VIL(IN+1)) GO TO 11                                 CONT0015
      IF(VIL(IN).LT.VIL(IN-1)) GO TO 20                                 CONT0016
      GO TO 12                                                          CONT0017
11    IF(VIL(IN).GE.VIL(IN-1 )) GO TO 20                                CONT0018
   12 IN=IN+1                                                           CONT0019
      IF (IN.NE.NIB) GO TO 10                                           CONT0020
      NIBC=INC                                                          CONT0021
   20 VILC(INC)=VIL(IN)                                                 CONT0022
      ZILC(INC)=ZIL(IN )                                                CONT0023
      LRCVNC(INC)=LRCVN(IN)                                             CONT0024
      LCONL(IN)=0                                                       CONT0025
      INC=INC+1                                                         CONT0026
      IF (IN.EQ.NIB ) RETURN                                            CONT0027
      GO TO 12                                                          CONT0028
      END                                                               CONT0029
```

```
      SUBROUTINE PLOTTER(SD,RD,N,RZ,DR,RMAX,MAX,TITLE,CELL,NDEPTH,
     1   NRANGE)
      DIMENSION CELL(NDEPTH,NRANGE)
      DIMENSION TITLE ( 8)
      DIMENSION RD(20),SYM(20)
      DIMENSION X1(400),Y1(400)
      INTEGER SYM
      COMMON/GRAPH/IDR,IR2,ILG,JDY,JY2,JHGT
      DATA(LY1=50),(LDY=10),(LY2=130),(LHGT=8)
      DATA (SYM=0,1,2,3,4,5,6,7,8,9,10,11,12,14,19,21,25,0,1,2)
C*****   THIS ROUTINE PRODUCES A CALCOMP PLOT OF TRANSMISSION LOSS, IF SPECIFIED
      LLHGT=LHGT-.2
      XSCALE=(.001*ILG)/IR2
    7 Y=.4*(N+3)+1.
      CALL SYMBOL(0.,Y,.14,3HSD=,0.,3)
      CALL NUMBER(.36,Y,.14,SD,0.,4HF9.2)
      Y=Y-.4
      CALL SYMBOL(.12,Y,.14,3HRCD,0.,3)
      CALL SYMBOL(.96,Y,.14,3HSYM,0.,3)
      DO 10 I=1,N
      Y=Y-.4
      CALL NUMBER(0.,Y,.105,RD(I),0.,4HF7.2)
10    CALL SYMBOL(1.08,Y,.105,SYM(I),0.0,-1)
      CALL PLOT(4.,.5,-3)
      CALL AXES(0,IDR,IR2,ILG,LY2,-LDY,LY1,LHGT,0,
     1   12HRANGE IN KMS,24HTRANSMISSION LOSS IN DBS,4)
      CALL SYMBOL(.5*ILG-4.,9.,.21,TITLE,0.,64)
      X1=XSCALE*RZ
      DX=XSCALE*DR
      DO 31 I=2,MAX
   31 X1(I)=X1(I-1)+DX
      DO 41 I=1,N
      DO 40 J=1,MAX
      Y1(J)=13.+.1*CELL(I,J)
      IF(Y1(J).LT..2)Y1(J)=.2
      IF(Y1(J).GT.LLHGT)Y1(J)=LLHGT
40    CONTINUE
      CALL LINE(X1,Y1,MAX,1,SYM(I),.105,5)
   41 CONTINUE
      CALL PLOT(ILG*3.,0.,3)
      RETURN
      END
```

```
      SUBROUTINE PRNTPLTR(SD,RCD,N)
      COMMON/TWO CARDS/GAMUD,GAMDD,DGAMD,RF,DR,RL,F KHZ,IATT,SLDB,SORLEV
     1   ,ISCP,IPER,IPRAY,IPL,IM,NSRM,NBRM,ITBP,IRBP,DATD,DARD
      TYPE INTEGER BUF,SYM,DOT,BLNK
      DIMENSION RCD(20),SYM(20),BUF(121)
      DATA (BLNK=1H ),(SYM=1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9,1HA,
     1   1HB,1HC,1HD,1HE,1HF,1HG,1HH,1HI,1HJ,1HK),(DOT=1H.)
