CIRCULATION COPY
SUBJECT TO RECALL
IN TWO WEEKS

UCID-19030

BIFUR III

A Program for Calculating Borehole and Surface Gravity Caused by Two-dimensional Structure in Cartesian or Cylindrical Coordinates

Joseph R Hearst

May 4, 1981

Lawrence Livermore Laboratory

This is an informal report intended primarily for internal or limited external distribution. The opinions and conclusions stated are those of the author and may or may not be those of the Laboratory.

Work performed under the auspices of the U.S. Department of Energy by the Lawrence Livermore Laboratory under Contract W-7405-Eng-48.

DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial products, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement recommendation, or favoring of the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

BIFUR III

A Program for Calculating Borehole and Surface Gravity Caused by Two-dimensional Structure in Cartesian or Cylindrical Coordinates

Joseph R Hearst

ABSTRACT

An interactive computer program, BIFUR III, has been written. It computes the gravitational effect of two-dimensional structures bounded by planes (Program BIFUR3) or co-axial cylindrical surfaces (Program BIFURC). The calculated gravimetric density can be compared to density calculated from measurements of gravity in a borehole and from a density log. The user can change the structure online, and compare the results of different changes. The primary application is to borehole gravimetry, but surface gravity problems can also be investigated.

BIFUR III

A Program for Calculating Borehole and Surface Gravity Caused by Two-dimensional Structure in Cartesian or Cylindrical Coordinates

Joseph R Hearst

DESCRIPTION

Borehole gravity data are often affected by material not immediately adjacent to the hole. We have studied these effects (1,2) and developed a computer program to calculate them in the case of beds that are infinite in one horizontal direction (that is, in Cartesian coordinates). That program, BIFUR, used equations developed by Elkins (3) which required that the beds have vertical ends. This requirement made the input cumbersome and inconvenient.

At the suggestion of P. Kasameyer, a new program, BIFUR II,(4) was written.BIFUR II also calculated the effect of beds that were infinite in one horizontal dimension, but used Talwani's (5) method of calculating the effect of a polygon that is infinite in one horizontal dimension. Therefore it permitted the shape of the cross section that was used for input to be arbitrary as long as it was made up of straight lines. R. Carlson made a further suggestion that greatly simplified input. He pointed out that it was not necessary to specify the density of each polygon in the cross section, but only the density contrast across each line.

BIFUR II has now been updated to permit the user to modify the cross section while the program is being run. The new version, BIFUR III, also plots the cross section, displaying the density in each polygon. It also calculates a structure correction to the measured gravimetric density. This correction term is the difference between the density in the polygon in which a pair of measuring stations is located and the density obtained from the calculated gravity difference between the pair of measuring stations . It then plots the measured gravimetric density vs depth and overplots the corrected measured gravimetric density and the density from a density log.This permits the user to see how well the correction to the measured gravity calculated from the cross section reproduces the log density.

Following a further suggestion from Kasameyer, another version of the code, BIFURC, calculates gravimetric density from a cross section described by cylindrical symmetry. In this case, the cross section is again described by lines, but instead of being infinite in the third dimension, the cross section is rotated about an axis of symmetry at the borehole. The code uses an expression derived by Talwani (6) for the gravitational effect of a vertical cone.

The input consists of a title, a cross section, and either a series of depths between which gravimetric density is to be calculated and a series of surface points at which gravity is to be calculated, or the output from a code that calculates measured gravity vs depth and from another that calculates density from a density log.

The output consists of printout describing the cross section and the calculated gravity and gravimetric density, the plot of the cross section, the

plot comparing the measured gravimetric density with the corrected measured gravimetric density and the density from a density log (if these have been input), and two files for later use as input.

INPUT

The first card of the input file is the title card, and may have up to 80 columns. Certain words, used as mnemonics in the code, are forbidden in the title; They are : LOG, DPT, DEPTH, LOGS, SUR, SURF,ADD,DEN,END,GO, EOF, FILE, INT, ZEP, ZPT, and TTY.

The next set of cards describe the cross section. They will be described for the Cartesian case. All input is in meters. The input must describe closed polygons, and ordinarily they should fill the entire space. Left and right hand boundaries will be described below. All polygons must be continuous; that is, they must be such that one can draw a line all around the polygon without lifting one's pencil from the paper. This implies that there can be no polygon completely enclosed in another with no common boundary, because then the outer boundary of the inner polygon would be a boundary of the outer polygon but not continuous with it. This constraint exists because the code must be able to identify each polygon so that its center can be located and information about it plotted. (The constraint is not required for the calculation of the gravity itself; if the constraint is not acceptable BIFUR II can be used .) Also, each intersection of two or more lines must be specified at least twice, once for each line that meets at that intersection. This requirement ensures that there are no unconnected lines (the code checks for it.)

A convenient way to prepare the input is to draw the cross section on graph paper, or draw it with the coordinates of each intersection of lines labelled with the X and Z values; and with all the densities written in the polygons. Such a sketch is shown in Fig. 1. In the figure the boundaries of the polygons, (or bed boundaries) are numbered. These numbers will be listed after the card that describes each boundary.

There are two kinds of cards describing lines in the cross section: scarp cards and bed-boundary cards. Figure 1 and Table 1 illustrate these cards. The first line in Table 1 is the title card. The next two are scarp cards, describing the two scarps shown as dashed lines on Fig. 1. The scarps are optional, and are not boundaries of polygons, but any point on a scarp can be used as a point on a bed boundary by specifying the scarp and either a depth or horizontal position.

Both scarps and bed-boundaries are described by five blank-delimited items of information. If there are fewer than five data, the card will be misinterpreted. The five items or data are called XA, ZA, XB,ZB, and RHO. A scarp is distinguished by the fifth datum, RHO, being a letter ,rather than a number. This letter, which may be any letter but R or L, is the "scarp letter", and must be different for each scarp. XA and ZA are the horizontal position (X) and the vertical position (Z) of a point on the scarp. Z is positive downwards, and X is positive to the right of the borehole and negative to the left of the borehole. ANY point on the scarp may be specified by XA,ZA; it need not be within the boundaries of the problem. The slope of the scarp may be specified by the data XB,ZB in two ways. The first is by letting XB,ZB be another point on the scarp line. The second is by letting XB be the angle of the scarp with the horizontal between 90 and -90 degrees

(positive in the first quadrant-- that is, to the right of the vertical and above the horizontal.) In that case, ZB should be a letter-- any letter. It is recommended, but not mandatory, that all scarps be input before any bed boundaries. It is mandatory that a scarp be input before any bed that it terminates. The second line in Table 1:

-650 0 -250 1200 A

describes a scarp (Scarp A) with points at X=-650, Z=0. and X= -250, Z=1200 and the third line

420 0 85 Q B

describes another scarp (scarp B) with a point at X=420, Z=0 and a slope of 85 degrees above the horizontal. These scarps are shown in Fig. 1.

The bed boundaries (or polygon boundaries) are normally specified by two X,Z pairs, XA,ZA and XB,ZB . These specify the vertical and horizontal positions of the ends of the boundary. RHO is the difference in density between the right and left sides of the boundary, with "right" defined for an observer standing at XA,ZA and facing in the direction of XB,ZB. That is, RHO is the density on the right side of the boundary minus that on the left side of the boundary. If XA or XB is the letter L, that X-value will be taken as -10^7, if it is the letter R, the X value will be $+10^7$. This permits specifying the boundaries of the problem. It is not necessary to input vertical lines at the left and right boundaries of the problem; the code will supply them.

If either the X or the Z of an X,Z pair is a letter (except L or R),instead of a number ,that letter will be interpreted as a scarp letter. Then the missing value will be calculated as the point on the scarp described by the existing value ; i.e., if X is a number, the Z will be calculated as the value of the Z on the scarp corresponding to the input X. Both ends of a bed-boundary can be described by scarp lines, but you cannot replace both X and Z of a pair by letters as the code will have no way to decide where to go.

The top and bottom of the problem must always be described by bed-boundaries going from L to R (they need not be horizontal.) Otherwise the code cannot compute the densities in the polygons. And the densities in a vertical column from outside the bottom boundary to outside the top boundary must always sum to zero. They will do so automatically if you assume the density above the top boundary and below the bottom boundary to be equal to zero and input RHO accordingly.

There is no required order of the bed-boundary cards and no requirement that the XA,ZA point be the left-hand point of the line, but it is usually convenient to work from left to right.

The next 13 lines in Table 1 are bed-boundary cards, shown with the corresponding line number from Fig. 1.

L 840 A 840 1.2	(1)
A 840 A 910 1.2	(2)
A 910 -286 A .7	(3)

Note that an X-value is specified instead of a Z- value at the second intersection.

```
A 910 B 1015 .5          (4)
A 1120 B 1070 .7         (5)
```

Note that the point "A 1120" is the same as point "-286 A" for line (3).

```
B 1070 B 1015 .7         (6)
B 1015 B 930 1.2         (7)
B 930 R 930 1.2          (8)
```

These complete the lines terminated by scarps.

```
L 500 -700 500 .4        (9)
-700 500 1000 400 .4     (10)
1000 400 R 400 .4        (11)
```

describe the crooked boundary going from left to right near the top and not terminted by the scarp lines. Finally

```
L 0 R 0 1.1              (12)
L 1200 R 1200 -2.7       (13)
```

are the upper and lower boundaries of the problem.

The cross-section input for the cylindrical geometry is identical to that for Cartesian geometry, except that no negative values of X are permitted, and an input X of L is identical to an input X of 0, and signifies the axis of symmetry. Input for a cylindrical problem is shown in Table 2

To summarize; a scarp line is recognized by its fifth datum being a letter; it is then identified by that letter. The first two data describe one point on the scarp line, the second pair either another point or , if the fourth datum is a letter, the angle of that line with respect to the horizontal. Bed-boundary lines are described by four data points representing the two ends and a fifth datum describing the density contrast across the line. Any, but not all, of the first four may be letters. If the letters are L or R they specify the left and right boundaries of the problem, otherwise they specify intersections with the denoted scarp line. The maximum number of scarps is 24, of boundaries 100 and of polygons 50, with 25 boundaries maximum for each..

The input of the cross section description is terminated by a line with fewer than five data, and with the first datum alphabetic. This datum is a mnemonic and may be either DPT, SUR, or LOG. If it is DPT, as in Table 1, then the next cards contain space-delimited lists of depth points (up to 100). If,either without DPT or after DPT the mnemonic SUR is on a card, the next cards consist of space-delimited lists of horizontal surface positions (with the borehole at zero) at which gravity is to be calculated. A blank card then ends the problem.

If the mnemonic is LOG, the next two data on the card must be the names of two files in memory. Table 3 shows an input set with the LOG mnemonic. The first of the two files (a typical one is shown in Table 4) is the output

from a code that calculates density from the data from a borehole gravimeter, and has three lines of header and then data in format E12.8, 3E13.8, consisting of two depths (in feet) and the density calculated from the gravity measured between those depths. The second file, (shown in Table 5) is the output from a density-log code ,and has 13 lines of header, and then, in format E12.8,5E13.8 has the depth (again in feet) in the first field and the integral of the density in the sixth field. (The intervening fields are not used by BIFUR.) This integral is the sum of the products of the density at the lower of two depths and the distance between those depths displayed at the lower depth. If the density between two depths is zero, then the integral does not change. This,then, completes the input to BIFUR.

OUTPUT

The output consists both of plots and of printout. The plots are compatible with the LLNL UX80 system. The code prints out the name of the plot file on the teletype, and they can be observed with the routine UXTV (q.v.) obtained from MJBLIB. Figure 2 is the plot obtained from the input in Table 1, and is essentially identical to Figure 1. (However, BIFUR renumbers the lines.) Near the center of each polygon are two numbers, an integer and a decimal number. The integer is the polygon number (in the code this is called a volume number.) The decimal number is the density of the material in the polygon. Near each intersection is another integer. This is the intersection number.(Figure 3 is a plot of the output from the cylindrical input in Table 2 .)

Table 6 is the printout from the problem described by the input in Table 1 and Figures 1 and 2. It is file HSPO. It is divided into four sections. The first, at the top, lists all the lines in the problem. First is the line number, then the X and Z coordinates of the left end of the line (if you input the right end first,the code switches them), and then the number of the intersection at the left end of the line. Next are the X, Z and intersection number for the right end of the line. The intersection numbers are those shown in the plot. Next are the volume numbers or polygon numbers associated with the left and right sides of the line (looking from its left end to its right end) and finally the density contrast across the line.

The next section of the output describes the volumes, or polygons. First is the number of the polygon ,or volume number, and then a list of all the lines bounding it; ten on the first line of the output, and another line of output if more are required. Next is the density in the polygon and finally the approximate X and Z positions of its midpoint .

The third part of the output is the calculated gravimetric densities. The first line, designated by VAG, is the change in calculated density that would be caused by correcting the vertical air gradient, or free air gradient, used in calculating the density in the borehole for the effect of the cross section. The other lines show first the two depths of the gravity stations, then the density that was calculated between those stations (RHO), and then the density in the polygon in which the stations lie (or the average between them if they straddle a boundary) (RHO NOM) Next are the corrected density (only shown if the measured gravimetric density is input by a LOG) and finally the correction to the density: RHO NOM - RHO -VAG . This correction is added to the measured gravimetric density to get RHO CORR. The last column is G, the calculated gravity at the lower of the two depths.

The last part of the printout is the surface gravity, and merely shows the horizontal position, the gravity, and the gravity difference between stations.

There are two other BCD files output. They are called BIFINP and BIFOUT. BIFINP, shown in Table 7, is a restart file. Each time a new version of the cross section is produced by interaction (see below) another set of input is put into BIFINP. There are no depth data in BIFINP. To use BIFINP for input to a new problem, use TRIX to select the set of input you desire, and then add either DPT or LOG cards at the end. BIFOUT is formatted for storage in the K-division data base, and is shown in Table 8.

If LOG input is used, an additional output plot is produced. The cross section plot produced by the input in Table 3 is given in Figure 4. It shows another feature of the cross section plot; the ability to have pointers to small polygons. The second plot, shown in Fig. 5,is the density plot. There are three sets of vertical line segments on this plot. The solid lines represent the density calculated from measured borehole gravity. The closely-dotted lines (labelled LOG) represent the average density from the gamma-gamma density log. The lines with dots having greater spacing (Labelled CORR 0) represent the corrected gravimetric density described above. These plots permit the user to see how well the input cross section accounts for the difference between the gravimetric density and the log density.

The last kind of output from BIFUR are error messages on the teletype. I have tried (with indifferent success) to make them self-explanatory. If you get an error message, try to look it up in the listing. Most of them will come from subroutine VOLUMES or CHECK, and usually the place in the code at which the error occurred is listed below the GO TO statement. If the code throws you off the machine with the error, look at the printout HSPO , which will be present and will contain results of all the calculations made thus far. This may help you understand the error.

INTERACTION

One can modify the cross section in three ways: by moving an intersection, by adding a constant to all X-values, and by changing the density in a polygon. When you have executed BIFUR the teletype will print

PLOT FILE IS BIFA80

This file can be viewed with UXTV. At LLNL we can view the output while we also interact with BIFUR. It next types:

TYPE -DEN-INT-ADD-, THEN GO----OR END

and then two "greater than" symbols. It then is ready to accept changes; as many as you like. The first problem described above has been changed by each of the methods, and the results are shown below.

INT moves an intersection. You type

INT NNN X Z

where NNN is the intersection number and X,Z the new position of the intersection. You may make as many changes as you like (and mix the types of changes) each time you interact. But if you make a change in the position of

an intersection such that you produce an impossible cross-section, the code will complain. In some cases it will throw you off and you will have to restart from your input or BIFINP, in some it will simply again type "greater than" signs. In the latter case, it has forgotten your changes but is ready for more.

After all changes are typed, type GO, and the problem will run again, and tell you

PLOT FILE IS BIFB80

Fig. 6 shows the revised output from the first simple problem (Figs 1,2) after typing

INT 6 200 900

thus moving intersection number 6 to position 200,900.
In all cases

ADD NNN

adds that value to all X. For example, the next change was

ADD 200

This added 200 to all X values, and the result is shown in Fig. 7.
Finally,

DEN NNN RHO

changes the density of polygon NNN to RHO. Fig.8 shows the result of

DEN 2 3.

When interaction is performed while using the LOG mode, two plots are produced each time. Figure 9 shows the result of

DEN 6 2.15.

on the complicated Cartesian problem described in Table 3 and Figure 4.A new corrected density, labelled CORR 1, is plotted as a dotted line with a dot pattern different from the first. This permits the user to overlay the plots from successive changes using UXEDIT. Any time interactive changes are made in the LOG mode, new density plots are produced. However ,after a while the dot pattern becomes too sparse to see.

THE MOVIE VERSION

There is a version of both BIFUR3 and BIFURC that produces color plots for movie frames. In that version both the cross section and density plots are output on one frame, with color corresponding to density in the cross section. The input consists of two cross sections and the code interpolates between them an input number of times to produce frames for a movie showing changes in calculated gravimeteric density as the cross section changes. Depths are calculated by the code, and LOG input is not used.

LISTINGS

The listing of BIFUR3 and all of the subroutines of BIFURC not identical to those in BIFUR3 follow the figures and tables. The program is written in LRLTRAN, an LLNL version of FORTRAN. The subroutine DATA and about half of the main code was written by D. Burton. The library subroutines called are in LLNL's ORDERLIB, TV80LIB, and UX80LIB, available on the LLNL 7600 computers.

REFERENCES

1. J. R Hearst and H.L.McKague: "Structure Elucidation with Borehole Gravimetry", UCRL 76316 (for listing), Geophysics v 41 p 491-505 (1976)
2. J. R Hearst :"Estimation of Dip and Lateral Extent of Beds with Borehole Gravimetry" Geophysics v 42, p 990-994 (1977)
3. T.A. Elkins, Personal Communication
4. J.R Hearst :" BIFUR II, A Program for Calculating Borehole Gravity Caused by Two-Dimensional Structure" UCID 17852 (1978)
5. M. Talwani, J. Worzel and M. Landisman: "Rapid Gravity Computations for Two-dimensional Bodies with Application to the Mendocino Submarine Fracture Zone " Jour. Geop. Res. v 64 p 49
6. M.Talwani :" Computer Usage in the Computation of Gravity Anomalies" in Methods of Computational Physics 13, Edited by B.Bolt pp 343-389 ,Academic, New York, 1973.

TABLE 1 -Input for Problem described by Figs 1 and 2

```
TEST PROBLEM FOR DEMONSTRATION
-650 0 -260 1200 A
420 0 85 Q B
L 840 A 840 1.2
A 840 A 910 1.2
A 910 -286 A .7
A 910 B 1015 .5
A 1120 B 1070 .7
B 1070 B 1015 .7
B 1015 B 930 1.2
B 930 R 930 1.2
L 500 -700 500 .4
-700 500 1000 400 .4
1000 400 R 400 .4
L 1200 R 1200 -2.7
L 0 R 0 1.1
DPT
1000 950 900 850 800 750 700
650 600 550 500 450 400 350
300 250 200
```

TABLE 2 Input for problem described by Fig. 3

```
CYLINDRICAL DEMONSTRATION
L 0 R 0 1
L 100 500 150 1
500 150 700 350 1
700 350 0 500 1
L 1000 R 1000 -1
DPT
100 200 300 400 500 600 700 800
```

TABLE 3 Input for LOG problem described by Fig. 4

```
U2FE/GIN1   ORIGINAL CROSS SECTION FROM PROSPECTUS
-405 0 30 1200 A
-205 0 228 1200 B
500 0 938 1200 C
L 0 R 0 2.08
L 363 -555 363 .57
-555 363 A 380 .57
A 380 B 472 -.33
B 472 C 555 -.21
C 555 C 588 -.21
C 588 1505 536 -.21
1505 536 R 536 -.21
A 380 A 465 .90
A 465 B 472 -.20
A 465 A 485 1.10
A 485 B 500 .20
B 500 B 472 .32
B 500 B 625 -.12
A 485 A 610 .90
A 610 B 625 .90
B 625 B 823 .78
B 823 C 905 .78
C 905 1505 878 .78
1505 878 R 878 .78
L 1200 R 1200 -2.65
LOG MOROUT U2FE/DEN3
```

TABLE 4 File of gravimeter output used as input for LOG-type problem

U2FE GRAVITY BH RUN 1 4-7-80 LLL-N TERRAIN BY HIGUERA

1ST DEPTH	2ND DEPTH	DENSITY	CORR. DENS 0.09200000
1.4500E+03	1.4000E+03	2.0243E+00	2.0243E+00
1.4000E+03	1.3500E+03	2.1138E+00	2.1138E+00
1.3500E+03	1.3000E+03	1.9980E+00	1.9980E+00
1.3000E+03	1.2500E+03	2.0302E+00	2.0302E+00
1.2500E+03	1.2000E+03	2.0287E+00	2.0287E+00
1.2000E+03	1.1500E+03	2.0890E+00	2.0890E+00
1.1500E+03	1.1000E+03	2.0678E+00	2.0678E+00
1.1000E+03	1.0500E+03	2.0323E+00	2.0323E+00
1.0500E+03	1.0000E+03	2.0988E+00	2.0988E+00
1.0000E+03	9.0000E+02	2.1271E+00	2.1271E+00
9.0000E+02	8.0000E+02	2.1689E+00	2.1689E+00
8.0000E+02	7.0000E+02	2.1627E+00	2.1627E+00
7.0000E+02	6.0000E+02	2.1664E+00	2.1664E+00
6.0000E+02	5.0000E+02	2.1390E+00	2.1390E+00
5.0000E+02	4.0000E+02	2.1764E+00	2.1764E+00
4.0000E+02	3.0000E+02	2.1483E+00	2.1483E+00
3.0000E+02	2.0000E+02	2.1047E+00	2.1047E+00
2.0000E+02	1.0000E+02	2.0155E+00	2.0155E+00
1.0000E+02	1.0000E+01	1.8046E+00	1.8046E+00

TABLE 5

File of density log output used as input in LOG-type problem
(First page only)


```
U2FE   DENSITY BH   RUN 3   2-27-80
 DENSITY   VERSION 12   COMPILED 3-3-77
 RUN BY:  NORMAN          U 04/08/80  14:25:03
 LOG-TYPE:   DENSITY BH          SOURCE :     COBALT-60
 FLUID LEVEL:    1460.0
 DEAD TIME:   0.00083 MILLIMINUTES,            DEPTH-CONSTANT:     1.00
 DENSITY CALIBRATION VALUES:        1.740     1.720     2.620     2.520
 KILOCOUNT CALIBRATION VALUES:       10.470     3.510
   U2FE/PX3 --  PROXIMITY FILE     2949 PTS       74 DUP        0 REVERSALS
   U2FE/KC3 --  KILOCOUNT FILE
  U2FE/DEN3 --  OUTPUT FILE                                STRIP-LEVEL = 1.00
```

DEPTH..FT	PROX..IN	RAW DENS	DENS..GM/CC	STRIPPED	INTEGRATION
1.05969E+02	0.	1.210765E+00	0.	0.	0.
1.06964E+02	5.396452E-01	1.573364E+00	1.872504E+00	1.872504E+00	9.315706E-01
1.08363E+02	7.533938E-01	1.414780E+00	1.802228E+00	1.802228E+00	3.502045E+00
1.08634E+02	6.911721E-01	1.420700E+00	1.759195E+00	1.759195E+00	3.984618E+00
1.10224E+02	3.829258E-01	1.634936E+00	1.840138E+00	1.840138E+00	6.846087E+00
1.10494E+02	3.483201E-01	1.651265E+00	1.837373E+00	1.837373E+00	7.342551E+00
1.11852E+02	3.029331E-01	1.676396E+00	1.838438E+00	1.838438E+00	9.838427E+00
1.12118E+02	2.400228E-01	1.720612E+00	1.850677E+00	1.850677E+00	1.032908E+01
1.12390E+02	2.062452E-01	1.720582E+00	1.829901E+00	1.829901E+00	1.082964E+01
1.13765E+02	4.270505E-01	1.650940E+00	1.892544E+00	1.892544E+00	1.338882E+01
1.15635E+02	2.657972E-01	1.907530E+00	2.087431E+00	2.087431E+00	1.711010E+01
1.16180E+02	2.047534E-01	1.907452E+00	2.040567E+00	2.040567E+00	1.823498E+01
1.18948E+02	4.086353E-01	1.634090E+00	1.856995E+00	1.856995E+00	2.362920E+01
1.19221E+02	4.477110E-01	1.634064E+00	1.885579E+00	1.885579E+00	2.414006E+01
1.20808E+02	1.530827E-01	1.969102E+00	2.071307E+00	2.071307E+00	2.727985E+01
1.21085E+02	1.225700E-01	1.931010E+00	2.008451E+00	2.008451E+00	2.784490E+01
1.23024E+02	2.900126E-01	1.710358E+00	1.870846E+00	1.870846E+00	3.160588E+01
1.23570E+02	3.870875E-01	1.710299E+00	1.939697E+00	1.939697E+00	3.264615E+01
1.23836E+02	4.178037E-01	1.747228E+00	2.012065E+00	2.012065E+00	3.317174E+01
1.24383E+02	3.335591E-01	1.737743E+00	1.934415E+00	1.934415E+00	3.425110E+01
1.26089E+02	5.836572E-01	1.396031E+00	1.643324E+00	1.643324E+00	3.730291E+01
1.26640E+02	7.824327E-01	1.373324E+00	1.754355E+00	1.754355E+00	3.823897E+01
1.27742E+02	8.679601E-01	1.329984E+00	1.751010E+00	1.751010E+00	4.017043E+01
1.29294E+02	4.115671E-01	1.683209E+00	1.923259E+00	1.923259E+00	4.302166E+01
1.29300E+02	4.105956E-01	1.718732E+00	1.968896E+00	1.968896E+00	4.303334E+01
1.29839E+02	3.802372E-01	1.718672E+00	1.945205E+00	1.945205E+00	4.408819E+01
1.30940E+02	4.757534E-01	1.657536E+00	1.939400E+00	1.939400E+00	4.622666E+01
1.31995E+02	3.789076E-01	1.893397E+00	2.167051E+00	2.167051E+00	4.839282E+01
1.34797E+02	6.071410E-01	1.461741E+00	1.759024E+00	1.759024E+00	5.389325E+01
1.35847E+02	5.204458E-01	1.657045E+00	1.976348E+00	1.976348E+00	5.585432E+01
1.36389E+02	2.555379E-01	1.673875E+00	1.805813E+00	1.805813E+00	5.687929E+01
1.36932E+02	2.820795E-01	1.691120E+00	1.842690E+00	1.842690E+00	5.786985E+01
1.38271E+02	2.480267E-01	1.847292E+00	2.003267E+00	2.003267E+00	6.044472E+01
1.38544E+02	2.367597E-01	1.847256E+00	1.995035E+00	1.995035E+00	6.099049E+01
1.39932E+02	2.988307E-01	1.682105E+00	1.842674E+00	1.842674E+00	6.365386E+01
1.40756E+02	3.742621E-01	1.648267E+00	1.851182E+00	1.851182E+00	6.517573E+01
1.41592E+02	4.064099E-01	1.555700E+00	1.753391E+00	1.753391E+00	6.668244E+01
1.41864E+02	4.807158E-01	1.555677E+00	1.803146E+00	1.803146E+00	6.716613E+01
1.42396E+02	6.918403E-01	1.623820E+00	2.095111E+00	2.095111E+00	6.820307E+01
1.43500E+02	6.740597E-01	1.555535E+00	1.964550E+00	1.964550E+00	7.044400E+01
1.44034E+02	7.505249E-01	1.615754E+00	2.150879E+00	2.150879E+00	7.154282E+01
1.45724E+02	6.481022E-01	1.372048E+00	1.648592E+00	1.648592E+00	7.475337E+01
1.47058E+02	5.573557E-01	1.493099E+00	1.769255E+00	1.769255E+00	7.703307E+01
1.47333E+02	5.061298E-01	1.479985E+00	1.715151E+00	1.715151E+00	7.751218E+01
1.47605E+02	5.561494E-01	1.486479E+00	1.758687E+00	1.758687E+00	7.798462E+01
1.50257E+02	1.415194E-01	1.926702E+00	2.016789E+00	2.016789E+00	8.299090E+01
1.50258E+02	1.415807E-01	1.914541E+00	2.003603E+00	2.003603E+00	8.299291E+01

TABLE 6

Output file (HSPO) From first problem (Fig 1,2, Table 1)

OUTPUT W. FORM

1 TEST PROBLEM FOR DEMONSTRATION

LINE	LEFT X	LEFT Z	LFT INT	RT X	RT Z	RT INT	VOL ON L	VOL ON R	DELTA RHO
1	-1.0000E+07	8.4000E+02	1	-3.7700E+02	8.4000E+02	2	1	4	1.200
2	-3.7700E+02	8.4000E+02	2	-3.5425E+02	9.1000E+02	3	1	4	1.200
3	-3.5425E+02	9.1000E+02	3	-2.8600E+02	1.1200E+03	4	2	4	0.700
4	-3.5425E+02	9.1000E+02	3	3.3091E+02	1.0150E+03	6	1	2	0.500
5	-2.8600E+02	1.1200E+03	4	3.2609E+02	1.0700E+03	5	2	4	0.700
6	3.2609E+02	1.0700E+03	5	3.3091E+02	1.0150E+03	6	2	4	0.700
7	3.3091E+02	1.0150E+03	6	3.3837E+02	9.3000E+02	7	1	4	1.200
8	3.3837E+02	9.3000E+02	7	1.0000E+07	9.3000E+02	13	1	4	1.200
9	-1.0000E+07	5.0000E+02	8	-7.0000E+02	5.0000E+02	9	3	1	0.400
10	-7.0000E+02	5.0000E+02	9	1.0000E+03	4.0000E+02	10	3	1	0.400
11	1.0000E+03	4.0000E+02	10	1.0000E+07	4.0000E+02	14	3	1	0.400
12	-1.0000E+07	1.2000E+03	11	1.0000E+07	1.2000E+03	15	4	0	-2.700
13	-1.0000E+07	0.	12	1.0000E+07	0.	16	0	3	1.100