C*****   PRINT-PLOTS TRANSMISSION LOSS FOR AS MANY AS 20 RECEIVER DEPTHS, BUT
C    THIS PRINT-PLOT IS NOT RECOMMENDED FOR MORE THAN FOUR OR FIVE DEPTHS.
C    IF THERE IS ONLY ONE RECEIVER DEPTH THIS WILL BE A BAR GRAPH.
C*****   NOTE THAT THE SYMBOLS FOR DEEPER RECEIVERS MAY OVERLAY SHALLOWER ONES.
      PRINT 900
      PRINT 901,SD,(RCD(I),SYM(I),I=1,N)
      NREF=(SORLEV+10.)/20.
      NREF=NREF*20
      BUF=NREF-160
      PEG VALUE=161.5-NREF
      DO 10 I=2,7
   10 BUF(I)=BUF(I-1)+20
      PRINT 902,(BUF(I),I=1,7)
      RETURN
      ENTRY PRPLOT1
      BUF=DOT
      II=1
      DO 20 I=1,12
      DO 15 IJ=1,9
      II=II+1
   15 BUF(II)=BLNK
      II=II+1
   20 BUF(II)=DOT
      DO 30 I=1,N
      J=PEG VALUE+RCD(I)
      IF(J.LT.1) J=1
      IF(J.GT.121) J=121
      BUF(J)=SYM(I)
      IF(N.GT.1)GO TO 30
      DO 25 JJ=1,J
   25 BUF(JJ)=SYM(I)
30    CONTINUE
      PRINT 903,SD,BUF
      RETURN
900   FORMAT(1H1,30X,*RECEIVED LEVEL VS RANGE*)
901   FORMAT(*0SD=*,F10.2/*0RECEIVER AT DEPTH      PLOTS AS*//
     1   (8X,F10.2,12X,A1))
902   FORMAT(*0R(KM) *,60X,*LEVEL(DB) *//I14,6I20//)
  903 FORMAT(F8.2,5X,121A1)
      END
```

```
      DIMENSION DEPTH(20),GRAD(20)
      COMMON/GRAPH/IDR,IR2,ILG,JDY,JY2,JHGT,ZBOT
      DATA(DG=.08),(GMAX=3.05),(NH=-4),(IFIRST=1)
C*****   RAY TRAJECTORY PLOTTING SUBROUTINE FOR RTRACE.  A PIECEWISE CUBIC IS
C   FITTED FOR EACH LAYER SEGMENT WITH PRESCRIBED POSITION AND SLOPE AT END-
C   POINTS.  E B WRIGHT MARCH 1973.
C*****   SINCE THE INTERPOLATING FUNCTION IS NOT THE REAL RAY SOLUTION,THE
C   PLOTTED TRAJECTORIES ARE NOT THE EXACT RAY PATHS.  THE PLOTTED POSITIONS
C   WILL BE MOST ACCURATE NEAR DEPTHS ON THE CONTRACTED PROFILE.
      IF(JHGT.EQ.0)RETURN
      R=RR
      IF(R.GT.0.)GO TO 100
C*****   THIS SECTION FOR INITIALIZING EACH RAY.
      XLG=ILG
      RSCALE=(.001*ILG)/IR2
      ZSCALE=FLOAT(JHGT)/JY2
      R=ABSF(R)
      RL=R
      ZL=Z
      NFIDDLE=0
      IF(IFIRST.EQ.0)GO TO 50
C*****   IF THIS IS THE FIRST RAY, DRAW AND LABEL THE PLOT AXES.
      IFIRST=0
      CALL PLOT(0.,.5,-3)
      CALL AXES(0,IDR,IR2,ILG,JY2,-JDY,0,JHGT,0,10HRANGE(KM)$,
     1   9HDEPTH(M)$,NH)
   50 X=R*RSCALE
      Y=ZSCALE*(JY2-Z)
      CALL PLOT(X,Y,3)
      SECANT=CM/C
      GL=SQRT(SECANT*SECANT-1.)
      IF(NDR.NE.0)GL=-GL
      YL=Y
      RETURN
C*****   COMPUTE THE MEAN SLOPE FOR THE SEGMENT AND THE NORMALIZED SLOPES AT
C   THE ENDPOINTS.
  100 GBAR=(ZL-Z)/(R-RL)
      IF(NFIDDLE.NE.0)GO TO 86
  110 SECANT=CM/C
      G=SQRT(SECANT*SECANT-1.)
  120 IF(GBAR.LT.0.)G=-G
      G0=GL/GBAR
      G1=G/GBAR
C*****   A WIDE DISPARITY IN THE NORMALIZED ENDPOINT SLOPES MAY CAUSE UNNATURAL
C   WIGGLES IN THE INTERPOLATED TRAJECTORY. ADJUST G0 OR G1 IF NECESSARY.
C   THE INTERPOLATING CUBIC WILL USUALLY BE WELL-BEHAVED BECAUSE THE KNOTS ARE
C   ON A SET OF DEPTHS INCLUDING ALL EXTREMA ON THE SOUNDSPEED PROFILE.
      IF(G0.GT.DG.OR.G1.LT.GMAX)GO TO 140
  130 G1=GMAX
      G=G1*GBAR
      NFIDDLE=NFIDDLE+1
      GRAD(NFIDDLE)=ABSF(G)
      DEPTH(NFIDDLE)=Z
      GO TO 150
```

```
  140 IF(G1.GT.DG.OR.G0.LT.GMAX)GO TO 150
      G0=GMAX
      NFIDDLE=NFIDDLE+1
      GRAD(NFIDDLE)=ABSF(G0*GBAR)
      DEPTH(NFIDDLE)=ZL
C*****   PLOT IN STEPS OF .01 INCH
  150 N=100.*(R-RL)*RSCALE+.5
      DY=(ZL-Z)*ZSCALE
      DO 200 I=1,N
      X=X+.01
      XN=FLOAT(I)/N
C*****   THIS IS THE INTERPOLATING CUBIC.
      Y=YL+XN*(G0+XN*((3.-2.*G0-G1)+XN*(G0+G1-2.)))*DY
      IF(X.GT.XLG.OR.Y.LT.0.)GO TO 200
      CALL PLOT(X,Y,2)
  200 CONTINUE
      GL=G
      IF(Z.EQ.0..OR.Z.EQ.ZBOT)GL=-GL
      RL=R
      ZL=Z
      YL=Y
      RETURN
   86 DO 300 ND=1,NFIDDLE
      IF(Z.NE.DEPTH(ND))GO TO 300
      G=GRAD(ND)
      GO TO 120
  300 CONTINUE
      GO TO 110
      END
```

```
      SUBROUTINE AXES(IX1,IDX,IX2,ILG,JY1,JDY,JY2,JHGT,                 AXS00100
     1   LABEL,LABEL X,LABEL Y,NH)                                      AXS00200
C*****   THIS SUBROUTINE PLOTS A SET OF RECTILINEAR AXES WITH LENGTH ALONG THE
C   X-AXIS EQUAL TO ILG AND HEIGHT ALONG Y-AXIS EQUAL TO JHGT (BOTH IN  INCHES),
C   THE X-AXIS IS NUMBERED AT EQUAL INTERVALS FROM IX1 TO IX2,WITH STEP IDX,
C   THE Y-AXIS IS NUMBERED AT EQUAL INTERVALS FROM JY1 TO JY2 WITH STEP JDY,
C   NOTE THAT THESE INPUT PARAMETERS ARE ALL TYPE INTEGER,
C   THE ARRAYS LABEL(32 CHARACTERS) AND LABELX AND LABELY(24 EACH) ARE USED TO
C   TITLE THE GRAPH AND TO LABEL THE AXES, THE PARAMETER NH CONTROLS THE
C   HEIGHT OF THE LETTERING AND OTHER SPACING PARAMETERS. IF NH>0 THE X-AXIS
C   IS LABELED ALONG THE BOTTOM AND THE TITLE APPEARS AT THE TOP. IF NH<0
C   THE X-AXIS IS LABELED ALONG THE TOP AND THE TITLE APPEARS BELOW THE GRAPH,
C   IN EITHER CASE THE LETTER HEIGHT IS IABS(NH)*.035 INCHES,
C   ANY OF THE LABELS CAN BE TERMINATED BY THE CHARACTER $ OR OMITTED ENTIRELY
C   BY MAKING THE FIRST LOCATION OF THE ARRAY CONTAIN ZERO.
C   PROGRAMMED BY E B WRIGHT  NRL CODE 8160   JAN 1972.