VOL	L1	L2	L3	L4	L5	L6	L7	L8	L9	L10	RHO	XMD	ZMD
1	1	2	4	7	8	9	10	11	0	0	1.500	1.6500E+02	-7.1935E+02
2	3	4	5	6	0	0	0	0	0	0	2.000	-1.1669E+01	-1.0300E+03
3	9	10	11	13	0	0	0	0	0	0	1.100	1.6500E+02	-2.2456E+02
4	1	2	3	5	6	7	8	12	0	0	2.700	1.6500E+02	-1.1416E+03

DEPT 1	DEPT 2	RHO	RHO NOM	RHO CORR	CORR	
VAG					-0.0202	
250.0	200.0	1.1318	1.1000	0.	-0.0116	5.8333E+01
300.0	250.0	1.1356	1.1000	0.	-0.0154	5.3573E+01
350.0	300.0	1.1400	1.1000	0.	-0.0198	4.8794E+01
400.0	350.0	1.1452	1.1000	0.	-0.0250	4.3994E+01
450.0	400.0	1.1513	1.1000	0.	-0.0312	3.9167E+01
500.0	450.0	1.4868	1.4294	0.	-0.0372	3.2935E+01
550.0	500.0	1.5657	1.5000	0.	-0.0455	2.6371E+01
600.0	550.0	1.5757	1.5000	0.	-0.0556	1.9766E+01
650.0	600.0	1.5875	1.5000	0.	-0.0673	1.3111E+01
700.0	650.0	1.6009	1.5000	0.	-0.0807	6.4003E+00
750.0	700.0	1.6158	1.5000	0.	-0.0956	-3.7320E-01
800.0	750.0	1.6317	1.5000	0.	-0.1115	-7.2131E+00
850.0	800.0	1.6474	1.5000	0.	-0.1273	-1.4119E+01
900.0	850.0	1.6616	1.5000	0.	-0.1415	-2.1085E+01
950.0	900.0	1.6725	1.5000	0.	-0.1524	-2.8096E+01
1000.0	950.0	2.0273	1.8571	0.	-0.1500	-3.6594E+01

NOTE: CORR= RHO NOM - RHO -VAG CORR

X-POSITION	G	DG
1.200000E+03	8.229575E+01	0.
0.	8.193583E+01	-3.599177E-01
2.000000E+02	8.184503E+01	-9.080588E-02
4.000000E+02	8.186605E+01	2.102131E-02
6.000000E+02	8.196144E+01	9.538564E-02
8.000000E+02	8.208667E+01	1.252351E-01
1.000000E+03	8.220493E+01	1.182632E-01

TABLE 7

Output file BIFINP from first problem. Used for restarting


```
TEST PROBLEM FOR DEMONSTRATION
-1.0000E+07  8.4000E+02 -3.7700E+02  8.4000E+02  1.2000E+00
-3.7700E+02  8.4000E+02 -3.5425E+02  9.1000E+02  1.2000E+00
-3.5425E+02  9.1000E+02 -2.8600E+02  1.1200E+03  7.0000E-01
-3.5425E+02  9.1000E+02  3.3091E+02  1.0150E+03  5.0000E-01
-2.8600E+02  1.1200E+03  3.2609E+02  1.0700E+03  7.0000E-01
 3.2609E+02  1.0700E+03  3.3091E+02  1.0150E+03  7.0000E-01
 3.3091E+02  1.0150E+03  3.3837E+02  9.3000E+02  1.2000E+00
 3.3837E+02  9.3000E+02  1.0000E+07  9.3000E+02  1.2000E+00
-1.0000E+07  5.0000E+02 -7.0000E+02  5.0000E+02  4.0000E-01
-7.0000E+02  5.0000E+02  1.0000E+03  4.0000E+02  4.0000E-01
 1.0000E+03  4.0000E+02  1.0000E+07  4.0000E+02  4.0000E-01
-1.0000E+07  1.2000E+03  1.0000E+07  1.2000E+03 -2.7000E+00
-1.0000E+07  0.          1.0000E+07  0.          1.1000E+00
TEST PROBLEM FOR DEMONSTRATION
-1.0000E+07  8.4000E+02 -3.7700E+02  8.4000E+02  1.2000E+00
-3.7700E+02  8.4000E+02 -3.5425E+02  9.1000E+02  1.2000E+00
-3.5425E+02  9.1000E+02 -2.8600E+02  1.1200E+03  7.0000E-01
-3.5425E+02  9.1000E+02  2.0000E+02  9.0000E+02  5.0000E-01
-2.8600E+02  1.1200E+03  3.2609E+02  1.0700E+03  7.0000E-01
 2.0000E+02  9.0000E+02  3.2609E+02  1.0700E+03 -7.0000E-01
 2.0000E+02  9.0000E+02  3.3837E+02  9.3000E+02  1.2000E+00
 3.3837E+02  9.3000E+02  1.0000E+07  9.3000E+02  1.2000E+00
-1.0000E+07  5.0000E+02 -7.0000E+02  5.0000E+02  4.0000E-01
-7.0000E+02  5.0000E+02  1.0000E+03  4.0000E+02  4.0000E-01
 1.0000E+03  4.0000E+02  1.0000E+07  4.0000E+02  4.0000E-01
-1.0000E+07  1.2000E+03  1.0000E+07  1.2000E+03 -2.7000E+00
-1.0000E+07  0.          1.0000E+07  0.          1.1000E+00
TEST PROBLEM FOR DEMONSTRATION
-9.9998E+06  8.4000E+02 -1.7700E+02  8.4000E+02  1.2000E+00
-1.7700E+02  8.4000E+02 -1.5425E+02  9.1000E+02  1.2000E+00
-1.5425E+02  9.1000E+02 -8.6000E+01  1.1200E+03  7.0000E-01
-1.5425E+02  9.1000E+02  4.0000E+02  9.0000E+02  5.0000E-01
-8.6000E+01  1.1200E+03  5.2609E+02  1.0700E+03  7.0000E-01
 4.0000E+02  9.0000E+02  5.2609E+02  1.0700E+03 -7.0000E-01
 4.0000E+02  9.0000E+02  5.3837E+02  9.3000E+02  1.2000E+00
 5.3837E+02  9.3000E+02  1.0000E+07  9.3000E+02  1.2000E+00
-9.9998E+06  5.0000E+02 -5.0000E+02  5.0000E+02  4.0000E-01
-5.0000E+02  5.0000E+02  1.2000E+03  4.0000E+02  4.0000E-01
 1.2000E+03  4.0000E+02  1.0000E+07  4.0000E+02  4.0000E-01
-9.9998E+06  1.2000E+03  1.0000E+07  1.2000E+03 -2.7000E+00
-9.9998E+06  0           1.0000E+07  0.          1.1000E+00
TEST PROBLEM FOR DEMONSTRATION
-9.9998E+06  8.4000E+02 -1.7700E+02  8.4000E+02  1.2000E+00
-1.7700E+02  8.4000E+02 -1.5425E+02  9.1000E+02  1.2000E+00
-1.5425E+02  9.1000E+02 -8.6000E+01  1.1200E+03 -3.0000E-01
-1.5425E+02  9.1000E+02  4.0000E+02  9.0000E+02  1.5000E+00
-8.6000E+01  1.1200E+03  5.2609E+02  1.0700E+03 -3.0000E-01
 4.0000E+02  9.0000E+02  5.2609E+02  1.0700E+03  3.0000E-01
 4.0000E+02  9.0000E+02  5.3837E+02  9.3000E+02  1.2000E+00
 5.3837E+02  9.3000E+02  1.0000E+07  9.3000E+02  1.2000E+00
-9.9998E+06  5.0000E+02 -5.0000E+02  5.0000E+02  4.0000E-01
-5.0000E+02  5.0000E+02  1.2000E+03  4.0000E+02  4.0000E-01
 1.2000E+03  4.0000E+02  1.0000E+07  4.0000E+02  4.0000E-01
-9.9998E+06  1.2000E+03  1.0000E+07  1.2000E+03 -2.7000E+00
-9.9998E+06  0.          1.0000E+07  0.          1.1000E+00
```

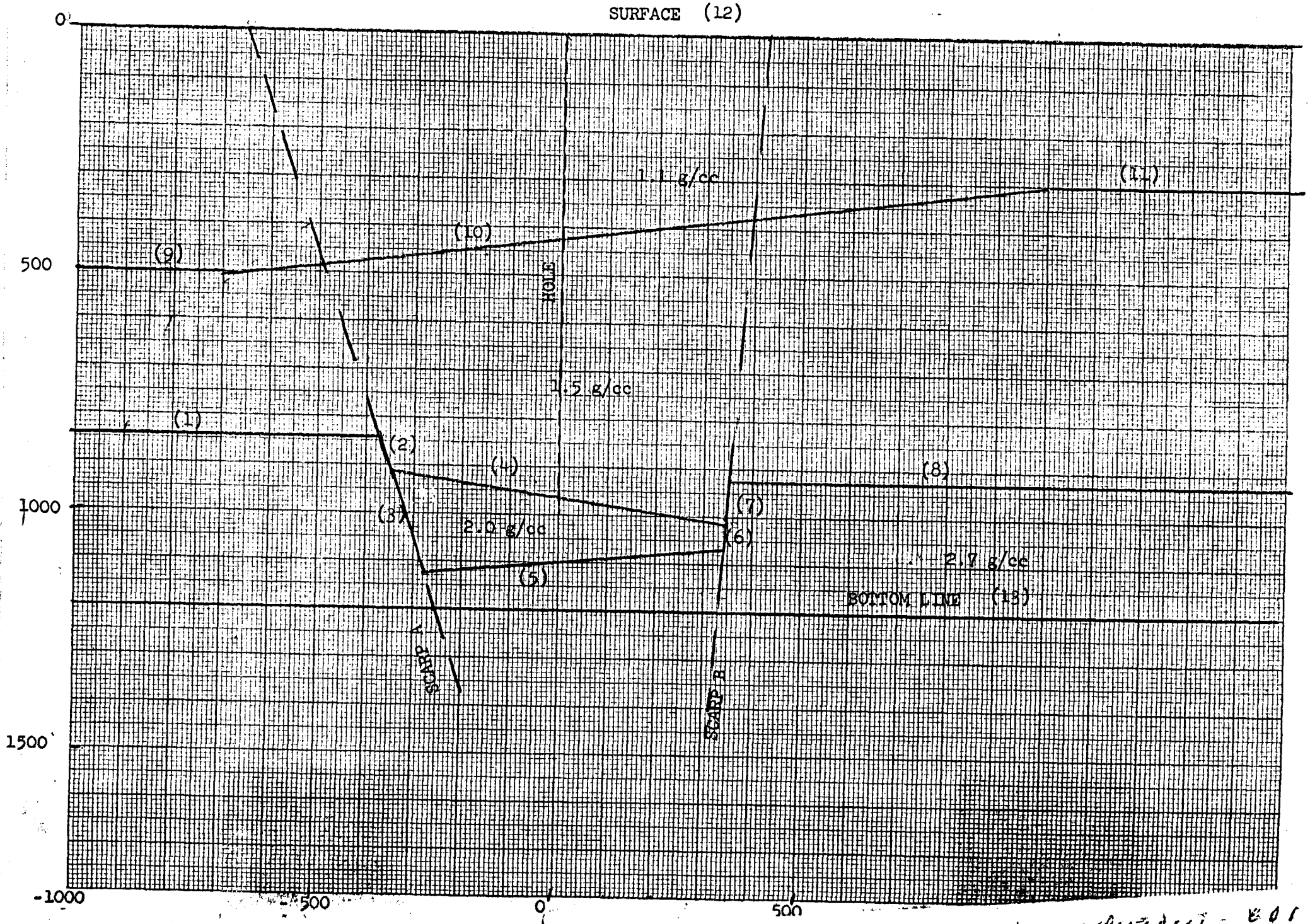
OUTPUT - BIF

TABLE 8

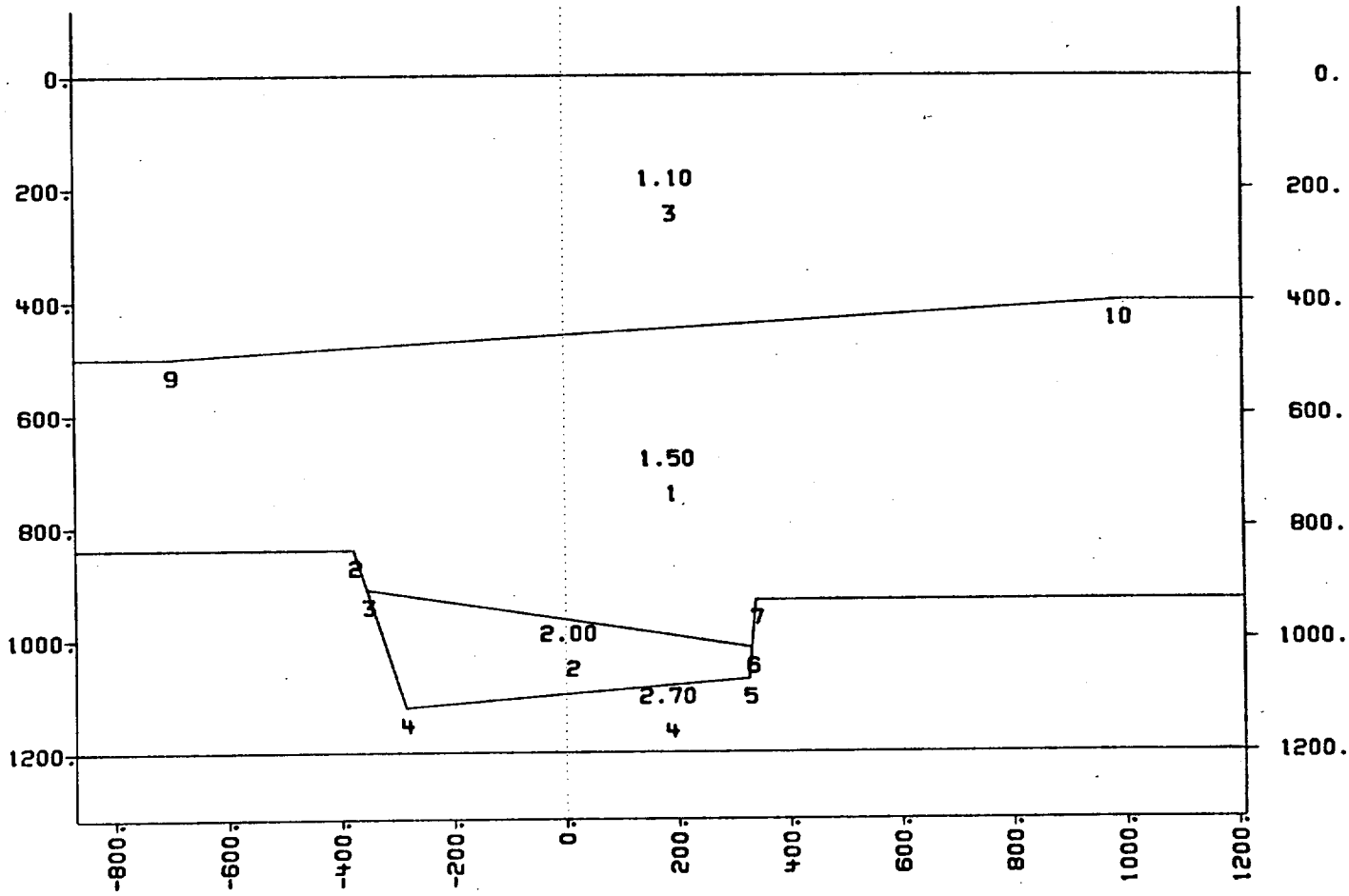
Output file BIFOUT from first problem. For storage

2.0000E+02	2.5000E+02	-1.1607E-02	0.
2.5000E+02	3.0000E+02	-1.5397E-02	0.
3.0000E+02	3.5000E+02	-1.9829E-02	0.
3.5000E+02	4.0000E+02	-2.5035E-02	0.
4.0000E+02	4.5000E+02	-3.1174E-02	0.
4.5000E+02	5.0000E+02	-3.7203E-02	0.
5.0000E+02	5.5000E+02	-4.5507E-02	0.
5.5000E+02	6.0000E+02	-5.5582E-02	0.
6.0000E+02	6.5000E+02	-6.7315E-02	0.
6.5000E+02	7.0000E+02	-8.0735E-02	0.
7.0000E+02	7.5000E+02	-9.5647E-02	0.
7.5000E+02	8.0000E+02	-1.1150E-01	0.
8.0000E+02	8.5000E+02	-1.2727E-01	0.
8.5000E+02	9.0000E+02	-1.4148E-01	0.
9.0000E+02	9.5000E+02	-1.5237E-01	0.
9.5000E+02	1.0000E+03	-1.4998E-01	0.
2.0000E+02	2.5000E+02	-9.2692E-03	0.
2.5000E+02	3.0000E+02	-1.2318E-02	0.
3.0000E+02	3.5000E+02	-1.5887E-02	0.
3.5000E+02	4.0000E+02	-2.0089E-02	0.
4.0000E+02	4.5000E+02	-2.5057E-02	0.
4.5000E+02	5.0000E+02	-2.9730E-02	0.
5.0000E+02	5.5000E+02	-3.6475E-02	0.
5.5000E+02	6.0000E+02	-4.4792E-02	0.
6.0000E+02	6.5000E+02	-5.4607E-02	0.
6.5000E+02	7.0000E+02	-6.6058E-02	0.
7.0000E+02	7.5000E+02	-7.9156E-02	0.
7.5000E+02	8.0000E+02	-9.3643E-02	0.
8.0000E+02	8.5000E+02	-1.0876E-01	0.
8.5000E+02	9.0000E+02	-1.2293E-01	0.
9.0000E+02	9.5000E+02	-1.3353E-01	0.
9.5000E+02	1.0000E+03	-1.3877E-01	0.
2.0000E+02	2.5000E+02	-7.5187E-03	0.
2.5000E+02	3.0000E+02	-1.0030E-02	0.
3.0000E+02	3.5000E+02	-1.3008E-02	0.
3.5000E+02	4.0000E+02	-1.6587E-02	0.
4.0000E+02	4.5000E+02	-2.0959E-02	0.
4.5000E+02	5.0000E+02	-2.5524E-02	0.
5.0000E+02	5.5000E+02	-3.1824E-02	0.
5.5000E+02	6.0000E+02	-4.0812E-02	0.
6.0000E+02	6.5000E+02	-5.2776E-02	0.
6.5000E+02	7.0000E+02	-6.9005E-02	0.
7.0000E+02	7.5000E+02	-9.1091E-02	0.
7.5000E+02	8.0000E+02	-1.2014E-01	0.
8.0000E+02	8.5000E+02	-1.5470E-01	0.
8.5000E+02	9.0000E+02	-1.8902E-01	0.
9.0000E+02	9.5000E+02	-2.1591E-01	0.
9.5000E+02	1.0000E+03	-2.3270E-01	0.
2.0000E+02	2.5000E+02	1.0906E-03	0.
2.5000E+02	3.0000E+02	1.2971E-03	0.
3.0000E+02	3.5000E+02	1.4907E-03	0.
3.5000E+02	4.0000E+02	1.6371E-03	0.
4.0000E+02	4.5000E+02	1.6753E-03	0.
4.5000E+02	5.0000E+02	2.3745E-03	0.
5.0000E+02	5.5000E+02	2.4246E-03	0.
5.5000E+02	6.0000E+02	1.1958E-03	0.
6.0000E+02	6.5000E+02	-1.1243E-03	0.
6.5000E+02	7.0000E+02	-5.0892E-03	0.
7.0000E+02	7.5000E+02	-1.1112E-02	0.
7.5000E+02	8.0000E+02	-1.8438E-02	0.

GRAPHIC CONTROLS CORPORATION Buffalo, New York
Printed in U.S.A.
SQUARE 10 X 10 TO THE CENTIMETER AS-8014-



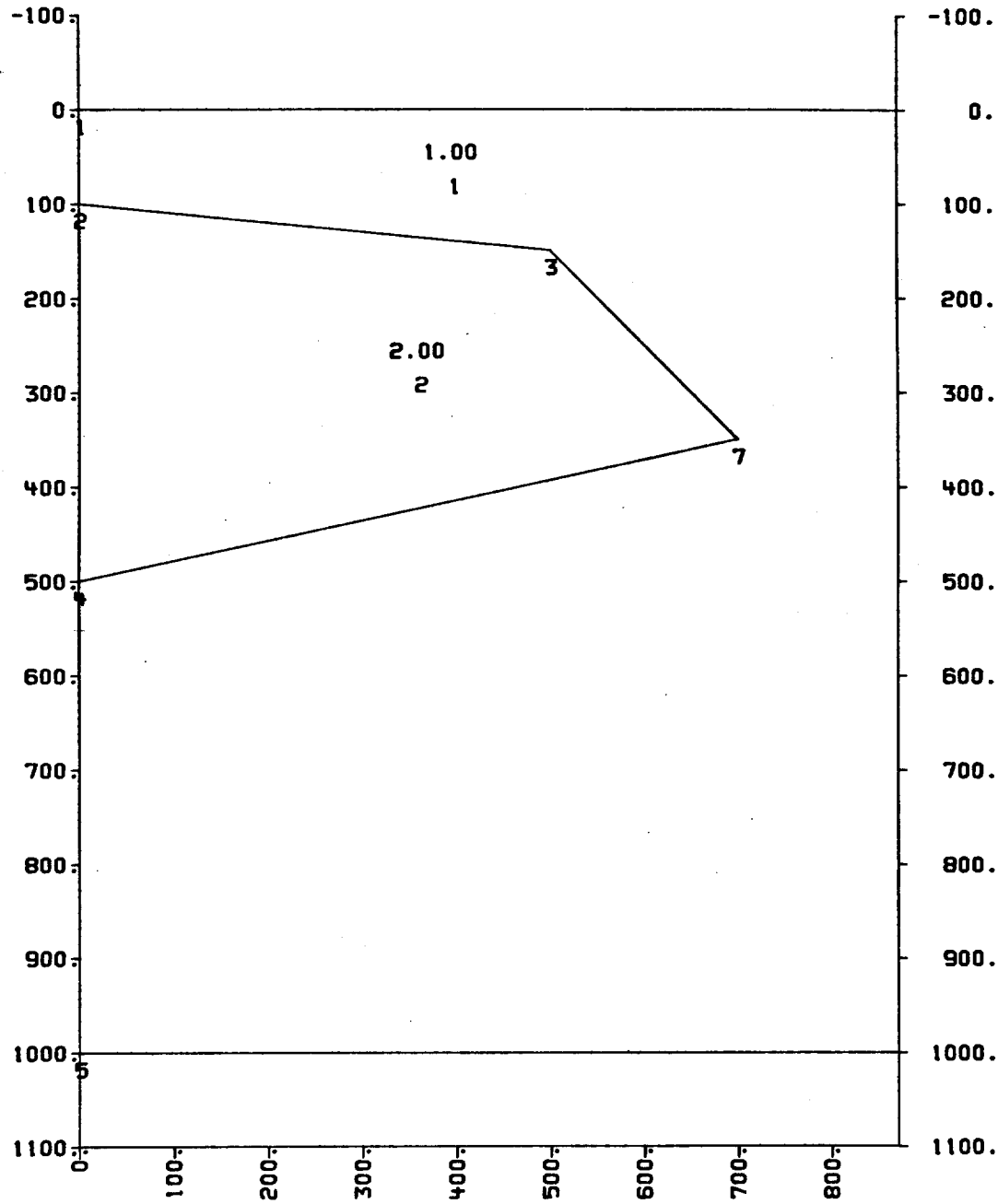
OUTPUT - EDI



TEST PROBLEM FOR DEMONSTRATION

FIGURE 2

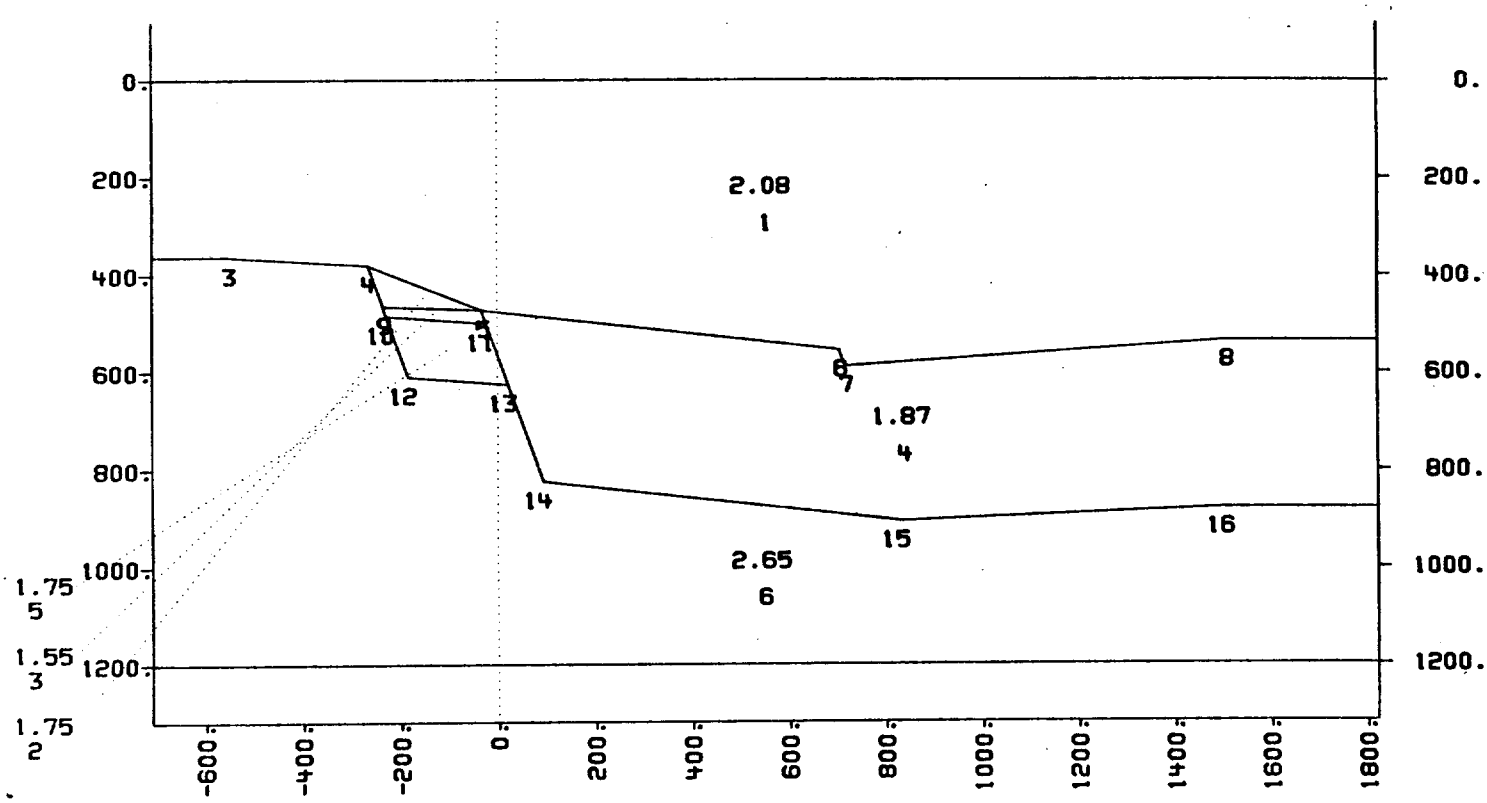
Output from first problem (described in Fig 1, input Table 1)