      DIMENSION LABEL(4),LABELX(3),LABELY(3),ICHAR(32)                  AXS00300
      DATA(MAX=32),(MAXA=24),(NFORMAT=2HI4)                             AXS00400
C*****   THE FIRST SECTION COMPUTES SPACING PARAMETERS.
      NNH=IABS(NH)                                                      AXS00500
      IF(NNH.GT.20)NNH=.25*JHGT+2.                                      AXS00600
      H=NNH*.035   $   TWO H=H+H   $   HALF H=.25*TWO H                 AXS00700
      DT=3.*HALF H                                                      AXS00800
      D=1.25*DT                                                         AXS00900
      TWO D=D+D                                                         AXS01000
      TDH=TWO D-H                                                       AXS01100
      NX=(IX2-IX1)/IDX                                                  AXS01200
      NY=(JY2-JY1)/JDY                                                  AXS01300
      DX=FLOAT(ILG)/NX                                                  AXS01400
      DY=FLOAT(JHGT)/NY                                                 AXS01500
C*****   THE NEXT SECTION DRAWS AND NUMBERS (IF SPECIFIED) THE LOWER X-AXIS.
      XX=-TWO H                                                         AXS01600
      YY=-D                                                             AXS01700
      NO NUMBER=1
      IF(NH.GT.0)CALL NUMBER(XX,YY,H,IX1,0.,NFORMAT)                    AXS01800
      N=IX1+IDX                                                         AXS01900
      XX=XX+DX                                                          AXS02000
      CALL PLOT(0.,0.,3)                                                AXS02100
      X=DX                                                              AXS02200
      DO 100 I=2,NX                                                     AXS02300
      CALL PLOT(X,0.,2)                                                 AXS02400
      CALL PLOT(X,DT,2)                                                 AXS02500
      IF(NH.GT.0.AND.NO NUMBER.NE.1)CALL NUMBER(XX,YY,H,N,0.,NFORMAT)
      NO NUMBER=-NO NUMBER
      CALL PLOT(X,0.,3)                                                 AXS02700
      N=N+IDX                                                           AXS02800
      XX=XX+DX                                                          AXS02900
  100 X=X+DX                                                            AXS03000
      CALL PLOT(X,0.,2)                                                 AXS03100
      IF(NH.GT.0)CALL NUMBER(XX,YY,H,N,0.,NFORMAT)                      AXS03200
C*****   THE NEXT SECTION DRAWS THE RIGHT-HAND Y-AXIS.   THE THREE COMMENTS
C   ARE STATEMENTS WHICH COULD BE USED TO NUMBER THIS AXIS.
      XX=X+H                                                            AXS03300
      YY=-HALF H                                                        AXS03400
C     CALL NUMBER(XX,YY,H,JY1,0.,NFORMAT)                               AXS03500
      N=JY1+JDY                                                         AXS03600
```

```
      YY=YY+DY                                                          AXS03700
      CALL PLOT(X,0.,3)                                                 AXS03800
      Y=DY                                                              AXS03900
      XDT=X+DT                                                          AXS04000
      DO 200 J=2,NY                                                     AXS04100
      CALL PLOT(X,Y,2)                                                  AXS04200
      CALL PLOT(XDT,Y,2)                                                AXS04300
C     CALL NUMBER(XX,YY,H,N,0.,NFORMAT)                                 AXS04400
      CALL PLOT(X,Y,3)                                                  AXS04500
      N=N+JDY                                                           AXS04600
      YY=YY+DY                                                          AXS04700
  200 Y=Y+DY                                                            AXS04800
      CALL PLOT(X,Y,2)                                                  AXS04900
C     CALL NUMBER(XX,YY,H,N,0.,NFORMAT)                                 AXS05000
C*****   THE NEXT SECTION DRAWS AND NUMBERS (IF SPECIFIED) THE UPPER X-AXIS.