CYLINDRICAL DEMONSTRATION

FIGURE 3

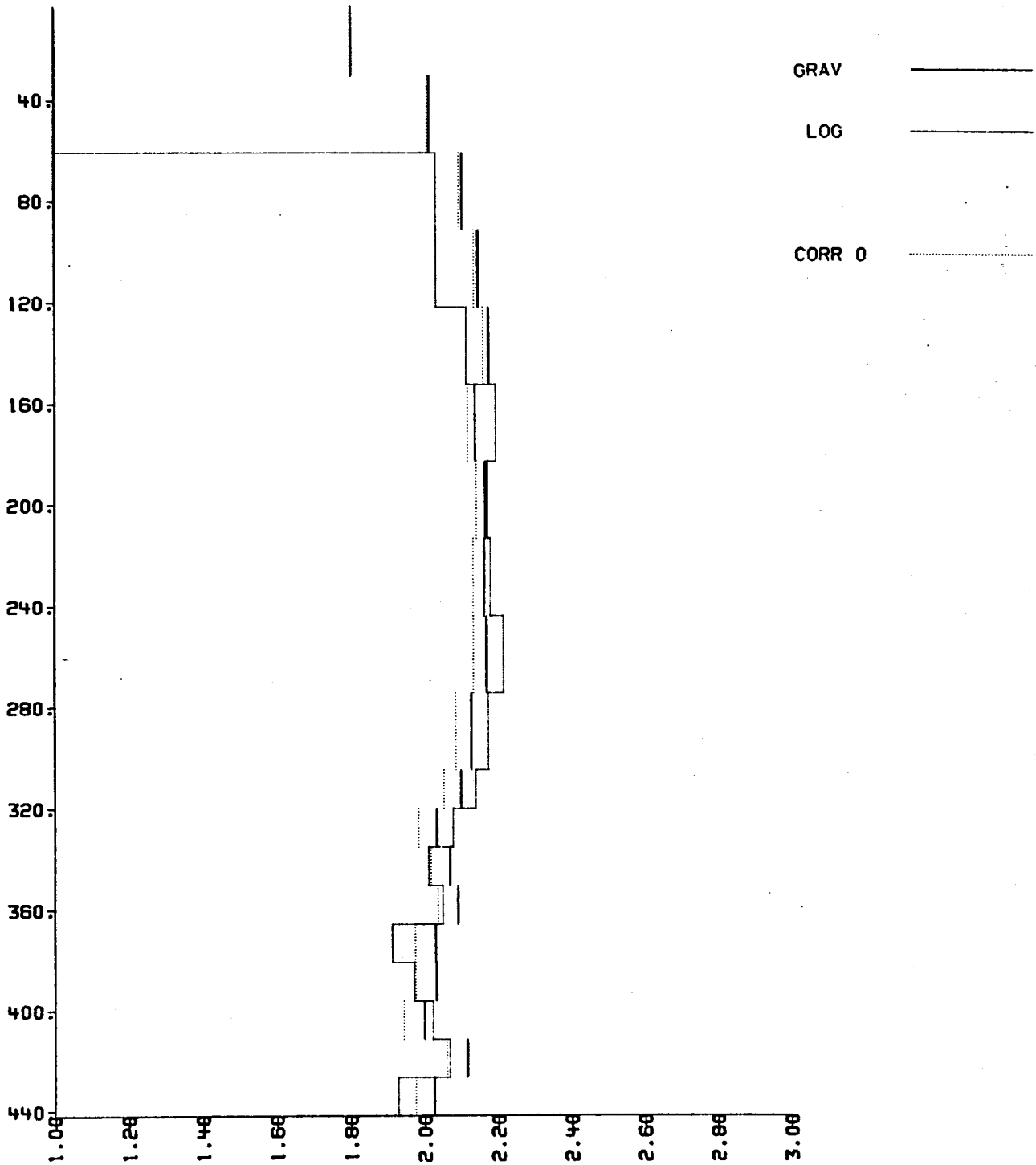
Output from cylindrical problem (Input Table 2)



2FE/GIN1 ORIGINAL CROSS SECTION FROM PROSPECTUS

FIGURE 4

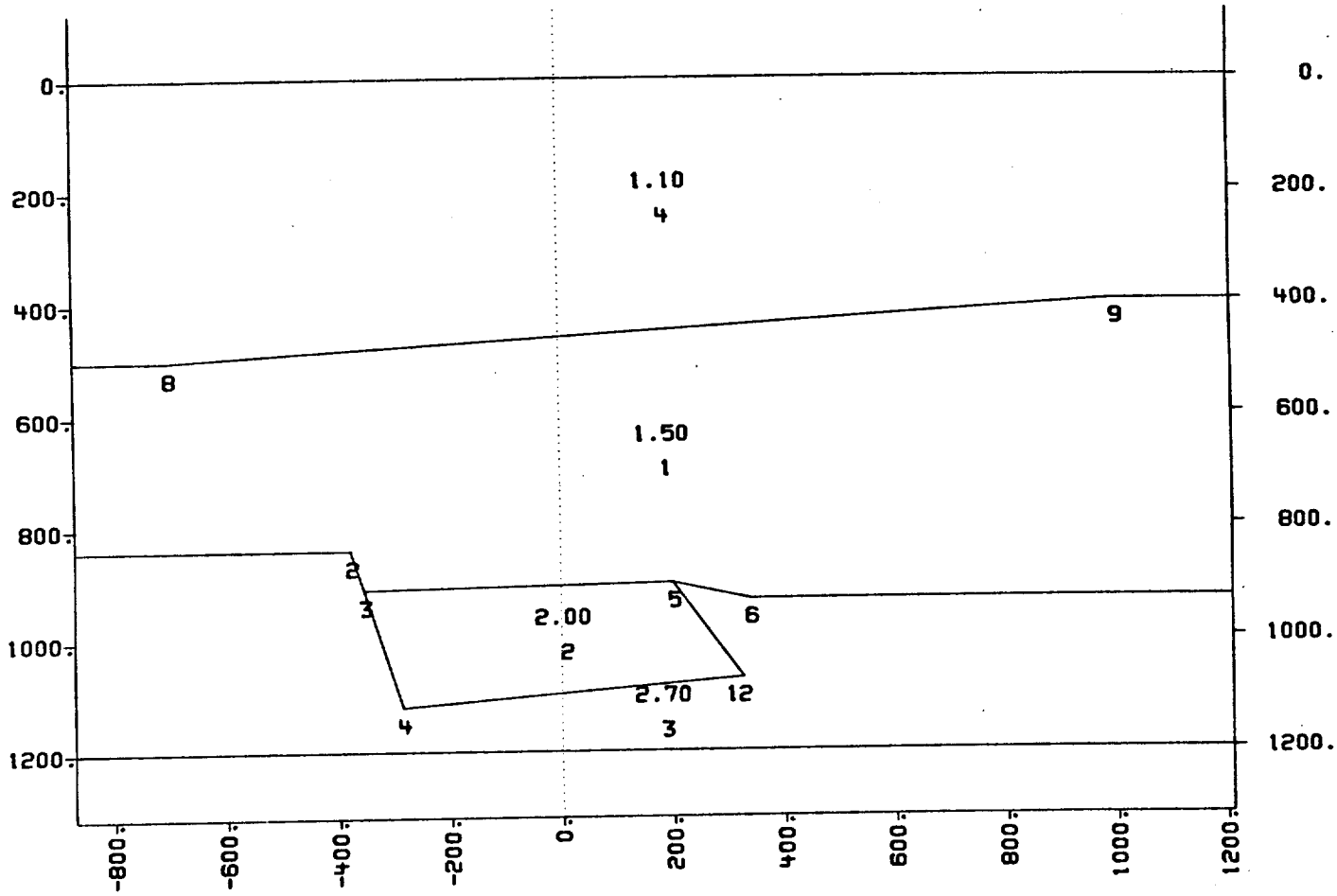
Cross section output from LOG-type problem (Input Table 3)



U2FE/G1N1 ORIGINAL CROSS SECTION FROM PROSPECTUS

FIGURE 5

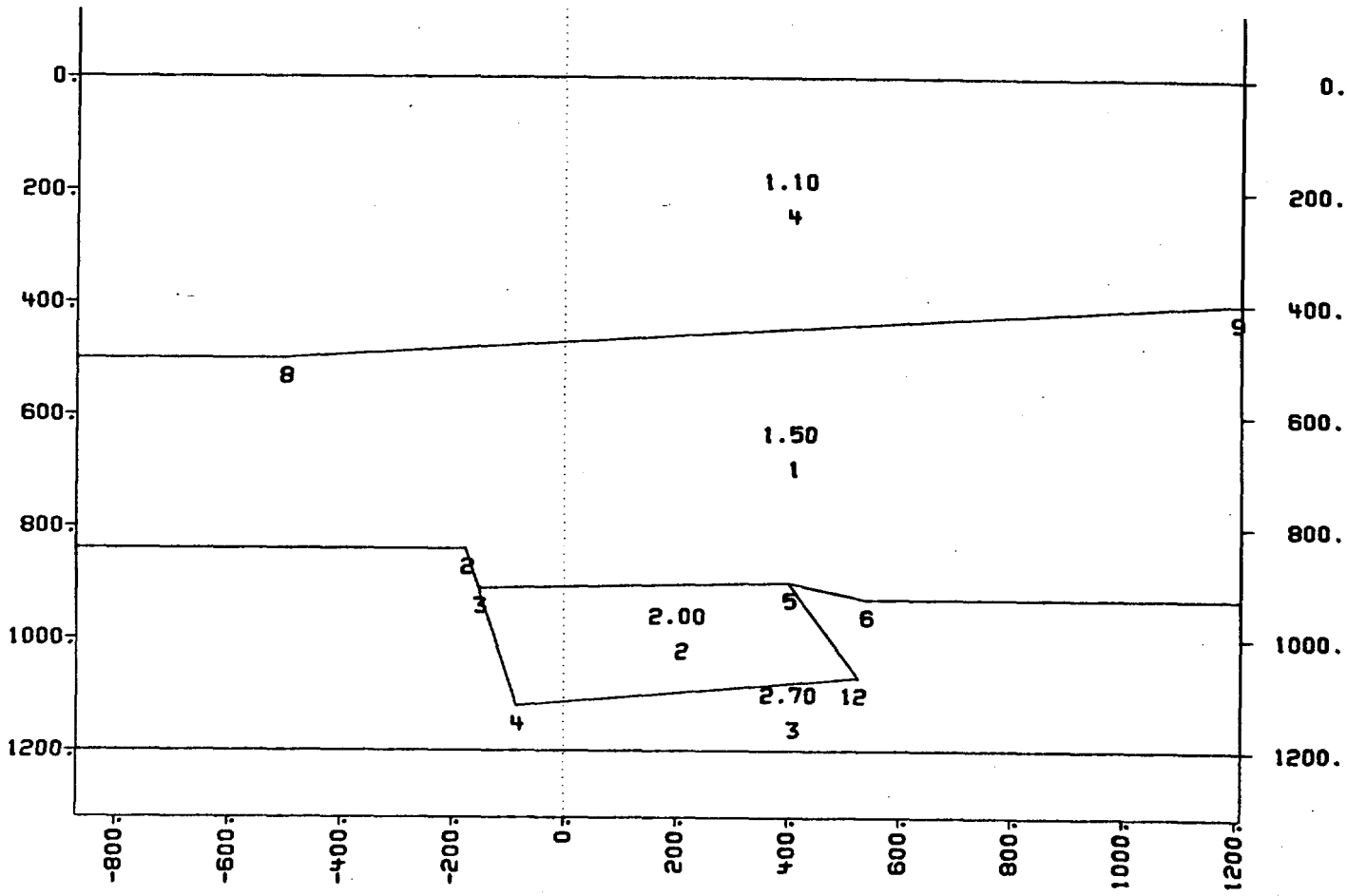
Density output from LOG-type problem (Input table 3)



TEST PROBLEM FOR DEMONSTRATION

FIGURE 6

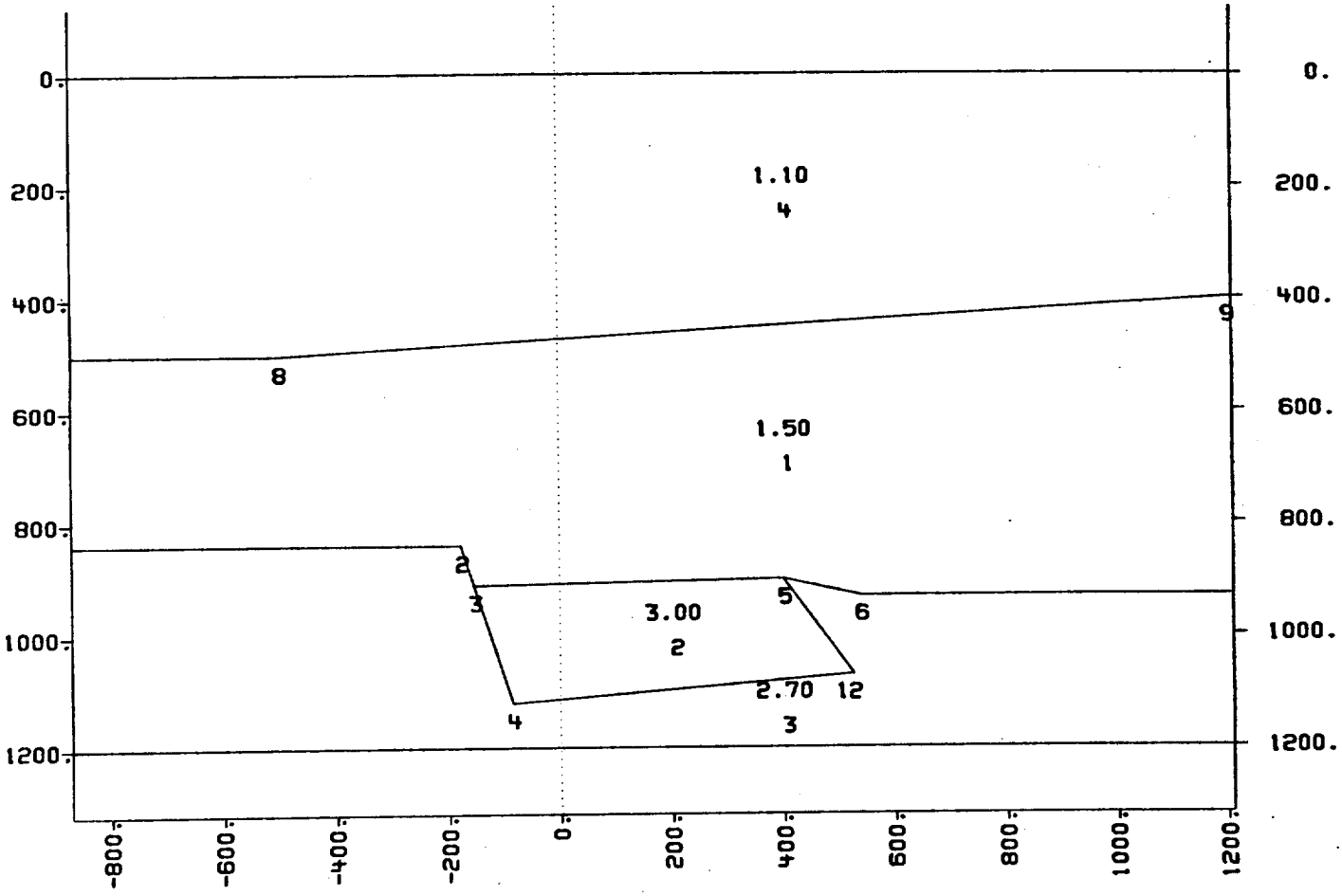
Output from first problem (Fig. 2) after INT interaction



TEST PROBLEM FOR DEMONSTRATION

FIGURE 7

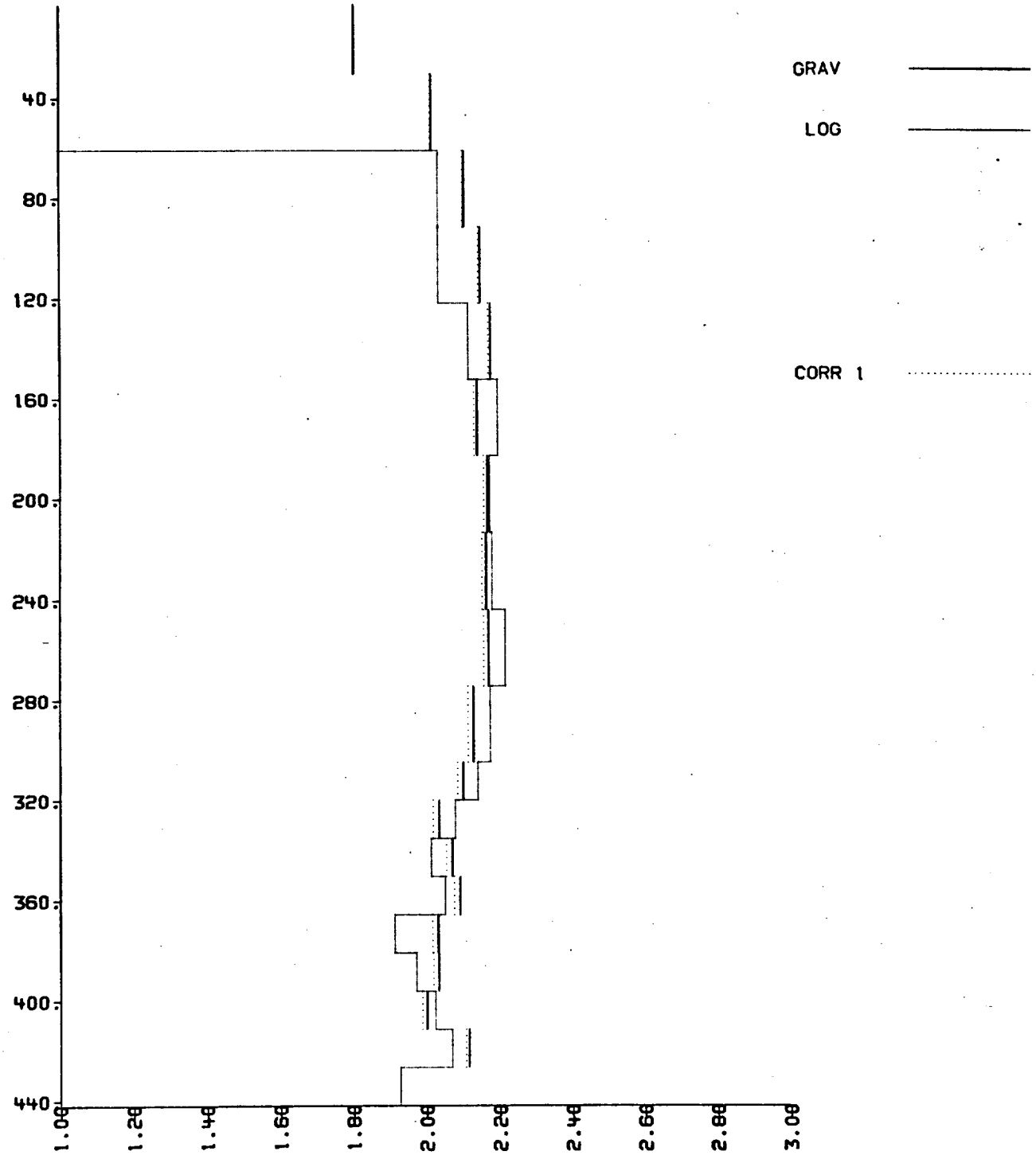
Output from first problem (Fig. 2) after INT and ADD interactions