      NO NUMBER=1
      XX=X-TWO H                                                        AXS05100
      YY=Y+D-H                                                          AXS05200
      IF(NH.LT.0)CALL NUMBER(XX,YY,H,IX2,0.,NFORMAT)                    AXS05300
      N=IX2*IDX                                                         AXS05400
      XX=XX+DX                                                          AXS05500
      CALL PLOT(X,Y,3)                                                  AXS05600
      X=X+DX                                                            AXS05700
      XDT=X+DT                                                          AXS05800
      DO 300 I=2,NX                                                     AXS05900
      CALL PLOT(X,Y,2)                                                  AXS06000
      CALL PLOT(X,YDT,2)                                                AXS06100
      IF(NH.LT.0.AND.NO NUMBER.NE.1)CALL NUMBER(XX,YY,H,N,0.,NFORMAT)
      NO NUMBER=-NO NUMBER
      CALL PLOT(X,Y,3)                                                  AXS06300
      N=N+IDX                                                           AXS06400
      XX=XX+DX                                                          AXS06500
  300 X=X+DX                                                            AXS06600
      CALL PLOT(X,Y,2)                                                  AXS06700
      IF(NH.LT.0)CALL NUMBER(XX,YY,H,N,0.,NFORMAT)                      AXS06800
C*****   THE NEXT SECTION DRAWS AND NUMBERS THE LEFT-HAND Y-AXIS.
      XX=-5.*H                                                          AXS06900
      YY=Y-HALF H                                                       AXS07000
      CALL NUMBER(XX,YY,H,JY2,0.,NFORMAT)                               AXS07100
      N=JY2*JDY                                                         AXS07200
      YY=YY+DY                                                          AXS07300
      CALL PLOT(X,Y,3)                                                  AXS07400
      NO NUMBER=1
      Y=Y+DY                                                            AXS07500
      DO 400 J=2,NY                                                     AXS07600
      CALL PLOT(0.,Y,2)                                                 AXS07700
      CALL PLOT(DT,Y,2)                                                 AXS07800
      IF(NO NUMBER.NE.1)CALL NUMBER(XX,YY,H,N,0.,NFORMAT)
      NO NUMBER=-NO NUMBER
      CALL PLOT(0.,Y,3)                                                 AXS08000
      N=N+JDY                                                           AXS08100
      YY=YY+DY                                                          AXS08200
  400 Y=Y+DY                                                            AXS08300
      CALL PLOT(0.,0.,2)                                                AXS08400
      CALL NUMBER(XX,YY,H,N,0.,NFORMAT)                                 AXS08500
C*****   THE NEXT THREE SECTIONS WRITE THE TITLE AND AXIS LABELS.
```

```
      IF(LABEL.EQ.0)501,499                                             AXS08600
  499 NL=MAX                                                            AXS08700
      DECODE(MAX,900,LABEL)ICHAR                                        AXS08800
      DO 500 I=1,NL                                                     AXS08900
      IF(ICHAR(I).EQ.1HS)NL=I-1                                         AXS09000
  500 CONTINUE                                                          AXS09100
      XX=.5*ILG=.015*NNH*NL                                             AXS09200
      YY=JHGT*H
      IF(NH.GT.0.AND.JHGT.GT.2)YY=.9*JHGT
      IF(NH.LT.0)YY=-TWO H
      CALL SYMBOL(XX,YY,H,LABEL,0.,NL)                                  AXS09300
  501 IF(LABELX.EQ.0)601,599                                            AXS09400
  599 NX=MAX A                                                          AXS09500
      DECODE(MAXA,900,LABEL X)(ICHAR(I),I=1,MAXA)                       AXS09600
      DO 600 I=1,NX                                                     AXS09700
      IF(ICHAR(I).EQ.1HS)NX=I-1                                         AXS09800
  600 CONTINUE                                                          AXS09900
      XX=.5*ILG=.015*NNH*NX                                             AXS10000
      IF(NH.GT.0)YY=-TWO D                                              AXS10100
      IF(NH.LT.0)YY=JHGT+TCH
      CALL SYMBOL(XX,YY,H,LABELX,0.,NX)                                 AXS10200
  601 IF(LABELY.EQ.0)701,699                                            AXS10300
  699 NY=MAXA                                                           AXS10400
      DECODE(MAXA,900,LABEL Y)(ICHAR(I),I=1,MAXA)                       AXS10500
      DO 700 I=1,NY                                                     AXS10600
      IF(ICHAR(I).EQ.1HS)NY=I-1                                         AXS10700
  700 CONTINUE                                                          AXS10800
      XX=-7.*H                                                          AXS10900
      YY=.5*JHGT=.428571*H*NY                                           AXS11000
      CALL SYMBOL(XX,YY,H,LABELY,90.,NY)                                AXS11100
  701 RETURN                                                            AXS11200
  900 FORMAT(32A1)                                                      AXS11300
      END                                                               AXS11400
```