TEST PROBLEM FOR DEMONSTRATION

FIGURE 8

Output from first problem (Fig. 2) after INT, ADD and DEN interactions



U2FE/GIN1 ORIGINAL CROSS SECTION FROM PROSPECTUS

FIGURE 9

Density plot from third problem (Figs 4,5) after DEN interaction

BIFUR3 LISTING

```
*     FORTRAN            MAIN
      PROGRAM TENPLOT(HSP,TAPE2,TAPE63=25,TAPE17,TAPE3,TAPE4,TAPE5)
C
      CLICHE DATACLICHE
      CODE ANALYSIS
      PARAMETER (MNEDIM=201)
      ADDRESS EBRAN,KBRAN,DBRAN,LABELS
      COMMON/CSCAN/INPUT(100),IDEC(100),NMESIN
      COMMON/CDATA1/LASTCARD(8),KSCAN(10)
      COMMON /CDATA2/ MNEMONIC,NEXTM,NWORD
      COMMON /BIFDAT/ CORR(100),CRHO(50),DPT(100),DRHO(100),EX,
     1 G(100),ILT,ILV(100),INA(100,5),INB(100,5),IRV(100),ISIG(26),JDP,
     2 JZX,JZY,KIT(8),KK,KV(25,50),LDEP,LER,LZEP,NA(100),NB(100),NI(50),
     3 NLFF(25),NLFT(25),NP,NRT(25),NRTT(25),NV(25,50),R(100),RCOR(100),
     4 RHOL(100),RMD(50),TST,TSTM,TTH(26),XO(26),XA(100),XB(100),
     5 XMD(50),XMN(50),XMX(50),XN(50),XAFT,XSHFT,ZO(26),ZA(100),
     6 ZB(100),ZLFT(50),ZMD(50),ZMN(50),ZMX(50),ZN(50),ZORD(50)
      COMMON /MORE/ ZPT(100),ZRT(25),ZSHFT,RHL(100),
     1LLOG,DD1(100),DD2(100),DDR(100)
      EQUIVALENCE (FINPUT,INPUT)
      DIMENSION FINPUT(1)
      ENDCLICHE
C
      USE DATACLICHE
      DIMENSION MNES(MNEDIM),LABELS(1),IPROMPT(2)
      EQUIVALENCE (LABELS,MNES(2))
C
C*********************************************************************
C
C INPUT ASSUMPTIONS
C
C 100 LINES MAXIMUM
C 50 VOLUMES MAX WITH MAX OF 25 LINES PER VOLUME
C MAX OF 100 INTERSECTIONS WITH  MAX OF 5 LINES PER INTERSECTION
C XA,ZA,XB,ZB INPUT COORDINATES OF LINE ENDS,THEN CODE SORTS SO
C XA,ZA ARE COORDINATES OF LEFT (MOST NEGATIVE X ) END OF LINE
C  XB,ZB OF RIGHT END-
C Z INCREASES DOWNWARDS
C DRHO IS ORIGINALLY DENSITY ON RIGHT MINUS DENSITY ON LEFT AS INPUT
C CODE WILL CHANGE TO MAKE CONSISTENT WITH XA LEFTMOST POINT
C
C SEE USERS MANUAL FOR DETAILS
C
C DEFINITIONS
C
C CORR=DIFFERENCE BETWEEN CALCULATED GRAVIMETRIC DENSITY AND
C       COMPUTED "LOG" DENSITY ---ALSO USED AS TEMP. PARAMETER
C CRHO=SORTING PARAM
C DD1,DD2 ARE INPUT GRAVITY MEASUREMENT DEPTHS
C DDR IS DENSITY MEASURED BETWEEN DD1 AND DD2
C DPT=GRAVITY MEASURING DEPTHS
C DRHO=DENSITY ON RIGHT OF INPUT LINE MINUS THAT ON LEFT
C EX IS SIGNAL TO EXIT FROM LOGDEN OR AN ERROR IN INPUT
C G=CALCULATED GRAVITY
C IDEC=1,2 OR 3 DEPENDING ON WHETHER INPUT IS --,NUMBER,OR LETTER
C IDS IS A TEMPORARY VERSION OF IDEC
C ILT=NUMBER OF VOLUMES
C ILV=VOLUME ON LEFT OF EACH LINE
C INA(K,L) ARE THE L LINES THAT HAVE LEFT ENDS AT INTERSECTION K
C INB(K,L) ARE THE L LINES THAT HAVE RIGHT ENDS AT INTERSECTION K
```

```
C INEW,INEW2 ARE LOG FILENAMES
C INPUT,FINPUT ARE FIXED,FLOATING VALUES OF INPUT DATA
C IRV=VOLUME ON RIGHT OF EACH LINE
C ISIG=1 IF SCARP CORRESPONDING TO THAT LETTER EXISTS
C JDP IS INDEX OF INPUT DEPTHS--AND MAXIMUM VALUE
C JZX=NUMBER OF LINES
C JZY=JZX PLUS ARTIFICIAL END LINES
C KIT IS TITLE
C KK=NUMBER OF LINES CROSSED GOING UP CALCULATING DENSITY
C KV(K,L)=1 IF VOLUME IS ON LEFT OF LINE IN QUESTION (WHILE PROCESSING
C        THE VOLUME)
C LDEP IS THE NUMBER OF MEASUREMENT DEPTHS
C LDP IS TEMPORARY VALUE OF LDEP
C LDER IS INDEX FOR INPUT GRAVIMENTRIC DENSITY
C LER IS THE NUMBER OF THE PASS THRU THE CODE WITH CHANGES
C LZEP=NUMBER OF SURFACE MEASURING POINTS
C NA=INTERSECTION NO. OF LEFT END OF LINE
C NB=INTERSECTION NO. OF RIGHT END OF LINE (MAXIMUM X)
C NI=NUMBER OF LINES FOR THE VOLUME
C NLFF IS SORTING VARIABLE
C NLFT IS SORTING VARIABLE
C NP=NUMBER OF INTERSECTIONS
C NRT IS SORTING VARIABLE
C NRTT IS SORTING VARIABLE
C NV(K,L)=LINE NUMBERS (K OF THEM) FOR LTH VOLUME
C R=CALCULATED GRAVIMETRIC DENSITY
C RCOR IS CORRECTED GRAVIMETRIC DENSITY
C RHI,RHIT ARE TEMPORARY INTEGRATED LOG DENSITY
C RHL=INTEGRATED LOG DENSITY
C RHOL=CALCULATED LOG DENSITY FROM INPUT DENSITIES
C RMD=DENSITY OF THE VOLUME
C TST= LARGE NUMBER,TSTM ITS NEGATIVE
C TTH=ANGLE OF SCARP
C XO=INITIAL X OF SCARP
C XA=X VALUE OF LEFT END OF LINE (MINIMUM)
C XB=X VALUE OF RIGHT END OF LINE
C XMD=X OF MIDDLE OF VOLUME
C XMN=MIN X OF VOLUME
C XMX=MAX X OF VOLUME
C XN=SORTING VARIABLE
C XAFT=X OF VOLUME FOR DENSITY CALC
C XSHFT=X ADDED TO ALL INPUT X VALUES. THEREFORE HOLE SHIFTED -XSHFT
C ZO=INITIAL Z OF SCARP
C ZA=LEFT Z OF LINE
C ZB=RIGHT Z OF LINE
C ZLFT IS SORTING PARAMETER
C ZMD=Z OF MIDDLE OF VOLUME
C ZMN IS MINIMUM Z OF VOLUME
C ZMX=MAX Z OF VOLUME
C ZN IS SORTING PARAMETER
C ZORD IS SORTING PARAMETER
C ZPT IS HORIZONTAL POSITION OF SURFACE MEASURING POINT
C ZRT IS SORTING PARAMETER
C ZSHFT=CONSTANT ADDED TO ALL  INPUT VALUES OF Z  FOR SHIFT
C
C  ************************************************************
C
C
C
      DATA (MASKL=77777777770000000000B),(MASKR=7777777777B),
```

```
     1 (MASK8=377B)
      DATA (INITCRT=1)
      CALL CHANGE (4R+INP)
      CALL ASSIGN (3,15)
      CALL ASSIGN (4,0,6RBIFOUT)
      CALL KEEP80(1,1)
      CALL ASSIGN (5,0,6RBIFINP)
      CALL UXFNAME(3HBIF)
      CALL UX80ID("BOX U79 HEARST")
C
C
C*****PUT OPTION DIRECTORY HERE
C     TRIX TABS AT:  9 29 49
      DATA (MNES=3RADD,.LOC.AD1,3RDEN,.LOC.DEN1,5RDEPTH,.LOC.DP01,
     13RDPT,.LOC.DP01,3REND,.LOC.C22,3REOF,.LOC.E11,4RFILE,.LOC.Z21,
     22RGO,.LOC.GO1,2RGO,.LOC.GO1,3RINT,.LOC.INT1,3RLOG,.LOC.LOG1,
     34RLOGS,.LOC.LOG1,3RSUR,.LOC.DP30,3RTTY,.LOC.B11,3RZEP,.LOC.DP30,
     1 3RZPT,.LOC.DP30,4RSURF,.LOC.DP30)
      DATA (NUMOPTS=0)
      DATA (TST=1000000.),(TSTM=-1000000.)
C
C
C
C     **********
C     **********
C     INITIALIZE
C     **********
C     **********
C
      RHI=0.
      RHL(1)=0.
      JDP=LLOG=LDP=0
      LDR=LD1=0
      IQUIT=1
      LER=LMN=0
      LMNN=LMNM=0
      KEPT=0
      ISCALE=1        $$$ TEMP
      IEDFLAG=0       $$$ TEMP
      INTYP=1         $$$ "FILE" DEFAULT
      IERR=NMESIN=0
      IPROMPT=36360000760000000000B
      INUNIT=59
      INFILE=IOLDFILE=3RTTY
      IF (NUMOPTS) ,,LP02
      DO D102 I=1,MNEDIM,2
      IF (MNES(I).EQ.00B) A102,
 D102 NUMOPTS=NUMOPTS+1
 A102 CALL DATALST(I,MNES,2)
      CALL SETDATA(INUNIT,"TYPE FILE NAME_")
      IFC=0
      GO TO LP00
C
C
C     **********
C     **********
C     LOOP
C     **********
C     **********
```

```
C
 LP02 CALL SETDATA(INUNIT,IPROMPT)
      GO TO LP00
 LP03 CALL SETPROMPT(IPROMPT)
C
 LP00 CALL DATA(99.,IDEC,INPUT)
      IDX=2
C
C
C     BRANCH LOOP
C
      IF(LMN.EQ.1) GO TO 200
      DO IN00 IJ=1,MNEDIM,2
      IF (MNES(IJ).EQ.00B) 200,
      IF (INPUT.EQ.MNES(IJ)) GO TO LABELS(IJ)
 IN00 CONTINUE
C
C     INPUT DOES NOT BEGIN WITH RECOGNIZED LITERAL
C
  200 IF (INTYP.EQ.2) J01,
      DO D201 I=1,98
      J=99-I
      INPUT(J+1)=INPUT(J)
 D201 IDEC(J+1)=IDEC(J)
      GO TO (Z21,J01,ER5),INTYP
C * * * * * * * * * * * * * * * * *
C      INPUT FOR NEW BIFUR
C * * * * * * * * * * * * * * * * *
  J01 IF(LLOG-1)J02 ,LOG2,LOG3
  J02 IF(JZX.NE.0) GO TO J10
      IFC=1
      JZX=1
      DO J05 K=1,8
  J05 KIT(K)=KSCAN(K)
      GO TO LP00
  J10 IF(LZEP.NE.0) GO TO DP40
      IF(LDEP.NE.0) GO TO DP10
      IF(NWORD.NE.5) GO TO DP01
      LSUM=0
      DO J15 K=1,5
  J15 LSUM=LSUM+IDEC(K)
      IF (LSUM.LT.10) GO TO ER14
      IF(LSUM.GT.10)  ,J25
C
C      SCARP OR LINE ENDING ON SCARP
C
      CALL UNCODE
      JZX=JZX+1
      GO TO LP00
C
C    NORMAL LINE INPUT
C
  J25 XA(JZX)=FINPUT(1)
      XB(JZX)=FINPUT(3)
      ZA(JZX)=FINPUT(2)
      ZB(JZX)=FINPUT(4)
      DRHO(JZX)=FINPUT(5)
      JZX=JZX+1
      GO TO LP00
C
```

```
C     SWITCH TO DEPTHS INPUT
C
 DP01 IF(NWORD.GT.2) GO TO ER15
      IF(IDEC(1).NE.3) GO TO ER15
      JZX=JZX-1
      IF(NWORD.EQ.2) XSHFT=FINPUT(2)
      LDEP=1
      GO TO LP00
 DP10 IF(IDEC(1).EQ.3) GO TO DP30
      DO DP20 L=1,NWORD
      DPT(LDEP)=FINPUT(L)
 DP20 LDEP=LDEP+1
      GO TO LP00
C
C       THIS IS SURFACE INPUT
C
 DP30 LZEP=1
      IF(LDEP.LE.0) JZX=JZX-1
      IF(NWORD.EQ.2) ZSHFT=FINPUT(2)
      GO TO LP00
 DP40 DO DP50 L=1,NWORD
      ZPT(LZEP)=FINPUT(L)
 DP50 LZEP=LZEP+1
      GO TO LP00
C * * * * * * * * * * * * * * * * *
C * * * * * * * * * * * * * * * * *
C
C
C
C*************************
C*                       *
C*     OPTIONS FOLLOW    *
C*                       *
C*************************
C
C*****TTY EOF TTYON
  B11 IF (INFILE.EQ.3RTTY) LP02,
      ITTYON=0
      IOLDFILE=INFILE
      CALL DEVICE(5HCLOSE,IOLDFILE)
      INFILE=3RTTY
      INUNIT=59
      GO TO LP02
C
C COME HERE IF AN ERROR IN A LINE
C
  TER IF(LER.EQ.0) GO TO FOP
      EX=0.
      LER=-LER
      CALL CINT
      LER=-LER
      GO TO LP03
C
C IF AN INPUT ERROR QUIT IF IN FILE,FIX IF IN CHANGE
C
 DEN1 CALL CDEN
      GO TO LP03
 INT1 CALL CINT
      GO TO LP03
  AD1 DO AD2 J=1,JZX
```

```
      XA(J)=XA(J)+FINPUT(2)
  AD2 XB(J)=XB(J)+FINPUT(2)
      GO TO LP03
C
C COME TO ABOVE FOR CHANGES, THEN BELOW TO RESTART
C
 LOG2 IF(LMNN.NE.1) LMN=1
      IF(IDEC(1).NE.2) GO TO LP02
      LMN=0
      LMNN=1
      JDP=JDP+1
C
C READ MORIA OUTPUT FILE,CONVERT FT TO M,KEEP FIRST AND SECOND
C DEPTHS FOR PLOTTING GRAVIMETRIC DENSITY.  ALSO PUT UNDUPLICATED
C DEPTHS INTO DPT FOR CALCULATIONS.STORE GRAVIMETRIC DENSITY
C
      FINPUT(1)=FINPUT(1)*TMUL
      FINPUT(2)=FINPUT(2)*TMUL
      DD1(JDP)=FINPUT(1)
      DD2(JDP)=FINPUT(2)
      DDR(JDP)=FINPUT(4)
      IF(LDP.EQ.0) GO TO LL3
      TT1=FINPUT(1)-DPT(LDP)
      IF(ABS(TT1).LT.0.1) GO TO LL4
      LDP=LDP+1
      DPT(LDP)=FINPUT(1)
      TT1=FINPUT(2)-DPT(LDP-1)
      GO TO LL5
  LL4 TT1=FINPUT(1)-FINPUT(2)
      IF(ABS(TT1).LT.0.1) GO TO LL8
      TT1=FINPUT(2)-DPT(LDP)
  LL5 IF(ABS(TT1).LT.0.1) GO TO LL8
      LDP=LDP+1
      DPT(LDP)=FINPUT(2)
      GO TO LL8
  LL3 LDP=1
      DPT(1)=FINPUT(1)
      TT1=FINPUT(1)-FINPUT(2)
      IF (ABS(TT1).LT.0.1) GO TO LL8
      LDP=2
      DPT(2)=FINPUT(2)
  LL8 GO TO LP00
 LOG3 IF(LMNM.NE.1) LMN=1
      IF(IDEC(1).NE.2) GO TO LP02
      LMN=0
      LMNM=1
C
C READ ALEXFILE OF INTEGRATED LOG DENSITIES
C CHOOSE LOG DEPTHS MATCHING GRAVIMETRIC DENSITY DEPTHS
C GET INTEGRATED DENSITY BETWEEN THEM
C
      DRD=FINPUT(1)*TMUL
      IF (LDR.EQ.0) GO TO LO6
  LO1 IF(DRD.LT.DPT(LD1)) GO TO LP00
      IF(RHI.NE.0.) GO TO LO2
      RHI=FINPUT(6)
      DRI=DRD
  LO2 CONTINUE
C
C INITIALIZING INTEGRATED DENSITY
```

```
C
      IF(DRD.LT.DPT(LDR)) GO TO LP00
      RHIT=FINPUT(6)
      DRH=RHIT-RHI
      DRIT=DRD
      DDP=DRIT-DRI
      IF(DDP.GT.0.) GO TO LO4
      RHL(LDR)=0.
      GO TO LO5
  LO4 RHL(LDR)=DRH*TMUL/DDP
  LO5 RHI=RHIT
      DRI=DRIT
      LDR=LDR+1
      IF(LDR.GT.LDEP) GO TO GPL
      GO TO LP00
  LO6 IF(DRD.LT.DPT(1)) GO TO LO8
      DO LO7 L=2,LDEP
      IF(DRD.GT.DPT(L)) GO TO LO7
      LDR=L+1
      LD1=L
      GO TO LO1
  LO7 CONTINUE
  LO8 LD1=1
      LDR=2
      GO TO LO1
C
C FINDING FIRST TWO POINTS BELOW FIRST LOG DEPTH POINT
C
C
C END OF LOG INPUT
C
  GO1 CALL UXFNAME(3HBIF)
      CALL UX80ID("BOX U79 HEARST")
      GO TO GO2
C
C IF EOF IS PRESENT EITHER END OF NORMAL INPUT,GRAVIMETRIC
C  DENSITY INPUT, OR LOG DENSITY INPUT. VALUE OF LLOG SHOWS WHICH
C 0 NORMAL, 1 GRAV DENS,2 LOG DENS
C
  E11 IF (LLOG.EQ.1) LDEP=LDP+1
      IF(LLOG.EQ.2) GO TO GPL
      LDEP=LDEP-1
      LZEP=LZEP-1
  GO2 WOT 3,FW4,KIT
      EX=0.
      CALL CHECK
      IF(EX.GT.0.) GO TO TER
      CALL VOLUMES
      IF(EX.GT.0.) GO TO TER
      CALL BIFPLT
  FOP WOT 3,FWO
      DO FOT J=1,JZX
  FOT WOT 3,FW2,J,XA(J),ZA(J),NA(J),XB(J),ZB(J),NB(J),ILV(J),
     1 IRV(J),DRHO(J)
      WOT 3,FVO
      DO FVT L=1,ILT
      NIL=NI(L)
      WOT 3,FV1,L,(NV(K,L),K=1,10),RMD(L),XMD(L),ZMD(L)
  FVT IF(NIL.GT.10) WOT 3,FV2,(NV(J,L),J=11,NIL)
      IF(EX.EQ.1.) CALL QUIT(1)
```

```
      CALL LOGDEN
      CALL GEXEC
      IF(LLOG.EQ.0) GO TO GTB
      IF(LER.EQ.0)  ,GPL
      LLOG=2
      IF(IDS.NE.3) GO TO ER5
      GO TO Z22
  GPL CALL GPLOT
  GTB CALL GTABLS(IFIL,1,2)
      WOT 59,FILL,IFIL
C
C MAKE AN INPUT-TYPE FILE OF CURRENT LINE DATA
C
      WOT 5,FW5,KIT
      WOT 5,FW3,(XA(J),ZA(J),XB(J),ZB(J),DRHO(J),J=1,JZX)
      CALL PLOTE
      CALL EMPTY(3)
      WOT 59,FILT
      LER=LER+1
      GO TO B11
 FILL FORMAT("PLOT FILE IS ",A8,"80")
 FILT FORMAT("TYPE-DEN-INT-ADD-, THEN GO----OR END")
  FVO FORMAT(//" VOL  L1  L2  L3  L4  L5  L6  L7  L8  L9 L10  RHO",
     1 "     XMD              ZMD   ")
  FV1 FORMAT(11I4,F6.3,2E12.4)
  FV2 FORMAT(4X,10I4)
  FWO FORMAT(" LINE    LEFT X     LEFT Z    LFT INT    RT X       ",
     1 "RT Z      RT INT  VOL ON L  VOL ON R  DELTA RHO")
  FW2 FORMAT(I5,1X,2E12.4,2X,I4,2X,2E12.4,2X,I4,4X,I4,5X,I4,5X,F7.3)
  FW3 FORMAT(5E12.4)
  FW4 FORMAT ("1",8A10,//)
  FW5 FORMAT(8A10)
C
C
C*****FILE
 LOG1 LLOG=1
      JZX=JZX-1
C
C THIS BIT READS FIRST THE GRAVIMETRIC DENSITY FILE  (NAMED BY
C SECOND WORD ON CARD WITH FIRST WORD "LOGS" AND LOG DENSITY FILE
C NAMED BY THIRD WORD. THE FOURTH WORD IS A MULTIPLIER THAT SHOULD
C BE 1. IF INPUT IS IN METRES AND OUTPUT WANTED IN METRES
C OTHERWISE IT IS ASSUMED THAT LOG INPUT IS IN FEET AND
C GEOLOGY INPUT IN METRES
C THEN THE REST OF THIS SORTS OUT INPUT FILES
C
      IDS=IDEC(3)
      INEW2=INPUT(3)
      TMUL=FINPUT(4)
      IF(NWORD.LT.4) TMUL=.3048
  Z21 IF(IDEC(2).NE.3) ER5,
  Z22 INTYP=2
      IOPENED=00B
      INEW=INPUT(2)
      IF (LLOG.GT.1) INEW=INEW2
      GO TO L21
  L21 IOLDFILE=INFILE
      INFILE=INEW
      IF (INFILE.EQ.3RTTY) GO TO M21
      CALL DEVICE("CLOSE",IOLDFILE,IERR)
```

```
C
      CALL DEVICE("OPEN",INFILE,KEPT,IERR)
      IF (IERR.EQ.0.OR.IERR.EQ.2) ,ER10
      IF (INUNIT.EQ.59) INUNIT=2
      CALL ASSIGN(INUNIT,INFILE)
      CALL SETDATA(INUNIT)
  M21 GO TO LP00
C
C*****END SAVE
  C22 CALL QUIT(IQUIT)
C
C
C*****ERROR RETURNS
  ER5 WOT 59,FER5,ISCALE,INPUT(1),FINPUT(IDX)
 FER5 FORMAT(":UNACCEPTABLE VALUE:  ",I4,A11,":  ",E10.3)
      GO TO EROR
 ER10 WOT 59,FER10,INFILE
FER10 FORMAT(":CANNOT OPEN ",A10)
      GO TO EROX
FER14 FORMAT ("FISHY INPUT COUNT WRONG",I4,A11)
      GO TO EROR
 ER15 WOT 59,FER15,JZX,INPUT(1)
FER15 FORMAT ("EXPECT ONE WORD FOR DEPTH SIGNAL",I4,A11)
      GO TO EROR
 ER14 WOT 59,FER14,JZX,INPUT(1)
      GO TO EROR
C
 EROR IPROMPT=":?  ^"
      GO TO LP03
 EROX IPROMPT=":?  ^"
      GO TO LP02
      END
```

```
      SUBROUTINE BIFPLT
      USE DATACLICHE
    1 FORMAT("VOLUME ",I3," HAS DENSITY LE ZERO")
      DIMENSION XM(2),ZM(2)
C
C PLOTS GRID ON ONE FRAME WITH UX80
C
      IF(NFIR.NE.0) GO TO A20
      NFIR=1
      DO A10 K=1,ILT
  A10 ZN(K)=XMX(K)-XMN(K)
C
C USING ZN SLOPPING OVER INTO ZORD AS SORTING VARIABLE
C
      DXMN=AMINAF(ZN,1,ILT,1,M)
      XMXX=AMAXAF(XMX,1,ILT,1,M)
      XMNN=AMINAF(XMN,1,ILT,1,M)
      ZMXX=AMAXAF(ZMX,1,ILT,1,M)
      ZMNN=AMINAF(ZMN,1,ILT,1,M)
      IF(XMNN.EQ.0.) XMNN=-0.1*XMXX
      DXMX=XMXX-XMNN
      IF(ZMNN.LT.0.) ZMNN=1.1*ZMNN
      IF(ZMNN.GT.0) ZMNN=.9*ZMNN
      IF(ZMNN.EQ.0.) ZMNN=-0.1*ZMXX
      ZMXX=1.1*ZMXX
      DZMX=ZMXX-ZMNN
      IF(DXMX.EQ.0.) DXMX=.001
      CHARN=5.*DXMX/DXMN
      CALL DDERS(-1)
      XM1=XMXX+100.
      XM2=XMNN-100.
      XMXX=1.1*XMXX
      XMNN=1.1*XMNN
      XMXX=AMAX1(XMXX,XM1)
      XMNN=AMIN1(XMNN,XM2)
      DXMX=XMXX-XMNN
      RATIO=DZMX/DXMX
      CHARW=DXMX/CHARN
      IF(RATIO.GT.1.) CHARN=CHARN* RATIO
      IF(CHARN.LT.100.) CHARN= 100.
      CALL UXCHSIZ(CHARN)
      CHARW=DXMX/CHARN
      IF(RATIO.GT.1.) CHARW=CHARW*RATIO
      CW6=6.*CHARW
      ZMNN=-ZMNN
      ZMXX=-ZMXX
      XM(1)=XMNN
      XM(2)=XMXX
      ZM(1)=-ZMNN
      ZM(2)=-ZMXX
  A20 CALL DDERS (-1)
      DST=DZMX/10.
      STT=ZMXX-DST
      DXT=DXMX/8.
      DXTT=XMNN-(DXT/2.)
      DXT=XMNN-DXT
      DO A21 K=1,ILT
  A21 ZN(K)=ZMX(K)-ZMN(K)
C
C    NOW USING ZN FOR VOLUME HEIGHT
```

```
C
      CALL SETCH(1.,1.,0,0,0,0)
      WOT 100,2,KIT
    2 FORMAT(8A10)
      CALL UXCHSIZ(CHARN)
      DO A30 J=1,JZX
      ZA(J)=-ZA(J)
  A30 ZB(J)=-ZB(J)
      DO A35 J=1,ILT
  A35 ZMD(J)=-ZMD(J)
      SMIN=.1
      SMAX=.9
      TMIN=.1
      TMAX=.9
      IF (RATIO.GE.1.) GO TO A40
      TMAX=TMIN+.8*RATIO
      GO TO A50
  A40 SMAX=SMIN+.8/RATIO
  A50 CALL MAP(XMNN,XMXX,ZMXX,ZMNN,SMIN,SMAX,TMIN,TMAX)
C
C WE HAVE SET UP A GRID INCREASING X AND Z-DIMENSIONS,OTHER THAN
C BOUNDARIES BY 10 PERCENT. CHARACTER SIZE IS SPECIFIED SO THAT
C 4 CHARACTERS FIT IN X-DIMENSION OF NARROWEST VOLUME.
C SCALE IS SAME IN BOTH DIMENSIONS. Z IS SET NEGATIVE FOR PLOTS
C
      CALL GAXIS(XMNN,ZMXX,XMXX,ZMXX,1,0,0,"F7.0",0,XM)
      CALL GAXIS(XMNN,ZMNN,XMNN,ZMXX,0,0,0,"F7.0",0,ZM)
      CALL GAXIS(XMXX,ZMNN,XMXX,ZMXX,0,0,1,"F7.0",0,ZM)
      CALL LINEP(0.,ZMXX,0.,ZMNN,3)
      DO A60 J=1,JZX
  A60 CALL LINE (XA(J),ZA(J),XB(J),ZB(J))
      DO A70 K=1,ILT
      IF(RMD(K).LE.0.001) WOT 59,1,K
      IF(RMD(K).LE.0.001) GO TO A70
      IF(ZN(K).GT.CW6) GO TO A61
      STT=STT+DST
      CALL SETLCH(DXT,STT)
      WOT 100,F4,RMD(K)
      WOT 100,F3,K
      CALL DDERS(1)
      CALL LINEP (DXTT,STT,XMD(K),ZMD(K))
      CALL DDERS(-1)
      GO TO A70
  A61 Z1=ZMD(K)-CHARW
      X1=XMD(K)
      CALL SETLCH (X1,Z1)
C
C WRITING VOLUME NUMBER
C
      WOT 100,F1,K
   F1 FORMAT(I2)
      X1=X1-CHARW
      Z1=Z1+3.*CHARW
      CALL SETLCH(X1,Z1)
C
C WRITING DENSITY
C
      WOT 100,F2,RMD(K)
   F2 FORMAT (F4.2)
   F3 FORMAT (1X,I2)
```

```
   F4 FORMAT(F5.2)
  A70 CONTINUE
      DO A80 K=1,NP
      LA=INA(K,1)
      IF(LA.EQ.0) GO TO A72
      X1=XA(LA)
      IF(LA.GT.JZX) GO TO A80
      IF(X1.LT.TSTM) GO TO A80
      IF(X1.GT.TST) GO TO A80
      Z1=ZA(LA)
      GO TO A75
  A72 LA=INB(K,1)
      X1=XB(LA)
      IF(LA.GT.JZX) GO TO A80
      IF(X1.GT.TST) GO TO A80
      IF(X1.LT.TSTM) GO TO A80
      Z1=ZB(LA)
  A75 X1=X1-CHARW
      Z1=Z1-1.5*CHARW
      CALL SETLCH(X1,Z1)
C
C WRITING INTERSECTION NUMBER
C
      WOT 100,F1,K
  A80 CONTINUE
      DO A90 J=1,JZX
      ZA(J)=-ZA(J)
  A90 ZB(J)=-ZB(J)
      CALL FRAME
      RETURN
      END
```

```
      SUBROUTINE CINT
      CODE ANALYSIS
      USE DATACLICHE
      BIG=1.E+7
      SML=-BIG
C
C CHANGES LOCATION OF AN INTERSECTION
C TTY INPUT IS "INT", INTERSECTION NUMBER,NEW X,NEW Z SPACE DELIMITED
C USING NORMAL SCARP CONVENTION SO X IS INPUT(3),Z IS INPUT(4)
C IF NO SCARP USED
C
      IF(LER.GE.0) GO TO A10
      XX=XXX
      ZZ=ZZZ
      XXX=ZZZ=0.
      GO TO B01
C
C THIS WILL MAKE S CODE FORGET LAST INT CHANGE
C
C TEST FOR SCARP,L ,OR R
C
  A10 XXX=ZZZ=0.
      IF((IDEC(3).NE.2).OR.(IDEC(4).NE.2)) GO TO A50
      XX =FINPUT(3)
      ZZ=FINPUT(4)
      GO TO A100
  A50 IF (IDEC(3).EQ.3)  ,A70
C
C X IS ON SCARP UNLESS L OR R
C
      LET=INPUT(3)-40B
      IF (LET.EQ.12)  ,A51
      XX =SML
      ZZ=FINPUT(4)
      GO TO A100
  A51 IF (LET.EQ.18)  ,A52
      XX=BIG
      ZZ=FINPUT(4)
      GO TO A100
  A52 ZZ=FINPUT(4)
      ZT=ZZ
      CALL LOCATE (3,LET,ZT,2)
      XX=ZT
      GO TO A100
  A70 XX=FINPUT(3)
C
C HERE Z IS ON SCARP
C
      ZT=XX
      CALL LOCATE (3,LET,ZT,1)
      ZZ=ZT
      GO TO A100
C
C NOW WE HAVE NEW X AND Z
C
 A100 IS=FINPUT(2)
      IT=0
  B01 DO B10 K=1,6
      IF(INA(IS,K).EQ.0) GO TO B20
      NL=INA(IS,K)
```

```
      XXX=XA(NL)
      ZZZ=ZA(NL)
      XA(NL)=XX
      IT=IT+1
  B10 ZA(NL)=ZZ
      GO TO ERR1
  B20 DO B30 K=1,6
      IF(INB(IS,K).EQ.0) GO TO B40
      NL=INB(IS,K)
      IF(ZZZ.EQ.0.)  ZZZ=ZB(NL)
      IF(XXX.EQ.0.) XXX=XB(NL)
      XB(NL)=XX
      IT=IT+1
  B30 ZB(NL)=ZZ
      GO TO ERR1
  B40 IF(IT.LT.2) GO TO ERR2
      RETURN
      ENTRY CDEN
C
C CHANGES DENSITY
C INPUT IS  "DEN" ,VOLUME NUMBER,NEW DENSITY, SPACE DELIMITED
C
      NVV=FINPUT(2)
      DDR=RMD(NVV)-FINPUT(3)
      DO C20 J=1,JZX
      IF(ILV(J).EQ.NVV) DRHO(J)=DRHO(J)+DDR
  C20 IF(IRV(J).EQ.NVV) DRHO(J)=DRHO(J)-DDR
      RETURN
 ERR1 WOT 59,FR1,IS
      RETURN
 ERR2 WOT 59,FR2,IS
      RETURN CDEN
  FR1 FORMAT(" TOO MANY LINES AT INT ",I3)
  FR2 FORMAT(" NOT ENOUGH LINES AT INT ",I3)
      END
```

```
*     FORTRAN            DATA
      SUBROUTINE DATA(WORD,TYPE)
      LCM DATA
C
C     SETDATA DECLARATIONS
      TYPE BYTE MEOM(6),MMSG(6)
      DIMENSION MMSG(1)
      DIMENSION MEOM(1)
      DIMENSION MSG(8)
      EQUIVALENCE (MMSG(1),MSG(1))
      EQUIVALENCE (NEOM,MEOM)
      DATA (NEOM=40777740020400000000B)
C
C     DATA DECLARATIONS
      ABSOLUTE IAA(0)
      ADDRESS BRANO
      ADDRESS NMEADD
      ADDRESS LOOB,L40B,L33B,L01B,L04B,L06B,L02B,L07B,
     1 LI0,LI1,LI2,LI3,LI4,LI5,LI6,LI7,LI8,LI9,
     2 LOR,LOP,LOM,LOE,LOB,LOD,L14B
      TYPE BYTE KOL(6),LCHAR(6),MNES(6)
      TYPE ADDRESS LOCS,BRAN1
      ADDRESS BRAN2
      ADDRESS INTOB,INTOR
      TYPE ADDRESS INTA,INTB,MOCS
      DIMENSION FMT(7)
      DIMENSION IAA(1)
      DIMENSION NMEADD(1)
      DIMENSION TYPE(1)
      DIMENSION MANT(3),LINT(3),NSIGN(2)
      DIMENSION MOCS(30),MNES(30)
      DIMENSION KOL(10),LCHAR(10)
      DIMENSION IWORD(1),WORD(1),NTYPE(1)
      DIMENSION NSYMB(8)
      COMMON/CDATA1/LSTLINE(8),LSCAN(10)
      COMMON/CDATA0/LOCS(1),L40B,L33B,L01B,L04B,L06B,L02B,L07B,
     1 LI0,LI1,LI2,LI3,LI4,LI5,LI6,LI7,LI8,LI9,
     2 LOR,LOP,LOM,LOE,LOB,LOD,L14B
      COMMON/CDATA2/MNEMONIC,NEXTM,NWORD
      EQUIVALENCE (TYPE(1),NTYPE(1))
      EQUIVALENCE (LSCAN(1),KOL(1)),(NSYMB,LCHAR(1))
      EQUIVALENCE (ISIGN,SKEPT),(SYMB,NSYMB)
      EQUIVALENCE (WORD,IWORD)
      EQUIVALENCE (NMEADD,WORD)
      EQUIVALENCE (MAX,XMAX),(LOCS(1),LOOB),(FLOT,IFX)
      DATA (FMT=10B,5(00B),1411B)
      DATA (LSCAN(9)=10H$ $ $ $ $  )
      DATA (LSTLINE=0,0,0,0,0,0,0,0)
      DATA (MNEMONIC=0),(NEXTM=6R.INIT.)
      DATA (NSYS=1601B)
      DATA (BRAN2=.LOC.E500)
      DATA (NWORD=0)
      DATA (INTA=.LOC.C500)
      DATA (MNES= 00B, 40B, 33B,01B, 1R$, 1R&, 1R", 1R',
     1 1R0, 1R1, 1R2, 1R3, 1R4, 1R5, 1R6, 1R7, 1R8, 1R9,
     1 1RR, 1R+, 1R-, 1RE, 1RB, 1R., 1R, )
      DATA (MOCS=.LOC.A500, .LOC.C550, 2(.LOC.B552), .LOC.Z500,
     1        .LOC.E500, .LOC.C511, .LOC.C512,
     2 8(.LOC.C501), 2(.LOC.C502),
     2 .LOC.C500, 2(.LOC.C506), 2(.LOC.C500), .LOC.C510,
```

```
     1 .LOC.C500 )
C      NOTE:  COMMA OPTION DISABLED ABOVE
       DATA (IFMT=-1000000)
       DATA (LINT=20,2,3)
       DATA (MASK1=77770000000000000000B)
       DATA (MSG=36360000004002040000B)
       DATA (MASK2=00010000000000000000B)
       DATA (MASK3=36000000000000000000B)
C
C      DATALST DECLARATIONS
       DIMENSION LST(1),LABS(1)
       DATA (NLST=0),(LSTSKP=1)
C
       CODE ANALYSIS
C
       IGO=IQ8QARGS+2
       IF (IGO.GT.5) IGO=4
       GO TO (A300,B300,C300,B300,E300), IGO
C      IGO = 1   SCAN RETURN
C          = 2    DATA RETURN
C          = 3    NAMELIST INPUT
C          = 4    CALL DATA
C          = 5    CALL SCAN
C
C***********
       ENTRY DATALST(NOLST,LST,LABS)
       NLST=NOLST
       LSTSKP=1
       IF (IQ8QARGS.EQ.3) LSTSKP=LABS
       RETURN DATALST
C
C
CALL DATA(NAMELIST)
 C300 ISCAN=-1
       IFLOAT=0
       NWORD=IWORD(2)-1
       IF (MASK3.INT.IWORD(3).EQ.1H*) ,A301
       IFLOAT=1
       GO TO A301
C
CALL SCAN(N,NTYPE,WORD)
C      NTYPE=3 ASCII
C           =1 INTEGER, OCTAL
C           =2 FLOATING
C           =4 LAST WORD +1
C           =5 SKIPPED FIELD (COMMA)
C
 E300 MAX=IWORD
       ISCAN=1
       IQ8QARGS=-1
       RETURN
C
CALL DATA(V1,N1,V2,N2,...)
C
C      DATA RETURN
 B300 MAX=NTYPE
       ISCAN=0
       IQ8QARGS=0
C
C      SCAN RETURN
```

```
 A300 IFLOAT=0
      NWORD=0
      IF (MAX.INT.MASK1.NE.0) ,A301
      MAX=XMAX
      IFLOAT=1
 A301 MNEMONIC=NEXTM
      GO TO BRAN2
C
C
C*****READ INPUT
 E500 DO E511 I=1,8
      LSTLINE(I)=LSCAN(I)
 E511 LSCAN(I)=00B
      IF (ITTYSEN.EQ.59) ,E510
      IER=0
      NSYS=1601B
 E501 KEEP=0
      CALL GOB(NSYS,KEEP,1000B,LSCAN)
      IF (KEEP.EQ.1) ,E502
      IF (MSG(1)) ,E512,
      KEEP=0
      CALL GOB(1412B,KEEP,10B,MSG)
      IF (KEEP.EQ.1) E512,
      NSYS=1600B
      GO TO E501
 E512 IER=IER+1
      IF (IER.GT.5) E509,E501
C
 E510 READ (ITTYSEN,F2) (LSCAN(I),I=1,8)
   F2 FORMAT(8A10)
      IF (EOF,ITTYSEN) ,E502
 E509 NSYMB=3REOF
      GO TO 760
C
C
C     INITIALIZE
 E502 NCOL=0
      ILIT=0
      IF(ILIT.EQ.0) GO TO Q651
C
C**********
C**********
C     START NCOL LOOP
      DO A500 NCOL=1,82
      KHAR=KOL(NCOL)
      KINT=KHAR-20B
      BRAN1=.LOC.A500

      DO 520 J=1,25
      IF (KHAR.EQ.MNES(J)) GO TO LOCS(J)
  520 CONTINUE
C
C
C     ASCII
 C500 JTYPE=5
      LOR=INTOR=INTA
      IF (NCHAR.GE.9) ,B500
      IF((KOL(NCOL+1)).NE.00B)  ILIT=-1
      BRAN1=.LOC.A600
C
```

```
 B500 NCHAR=NCHAR+1
      LCHAR(NCHAR)=KHAR
      LOOB=.LOC.A600
      L33B=L01B=L04B=L06B=L14B=.LOC.C552
      IF (JTYPE.GT.ITYPE) ITYPE=JTYPE
      GO TO BRAN1        $$$ A500,A600
C
C     INTEGER
 C502 INTOB=INTA
 C501 NOCT=(NOCT.SHL.3).UN.KINT
      MSHF=MSHF-IDEC
      LOP=LOM=.LOC.C505
      LOE=.LOC.C509
      LOB=INTOB
 G501 LOR=INTOR
C
 C503 MANT(IEXP)=MANT(IEXP)*10+KINT
      INT=INT+1
      JTYPE=1
      IF (INT.GT.LINT(IEXP)) C500,B500
C
C     + OR -
 C505 IEXP=2
      INTB=.LOC.G501
      GO TO B506
 C506 INTB=.LOC.C502
 B506 NSIGN(IEXP)=-(KHAR-14B)
      JTYPE=4
      KL=19
      KU=23
      GO TO Q600
C
C     RESET BRANCHES
C     +   R + - E B
C     -   R + - E B
C     R   R + - E B .
C     B     + - E B .
C     E         E B .
C     .           B .
 N600 JTYPE=4
 P600 INTB=.LOC.C503
 M600 KU=24
 Q600 DO K600 K=KL,KU
 K600 LOCS(K)=INTA
      DO L600 K=9,18
 L600 LOCS(K)=INTB
      INT=0
      GO TO B500
C
C     OCTAL B
 C507 INTB=INTA
      JTYPE=2
      KL=20
      GO TO M600
C
C     REPEAT R
 C508 IEXP=3
      KL=19
      GO TO P600
C
```

```
C       E
 C509 IEXP=2
      LOP=LOM=.LOC.C505
      KL=22
      GO TO N600
C
C       DEC (.)
 C510 IDEC=1
      INTB=.LOC.C502
      KL=23
      JTYPE=4
      GO TO M600
C
C       (") LF JUSTIFIED
 C511 ILIT=-1
      GO TO A511
C
C       (') RT JUSTIFIED
 C512 ILIT=1
 A511 JTYPE=4
      DO D511 K=9,25
 D511 LOCS(K)=INTA
      L02B=L07B=.LOC.B511
      GO TO A500
 B511 L02B=.LOC.C511
      L07B=.LOC.C512
      ILIT=ILIT*2
      GO TO A600
C
C       (COMMA 14B)
 C514 ITYPE=5
      GO TO 650
C
C       40B
 C550 NCOL=NCOL+1
      GO TO A500
C
C       $ (04B EOM)
C       GO TO Z500
C
C
C       (33B ESC)
 C552 NCOL=NCOL-1
      GO TO A600
 B552 NSYMB=01B
      BRAN2=.LOC.Q651
      GO TO 763
C
C       &
C       GO TO E500
C
C       00B
 A600 GO TO (634,632,633,633,631), ITYPE
C
C
C*****PROCESS WORD
C
C       OCTAL
  632 IFX=NOCT
      AFMT="1X,O10,"
```

```
 A632 ITYPE=1
      GO TO 650
C
C     INTEGER
  634 IFX=NSIGN*MANT
      AFMT="1X,I10,"
      IF (IFLOAT) ,A632,
      FLOT=IFX
      GO TO 636
C
C    FLOAT
  633 FLOT=NSIGN*MANT
      SYMB=NSIGN(2)*MANT(2)+MSHF
      SKEPT=10.**ABS(SYMB)
      IF (SYMB) ,636,635
      FLOT=FLOT/SKEPT
      GO TO 636
  635 FLOT=FLOT*SKEPT
  636 AFMT="1X,E10.3,"
  637 ITYPE=2
      GO TO 650
C
C     ASCII
C     SHIFT RIGHT
  631 ITYPE=3
      IF (ILIT) Q642,,
      LSYMB=NSYMB
      DO 640 J=1,10
      IF (LCHAR(1).EQ.OOB) A642,
  640 NSYMB=NSYMB.SHL.6
C
C*****CHECK MNEMONIC LIST
 A642 BRAN2=.LOC.Q651
      IF (NSYMB.EQ.6R.DBUG.) ,C642
      IFMT=0
      GO TO E500
 C642 IF (ISCAN) ,763,B642
C
C     NAMELIST
      DO 621 K=1,1000,3
      KEEP=IWORD(K)
      IF (KEEP.EQ.6H$E$N$D ) 763,
      IF (LSYMB.EQ.KEEP) A621,
  621 CONTINUE
 A621 NWORD=IWORD(K+1)
      KEEP=IWORD(K+2)
      IF (MASK2.INT.KEEP.EQ.MASK2) GO TO NMEADD(K+1)
      IFLOAT=0
      IF (MASK3.INT.KEEP.EQ.1H*) ,Q651
      IFLOAT=1
      GO TO Q651
C
C     DATA
C     GO TO 763
C
C     SCAN
 B642 BRAN2=.LOC.P642
      IF (NWORD) Q642,Q642,
      DO 644 K=1,NLST,LSTSKP
```

```
      IF (LST(K).EQ.00B ) Q642,
      IF (NSYMB.EQ.LST(K)) 763,
  644 CONTINUE
      GO TO Q642
C
 P642 NWORD=0
 Q642 IFX=NSYMB
      AFMT="1X,A10,"
C
C
C**********
C**********
C     STORE IN PROPER ARRAYS
  650 BRAN2=.LOC.Q651
      K=0
      IF(K.EQ.0) GO TO 666
C
      DO 665 K=1,MANT(3)-1
  666 NWORD=NWORD+1
      IFMT=IFMT+1
      IF (IFMT.GT.0) FMT(IFMT+1)=AFMT
      IF (ISCAN) ,669,A668
      IF (ITYPE.NE.5) IAA(NWORD)=IFX
      ASSIGN 665 TO BRANO
      GO TO 812
C
 A668 NTYPE(NWORD)=ITYPE
  669 IF (ITYPE.NE.5) IWORD(NWORD)=IFX
      ASSIGN 667 TO BRANO
      GO TO 811
  667 IF (NWORD.LT.MAX) ,1001
  665 CONTINUE
C
C     INITIALIZE
 Q651 MANT=MANT(2)=MANT(3)=NOCT=MSHF=0
      NSYMB=NCHAR=INT=IDEC=ITYPE=0
      IEXP=NSIGN=NSIGN(2)=1
      IF (IABS(ILIT).GT.1) ILIT=0
      INTOB=.LOC.C507
      INTOR=.LOC.C508
      DO 646 K=1,25
  646 LOCS(K)=MOCS(K)
C
C
 A500 CONTINUE
 Z500 NSYMB=1R$
      IF (ISCAN) E500,E500,
C
C
C
C
C*****RETURN
C
  760 BRAN2=.LOC.E500
  763 ASSIGN 1003 TO BRANO
      IF (ISCAN) ,1002,764
      IF (IFMT) 1003,1003,815
C
  764 IF (NWORD.LT.MAX) ,1002
      NTYPE(NWORD+1)=4
```

```
      IWORD(NWORD+1)=NSYMB
 1002 IF (IFMT) ,,814
 1003 NEXTM=NSYMB
 1000 RETURN
C
 1001 NSYMB=1R/
      GO TO 763
C
C**********
C     DEBUG.
  811 IF (IFMT.EQ.5) ,BRANO
  814 WOT 59,FMT,(IWORD(KK),KK=NWORD-IFMT+1,NWORD)
  813 IFMT=0
      GO TO BRANO
  812 IF (IFMT.EQ.5) ,BRANO
  815 WOT 59,FMT,(IAA(KK),KK=NWORD-IFMT+1,NWORD)
      GO TO 813
C
C
C
C*****
      ENTRY SETDATA(INUNIT,IPROMPT)
      TYPE BYTE MPROMPT(6)
      DIMENSION IPROMPT(1),MPROMPT(1)
      EQUIVALENCE (IPROMPT(1),MPROMPT(1))
C
      GO TO (1,8,3), IQ8QARGS+1
    3 CALL SETPROMPT(IPROMPT)
      GO TO 2
    8 IF (INUNIT.EQ.59) MSG(1)=00B
    2 ITTYSEN=INUNIT
    1 BRAN2=.LOC.E500      $$$ CLEAR BUFFER
      RETURN SETDATA
C
      ENTRY SETPROMPT(IPROMPT)
      J=0
      IB=0
      DO 4 I=1,80
      KEEP=MPROMPT(I)
      IF (KEEP.EQ.1R ) A2,
      IB=0
      IF (KEEP.EQ.1R^) A3,
      IF (KEEP.EQ.1R_) A1,
   B1 J=J+1
      MMSG(J)=KEEP
      GO TO 4
C
   A2 IB=IB+1
      IF (IB.GT.5) ,B1
      J=J-5
   A3 KL=4
      KU=6
      I=85
      GO TO B2
C
   A1 KL=1
      KU=6
   B2 CONTINUE
      DO 5 II=KL,KU
      J=J+1
```

```
5 MMSG(J)=MEOM(II)
4 CONTINUE
  RETURN SETPROMPT
  END
```

```
      SUBROUTINE GPLOT
      USE DATACLICHE
      DIMENSION XM(2),ZM(2)
C
C PLOTS MEASURED LOG DENSITY
C       MEASURED GRAVIMETRIC DENSITY
C       CORRECTED GRAVIMETRIC DENSITY
C ON SAME SCALE. CHANGES CORRECTED DENSITY DOT SPACEING FOR EACH RUN
C
      DSM=DPT(1)
      DBG=DPT(LDEP)
      DNM=XM(1)=1.
      DNX=XM(2)=3.
      ZM(1)=DSM
      ZM(2)=DBG
      CALL MAP (DNM,DNX,DBG,DSM,.2,.8,.1,1.)
      CALL GAXIS (DNM,DBG,DNX,DBG,1,0,0,"F7.2",0,XM)
      CALL GAXIS (DNM,DSM,DNM,DBG,0,0,1,"F7.0",0,ZM)
C
C NOW WE HAVE AXES WITH Z INCREASING DOWN
C
      LS=LER+2
      LSS=10-LS
      SS=LSS
      DO 100 J=1,JDP
      CALL LINE(DDR(J),DD1(J),DDR(J),DD2(J))
  100 CALL LINEP(RCOR(J),DD1(J),RCOR(J),DD2(J),LS)
      LDD=LDEP-1
      DO 200 L=1,LDD
  200 CALL LINEP(RHL(L),DPT(L),RHL(L+1),DPT(L),1)
      DO 300 L=2,LDEP
  300 CALL LINEP(RHL(L),DPT(L-1),RHL(L),DPT(L),1)
      CALL MAP (0.,10.,0.,10.)
      CALL SETLCH (8.1,9.5,0,0,0,0)
      WOT 100,1
      CALL SETLCH(8.1,9.)
      WOT 100,2
      CALL SETLCH(8.1,SS)
      WOT 100,3,LER
      CALL LINE(9.,9.5,10.,9.5)
      CALL LINEP(9.,9.,10.,9.,1)
      CALL LINEP(9.,SS,10.,SS,LS)
      CALL SETLCH (.5,.1)
      WOT 100,4,KIT
      CALL FRAME
      RETURN
    1 FORMAT("GRAV")
    2 FORMAT(" LOG")
    3 FORMAT ("CORR",I2)
    4 FORMAT(8A10)
      END
```

```
      SUBROUTINE GEXEC
C
C        MANAGES CALCULATION OF G
C
      USE DATACLICHE
      DIMENSION RCC(100)
      KEND=JZX
      DO 50 K=1,JZX
      ZA(K)=ZA(K)+ZSHFT
   50 ZB(K)=ZB(K)+ZSHFT
   60 DO 70 K=1,KEND
      XA(K)=XA(K)+XSHFT
   70 XB(K)=XB(K)+XSHFT
C
C   CALCULATE CORRECTION TO FAG
C
      E1=0.
      DO 90 L=1,2
      G(L)=0.
      DO 85 K=1,KEND
      X1=XA(K)
      X2=XB(K)
      Z1=ZA(K)-E1
      Z2=ZB(K)-E1
      G(L)=G(L)+DRHO(K)*DG(X1,X2,Z1,Z2)
   85 CONTINUE
      E1=-15.
   90 CONTINUE
      FCOR=0.795165*(G(1)-G(2))
C
C     CALCULATE SUBSURFACE GRAVITY
C
      DO 300 L=1,LDEP
      G(L)=0.
      DO 200 K=1,KEND
      X1=XA(K)
      X2=XB(K)
      Z1=ZA(K)-DPT(L)
      Z2=ZB(K)-DPT(L)
  180 G(L)=G(L)+DRHO(K)*DG(X1,X2,Z1,Z2)
  200 CONTINUE
  300 CONTINUE
C
C     CALCULATE DENSITY
C
      WOT 3,2
      WOT 3,6,FCOR
      DO 400 L=2,LDEP
      IF (DPT(L).EQ.DPT(L-1)) GO TO 400
      R(L)=-11.92748*(G(L)-G(L-1))/(DPT(L)-DPT(L-1))
      CORR(L)=RHOL(L)-R(L)-FCOR
  400  CONTINUE
    1 FORMAT(8A10)
    2 FORMAT("   DEPT 1    DEPT 2    RHO       RHO NOM    RHO CORR CORR")
    3 FORMAT(2F10.1,4F10.4,E12.4)
    4 FORMAT(E13.4,3E12.4)
    5 FORMAT(///"   X-POSITION            G       DG")
    6 FORMAT ("   VAG     ",40X,F10.4)
    7 FORMAT (3E15.6)
    8 FORMAT( /,"NOTE: CORR= RHO NOM - RHO -VAG CORR")
```

```
    9 FORMAT("   DEPTH 1       DEPTH 2       CORR          RHO CORR ")
C
C CALCULATE CORRECTION TO MEASURED GRAVIMETRIC DENSITY
C MUST MATCH CALCULATED AND MEASURED DEPTHS
C PREVIOUSLY DID CALCULATIONS ON ALL MEASURED DEPTH POINTS
C BUT SOME MEASURED PAIRS MAY COVER MORE THAN ONE CALCULATED PAIR
C
      IF(LLOG.EQ.0) GO TO 475
      DO L100 J=1,JDP
      L1=0
      IF(DD1(J).EQ.DD2(J)) GO TO L100
      L2=0
      DDX=AMAX1(DD1(J),DD2(J))
      DDM=AMIN1(DD1(J),DD2(J))
      RC=0.
C
C PICKED TOP AND BOTTOM OF INTERVAL. EACH SHOULD BE A
C CALCULATED DEPTH
C
      DO L20 L=1,LDEP
      IF(L1.GT.0) GO TO L05
      DIM=DDM-DPT(L)
      IF(DIM.LT.1.) L1=L
  L05 IF(L2.GT.0) GO TO L19
      DIX=DDX-DPT(L)
      IF(DIX.GT.1.) GO TO L19
      L2=L
  L19 IF((L1.GT.0).AND.(L2.GT.0)) GO TO L30
  L20 CONTINUE
C
C NOW WE HAVE L VALUES OF ENDPOINTS
C L2 IS NOT NECESSARILY GRETER THAN L1
C
  L30 LX=MAX0(L2,L1)
      LM=MIN0(L2,L1)
      DL=LX-LM
      IF(DL-1) L100,   ,L40
      RCOR(J)=DDR(J)+CORR(LX)
      GO TO L100
  L40 L3=LM+1
      DO L50 L=L3,LX
  L50 RC=RC+CORR(L)*(DPT(L)-DPT(L-1))
C
C EQUAL DEPTH VALUES WERE REMOVED IN LOGDEN
C
      RCOR(J)=DDR(J)+RC/(DPT(LX)-DPT(LM))
 L100 RCC(LX)=RCOR(J)
C
C NOW WE HAVE THE CORRECTED GRAVIMETRIC DENSITIES
C
      WOT 4,1,KIT
      WOT 4,9
  475 DO 477 L=2,LDEP
      WOT 4,4,DPT(L-1),DPT(L),CORR(L),RCC(L)
      WOT 3,3,DPT(L),DPT(L-1),R(L),RHOL(L),RCC(L),CORR(L),G(L)
  477 CONTINUE
      WOT 3,8
      IF(LZEP.LE.0) RETURN
C
C     CALCULATE SURFACE GRAVITY
```

```
C
      WOT 3,5
      DDG=0.
      DO 600 L=1,LZEP
      G(L)=0.
      DO 500 K=1,JZX
      X1=XA(K)-ZPT(L)
      X2=XB(K)-ZPT(L)
      Z1=ZA(K)
      Z2=ZB(K)
  480 G(L)=G(L)+DRHO(K)*DG(X1,X2,Z1,Z2)
  500 CONTINUE
       IF(L.GT.1) DDG=G(L)-G(L-1)
  600 WOT 3,7,ZPT(L),G(L) ,DDG
      RETURN
      END
      FUNCTION DG (X1,X2,Z1,Z2)
C
C     CALCULATES GRAVITY AT A POINT CAUSED BY A SEGMENT
C
      DG=0.
   10 IF ((X1.EQ.0.).AND.(Z1 EQ.0.)) GO TO 50
      IF ((X2.EQ.0.).AND.(Z2.EQ.0.)) GO TO 50
      THO=1.570796327
      TH1=SIGN(THO,X1)
      TH2=SIGN(THO,X2)
   20 IF (Z2.NE.0.) TH2=ATAN2(X2,Z2)
      IF (Z1.NE.0.) TH1=ATAN2(X1,Z1)
      T2=TH2-TH1
      PI=2*THO
      IF(ABS(T2).LE.PI) GO TO 25
      TX=2.*PI-ABS(T2)
      T2=-SIGN(TX,T2)
   25 IF(Z2.NE.Z1) GO TO 40
   30 DG=-Z1*T2*.01334
      GO TO 50
   40 A=(X2-X1)/(Z2-Z1)
      B=(X1*Z2-X2*Z1)/(Z2-Z1)
      AB=B/(1.+A**2)
      T1=LOGF((X2**2+Z2**2)/(X1**2+Z1**2))
      DG=AB*(.5*T1+A*T2)*.01334
   50 RETURN
    1 FORMAT(8E12.4)
      END
```

```
*       FORTRAN
      SUBROUTINE LOCATE(K,LET,XZ,N)
      USE DATACLICHE
      BIG=100000.
      ISIG(12)=ISIG(18)=0
      IF(ISIG(LET).EQ.0) GO TO ERO1
      IF(N.EQ.1) ,ZO1
C
C      THIS IS FOR X INPUT
C
      IF(TTH(LET).GT.BIG) GO TO ERO2
      DX=XZ-XO(LET)
      ZZ=ZO(LET)-DX*TTH(LET)
      IF(K-2) ,N1,N3
      ZA(JZX)=ZZ
      RETURN
   N1 ZB(JZX)=ZZ
      RETURN
   N3 XZ=ZZ
      RETURN
C
C        THIS IS FOR Z INPUT
C
  ZO1 IF(TTH(LET).EQ.0) GO TO ERO3
      DZ=XZ-ZO(LET)
      XX=XO(LET)-DZ/TTH(LET)
      IF(K-2)  ,N2,N4
      XA(JZX)=XX
      RETURN
   N2 XB(JZX)=XX
      RETURN
   N4 XZ=XX
      RETURN
 ERO1 WOT 59,FER1,LET,JZX
 FER1 FORMAT("ILLEGAL SCARP TRIED LOCATE",A11,I5)
      CALL QUIT(1)
 ERO2 WOT 59,FER2,LET,JZX
 FER2 FORMAT ("INPUT Z AND HORIZONTAL LINE ",A11,I5)
      CALL QUIT(1)
 ERO3 WOT 59,FER3,LET,JZX
 FER3 FORMAT("INPUT X AND VERTICAL LIN ",A11,I5)
      CALL QUIT(1)
      END
```

```
      SUBROUTINE LOGDEN
      USE DATACLICHE
C
C     ORDERS DEPTH,CALCULATES LOG DENSITY BETWEEN DEPTHS
C
C     FIRST ORDER DEPTHS
C
      DO 50 K=1,LDEP-1
      LSAV=K
      DO 25 J=K+1,LDEP
      IF(DPT(J).LT.DPT(LSAV)) GO TO 22
      IF((DPT(J).NE.DPT(K-1)).OR.(K.EQ.1)) GO TO 25
   22 LSAV=J
   25 CONTINUE
      DSAV=DPT(LSAV)
      DPT(LSAV)=DPT(K)
      DPT(K)=DSAV
   50 CONTINUE
C
C      NOW DEPTHS ARE ORDERED
C NOW ELIMINATE DUPLICATES
C
      CORR(1)=DPT(1)
      LORR=1
      DO 60 K=2,LDEP
      IF(DPT(K).EQ.CORR(LORR)) GO TO 60
      LORR=LORR+1
      CORR(LORR)=DPT(K)
   60 CONTINUE
      LDEP=LORR
      DO 65 K=1,LDEP
      DPT(K)=CORR(K)
   65 CORR(K)=0.
C
C      SEARCH LINES FOR THOSE CROSSING HOLE AND STORE DENSITIES
C
      XAFT=XSHFT
      ENTRY VOLDEN
      KK=0
      DO 100 N=1,JZX
      DEL=1.
      IF((XA(N).LE.XAFT).AND.(XB(N).GE.XAFT))    ,100
      IF(XA(N).EQ.XB(N)) GO TO 70
      ZCT=(ZB(N)-ZA(N))*(XAFT-XA(N))/(XB(N)-XA(N))
      ZCT=ZA(N)+ZCT
      KK=KK+1
      ZORD(KK)=ZCT
      IF(XA(N).GT.XB(N)) DEL=-1.
      NRT(KK)=N
      CRHO(KK)=DEL*DRHO(N)
   70 IF((XA(N).EQ.XAFT).OR.(XB(N).EQ.XAFT)) WOT 3,800
  100 CONTINUE
C
C       WE HAVE SELECTED ALL LINES WHOSE INTERSECTION WITH HOLES IS
C       ABOVE MAX DEPTH. NOW ORDER THEM
C
      DO 150 L=1,KK-1
      LSAV=L
      DO 125 J=L+1,KK
      IF(ZORD(J).LT.ZORD(LSAV))   ,125
```

```
      LSAV=J
  125 CONTINUE
      NSAV=NRT(LSAV)
      DSAV=CRHO(LSAV)
      ZSAV=ZORD(LSAV)
      CRHO(LSAV)=CRHO(L)
      ZORD(LSAV)=ZORD(L)
      NRT(LSAV)=NRT(L)
      CRHO(L)=DSAV
      ZORD(L)=ZSAV
      NRT(L)=NSAV
  150 CONTINUE
C
C       NOW SUM DENSITIES TO GET DENSITY BELOW EACH LINE
C
      DO 200 L=2,KK
  200 CRHO(L)=CRHO(L)+CRHO(L-1)
      IF(EX.NE.0.) RETURN
C
C      ORDERED SET OF DENSITIES AND DEPTHS
C      NOW GET AVERAGE DENSITIES BETWEEN DEPTHS
C       START BY FINDING LINES BELOW TOP DEPTH AND ABOVE BOTTOM
C      DEPTH FOR EACH DEPTH PAIR. FIRST TOP ONE
C
      DO 400 L=1,LDEP-1
      MM=NN=MN=0
      IF(DPT(L).LE.ZORD(1))  ,240
      IF(DPT(L+1).LE.ZORD(1))  GO TO 400
      MM=1
      RH1=0.
      Z1=ZORD(1)-DPT(L)
      GO TO 270
  240 DO 250 K=2,KK
      IF(ZORD(K).GT.DPT(L))  ,250
      MM=K
      RH1=CRHO(K-1)
      Z1=ZORD(K)-DPT(L)
      GO TO 270
  250 CONTINUE
      GO TO 410
  270 DO 300 K=2,KK
      IF(ZORD(K).GE.DPT(L+1))  ,300
      NN=K-1
      MN=K
      RH2=CRHO(NN)
      Z2=DPT(L+1)-ZORD(NN)
      GO TO 320
  300 CONTINUE
      NN=KK
      RH2=0.
      Z2=DPT(L+1)-ZORD(KK)
C
C     NOW WE HAVE MM--LINE BELOW THE TOP MEASURING POINT
C      AND NN--LINE ABOVE BOTTOM POINT
C     NOW GET DENSITIES
C
  320 IF (MM.EQ.MN)  ,330
      RHOL(L+1)=RH1
      GO TO 400
  330 PRHO=0.
```

```
      IF(MM.EQ.NN)   ,350
  340 RHOL(L+1)=(PRHO+RH1*Z1+RH2*Z2)/(DPT(L+1)-DPT(L))
      GO TO 400
  350 DO 360 J=MM,NN-1
      DZ=ZORD(J+1)-ZORD(J)
  360 PRHO=PRHO+DZ*CRHO(J)
      GO TO 340
  400 CONTINUE
  410 RETURN
  800 FORMAT ("WARNING--LINE ENDS ON HOLE DENSITY BAD ")
      END
```

```
*       FORTRAN
        SUBROUTINE UNCODE
        USE DATACLICHE
        BIG=1.E+7
        SML=-1.E+7
        IF(IDEC(5).EQ.3) GO TO SC01
        DRHO(JZX)=FINPUT(5)
        LET=0
        IF(IDEC(1).EQ.3) ,NX01
C
C       FIRST X IS ON SCARP
C
        IF(IDEC(2).EQ.3) GO TO ER01
        LET=INPUT(1)-40B
        IF(LET.EQ.12)   ,X105
C
C        LEFT X INFINITE --LETTER IS L
C
        XA(JZX)=SML
        GO TO NX02
 X105   IF (LET.EQ.18)   ,X110
C
C     RIGHT X INFINITE--LETTER IS R
C
        XA(JZX)=BIG
        GO TO NX02
 X110   ZA(JZX)=FINPUT(2)
        CALL LOCATE(1,LET,ZA(JZX),2)
        GO TO NX03
 NX01   XA(JZX)=FINPUT(1)
        IF(IDEC(2).EQ.3)   ,NX02
C
C       FIRST Z IS ON THE SCARP
C
        LET=INPUT(2)-40B
        CALL LOCATE (1,LET,XA(JZX),1)
        GO TO NX03
 NX02   ZA(JZX)=FINPUT(2)
 NX03   IF (IDEC(3).EQ.3)   ,NX04
C
C         SECOND X IS ON THE SCARP
C
        IF (IDEC(4).EQ.3) GO TO ER01
        LET=INPUT(3)-40B
        IF(LET.EQ.12)    ,X205
C
C     LEFT X INFINITE--LETTER IS L
C
        XB(JZX)=SML
 X203   ZB(JZX)=FINPUT(4)
        RETURN
 X205   IF (LET.EQ.18)   ,X210
C
C    RIGHT X IS INFINITE--LETTER R
C
        XB(JZX)=BIG
        GO TO X203
 X210   ZB(JZX)=FINPUT(4)
        CALL LOCATE (2,LET,ZB(JZX),2)
        RETURN
```

```
 NX04 IF (IDEC(4).NE.3) GO TO NX05
C
C        SECOND Z IS ON SCARP
C
      LET=INPUT(4)-40B
      XB(JZX)=FINPUT(3)
      CALL LOCATE (2,LET,XB(JZX),1)
      RETURN
 NX05 IF (LET.EQ.0) GO TO ER03
      XB(JZX)=FINPUT(3)
      ZB(JZX)=FINPUT(4)
      RETURN
C
C       THIS CARD IS A SCARP LINE INPUT
C
 SC01 LET=INPUT(5)-40B
      JZX=JZX-1
      IF((LET.EQ.12).OR.(LET.EQ.18)) GO TO ER04
      ITST=IDEC(1)+IDEC(2)+IDEC(3)
      IF(ITST.NE.0) GO TO ER01
      ISIG(LET)=1
      XO(LET)=FINPUT(1)
      ZO(LET)=FINPUT(2)
      IF(IDEC(4).EQ.3)     ,L01
C
C     INPUT IS XO,ZO,ANGLE (DEGREES + TO -90)
C
      TH=FINPUT(3)
      IF(ABS(TH).GT.90.) GO TO ER05
      IF(ABS(TH).EQ.90.)   ,A05
      TTH(LET)=BIG
      RETURN
  A05 TH=TH*.01745
      TTH(LET)=TAN(TH)
      RETURN
C
C     INPUT IS XO,ZO,X1,Z1
C
  L01 DX=FINPUT(3)-XO(LET)
      DZ=FINPUT(4)-ZO(LET)
      IF (DZ.EQ.0)    ,L05
      TTH(LET)=0.
      RETURN
  L05 TTH(LET)=-(DZ/DX)
      RETURN
 ER01 WOT 59,FER1,JZX,(INPUT(L),L=1,5)
 FER1 FORMAT ("TWO CONSECUTIVE LETTER",I3,/,5A11)
      CALL QUIT(1)
 ER03 WOT 59,FER3,INPUT(1),JZX
 FER3 FORMAT ("BUG--NO LETTERS ON CARD",A11,I5)
      CALL QUIT(1)
 ER04 WOT 59,FER4,INPUT(5),JZX
 FER4 FORMAT ("ILLEGAL SCARP LETTER ",A11,I5)
      CALL QUIT(1)
 ER05 WOT 59,FER5,INPUT(3),JZX
 FER5 FORMAT("ANGLE MORE THAN 90 DEGREES ",A11,I5)
      CALL QUIT(1)
      END
```

```
      SUBROUTINE VOLUMES
      USE DATACLICHE
      DIMENSION MANG(5),DANG(5),NANG(5)
C
C THIS SR ASSIGNS ARTIFICIAL LINES TO LEFT AND RIGHT ENDS
C LISTS ALL LINES WITH COMMON INTERSECTIONS NA,NB
C FINDS ALL LINES SURROUNDING EACH VOLJME (STRICTLY,AREA)
C AND FINDS DENSITY OF EACH VOLUME
C
C FIRST DO ARTIFICIAL LINES
C
      DO A01 J=1,100
      DO A00 K=1,5
  A00 INA(J,K)=INB(J,K)=0
  A01 ILV(J)=IRV(J)=0
      AMX=1.E+7
      CALL AMINMX(ZA,1,JZX,1,ZAN,ZAX,M2,M1)
      CALL AMINMX(ZB,1,JZX,1,ZBN,ZBX,M4,M3)
      MX=M1
      IF(ZAX.LT.ZBX) MX=M3
      MN=M2
      IF(ZBN.LT.ZAN) MN=M4
      IF((XA(MX).LT.TSTM).AND.(XB(MX).GT.TST)) GO TO A03
C
C IF SO MX IS BOTTOM LINE
C
      JZX=JZX+1
      XA(JZX)=-AMX
      XB(JZX)=AMX
      ZA(JZX)=ZA(MX)+.01
      ZB(JZX)=ZA(JZX)
      DRHO(JZX)=0.
      NP=NP+1
      NA(JZX)=NP
      NP=NP+1
      NB(JZX)=NP
  A03 IF((XA(MN).LT.TSTM).AND.(XB(MN).GT.TST)) GO TO A05
      JZX=JZX+1
      XA(JZX)=-AMX
      XB(JZX)=AMX
      ZA(JZX)=ZA(MN)-.01
      ZB(JZX)=ZA(JZX)
      DRHO(JZX)=0.
      NP=NP+1
      NA(JZX)=NP
      NP=NP+1
      NB(JZX)=NP
  A05 CONTINUE
C
C HAVE PUT IN FAKE TOP AND BOTTOM LINES FROM L TO R IF NO REAL ONES
C
      I=1
      DO A10 J=1,JZX
      IF(XA(J).GT.TSTM) GO TO A10
      NLFT(I)=J
      ZLFT(I)=ZA(J)
      I=I+1
  A10 CONTINUE
      IEND=I-1
C
```

```
C NOW WE HAVE ALL LINES WITH XA ON LEFT BOUNDARY
C
      IF(IEND.LT.2) GO TO ERR1
      DO A20 I=1,IEND
      ZM=AMINAF(ZLFT,1,IEND,1,M)
      NLFF(I)=NLFT(M)
      ZLFT(M)=TST
  A20 CONTINUE
      JTP=NLFF(1)
C
C NOW NLFF LISTS LEFT-ENDING LINES ORDERED BY INCREASING Z
C
      JZY=JZX+1
      DO A30 I=1,IEND-1
      NA(JZY)=NA(NLFF(I))
      NB(JZY)=NA(NLFF(I+1))
  A30 JZY=JZY+1
      JZA=JZY-1
C
C NOW WE HAVE VERTICAL "LINES" WITH NA TOP,NB BOTTOM
C
      I=1
      DO A40 J=1,JZX
      IF(XB(J).LT.TST) GO TO A40
      NRT(I)=J
      ZRT(I)=ZA(J)
      I=I+1
  A40 CONTINUE
      IEND=I-1
C
C NOW WE HAVE LIST OF LINES ENDING ON RIGHT BOUNDARY
C
      IF (IEND.LT.2) GO TO ERR2
      DO A60 I=1,IEND
      ZM=AMAXAF(ZRT,1,IEND,1,M)
      NRTT(I)=NRT(M)
      ZRT(M)=TSTM
  A60 CONTINUE
C
C NOW ORDERED BY DECREASING Z
C
      JBT=NRTT(1)
      DO A70 I=1,IEND-1
      NA(JZY)=NB(NRTT(I))
      NB(JZY)=NB(NRTT(I+1))
      JZY=JZY+1
  A70 CONTINUE
      JZY=JZY-1
C
C NOW NEW LINES WITH NA BOTTOM,NB TOP
C
C NOW SORT ALL LINES BY INTERSECTION
C
      DO B50 J=1,NP
      DO B40 L=1,JZY
      IF(NA(L).NE.J) GO TO B30
      DO B20 K=1,5
      IF(INA(J,K).NE.0) GO TO B20
C
C J IS INTERSECTION NO.,L IS LINE NO. THIS IS KTH LINE WITH NA=J
```

```
C
      INA(J,K)=L
      GO TO B30
  B20 CONTINUE
  B30 IF(NB(L).NE.J) GO TO B40
      DO B35 K=1,5
      IF(INB(J,K).NE.0) GO TO B35
      INB(J,K)=L
      GO TO B40
  B35 CONTINUE
  B40 CONTINUE
  B50 CONTINUE
C
C INA CONTAINS NOS. OF ALL LINES WITH XA AT INT. J,INB SAME FOR XB
C
      DO B80 J=1,NP
      ITT=0
      DO B60 K=1,5
  B60 IF(INA(J,K).NE.0) ITT=ITT+1
      DO B70 K=1,5
  B70 IF(INB(J,K).NE.0) ITT=ITT+1
      IF(ITT.LT.2) GO TO ERR3
C
C THIS IS A TEST FOR AT LEAST 2 LINES AT ALL INTERSECTIONS
C IT IS REDUNDANT WITH SUBROUTINE CHECK
C
  B80 CONTINUE
      DO B90 J=1,JZX
      IF(XA(J).EQ.XB(J))  ,B90
      FINPUT(3)=XA(J)-.01
      FINPUT(4)=ZA(J)
      FINPUT(2)=NA(J)
      IDEC(3)=IDEC(4)=2
      CALL CINT
  B90 CONTINUE
C
C NOW FIND WHICH LINES GO WITH WHICH VOLUME
C
      I=1
      ILT=1
      DO C200 J=1,JZX
      IF(J.EQ.JTP) GO TO C200
      IF (ILV(J).NE.0) GO TO C200
      ILV(J)=ILT
C
C TAKING EACH LINE AND,IF IT IS NNOT ALREADY ASSOCIATED ON ITS LEFT
C WITH A VOLUME,GIVE IT A VOLUME NUMBER. FIND ALL OTHER LINES ON SAME
C VOLUME. ROCEED CCW AROUND VOLUME TESTING EACH INTERSECTION
C
      ICT=1
      JXX=J
      NR=NB(J)
      NAB=2
      DO C150 K=1,25
      IANG=1
      IF(JXX.GT.JZX) GO TO LFS
      IF (NAB.NE.1) GO TO C02
C
C NAB SAYS WHETHER END IN QUESTION IS RIGHT OR LEFT.  FOR FIRST
C LINE, INTERSECTION MUST BE ON RIGHT
```

```
C
      ANG=ATAN2((ZA(JXX)-ZB(JXX)),(XB(JXX)-XA(JXX)))
      GO TO C03
  C02 ANG=ATAN2((ZB(JXX)-ZA(JXX)),(XA(JXX)-XB(JXX)))
C
C ANG IS ANGLE OF FIRST LINE WITH HORIZONTAL
C MUST REMEMBER Z IS POSITIVE DOWNWARDS IN CALCULATING ANGLE
C
  C03 DO C10 L=1,5
      INP=INA(NR,L)
      IF(INP.EQ.0) GO TO C11
C
C NO MORE LINES AT THIS INTERSECTION
C
      IF(INP.EQ.JXX) GO TO C10
C
C THIS IS THE LINE WE STARTED WITH
C
      IF(INP.GT.JZX) GO TO LFT
C
C ABOVE ARE VERTICALS AT BOUNDARY SO MUST BE NEXT LINE
C
      ANH=ATAN2((ZA(INP)-ZB(INP)),(XB(INP)-XA(INP)))
  C04 DING=ANG-ANH
      IF(DING.LT.0) DING=DING+6.283
      DANG(IANG)=DING
C
C THIS IS ANGLE BETWEEN FIRST LINE AND THAT BEING TESTED
C
      NANG(IANG)=INP
      MANG(IANG)=2
C
C MANG GIVES NAB OF OTHER END OF THIS LINE
C
      IANG=IANG+1
  C10 CONTINUE
C
C NOW HAVE LOOKED AT ALL LINES WITH A END AT INTERSECTION
C
  C11 DO C20 N=1,5
      INP=INB(NR,N)
      IF(INP.EQ.0) GO TO C21
      IF(INP.EQ.JXX) GO TO C20
      IF(INP.GT.JZX) GO TO C20
      ANH=ATAN2((ZB(INP)-ZA(INP)),(XA(INP)-XB(INP)))
      DING=ANG-ANH
      IF(DING.LT.0.) DING=DING+6.283
      DANG(IANG)=DING
      NANG(IANG)=INP
      MANG(IANG)=1
      IANG=IANG+1
  C20 CONTINUE
  C21 IANG=IANG-1
      DO C30 I=1,IANG
      IF(DANG(IANG).LT.0.) GO TO ERR4
  C30 CONTINUE
C
C NOW WE HAVE ANGLES OF ALL LINES AT INTERSECTION WITH FIRST LINE
C
      AM=AMINAF(DANG,1,IANG,1,M)
```

```
      JXX=NANG(M)
      IF(JXX.EQ.J) GO TO C180
C
C WE HAVE CHOSEN THE LINE WITH MININUM CW ANGLE WITH RESPECT TO
C PREVIOUS LINE. IF JXX.EQ.J IT IS OUR ORIGINAL LINE AND WERE
C DONE WITH THIS VOLUME
C
      NAB=MANG(M)
      IF(NAB.EQ.1) GO TO C50
      IF(ILV(JXX).NE.0) GO TO ERR8
      ILV(JXX)=ILT
      NR=NB(JXX)
C
C INTERSECTION IS NA END OF NEW LINE,SO NEXT INTERSECTION IS NB
C
      GO TO C150
  C50 IF (IRV(JXX).NE.0) GO TO ERR8
      IRV(JXX)=ILT
      NR=NA(JXX)
C
C INTERSECTION IS NB END
C
      GO TO C150
  LFS ANG=1.57
      IF(JXX.GT.JZA) ANG=-ANG
      GO TO C03
  LFT ANH=1.57
      IF(INP.LE.JZA) ANH=-ANH
      GO TO C04
C
C THIS IS THE ARTIFICIAL VERTICAL LINE SETTING ANGLE TO  90 DEG
C
 C150 ICT=ICT+1
      GO TO ERR5
 C180 IF (ICT.LT.3) GO TO ERR5
      ILT=ILT+1
 C200 CONTINUE
      ILT=ILT-1
      IF(ILT.LT.2) GO TO ERR9
C
C NOW WE HAVE ILV AND IRV FOR ALL LINES
C THAT IS,WE KNOW WHICH VOLUME IS ON EACH SIDE OF EACH LINE
C
      DO C300 J=1,JZX
      IF((ILV(J).EQ.0).AND.(J.NE.JTP)) GO TO ERR6
      IF((IRV(J).EQ.0).AND.(J.NE.JBT)) GO TO ERR6
 C300 CONTINUE
      XLF=TST
      XRT=TSTM
      DO D20 J=1,JZX
      IF(XA(J).LT.TSTM) GO TO D10
      IF(XA(J).LT.XLF) XLF=XA(J)
  D10 IF(XB(J).GT.TST) GO TO D20
      IF(XB(J).GT.XRT) XRT=XB(J)
  D20 CONTINUE
C
C XRT IS LARGEST NON-BOUNDARY X, XLF SMALLEST
C
      DO D100 I=1,ILT
      NI(I)=1
```

```
      DO D50 J=1,JZX
      IF(ILV(J).NE.I) GO TO D40
      NV(NI(I),I)=J
      KV(NI(I),I)=1
      NI(I)=NI(I)+1
C
C ILV(J)IS I
C
      GO TO D50
  D40 IF(IRV(J).NE.I) GO TO D50
      NV(NI(I),I)=J
      KV(NI(I),I)=2
      NI(I)=NI(I)+1
  D50 CONTINUE
      NI(I)=NI(I)-1
 D100 CONTINUE
C
C WE HAVE NOW LISTED ALL LINES SURROUNDING EACH VOLUME (I ABOVE)
C AS NV(NI) WHERE THERE ARE NI LINES FOR EACH VOLUME
C (AND ILT VOLUMES)
C KV IS 1 IF VOLUME IS ON LEFT OF LINE,2 IF ON RIGHT
C
C NOW FIND DENSITIES AND MIDPOINTS OF VOLUMES
C
      DO E200 I=1,ILT
      NIT=2*NI(I)
      L=0
      DO E20 K=1,NI(I)
      L=L+1
      JK=NV(K,I)
      XN(L)=XA(JK)
      IF(XN(L).LT.TSTM) XN(L)=XLF-.1*ABS(XLF)
      ZN(L)=ZA(JK)
      L=L+1
      XN(L)=XB(JK)
      IF(XN(L).GT.TST) XN(L)=XRT+.1*ABS(XRT)
  E20 ZN(L)=ZB(JK)
      CALL AMINMX(XN,1,NIT,1,AN,AX,M1,M2)
      XMD(I)=(AX+AN)/2.
C
C MIDDLE X OF VOLUME
C
      XMX(I)=AX
      XMN(I)=AN
      CALL AMINMX (ZN,1,NIT,1,AN,AX,M1,M2)
      ZMX(I)=AX
      ZMN(I)=AN
      XAFT=XMD(I)
      EX=1.
      CALL VOLDEN
      EX=0.
C
C FIRST FIND A Z INSIDE VOLUME,THEN DENSITY
C
      DO E50 K=1,KK-1
      N1=NRT(K)
      N2=NRT(K+1)
      IF((IRV(N1).EQ.I).AND.(ILV(N2).EQ.I)) GO TO E60
  E50 CONTINUE
      GO TO ERR7
```

```
  E60 ZMD(I)=(ZORD(K+1)+ZORD(K))/2.
      RMD(I)=CRHO(K)
      GO TO E200
 E200 CONTINUE
      DO E300 J=1,JZX
      DELR=RMD(IRV(J))-RMD(ILV(J))
      DELT=DELR-DRHO(J)
 E300 IF (ABS(DELT).GT.0.01) GO TO ERRO
      RETURN
 ERR1 WOT 59,FER1
C    FROM NEAR A10
      GO TO F100
 ERR2 WOT 59,FER2
C   FROM NEAR A40
      GO TO F100
 ERR3 WOT 59,FER3,J
C    FROM B70
      GO TO F100
 ERR4 WOT 59,FER4,NR,INP,ILT,IANG,DANG
C    FROM C30
      GO TO C150
 ERR5 WOT 59,FER5,ILT,JXX,NR
C      FROM C180
      GO TO F100
 ERR6 WOT 59,FER6,J
C    FROM C300
      GO TO F100
 ERR7 WOT 59,FER7,I
C      FROM E54
      GO TO F100
 ERR8 WOT 59,FER8,JXX,ILT,ILV(JXX),IRV(JXX)
C     FROM C50
      GO TO F100
 ERRO WOT 59,FERO,J,ILV(J),IRV(J),DRHO(J),RMD(ILV(J)),RMD(IRV(J)),DELR
C     FROM E300
      GO TO F100
 ERR9 WOT 59,FER9
      GO TO F100
 FER1 FORMAT(" ONLY ONE POINT ON LEFT BOUNDARY")
 FER2 FORMAT(" ONLY ONE POINT ON RIGHT BOUNDARY")
 FER3 FORMAT(" ONLY ONE LINE AT INTERSECTION ",I3)
 FER4 FORMAT("NEGATIVE ANGLE AT INTERSECTION ",I3," LINE",I3,
     1 " VOLUME",I3," ANGLE",I3," IS",E12.4)
 FER5 FORMAT("ONLY 2 LINES FOR VOLUME",I3," LAST LINE WAS",I3,
     1 " LAST INT WAS",I3)
 FER6 FORMAT("LINE NO",I3," HAS NO VOLUME ON ONE SIDE")
 FER7 FORMAT ("CANT FIND DENSITY FOR VOLUME",I4)
 FER8 FORMAT("TRYING TO ASSIGN VOL" ,I3," TO LINE",I3," VOL",I3,
     1 " ALREADY ON L. OR ",I3," ON R")
 FER9 FORMAT("ONLY ONE VOLUME IN THE PROBLEM")
 FERO FORMAT("DENSITY CONTRAST ON LINE",I3," VOL",I3,"ON L AND",I3,
     1 " ON R IS ",E12.4 /,"DISAGREES WITH DIFF BETWEEN L AND R VALUES "
     2 ,2E12.4,/," YIELDING ",E12.4)
 F100 EX=1.
      RETURN
      END
```

```
      SUBROUTINE VOLUMES
      USE DATACLICHE
      DIMENSION MANG(5),DANG(5),NANG(5)
C
C THIS SR ASSIGNS ARTIFICIAL LINES TO LEFT AND RIGHT ENDS
C LISTS ALL LINES WITH COMMON INTERSECTIONS NA,NB
C FINDS ALL LINES SURROUNDING EACH VOLUME (STRICTLY,AREA)
C AND FINDS DENSITY OF EACH VOLUME
C
C FIRST DO ARTIFICIAL LINES
C
      DO A01 J=1,100
      DO A00 K=1,5
  A00 INA(J,K)=INB(J,K)=0
  A01 ILV(J)=IRV(J)=0
      AMX=1.E+7
      CALL AMINMX(ZA,1,JZX,1,ZAN,ZAX,M2,M1)
      CALL AMINMX(ZB,1,JZX,1,ZBN,ZBX,M4,M3)
      MX=M1
      IF(ZAX.LT.ZBX) MX=M3
      MN=M2
      IF(ZBN.LT.ZAN) MN=M4
      IF((XA(MX).LT.TSTM).AND.(XB(MX).GT.TST)) GO TO A03
C
C IF SO MX IS BOTTOM LINE
C
      JZX=JZX+1
      XA(JZX)=-AMX
      XB(JZX)=AMX
      ZA(JZX)=ZA(MX)+.01
      ZB(JZX)=ZA(JZX)
      DRHO(JZX)=0.
      NP=NP+1
      NA(JZX)=NP
      NP=NP+1
      NB(JZX)=NP
  A03 IF((XA(MN).LT.TSTM).AND.(XB(MN).GT.TST)) GO TO A05
      JZX=JZX+1
      XA(JZX)=-AMX
      XB(JZX)=AMX
      ZA(JZX)=ZA(MN)-.01
      ZB(JZX)=ZA(JZX)
      DRHO(JZX)=0.
      NP=NP+1
      NA(JZX)=NP
      NP=NP+1
      NB(JZX)=NP
  A05 CONTINUE
C
C HAVE PUT IN FAKE TOP AND BOTTOM LINES FROM L TO R IF NO REAL ONES
C
      I=1
      DO A10 J=1,JZX
      IF(XA(J).GT.TSTM) GO TO A10
      NLFT(I)=J
      ZLFT(I)=ZA(J)
      I=I+1
  A10 CONTINUE
      IEND=I-1
C
```

```
C NOW WE HAVE ALL LINES WITH XA ON LEFT BOUNDARY
C
      IF(IEND.LT.2) GO TO ERR1
      DO A20 I=1,IEND
      ZM=AMINAF(ZLFT,1,IEND,1,M)
      NLFF(I)=NLFT(M)
      ZLFT(M)=TST
  A20 CONTINUE
      JTP=NLFF(1)
C
C NOW NLFF LISTS LEFT-ENDING LINES ORDERED BY INCREASING Z
C
      JZY=JZX+1
      DO A30 I=1,IEND-1
      NA(JZY)=NA(NLFF(I))
      NB(JZY)=NA(NLFF(I+1))
  A30 JZY=JZY+1
      JZA=JZY-1
C
C NOW WE HAVE VERTICAL "LINES" WITH NA TOP,NB BOTTOM
C
      I=1
      DO A40 J=1,JZX
      IF(XB(J).LT.TST) GO TO A40
      NRT(I)=J
      ZRT(I)=ZA(J)
      I=I+1
  A40 CONTINUE
      IEND=I-1
C
C NOW WE HAVE LIST OF LINES ENDING ON RIGHT BOUNDARY
C
      IF (IEND.LT.2) GO TO ERR2
      DO A60 I=1,IEND
      ZM=AMAXAF(ZRT,1,IEND,1,M)
      NRTT(I)=NRT(M)
      ZRT(M)=TSTM
  A60 CONTINUE
C
C NOW ORDERED BY DECREASING Z
C
      JBT=NRTT(1)
      DO A70 I=1,IEND-1
      NA(JZY)=NB(NRTT(I))
      NB(JZY)=NB(NRTT(I+1))
      JZY=JZY+1
  A70 CONTINUE
      JZY=JZY-1
C
C NOW NEW LINES WITH NA BOTTOM,NB TOP
C
C NOW SORT ALL LINES BY INTERSECTION
C
      DO B50 J=1,NP
      DO B40 L=1,JZY
      IF(NA(L).NE.J) GO TO B30
      DO B20 K=1,5
      IF(INA(J,K).NE.0) GO TO B20
C
C J IS INTERSECTION NO.,L IS LINE NO. THIS IS KTH LINE WITH NA=J
```

```
C
      INA(J,K)=L
      GO TO B30
  B20 CONTINUE
  B30 IF(NB(L).NE.J) GO TO B40
      DO B35 K=1,5
      IF(INB(J,K).NE.0) GO TO B35
      INB(J,K)=L
      GO TO B40
  B35 CONTINUE
  B40 CONTINUE
  B50 CONTINUE
C
C INA CONTAINS NOS. OF ALL LINES WITH XA AT INT. J,INB SAME FOR XB
C
      DO B80 J=1,NP
      ITT=0
      DO B60 K=1,5
  B60 IF(INA(J,K).NE.0) ITT=ITT+1
      DO B70 K=1,5
  B70 IF(INB(J,K).NE.0) ITT=ITT+1
      IF(ITT.LT.2) GO TO ERR3
C
C THIS IS A TEST FOR AT LEAST 2 LINES AT ALL INTERSECTIONS
C IT IS REDUNDANT WITH SUBROUTINE CHECK
C
  B80 CONTINUE
C
C NOW FIND WHICH LINES GO WITH WHICH VOLUME
C
      I=1
      ILT=1
      DO C200 J=1,JZX
      IF(J.EQ.JTP) GO TO C200
      IF (ILV(J).NE.0) GO TO C200
      ILV(J)=ILT
C
C TAKING EACH LINE AND,IF IT IS NNOT ALREADY ASSOCIATED ON ITS LEFT
C WITH A VOLUME,GIVE IT A VOLUME NUMBER. FIND ALL OTHER LINES ON SAME
C VOLUME. ROCEED CCW AROUND VOLUME TESTING EACH INTERSECTION
C
      ICT=1
      JXX=J
      NR=NB(J)
      NAB=2
      DO C150 K=1,25
      IANG=1
      IF(JXX.GT.JZX) GO TO LFS
      IF (NAB.NE.1) GO TO C02
C
C NAB SAYS WHETHER END IN QUESTION IS RIGHT OR LEFT.  FOR FIRST
C LINE, INTERSECTION MUST BE ON RIGHT
C
      ANG=ATAN2((ZA(JXX)-ZB(JXX)),(XB(JXX)-XA(JXX)))
      GO TO C03
  C02 ANG=ATAN2((ZB(JXX)-ZA(JXX)),(XA(JXX)-XB(JXX)))
C
C ANG IS ANGLE OF FIRST LINE WITH HORIZONTAL
C MUST REMEMBER Z IS POSITIVE DOWNWARDS IN CALCULATING ANGLE
C
```

```
  C03 DO C10 L=1,5
      INP=INA(NR,L)
      IF(INP.EQ.0) GO TO C11
C
C NO MORE LINES AT THIS INTERSECTION
C
      IF(INP.EQ.JXX) GO TO C10
C
C THIS IS THE LINE WE STARTED WITH
C
      IF(INP.GT.JZX) GO TO LFT
C
C ABOVE ARE VERTICALS AT BOUNDARY SO MUST BE NEXT LINE
C
      ANH=ATAN2((ZA(INP)-ZB(INP)),(XB(INP)-XA(INP)))
  C04 DING=ANG-ANH
      IF(DING.LT.0) DING=DING+6.283
      DANG(IANG)=DING
C
C THIS IS ANGLE BETWEEN FIRST LINE AND THAT BEING TESTED
C
      NANG(IANG)=INP
      MANG(IANG)=2
C
C MANG GIVES NAB OF OTHER END OF THIS LINE
C
      IANG=IANG+1
  C10 CONTINUE
C
C NOW HAVE LOOKED AT ALL LINES WITH A END AT INTERSECTION
C
  C11 DO C20 N=1,5
      INP=INB(NR,N)
      IF(INP.EQ.0) GO TO C21
      IF(INP.EQ.JXX) GO TO C20
      IF(INP.GT.JZX) GO TO C20
      ANH=ATAN2((ZB(INP)-ZA(INP)),(XA(INP)-XB(INP)))
      DING=ANG-ANH
      IF(DING.LT.0.) DING=DING+6.283
      DANG(IANG)=DING
      NANG(IANG)=INP
      MANG(IANG)=1
      IANG=IANG+1
  C20 CONTINUE
  C21 IANG=IANG-1
      DO C30 I=1,IANG
      IF(DANG(IANG).LT.0.) GO TO ERR4
  C30 CONTINUE
C
C NOW WE HAVE ANGLES OF ALL LINES AT INTERSECTION WITH FIRST LINE
C
      AM=AMINAF(DANG,1,IANG,1,M)
      JXX=NANG(M)
      IF(JXX.EQ.J) GO TO C180
C
C WE HAVE CHOSEN THE LINE WITH MININUM CW ANGLE WITH RESPECT TO
C PREVIOUS LINE. IF JXX.EQ.J IT IS OUR ORIGINAL LINE AND WERE
C DONE WITH THIS VOLUME
C
      NAB=MANG(M)
```

```
      IF(NAB.EQ.1) GO TO C50
      IF(ILV(JXX).NE.0) GO TO ERR8
      ILV(JXX)=ILT
      NR=NB(JXX)
C
C INTERSECTION IS NA END OF NEW LINE,SO NEXT INTERSECTION IS NB
C
      GO TO C150
  C50 IF (IRV(JXX).NE.0) GO TO ERR8
      IRV(JXX)=ILT
      NR=NA(JXX)
C
C INTERSECTION IS NB END
C
      GO TO C150
  LFS ANG=1.57
      IF(JXX.GT.JZA) ANG=-ANG
      GO TO C03
  LFT ANH=1.57
      IF(INP.LE.JZA) ANH=-ANH
      GO TO C04
C
C THIS IS THE ARTIFICIAL VERTICAL LINE SETTING ANGLE TO  90 DEG
C
 C150 ICT=ICT+1
      GO TO ERR5
 C180 IF (ICT.LT.3) GO TO ERR5
      ILT=ILT+1
 C200 CONTINUE
      ILT=ILT-1
      IF(ILT.LT.2) GO TO ERR9
C
C NOW WE HAVE ILV AND IRV FOR ALL LINES
C THAT IS,WE KNOW WHICH VOLUME IS ON EACH SIDE OF EACH LINE
C
      DO C300 J=1,JZX
      IF((ILV(J).EQ.0).AND.(J.NE.JTP)) GO TO ERR6
      IF((IRV(J).EQ.0).AND.(J.NE.JBT)) GO TO ERR6
 C300 CONTINUE
      XLF=TST
      XRT=TSTM
      DO D20 J=1,JZX
      IF(XA(J).LT.TSTM) GO TO D10
      IF(XA(J).LT.XLF) XLF=XA(J)
  D10 IF(XB(J).GT.TST) GO TO D20
      IF(XB(J).GT.XRT) XRT=XB(J)
  D20 CONTINUE
C
C XRT IS LARGEST NON-BOUNDARY X, XLF SMALLEST
C
      DO D100 I=1,ILT
      NI(I)=1
      DO D50 J=1,JZX
      IF(ILV(J).NE.I) GO TO D40
      NV(NI(I),I)=J
      KV(NI(I),I)=1
      NI(I)=NI(I)+1
C
C ILV(J)IS I
C
```

```
      GO TO D50
  D40 IF(IRV(J).NE.I) GO TO D50
      NV(NI(I),I)=J
      KV(NI(I),I)=2
      NI(I)=NI(I)+1
  D50 CONTINUE
      NI(I)=NI(I)-1
 D100 CONTINUE
C
C WE HAVE NOW LISTED ALL LINES SURROUNDING EACH VOLUME (I ABOVE)
C AS NV(NI) WHERE THERE ARE NI LINES FOR EACH VOLUME
C (AND ILT VOLUMES)
C KV IS 1 IF VOLUME IS ON LEFT OF LINE,2 IF ON RIGHT
C
C NOW FIND DENSITIES AND MIDPOINTS OF VOLUMES
C
      DO E200 I=1,ILT
      NIT=2*NI(I)
      L=0
      DO E20 K=1,NI(I)
      L=L+1
      JK=NV(K,I)
      XN(L)=XA(JK)
      IF(XN(L).LT.TSTM) XN(L)=XLF-.1*ABS(XLF)
      ZN(L)=ZA(JK)
      L=L+1
      XN(L)=XB(JK)
      IF(XN(L).GT.TST) XN(L)=XRT+.1*ABS(XRT)
   E20 ZN(L)=ZB(JK)
      CALL AMINMX(XN,1,NIT,1,AN,AX,M1,M2)
      XMD(I)=(AX+AN)/2.
C
C MIDDLE X OF VOLUME
C
      XMX(I)=AX
      XMN(I)=AN
      CALL AMINMX (ZN,1,NIT,1,AN,AX,M1,M2)
      ZMX(I)=AX
      ZMN(I)=AN
      XAFT=XMD(I)
      EX=1.
      CALL VOLDEN
      EX=0.
C
C FIRST FIND A Z INSIDE VOLUME,THEN DENSITY
C
      DO E50 K=1,KK-1
      N1=NRT(K)
      N2=NRT(K+1)
      IF((IRV(N1).EQ.I).AND.(ILV(N2).EQ.I)) GO TO E60
   E50 CONTINUE
      GO TO ERR7
   E60 ZMD(I)=(ZORD(K+1)+ZORD(K))/2.
      RMD(I)=CRHO(K)
      GO TO E200
  E200 CONTINUE
      DO E300 J=1,JZX
      DELR=RMD(IRV(J))-RMD(ILV(J))
      DELT=DELR-DRHO(J)
  E300 IF (ABS(DELT).GT.0.01) GO TO ERRO
```

```
      RETURN
 ERR1 WOT 59,FER1
C   FROM NEAR A10
      GO TO F100
 ERR2 WOT 59,FER2
C  FROM NEAR A40
      GO TO F100
 ERR3 WOT 59,FER3,J
C   FROM B70
      GO TO F100
 ERR4 WOT 59,FER4,NR,INP,ILT,IANG,DANG
C   FROM C30
      GO TO C150
 ERR5 WOT 59,FER5,ILT,JXX,NR
C     FROM C180
      GO TO F100
 ERR6 WOT 59,FER6,J
C   FROM C300
      GO TO F100
 ERR7 WOT 59,FER7,I
C     FROM E54
      GO TO F100
 ERR8 WOT 59,FER8,JXX,ILT,ILV(JXX),IRV(JXX)
C    FROM C50
      GO TO F100
 ERRO WOT 59,FERO,J,ILV(J),IRV(J),DRHO(J),RMD(ILV(J)),RMD(IRV(J)),DELR
C    FROM E300
      GO TO F100
 ERR9 WOT 59,FER9
      GO TO F100
 FER1 FORMAT(" ONLY ONE POINT ON LEFT BOUNDARY")
 FER2 FORMAT(" ONLY ONE POINT ON RIGHT BOUNDARY")
 FER3 FORMAT(" ONLY ONE LINE AT INTERSECTION ",I3)
 FER4 FORMAT("NEGATIVE ANGLE AT INTERSECTION ",I3," LINE",I3,
     1 " VOLUME",I3," ANGLE",I3," IS",E12.4)
 FER5 FORMAT("ONLY 2 LINES FOR VOLUME",I3," LAST LINE WAS",I3,
     1 " LAST INT WAS",I3)
 FER6 FORMAT("LINE NO",I3," HAS NO VOLUME ON ONE SIDE")
 FER7 FORMAT ("CANT FIND DENSITY FOR VOLUME",I4)
 FER8 FORMAT("TRYING TO ASSIGN VOL" ,I3," TO LINE",I3," VOL",I3,
     1 " ALREADY ON L. OR ",I3," ON R")
 FER9 FORMAT("ONLY ONE VOLUME IN THE PROBLEM")
 FERO FORMAT("DENSITY CONTRAST ON LINE" I3," VOL",I3,"ON L AND",I3,
     1 " ON R IS ",E12.4 /,"DISAGREES WITH DIFF BETWEEN L AND R VALUES "
     2 ,2E12.4,/," YIELDING ",E12.4)
 F100 EX=1.
      RETURN
      END
```

```
      SUBROUTINE CHECK
      USE DATACLICHE
C
C     TESTS TO ASSURE ALL POINTS EXCEPT AT SIDE BOUNDARIES ARE DUPLICATED
C      IF NA AND NB GE 1 POINT IS OK
C      WE ARE SETTING NA AND NB TO INTERSECTION NUMBER IN THIS
C      REVISION.  NOT SET UNTIL EITHER TWO LINES MEET THERE
C      OR IT IS A RIGHT OR LEFT BOUNDARY
C      NEED PROVISION FOR TRIPLICATE
C
      NLA=NLB=0
      DO 10 N=1,100
   10 NA(N)=NB(N)=0
      NP=1
      TTT=.1
      TMX=100000.
      DO 20 N=1,JZX
      IF(XA(N)-XB(N))13,12,11
   11 SAX=XA(N)
      SAZ=ZA(N)
      XA(N)=XB(N)
      ZA(N)=ZB(N)
      XB(N)=SAX
      ZB(N)=SAZ
      DRHO(N)=-DRHO(N)
      GO TO 13
   12 XA(N)=XA(N)-.001
   13 IF(XA(N).EQ.0.) XA(N)=-.001
      IF (XB(N).EQ.0.) XB(N)=.001
   20 CONTINUE
C
C ABOVE ROUTINE MAKES ALL XA THE LEFT HAND (MINIMUM X) VALUE OF THE
C XA,XB PAIR AND ASSURES THAT THERE ARE NO VERTICAL LINES AND NO ZERO
C VALUES OF X (THOSE WOULD FOUL UP CALCULATION OF LOG DENSITY)
C
C NOW TEST FOR LINES INTERSECTING NOT AT PLANNED INTERSECTIONS
C G AND CORR USED TEMPORARILY AS COEFFICIENTS OF LINES
C
      DO C10 J=1,JZX
      G(J)=(ZB(J)-ZA(J))/(XB(J)-XA(J))
  C10 CORR(J)=ZB(J)-G(J)*XB(J)
C
C Z=G*X+CORR
C
      DO C50 J=1,JZX-1
      NN=J+1
      DO C40 N=NN,JZX
      IF(N.EQ.J) GO TO C40
      DDG=ABS(G(N)-G(J))
      IF(DDG.LT.0.01) GO TO C40
      XX=(CORR(N)-CORR(J))/(G(J)-G(N))
      X1=XX-XA(J)-.01
      X2=XX-XA(N)-.01
      IF((X1.LT.0.).OR.(X2.LT.0.)) GO TO C40
      X1=XX-XB(J)+.01
      X2=XX-XB(N)+.01
      IF((X1.GT.0.).OR.(X2.GT.0.)) GO TO C40
      GO TO CERR
  C40 CONTINUE
  C50 CONTINUE
```

```
      DO C60 J=1,100
      CORR(J)=0.
  C60 G(J)=0.
      DO 100 N=1,JZX
      IF(ABS(XA(N)).LT.TMX) GO TO 30
      IF(NA(N).GT.0) GO TO 100
      NA(N)=NP
      NP=NP+1
      GO TO 100
C
C TEST FOR XA EQ. R OR L END
C
  30  IF(NA(N).GE.1) GO TO 100
      JJ=N+1
      IF(N.EQ.JZX) GO TO 50
      DO 40 J=JJ,JZX
      T1=ABS(XA(N)-XA(J))
      T2=ABS(ZA(N)-ZA(J))
      IF((T1.GT.TTT).OR.(T2.GT.TTT)) GO TO 40
C
C COMPARING LEFT ENDS--IF NO COMMON INTERSECTION GO TO 40
C
      IF(NA(N).LE.0) GO TO 31
      NA(J)=NA(N)
      GO TO 40
   31 IF(NA(J).LE.0) GO TO 32
      NA(N)=NA(J)
      GO TO 40
   32 NA(N)=NA(J)=NP
      NP=NP+1
   40 CONTINUE
   50 DO 80 K=1,JZX
      T1=ABS(XA(N)-XB(K))
      T2=ABS(ZA(N)-ZB(K))
      IF((T1.GT.TTT).OR.(T2.GT.TTT)) GO TO 80
C
C COMPARING ONE LEFT TO OTHER RIGHT END--IF NO COMMON INT GO TO 80
C
      IF(NA(N).LE.0) GO TO 71
      NB(K)=NA(N)
      GO TO 80
   71 IF(NB(K).LE.0) GO TO 72
      NA(N)=NB(K)
      GO TO 80
   72 NA(N)=NB(K)=NP
      NP=NP+1
   80 CONTINUE
      IF(NA(N).LT.1) NLA=1
  100 CONTINUE
      DO 200 M=1,JZX
      IF(ABS(XB(M)).LT.TMX) GO TO 130
      NB(M)=NP
      NP=NP+1
      GO TO 200
  130  IF(NB(M).GE.1) GO TO 200
      IF(M.EQ.JZX) GO TO EROB
      LL=M+1
      DO 140 L=LL,JZX
      T1=ABS(XB(M)-XB(L))
      T2=ABS(ZB(M)-ZB(L))
```

```
      IF((T1.GT.TTT).OR.(T2.GT.TTT)) GO TO 140
C
C COMPARING RIGHT ENDS IF NO COMMON INTERSECTION GO TO 140
C
      IF(NB(M).LE.0) GO TO 131
      NB(L)=NB(M)
      GO TO 140
  131 IF(NB(L).LE.0) GO TO 132
      NB(M)=NB(L)
      GO TO 140
  132 NB(M)=NB(L)=NP
      NP=NP+1
  140 CONTINUE
      IF(NB(M).LT.1) NLA=1
  200 CONTINUE
      NP=NP-1
      IF(NLA.LT.1) GO TO 250
C
C ALL IS WELL
C
      DO 210 N=1,JZX
      NR=0
C
C FINDING THE BAD INTERSECTION
C
      IF(NA(N).GE.1) GO TO 205
      NR=1
      GO TO 208
  205 IF((NB(N).LT.1).AND.(NR.LT.1)) GO TO 208
      GO TO 210
  208 WOT 59,FMA,N,XA(N),ZA(N),XB(N),ZB(N)
  210 CONTINUE
  220 EX=1.
  250 RETURN
 EROB WOT 59,FMA,JZX,XA(JZX),ZA(JZX),XB(JZX),ZB(JZX)
      GO TO 220
  FMA FORMAT(" NOT DUPLICATED: ",I3,4E12.3)
 CERR WOT 59,CFT,J,N,XX
      GO TO 220
  CFT FORMAT (" LINES ",I2," AND ",I3," CROSS AT ",E12.4)
      END
```

LISTING OF SUBROUTINES IN BIFURC

THAT ARE DIFFERENT FROM THOSE IN BIFUR3

(Some END statements left out)

```
      SUBROUTINE BIFPLT
      USE DATACLICHE
    1 FORMAT("VOLUME ",I3," HAS DENSITY LE ZERO")
      DIMENSION XM(2),ZM(2)
C
C PLOTS GRID ON ONE FRAME WITH UX80
C
      IF(NFIR.NE.0) GO TO A20
      NFIR=1
      DO A10 K=1,ILT
  A10 ZN(K)=XMX(K)-XMN(K)
C
C USING ZN SLOPPING OVER INTO ZORD AS SORTING VARIABLE
C
      DXMN=AMINAF(ZN,1,ILT,1,M)
      XMXX=AMAXAF(XMX,1,ILT,1,M)
      XMNN=0.
      ZMXX=AMAXAF(ZMX,1,ILT,1,M)
      ZMNN=AMINAF(ZMN,1,ILT,1,M)
      IF(XMNN.EQ.0.) XMNN=-0.1*XMXX
      DXMX=XMXX-XMNN
      IF(ZMNN.LT.0.) ZMNN=1.1*ZMNN
      IF(ZMNN.GT.0) ZMNN=.9*ZMNN
      IF(ZMNN.EQ.0.) ZMNN=-0.1*ZMXX
      ZMXX=1.1*ZMXX
      DZMX=ZMXX-ZMNN
      IF(DXMX.EQ.0.) DXMX=.001
      CHARN=5.*DXMX/DXMN
      CALL DDERS(-1)
      XM1=XMXX+100.
      XMXX=1.1*XMXX
      XMXX=AMAX1(XMXX,XM1)
      DXMX=XMXX-XMNN
      RATIO=DZMX/DXMX
      CHARW=DXMX/CHARN
      IF(RATIO.GT.1.) CHARN=CHARN* RATIO
      IF(CHARN.LT.100.) CHARN= 100.
      CALL UXCHSIZ(CHARN)
      CHARW=DXMX/CHARN
      IF(RATIO.GT.1.) CHARW=CHARW*RATIO
      CW6=6.*CHARW
      ZMNN=-ZMNN
      ZMXX=-ZMXX
      XM(1)=XMNN
      XM(2)=XMXX
      ZM(1)=-ZMNN
      ZM(2)=-ZMXX
  A20 CALL DDERS (-1)
      DST=DZMX/10.
      STT=ZMXX-DST
      DXT=DXMX/8.
      DXTT=XMNN-(DXT/2.)
      DXT=XMNN-DXT
      DO A21 K=1,ILT
  A21 ZN(K)=ZMX(K)-ZMN(K)
C
C    NOW USING ZN FOR VOLUME HEIGHT
C
      CALL SETCH(1.,1.,0,0,0,0)
      WOT 100,2,KIT
```

```
    2 FORMAT(8A10)
      CALL UXCHSIZ(CHARN)
      DO A30 J=1,JZX
      ZA(J)=-ZA(J)
  A30 ZB(J)=-ZB(J)
      DO A35 J=1,ILT
  A35 ZMD(J)=-ZMD(J)
      SMIN=.1
      SMAX=.9
      TMIN=.1
      TMAX=.9
      IF (RATIO.GE.1.) GO TO A40
      TMAX=TMIN+.8*RATIO
      GO TO A50
  A40 SMAX=SMIN+.8/RATIO
  A50 CALL MAP(XMNN,XMXX,ZMXX,ZMNN,SMIN,SMAX,TMIN,TMAX)
C
C WE HAVE SET UP A GRID INCREASING X AND Z-DIMENSIONS,OTHER THAN
C BOUNDARIES BY 10 PERCENT. CHARACTER SIZE IS SPECIFIED SO THAT
C 4 CHARACTERS FIT IN X-DIMENSION OF NARROWEST VOLUME.
C SCALE IS SAME IN BOTH DIMENSIONS. Z IS SET NEGATIVE FOR PLOTS
C
      CALL GAXIS(XMNN,ZMXX,XMXX,ZMXX,1,0,0,"F7.0",0,XM)
      CALL GAXIS(XMNN,ZMNN,XMNN,ZMXX,0,0,0,"F7.0",0,ZM)
      CALL GAXIS(XMXX,ZMNN,XMXX,ZMXX,0,0,1,"F7.0",0,ZM)
      CALL LINEP(0.,ZMXX,0.,ZMNN,3)
      DO A60 J=1,JZX
  A60 CALL LINE (XA(J),ZA(J),XB(J),ZB(J))
      DO A70 K=1,ILT
      IF(RMD(K).LE.0.001) WOT 59,1,K
      IF(RMD(K).LE.0.001) GO TO A70
      IF(ZN(K).GT.CW6) GO TO A61
      STT=STT+DST
      CALL SETLCH(DXT,STT)
      WOT 100,F4,RMD(K)
      WOT 100,F3,K
C
C     WRITING ON THE SIDE
C
      CALL DDERS(1)
      CALL LINEP (DXTT,STT,XMD(K),ZMD(K))
      CALL DDERS(-1)
      GO TO A70
  A61 Z1=ZMD(K)-CHARW
      X1=XMD(K)
      CALL SETLCH (X1,Z1)
C
C WRITING VOLUME NUMBER
C
      WOT 100,F1,K
   F1 FORMAT(I2)
      X1=X1-CHARW
      Z1=Z1+3.*CHARW
      CALL SETLCH(X1,Z1)
C
C WRITING DENSITY
C
      WOT 100,F2,RMD(K)
   F2 FORMAT (F4.2)
   F3 FORMAT (1X,I2)
```

```
   F4 FORMAT(F5.2)
  A70 CONTINUE
      DO A80 K=1,NP
      LA=INA(K,1)
      IF(LA.EQ.0) GO TO A72
      X1=XA(LA)
      IF(LA.GT.JZX) GO TO A80
      IF(X1.LT.TSTM) GO TO A80
      IF(X1.GT.TST) GO TO A80
      Z1=ZA(LA)
      GO TO A75
  A72 LA=INB(K,1)
      X1=XB(LA)
      IF(LA.GT.JZX) GO TO A80
      IF(X1.GT.TST) GO TO A80
      IF(X1.LT.TSTM) GO TO A80
      Z1=ZB(LA)
  A75 X1=X1-CHARW
      Z1=Z1-1.5*CHARW
      CALL SETLCH(X1,Z1)
C
C WRITING INTERSECTION NUMBER
C
      WOT 100,F1,K
  A80 CONTINUE
      DO A90 J=1,JZX
      ZA(J)=-ZA(J)
  A90 ZB(J)=-ZB(J)
      CALL FRAME
      RETURN
      END
```

```
      SUBROUTINE GEXEC
C
C        MANAGES CALCULATION OF G
C    FOR CYLINDRICAL GEOMETRY
C
      USE DATACLICHE
      DIMENSION RCC(100)
      KEND=JZY
C
C   CALCULATE CORRECTION TO FAG
C
      E1=0.
      DO 90 L=1,2
      G(L)=0.
      DO 85 K=1,KEND
      X1=XA(K)
      X2=XB(K)
      Z1=ZA(K)-E1
      Z2=ZB(K)-E1
      G(L)=G(L)+DRHO(K)*DG(X1,X2,Z1,Z2)
   85 CONTINUE
      E1=-15.
   90 CONTINUE
      FCOR=0.795165*(G(1)-G(2))
C
C     CALCULATE SUBSURFACE GRAVITY
C
      DO 300 L=1,LDEP
      G(L)=0.
      DO 200 K=1,KEND
      X1=XA(K)
      X2=XB(K)
      Z1=DPT(L)-ZA(K)
      Z2=DPT(L)-ZB(K)
      IF (Z1.EQ.0.) Z1=.01
      IF (Z2.EQ.0.) Z2=.01
      IF(Z1.EQ.Z2) GO TO 200
      IF((X1.EQ.0.).AND.(X2.EQ.0.)) GO TO 200
      DAD=DRHO(K)
      ZAD=Z2
      XAD=X2
      IF(Z1-Z2) ,200,180
      Z2=Z1
      X2=X1
      Z1=ZAD
      X1=XAD
      DAD=-DAD
  180 Z1B=ABS(Z1)
      Z2B=ABS(Z2)
      IF((Z1.GT.0.).AND.(Z2.LT.0.)) GO TO 190
      G(L)=G(L)+DAD*CONE(X1,X2,Z1B,Z2B)
      GO TO 200
  190 RR=X1+Z1B*(X2-X1)/(Z1-Z2)
      G(L)=G(L)+DAD*(CONE(X1,RR,Z1B,0.)+CONE(RR,X2,0.,Z2B))
  200 CONTINUE
      G(L)=G(L)*.04192
  300 CONTINUE
C
C     CALCULATE DENSITY
      WOT 3,2
```

```
      WOT 3,6,FCOR
      DO 400 L=2,LDEP
      IF (DPT(L).EQ.DPT(L-1)) GO TO 400
      R(L)=-11.92748*(G(L)-G(L-1))/(DPT(L)-DPT(L-1))
      CORR(L)=RHOL(L)-R(L)-FCOR
  400  CONTINUE
    1 FORMAT(8A10)
    2 FORMAT("    DEPT 1     DEPT 2     RHO       RHO NOM     RHO CORR CORR")
    3 FORMAT(2F10.1,4F10.4,E12.4)
    4 FORMAT(E13.4,3E12.4)
    5 FORMAT(///"    X-POSITION              G       DG")
    6 FORMAT ("    VAG     ",40X,F10.4)
    7 FORMAT (3E15.6)
    8 FORMAT( /,"NOTE: CORR= RHO NOM - RHO -VAG CORR")
    9 FORMAT("    DEPTH 1       DEPTH 2       CORR          RHO CORR ")
C
C CALCULATE CORRECTION TO MEASURED GRAVIMETRIC DENSITY
C MUST MATCH CALCULATED AND MEASURED DEPTHS
C PREVIOUSLY DID CALCULATIONS ON ALL MEASURED DEPTH POINTS
C BUT SOME MEASURED PAIRS MAY COVER MORE THAN ONE CALCULATED PAIR
C
      IF(LLOG.EQ.0) GO TO 475
      DO L100 J=1,JDP
      L1=0
      IF(DD1(J).EQ.DD2(J)) GO TO L100
      L2=0
      DDX=AMAX1(DD1(J),DD2(J))
      DDM=AMIN1(DD1(J),DD2(J))
      RC=0.
C
C PICKED TOP AND BOTTOM OF INTERVAL. EACH SHOULD BE A
C CALCULATED DEPTH
C
      DO L20 L=1,LDEP
      IF(L1.GT.0) GO TO L05
      DIM=DDM-DPT(L)
      IF(DIM.LT.1.) L1=L
  L05 IF(L2.GT.0) GO TO L19
      DIX=DDX-DPT(L)
      IF(DIX.GT.1.) GO TO L19
      L2=L
  L19 IF((L1.GT.0).AND.(L2.GT.0)) GO TO L30
  L20 CONTINUE
C
C NOW WE HAVE L VALUES OF ENDPOINTS
C L2 IS NOT NECESSARILY GRETER THAN L1
C
  L30 LX=MAX0(L2,L1)
      LM=MIN0(L2,L1)
      DL=LX-LM
      IF(DL-1) L100,   ,L40
      RCOR(J)=DDR(J)+CORR(LX)
      GO TO L100
  L40 L3=LM+1
      DO L50 L=L3,LX
  L50 RC=RC+CORR(L)*(DPT(L)-DPT(L-1))
C
C EQUAL DEPTH VALUES WERE REMOVED IN LOGDEN
C
      RCOR(J)=DDR(J)+RC/(DPT(LX)-DPT(LM))
```

```
 L100 RCC(LX)=RCOR(J)
C
C NOW WE HAVE THE CORRECTED GRAVIMETRIC DENSITIES
C
      WOT 4,1,KIT
      WOT 4,9
  475 DO 477 L=2,LDEP
      WOT 4,4,DPT(L-1),DPT(L),CORR(L),RCC(L)
      WOT 3,3,DPT(L),DPT(L-1),R(L),RHOL(L),RCC(L),CORR(L),G(L)
  477 CONTINUE
      WOT 3,8
      IF(LZEP.LE.0) RETURN
C
C     CALCULATE SURFACE GRAVITY
C
      WOT 3,5
      DDG=0.
      DO 600 L=1,LZEP
      G(L)=0.
      DO 500 K=1,JZX
      X1=XA(K)-ZPT(L)
      X2=XB(K)-ZPT(L)
      Z1=ZA(K)
      Z2=ZB(K)
      DAD=DRHO(K)
      ZAD=Z2
      XAD=X2
      IF(Z1-Z2) ,500,480
      Z2=Z1
      X2=X1
      Z1=ZAD
      X1=XAD
      DAD=-DAD
  480 G(L)=G(L)+DAD*CONE(X1,X2,Z1,Z2)
  500 CONTINUE
       IF(L.GT.1) DDG=G(L)-G(L-1)
  600 WOT 3,7,ZPT(L),G(L) ,DDG
      RETURN
      END
      FUNCTION CONE(R1,R2,Z1,Z2)
      B=ATANF((R2-R1)/(Z2-Z1))
      CB=COSF(B)
      SB=SINF(B)
      CB2=CB**2
      SB2=SB**2
      RB=R1*SB*CB-Z1*SB2
      ZR1=SQRT(Z1**2+R1**2)
      ZR2=SQRT(Z2**2+R2**2)
      T1=Z2-Z1-CB2*(ZR2-ZR1)
      T2=R1*CB2*SB-Z1*CB*SB2
      T3=ZR2*CB+Z2+RB
      T4=ZR1*CB+Z1+RB
      IF(T4.EQ.0.) GO TO ERR
      TT=T3/T4
      IF(TT.LT.0.) GO TO ERR
      CONE=T1+T2*LOGF(T3/T4)
      RETURN
  ERR WOT 59,FERR,T3,T4
 FERR FORMAT("ERROR IN CONE FUNCTION", 2E12.4)
      CALL QUIT
```

```
      SUBROUTINE LOGDEN
      USE DATACLICHE
C
C     ORDERS DEPTH,CALCULATES LOG DENSITY BETWEEN DEPTHS
C
C     FIRST ORDER DEPTHS
C
      DO 50 K=1,LDEP-1
      LSAV=K
      DO 25 J=K+1,LDEP
      IF(DPT(J).LT.DPT(LSAV)) GO TO 22
      IF((DPT(J).NE.DPT(K-1)).OR.(K.EQ.1)) GO TO 25
   22 LSAV=J
   25 CONTINUE
      DSAV=DPT(LSAV)
      DPT(LSAV)=DPT(K)
      DPT(K)=DSAV
   50 CONTINUE
C
C      NOW DEPTHS ARE ORDERED
C NOW ELIMINATE DUPLICATES
C
      CORR(1)=DPT(1)
      LORR=1
      DO 60 K=2,LDEP
      IF(DPT(K).EQ.CORR(LORR)) GO TO 60
      LORR=LORR+1
      CORR(LORR)=DPT(K)
   60 CONTINUE
      LDEP=LORR
      DO 65 K=1,LDEP
      DPT(K)=CORR(K)
   65 CORR(K)=0.
C
C      SEARCH LINES FOR THOSE CROSSING HOLE AND STORE DENSITIES
C
      XAFT=0.
      ENTRY VOLDEN
      KK=0
      DO 100 N=1,JZX
      DEL=1.
      IF((XA(N).LE.XAFT).AND.(XB(N).GE.XAFT))    ,100
      IF(XA(N).EQ.XB(N)) GO TO 70
      ZCT=(ZB(N)-ZA(N))*(XAFT-XA(N))/(XB(N)-XA(N))
      ZCT=ZA(N)+ZCT
      KK=KK+1
      ZORD(KK)=ZCT
      IF(XA(N).GT.XB(N)) DEL=-1.
      NRT(KK)=N
      CRHO(KK)=DEL*DRHO(N)
   70 IF(XB(N).EQ.XAFT)WOT 3,800
  100 CONTINUE
C
C       WE HAVE SELECTED ALL LINES WHOSE INTERSECTION WITH HOLES IS
C       ABOVE MAX DEPTH. NOW ORDER THEM
C
      DO 150 L=1,KK-1
      LSAV=L
      DO 125 J=L+1,KK
      IF(ZORD(J).LT.ZORD(LSAV))   ,125
```

```
      LSAV=J
  125 CONTINUE
      NSAV=NRT(LSAV)
      DSAV=CRHO(LSAV)
      ZSAV=ZORD(LSAV)
      CRHO(LSAV)=CRHO(L)
      ZORD(LSAV)=ZORD(L)
      NRT(LSAV)=NRT(L)
      CRHO(L)=DSAV
      ZORD(L)=ZSAV
      NRT(L)=NSAV
  150 CONTINUE
C
C       NOW SUM DENSITIES TO GET DENSITY BELOW EACH LINE
C
      DO 200 L=2,KK
  200 CRHO(L)=CRHO(L)+CRHO(L-1)
      IF(EX.NE.0.) RETURN
C
C      ORDERED SET OF DENSITIES AND DEPTHS
C      NOW GET AVERAGE DENSITIES BETWEEN DEPTHS
C       START BY FINDING LINES BELOW TOP DEPTH AND ABOVE BOTTOM
C      DEPTH FOR EACH DEPTH PAIR. FIRST TOP ONE
C
      DO 400 L=1,LDEP-1
      MM=NN=MN=0
      IF(DPT(L).LE.ZORD(1))  ,240
      IF(DPT(L+1).LE.ZORD(1))  GO TO 400
      MM=1
      RH1=0.
      Z1=ZORD(1)-DPT(L)
      GO TO 270
  240 DO 250 K=2,KK
      IF(ZORD(K).GT.DPT(L))  ,250
      MM=K
      RH1=CRHO(K-1)
      Z1=ZORD(K)-DPT(L)
      GO TO 270
  250 CONTINUE
      GO TO 410
  270 DO 300 K=2,KK
      IF(ZORD(K).GE.DPT(L+1))  ,300
      NN=K-1
      MN=K
      RH2=CRHO(NN)
      Z2=DPT(L+1)-ZORD(NN)
      GO TO 320
  300 CONTINUE
      NN=KK
      RH2=0.
      Z2=DPT(L+1)-ZORD(KK)
C
C     NOW WE HAVE MM--LINE BELOW THE TOP MEASURING POINT
C      AND NN--LINE ABOVE BOTTOM POINT
C     NOW GET DENSITIES
C
  320 IF (MM.EQ.MN)  ,330
      RHOL(L+1)=RH1
      GO TO 400
  330 PRHO=0.
```

```
      IF(MM.EQ.NN)   ,350
  340 RHOL(L+1)=(PRHO+RH1*Z1+RH2*Z2)/(DPT(L+1)-DPT(L))
      GO TO 400
  350 DO 360 J=MM,NN-1
      DZ=ZORD(J+1)-ZORD(J)
  360 PRHO=PRHO+DZ*CRHO(J)
      GO TO 340
  400 CONTINUE
  410 RETURN
  800 FORMAT ("WARNING--LINE ENDS ON HOLE DENSITY BAD ")
      END
```

```
*       FORTRAN
      SUBROUTINE UNCODE
      USE DATACLICHE
      BIG=1.E+7
      SML=0.
      IF(IDEC(5).EQ.3) GO TO SC01
      DRHO(JZX)=FINPUT(5)
      LET=0
      IF(IDEC(1).EQ.3) ,NX01
C
C     FIRST X IS ON SCARP
C
      IF(IDEC(2).EQ.3) GO TO ER01
      LET=INPUT(1)-40B
      IF(LET.EQ.12)  ,X105
C
C      LEFT X CENTER --LETTER IS L
C
      XA(JZX)=SML
      GO TO NX02
 X105 IF (LET.EQ.18)  ,X110
C
C    RIGHT X INFINITE--LETTER IS R
C
      XA(JZX)=BIG
      GO TO NX02
 X110 ZA(JZX)=FINPUT(2)
      CALL LOCATE(1,LET,ZA(JZX),2)
      GO TO NX03
 NX01 XA(JZX)=FINPUT(1)
      IF(IDEC(2).EQ.3)  ,NX02
C
C     FIRST Z IS ON THE SCARP
C
      LET=INPUT(2)-40B
      CALL LOCATE (1,LET,XA(JZX),1)
      GO TO NX03
 NX02 ZA(JZX)=FINPUT(2)
 NX03 IF (IDEC(3).EQ.3)  ,NX04
C
C       SECOND X IS ON THE SCARP
C
      IF (IDEC(4).EQ.3) GO TO ER01
      LET=INPUT(3)-40B
      IF(LET.EQ.12)   ,X205
C
C    LEFT X INFINITE--LETTER IS L
C
      XB(JZX)=SML
 X203 ZB(JZX)=FINPUT(4)
      RETURN
 X205 IF (LET.EQ.18)  ,X210
C
C   RIGHT X IS INFINITE--LETTER R
C
      XB(JZX)=BIG
      GO TO X203
 X210 ZB(JZX)=FINPUT(4)
      CALL LOCATE (2,LET,ZB(JZX),2)
      RETURN
```

```
 NX04 IF (IDEC(4).NE.3) GO TO NX05
C
C        SECOND Z IS ON SCARP
C
      LET=INPUT(4)-40B
      XB(JZX)=FINPUT(3)
      CALL LOCATE (2,LET,XB(JZX),1)
      RETURN
 NX05 IF (LET.EQ.0) GO TO ER03
      XB(JZX)=FINPUT(3)
      ZB(JZX)=FINPUT(4)
      RETURN
C
C       THIS CARD IS A SCARP LINE INPUT
C
 SC01 LET=INPUT(5)-40B
      JZX=JZX-1
      IF((LET.EQ.12).OR.(LET.EQ.18)) GO TO ER04
      ITST=IDEC(1)+IDEC(2)+IDEC(3)
      IF(ITST.NE.6) GO TO ER01
      ISIG(LET)=1
      XO(LET)=FINPUT(1)
      ZO(LET)=FINPUT(2)
      IF(IDEC(4).EQ.3)     ,L01
C
C     INPUT IS XO,ZO,ANGLE (DEGREES + TO -90)
C
      TH=FINPUT(3)
      IF(ABS(TH).GT.90.) GO TO ER05
      IF(ABS(TH).EQ.90.)   ,A05
      TTH(LET)=BIG
      RETURN
  A05 TH=TH*.01745
      TTH(LET)=TAN(TH)
      RETURN
C
C     INPUT IS XO,ZO,X1,Z1
C
  L01 DX=FINPUT(3)-XO(LET)
      DZ=FINPUT(4)-ZO(LET)
      IF (DZ.EQ.0)    ,L05
      TTH(LET)=0.
      RETURN
  L05 TTH(LET)=-(DZ/DX)
      RETURN
 ER01 WOT 59,FER1,JZX,(INPUT(L),L=1,5)
 FER1 FORMAT ("TWO CONSECUTIVE LETTER",I3,/,5A11)
      CALL QUIT(1)
 ER03 WOT 59,FER3,INPUT(1),JZX
 FER3 FORMAT ("BUG--NO LETTERS ON CARD",A11,I5)
      CALL QUIT(1)
 ER04 WOT 59,FER4,INPUT(5),JZX
 FER4 FORMAT ("ILLEGAL SCARP LETTER ",A11,I5)
      CALL QUIT(1)
 ER05 WOT 59,FER5,INPUT(3),JZX
 FER5 FORMAT("ANGLE MORE THAN 90 DEGREES ",A11,I5)
      CALL QUIT(1)
      END
```

```
      SUBROUTINE VOLUMES
      USE DATACLICHE
      DIMENSION MANG(5),DANG(5),NANG(5)
C
C THIS SR ASSIGNS ARTIFICIAL LINES TO LEFT AND RIGHT ENDS
C LISTS ALL LINES WITH COMMON INTERSECTIONS NA,NB
C FINDS ALL LINES SURROUNDING EACH VOLUME (STRICTLY,AREA)
C AND FINDS DENSITY OF EACH VOLUME
C
C FIRST DO ARTIFICIAL LINES
C
      DO A01 J=1,100
      DO A00 K=1,5
  A00 INA(J,K)=INB(J,K)=0
  A01 ILV(J)=IRV(J)=0
      AMX=1.E+7
      CALL AMINMX(ZA,1,JZX,1,ZAN,ZAX,M2,M1)
      CALL AMINMX(ZB,1,JZX,1,ZBN,ZBX,M4,M3)
      MX=M1
      IF(ZAX.LT.ZBX) MX=M3
      MN=M2
      IF(ZBN.LT.ZAN) MN=M4
      IF((XA(MX).EQ.TSTM).AND.(XB(MX).GT.TST)) GO TO A03
C
C IF SO MX IS BOTTOM LINE
C
      JZX=JZX+1
      XA(JZX)=XA2(JZX)=0.
      XB(JZX)=XB2(JZX)=AMX
      ZA(JZX)=ZA(MX)+.01
      ZB(JZX)=ZA2(JZX)=ZB2(JZX)=ZA(JZX)
      DRHO(JZX)=.0001
      NP=NP+1
      NA(JZX)=NP
      NP=NP+1
      NB(JZX)=NP
  A03 IF((XA(MN).EQ.TSTM).AND.(XB(MN).GT.TST)) GO TO A05
      JZX=JZX+1
      XA(JZX)=XA2(JZX)=0.
      XB(JZX)=XB2(JZX)=AMX
      ZA(JZX)=ZA(MN)-.01
      ZB(JZX)=ZA2(JZX)=ZB2(JZX)=ZA(JZX)
      DRHO(JZX)=.0001
      NP=NP+1
      NA(JZX)=NP
      NP=NP+1
      NB(JZX)=NP
  A05 CONTINUE
C
C HAVE PUT IN FAKE TOP AND BOTTOM LINES FROM L TO R IF NO REAL ONES
C
      I=1
      DO A10 J=1,JZX
      IF(XA(J).GT.TSTM) GO TO A10
      NLFT(I)=J
      ZLFT(I)=ZA(J)
      I=I+1
  A10 CONTINUE
      IEND=I-1
C
```

```
C NOW WE HAVE ALL LINES WITH XA ON LEFT BOUNDARY
C
      IF(IEND.LT.2) GO TO ERR1
      DO A20 I=1,IEND
      ZM=AMINAF(ZLFT,1,IEND,1,M)
      NLFF(I)=NLFT(M)
      ZLFT(M)=TST
  A20 CONTINUE
      JTP=NLFF(1)
C
C NOW NLFF LISTS LEFT-ENDING LINES ORDERED BY INCREASING Z
C
      JZY=JZX+1
      DO A30 I=1,IEND-1
      NA(JZY)=NA(NLFF(I))
      NB(JZY)=NA(NLFF(I+1))
      XA(JZY)=XB(JZY)=0.
      ZA(JZY)=ZA(NLFF(I))
      ZB(JZY)=ZA(NLFF(I+1))
  A30 JZY=JZY+1
      JZA=JZY-1
C
C NOW WE HAVE VERTICAL "LINES" WITH NA TOP,NB BOTTOM
C
      I=1
      DO A40 J=1,JZX
      IF(XB(J).LT.TST) GO TO A40
      NRT(I)=J
      ZRT(I)=ZA(J)
      I=I+1
  A40 CONTINUE
      IEND=I-1
C
C NOW WE HAVE LIST OF LINES ENDING ON RIGHT BOUNDARY
C
      IF (IEND.LT.2) GO TO ERR2
      DO A60 I=1,IEND
      ZM=AMAXAF(ZRT,1,IEND,1,M)
      NRTT(I)=NRT(M)
      ZRT(M)=-1000.
  A60 CONTINUE
C
C NOW ORDERED BY DECREASING Z
C
      JBT=NRTT(1)
      DO A70 I=1,IEND-1
      NR1=NRTT(I)
      NR2=NRTT(I+1)
      NA(JZY)=NB(NR1)
      NB(JZY)=NB(NR2)
      XA(JZY)=XB(JZY)=1.E+7
      ZA(JZY)=ZB(NR1)
      ZB(JZY)=ZB(NR2)
      JZY=JZY+1
  A70 CONTINUE
      JZY=JZY-1
C
C NOW NEW LINES WITH NA BOTTOM,NB TOP
C
C NOW SORT ALL LINES BY INTERSECTION
```

```
C
      DO B50 J=1,NP
      DO B40 L=1,JZY
      IF(NA(L).NE.J) GO TO B30
      DO B20 K=1,5
      IF(INA(J,K).NE.0) GO TO B20
C
C J IS INTERSECTION NO.,L IS LINE NO. THIS IS KTH LINE WITH NA=J
C
      INA(J,K)=L
      GO TO B30
  B20 CONTINUE
  B30 IF(NB(L).NE.J) GO TO B40
      DO B35 K=1,5
      IF(INB(J,K).NE.0) GO TO B35
      INB(J,K)=L
      GO TO B40
  B35 CONTINUE
  B40 CONTINUE
  B50 CONTINUE
C
C INA CONTAINS NOS. OF ALL LINES WITH XA AT INT. J,INB SAME FOR XB
C
      DO B80 J=1,NP
      ITT=0
      DO B60 K=1,5
  B60 IF(INA(J,K).NE.0) ITT=ITT+1
      DO B70 K=1,5
  B70 IF(INB(J,K).NE.0) ITT=ITT+1
      IF(ITT.LT.2) GO TO ERR3
C
C THIS IS A TEST FOR AT LEAST 2 LINES AT ALL INTERSECTIONS
C IT IS REDUNDANT WITH SUBROUTINE CHECK
C
  B80 CONTINUE
C
C NOW FIND WHICH LINES GO WITH WHICH VOLUME
C
      I=1
      ILT=1
      DO C200 J=1,JZX
      IF(J.EQ.JTP) GO TO C200
      IF (ILV(J).NE.0) GO TO C200
      ILV(J)=ILT
C
C TAKING EACH LINE AND,IF IT IS NNOT ALREADY ASSOCIATED ON ITS LEFT
C WITH A VOLUME,GIVE IT A VOLUME NUMBER. FIND ALL OTHER LINES ON SAME
C VOLUME. ROCEED CCW AROUND VOLUME TESTING EACH INTERSECTION
C
      ICT=1
      JXX=J
      NR=NB(J)
      NAB=2
      DO C150 K=1,25
      IANG=1
      IF(JXX.GT.JZX) GO TO LFS
      IF (NAB.NE.1) GO TO C02
C
C NAB SAYS WHETHER END IN QUESTION IS RIGHT OR LEFT.  FOR FIRST
C LINE, INTERSECTION MUST BE ON RIGHT
```

```
C
      ANG=ATAN2((ZA(JXX)-ZB(JXX)),(XB(JXX)-XA(JXX)))
      GO TO C03
  C02 ANG=ATAN2((ZB(JXX)-ZA(JXX)),(XA(JXX)-XB(JXX)))
C
C ANG IS ANGLE OF FIRST LINE WITH HORIZONTAL
C MUST REMEMBER Z IS POSITIVE DOWNWARDS IN CALCULATING ANGLE
C
  C03 DO C10 L=1,5
      INP=INA(NR,L)
      IF(INP.EQ.0) GO TO C11
C
C NO MORE LINES AT THIS INTERSECTION
C
      IF(INP.EQ.JXX) GO TO C10
C
C THIS IS THE LINE WE STARTED WITH
C
      IF(INP.GT.JZX) GO TO LFT
C
C ABOVE ARE VERTICALS AT BOUNDARY SO MUST BE NEXT LINE
C
      ANH=ATAN2((ZA(INP)-ZB(INP)),(XB(INP)-XA(INP)))
  C04 DING=ANG-ANH
      IF(DING.LT.0) DING=DING+6.283
      DANG(IANG)=DING
C
C THIS IS ANGLE BETWEEN FIRST LINE AND THAT BEING TESTED
C
      NANG(IANG)=INP
      MANG(IANG)=2
C
C MANG GIVES NAB OF OTHER END OF THIS LINE
C
      IANG=IANG+1
  C10 CONTINUE
C
C NOW HAVE LOOKED AT ALL LINES WITH A END AT INTERSECTION
C
  C11 DO C20 N=1,5
      INP=INB(NR,N)
      IF(INP.EQ.0) GO TO C21
      IF(INP.EQ.JXX) GO TO C20
      IF(INP.GT.JZX) GO TO C20
      ANH=ATAN2((ZB(INP)-ZA(INP)),(XA(INP)-XB(INP)))
      DING=ANG-ANH
      IF(DING.LT.0.) DING=DING+6.283
      DANG(IANG)=DING
      NANG(IANG)=INP
      MANG(IANG)=1
      IANG=IANG+1
  C20 CONTINUE
  C21 IANG=IANG-1
      DO C30 I=1,IANG
      IF(DANG(IANG).LT.0.) GO TO ERR4
  C30 CONTINUE
C
C NOW WE HAVE ANGLES OF ALL LINES AT INTERSECTION WITH FIRST LINE
C
      AM=AMINAF(DANG,1,IANG,1,M)
```

```
      JXX=NANG(M)
      IF(JXX.EQ.J) GO TO C180
C
C WE HAVE CHOSEN THE LINE WITH MININUM CW ANGLE WITH RESPECT TO
C PREVIOUS LINE. IF JXX.EQ.J IT IS OUR ORIGINAL LINE AND WERE
C DONE WITH THIS VOLUME
C
      NAB=MANG(M)
      IF(NAB.EQ.1) GO TO C50
      IF(ILV(JXX).NE.0) GO TO ERR8
      ILV(JXX)=ILT
      NR=NB(JXX)
C
C INTERSECTION IS NA END OF NEW LINE,SO NEXT INTERSECTION IS NB
C
      GO TO C150
  C50 IF (IRV(JXX).NE.0) GO TO ERR8
      IRV(JXX)=ILT
      NR=NA(JXX)
C
C INTERSECTION IS NB END
C
      GO TO C150
  LFS ANG=1.57
      IF(JXX.GT.JZA) ANG=-ANG
      GO TO C03
  LFT ANH=1.57
      IF(INP.LE.JZA) ANH=-ANH
      GO TO C04
C
C THIS IS THE ARTIFICIAL VERTICAL LINE SETTING ANGLE TO  90 DEG
C
 C150 ICT=ICT+1
      GO TO ERR5
 C180 IF (ICT.LT.3) GO TO ERR5
      ILT=ILT+1
 C200 CONTINUE
      ILT=ILT-1
      IF(ILT.LT.2) GO TO ERR9
C
C NOW WE HAVE ILV AND IRV FOR ALL LINES
C THAT IS,WE KNOW WHICH VOLUME IS ON EACH SIDE OF EACH LINE
C
      DO C300 J=1,JZX
      IF((ILV(J).EQ.0).AND.(J.NE.JTP)) GO TO ERR6
      IF((IRV(J).EQ.0).AND.(J.NE.JBT)) GO TO ERR6
 C300 CONTINUE
      XLF=0.
      XRT=TSTM
      DO D20 J=1,JZX
      IF(XB(J).GT.TST) GO TO D20
      IF(XB(J).GT.XRT) XRT=XB(J)
  D20 CONTINUE
C
C XRT IS LARGEST NON-BOUNDARY X, XLF SMALLEST
C
      DO D100 I=1,ILT
      NI(I)=1
      DO D50 J=1,JZY
      IF(ILV(J).NE.I) GO TO D40
```

```
      NV(NI(I),I)=J
      KV(NI(I),I)=1
      NI(I)=NI(I)+1
C
C ILV(J)IS I
C
      GO TO D50
  D40 IF(IRV(J).NE.I) GO TO D50
      NV(NI(I),I)=J
      KV(NI(I),I)=2
      NI(I)=NI(I)+1
  D50 CONTINUE
      NI(I)=NI(I)-1
 D100 CONTINUE
C
C WE HAVE NOW LISTED ALL LINES SURROUNDING EACH VOLUME (I ABOVE)
C AS NV(NI) WHERE THERE ARE NI LINES FOR EACH VOLUME
C (AND ILT VOLUMES)
C KV IS 1 IF VOLUME IS ON LEFT OF LINE,2 IF ON RIGHT
C
C NOW FIND DENSITIES AND MIDPOINTS OF VOLUMES
C
      DO E200 I=1,ILT
      NIT=2*NI(I)
      L=0
      DO E20 K=1,NI(I)
      L=L+1
      JK=NV(K,I)
      XN(L)=XA(JK)
      IF(XN(L).GT.TST) XN(L)=XRT+.1*ABS(XRT)
      ZN(L)=ZA(JK)
      L=L+1
      XN(L)=XB(JK)
      IF(XN(L).GT.TST) XN(L)=XRT+.1*ABS(XRT)
  E20 ZN(L)=ZB(JK)
      CALL AMINMX(XN,1,NIT,1,AN,AX,M1,M2)
      XMD(I)=(AX+AN)/2.
C
C MIDDLE X OF VOLUME
C
      XMX(I)=AX
      XMN(I)=AN
      CALL AMINMX (ZN,1,NIT,1,AN,AX,M1,M2)
      ZMX(I)=AX
      ZMN(I)=AN
      XAFT=XMD(I)
      EX=1.
      CALL VOLDEN
      EX=0.
C
C FIRST FIND A Z INSIDE VOLUME,THEN DENSITY
C
      DO E50 K=1,KK-1
      N1=NRT(K)
      N2=NRT(K+1)
      IF((IRV(N1).EQ.I).AND.(ILV(N2).EQ.I)) GO TO E60
  E50 CONTINUE
      GO TO ERR7
  E60 ZMD(I)=(ZORD(K+1)+ZORD(K))/2.
      RMD(I)=CRHO(K)
```

```
      GO TO E200
 E200 CONTINUE
      JZB=JZA+1
      DO E250 J=JZB,JZY
 E250 DRHO(J)=-RMD(ILV(J))
C
C  ASSIGNING DELTA RHO TO OUTSIDE LINES. NEGATIVE BECAUSE
C    RIGHT IS OUTSIDE
C
      DO E300 J=1,JZX
      DELR=RMD(IRV(J))-RMD(ILV(J))
      DELT=DELR-DRHO(J)
 E300 IF (ABS(DELT).GT.0.01) GO TO ERRO
      RETURN
 ERR1 WOT 59,FER1
C    FROM NEAR A10
      GO TO F100
 ERR2 WOT 59,FER2
C   FROM NEAR A40
      GO TO F100
 ERR3 WOT 59,FER3,J
C    FROM B70
      GO TO F100
 ERR4 WOT 59,FER4,NR,INP.ILT,IANG,DANG
C    FROM C30
      GO TO C150
 ERR5 WOT 59,FER5,ILT,JXX,NR
C     FROM C180
      GO TO F100
 ERR6 WOT 59,FER6,J
C    FROM C300
      GO TO F100
 ERR7 WOT 59,FER7,I
C     FROM E54
      GO TO F100
 ERR8 WOT 59,FER8,JXX,ILT,ILV(JXX),IRV(JXX)
C    FROM C50
      GO TO F100
 ERRO WOT 59,FERO,J,ILV(J),IRV(J),DRHO(J),RMD(ILV(J)),RMD(IRV(J)),DELR
C    FROM E300
      GO TO F100
 ERR9 WOT 59,FER9
      GO TO F100
 FER1 FORMAT(" ONLY ONE POINT ON LEFT BOUNDARY")
 FER2 FORMAT(" ONLY ONE POINT ON RIGHT BOUNDARY")
 FER3 FORMAT(" ONLY ONE LINE AT INTERSECTION ",I3)
 FER4 FORMAT("NEGATIVE ANGLE AT INTERSECTION ",I3," LINE",I3,
     1 " VOLUME",I3," ANGLE",I3," IS",E12.4)
 FER5 FORMAT("ONLY 2 LINES FOR VOLUME",I3," LAST LINE WAS",I3,
     1 " LAST INT WAS",I3)
 FER6 FORMAT("LINE NO",I3," HAS NO VOLUME ON ONE SIDE")
 FER7 FORMAT ("CANT FIND DENSITY FOR VOLUME",I4)
 FER8 FORMAT("TRYING TO ASSIGN VOL" ,I3," TO LINE",I3," VOL",I3,
     1 " ALREADY ON L. OR ",I3," ON R")
 FER9 FORMAT("ONLY ONE VOLUME IN THE PROBLEM")
 FERO FORMAT("DENSITY CONTRAST ON LINE",I3," VOL",I3,"ON L AND",I3,
     1 " ON R IS ",E12.4 /,"DISAGREES WITH DIFF BETWEEN L AND R VALUES "
     2 ,2E12.4,/," YIELDING ",E12.4)
 F100 EX=1.
      RETURN
```

```
      SUBROUTINE CHECK
      USE DATACLICHE
C
C    TESTS TO ASSURE ALL POINTS EXCEPT AT SIDE BOUNDARIES ARE DUPLICATED
C     IF NA AND NB GE 1 POINT IS OK
C     WE ARE SETTING NA AND NB TO INTERSECTION NUMBER IN THIS
C     REVISION.  NOT SET UNTIL EITHER TWO LINES MEET THERE
C     OR IT IS A RIGHT OR LEFT BOUNDARY
C     NEED PROVISION FOR TRIPLICATE
C
      NLA=NLB=0
      DO 10 N=1,100
   10 NA(N)=NB(N)=0
      NP=1
      TTT=.1
      TMX=100000.
      DO 20 N=1,JZX
      IF(XA(N)-XB(N))20,12,11
   11 SAX=XA(N)
      SAZ=ZA(N)
      XA(N)=XB(N)
      ZA(N)=ZB(N)
      XB(N)=SAX
      ZB(N)=SAZ
      DRHO(N)=-DRHO(N)
      GO TO 20
   12 XB(N)=XB(N)+.001
   20 CONTINUE
C
C ABOVE ROUTINE MAKES ALL XA THE LEFT HAND (MINIMUM X) VALUE OF THE
C XA,XB PAIR
C
C NOW TEST FOR LINES INTERSECTING NOT AT PLANNED INTERSECTIONS
C G AND CORR USED TEMPORARILY AS COEFFICIENTS OF LINES
C
      DO C10 J=1,JZX
      G(J)=(ZB(J)-ZA(J))/(XB(J)-XA(J))
  C10 CORR(J)=ZB(J)-G(J)*XB(J)
C
C Z=G*X+CORR
C
      DO C50 J=1,JZX-1
      NN=J+1
      DO C40 N=NN,JZX
      IF(N.EQ.J) GO TO C40
      DDG=ABS(G(N)-G(J))
      IF(DDG.LT.0.01) GO TO C40
      XX=(CORR(N)-CORR(J))/(G(J)-G(N))
      X1=XX-XA(J)-.01
      X2=XX-XA(N)-.01
      IF((X1.LT.0.).OR.(X2.LT.0.)) GO TO C40
      X1=XX-XB(J)+.01
      X2=XX-XB(N)+.01
      IF((X1.GT.0.).OR.(X2.GT.0.)) GO TO C40
      GO TO CERR
  C40 CONTINUE
  C50 CONTINUE
      DO C60 J=1,100
      CORR(J)=0.
  C60 G(J)=0.
```

```
      DO 100 N=1,JZX
      IF(XA(N).GE.0.) GO TO C61
      WOT 59,FMB,N,XA(N),ZA(N),XB(N),ZB(N)
      GO TO 220
  C61 IF(XA(N).GT.0.) GO TO 30
      IF(NA(N).GT.0) GO TO 100
      NA(N)=NP
      NP=NP+1
      GO TO 100
C
C TEST FOR XA EQ. CENTER OR R END
C
   30 IF(NA(N).GE.1) GO TO 100
      JJ=N+1
      IF(N.EQ.JZX) GO TO 50
      DO 40 J=JJ,JZX
      T1=ABS(XA(N)-XA(J))
      T2=ABS(ZA(N)-ZA(J))
      IF((T1.GT.TTT).OR.(T2.GT.TTT)) GO TO 40
C
C COMPARING LEFT ENDS--IF NO COMMON INTERSECTION GO TO 40
C
      IF(NA(N).LE.0) GO TO 31
      NA(J)=NA(N)
      GO TO 40
   31 IF(NA(J).LE.0) GO TO 32
      NA(N)=NA(J)
      GO TO 40
   32 NA(N)=NA(J)=NP
      NP=NP+1
   40 CONTINUE
   50 DO 80 K=1,JZX
      T1=ABS(XA(N)-XB(K))
      T2=ABS(ZA(N)-ZB(K))
      IF((T1.GT.TTT).OR.(T2.GT.TTT)) GO TO 80
C
C COMPARING ONE LEFT TO OTHER RIGHT END--IF NO COMMON INT GO TO 80
C
      IF(NA(N).LE.0) GO TO 71
      NB(K)=NA(N)
      GO TO 80
   71 IF(NB(K).LE.0) GO TO 72
      NA(N)=NB(K)
      GO TO 80
   72 NA(N)=NB(K)=NP
      NP=NP+1
   80 CONTINUE
      IF(NA(N).LT.1) NLA=1
  100 CONTINUE
      DO 200 M=1,JZX
      IF(ABS(XB(M)).LT.TMX) GO TO 130
      NB(M)=NP
      NP=NP+1
      GO TO 200
  130  IF(NB(M).GE.1) GO TO 200
      IF(M.EQ.JZX) GO TO EROB
      LL=M+1
      DO 140 L=LL,JZX
      T1=ABS(XB(M)-XB(L))
      T2=ABS(ZB(M)-ZB(L))
```

```
      IF((T1.GT.TTT).OR.(T2.GT.TTT)) GO TO 140
C
C COMPARING RIGHT ENDS IF NO COMMON INTERSECTION GO TO 140
C
      IF(NB(M).LE.0) GO TO 131
      NB(L)=NB(M)
      GO TO 140
  131 IF(NB(L).LE.0) GO TO 132
      NB(M)=NB(L)
      GO TO 140
  132 NB(M)=NB(L)=NP
      NP=NP+1
  140 CONTINUE
      IF(NB(M).LT.1) NLA=1
  200 CONTINUE
      NP=NP-1
      IF(NLA.LT.1) GO TO 250
C
C ALL IS WELL
C
      DO 210 N=1,JZX
      NR=0
C
C FINDING THE BAD INTERSECTION
C
      IF(NA(N).GE.1) GO TO 205
      NR=1
      WOT 59,FMB,N,XA(N),ZA(N),XB(N),ZB(N)
      GO TO 210
  205 IF((NB(N).LT.1).AND.(NR.LT.1)) GO TO 208
      GO TO 210
  208 WOT 59,FMA,N,XA(N),ZA(N),XB(N),ZB(N)
  210 CONTINUE
  220 EX=1.
  250 RETURN
 EROB WOT 59,FMA,JZX,XA(JZX),ZA(JZX),XB(JZX),ZB(JZX)
      GO TO 220
  FMA FORMAT("LINE",I3," WITH ENDS AT",4E12.3," HAS BAD RIGHT INT")
  FMB FORMAT("LINE",I3," WITH ENDS AT",4E12.3," HAS BAD LEFT INT")
 CERR WOT 59,CFT,J,N,XX
      GO TO 220
  CFT FORMAT (" LINES ",I2," AND ",I3," CROSS AT ",E12.4)
      END
```