UNITED STATES DEPARTMENT OF THE INTERIOR

GEOLOGICAL SURVEY

IBM PC PROGRAMS FOR AUTOMATIC PROCESSING AND INTERPRETATION OF WENNER SOUNDING CURVES in QuickBASIC 4.0

By

Robert J. Bisdorf and Adel A.R. Zohdy

BOX 25046, MS 964; Denver, CO 80225

Open-File Report 90-211 A & B

1990

90-211A Manual, examples, and program listing.

90-211B Disk with computer programs (source and executable) and test examples.

For monthly list.

ATOWEN is an automatic interpretation program for Wenner sounding curves. Required equipment: IBM PC or compatible, CGA or EGA capability, 1-5 1/4" disk drive. Recommended equipment: math coprocessor. Optional equipment: Epson compatible printer. OF90-211A, Documentation and program listing, 49 p. OF90-211B Executable and source on 5 1/4" diskette.

IBM PC PROGRAMS FOR AUTOMATIC PROCESSING AND INTERPRETATION

OF WENNER SOUNDING CURVES in QuickBASIC 4.0

By

Robert J. Bisdorf and Adel A.R. Zohdy

SYSTEM REQUIREMENTS

These programs require an IBM PC or a compatible computer running under MS-DOS version 2.1 or higher. The computer must be equipped with an EGA (Enhanced Graphics Adaptor) or CGA (Color Graphics Adaptor) card and at least 256 K RAM (random access memory). A math co-processor chip is strongly recommended but is not required. If the proper graphics card is not available the program displays a message to that effect and then exits.

An Epson or compatible printer is required to dump graphics and obtain a hard copy.

INTRODUCTION

This open file is composed of two parts: (a) The manual you are reading now (including examples and program listings which are given in the appendix), and (b) the disk with computer programs and test examples.

Two computer programs are presented. The first program (ATOWEN.EXE) is based on a Schlumberger program (Zohdy, and Bisdorf, 1989) modified to use convolution filter coefficients for Wenner soundings (Koefoed, 1979, p. 95). This program is a Wenner implementation of a new method for the automatic interpretation of sounding curves obtained over horizontally stratified media (Zohdy, 1989) The second program (PICKCONT.EXE) is a utility program that reads layering files created by ATOWEN.EXE and automatically interpolates the depths at preselected resistivity contour values. The resulting list of depths and resistivities facilitates the construction of contoured geoelectric cross sections.

The programs were written in Microsoft QuickBASIC 4.0. Each program is composed of several modules. The complete listings of the source code for the various modules are given in appendices A and B.

DISK CONTENTS

The accompanying disk contains the following:

1) ATOWEN.EXE is the executable form of the automatic interpretation program.
2) ATOWEN.BAS is the main source code module of the automatic interpretation program written in QuickBASIC version 4.0.

It requires three modules: ATOSUB.BAS, DRCT.BAS, WENN.BAS. (See appendix A for a complete listing of ATOWEN.BAS, ATOSUB.BAS, DRCT.BAS, and WENN.BAS).

3) ATOSUB.BAS is a module containing several sub-programs that are used with ATOWEN.BAS.
4) DRCT.BAS is a second module containing several sub-programs that are used with ATOWEN.BAS.
5) WENN.BAS is third module containing several sub-programs that are used with ATOWEN.BAS.
6) ATO.MAK is a make file created by QuickBASIC 4.0 to call the modules ATOWEN.BAS, ATOSUB.BAS, DRCT.BAS, and WENN.BAS.
7) EGADUMPL.COM is a public domain (PD) program for dumping EGA graphics on an Epson (or compatible) printer.
8) PICKCONT.EXE is the executable form of the pick contour program.
9) PICKCONT.BAS is the main module of the pick contour program written in QuickBASIC 4.0. It requires the following modules: ATOSUB.BAS, CONTSUB.BAS, and DRCT.BAS. The first and third modules are the same as those used with ATOWEN.BAS. (see appendix B for a complete listing of PICKCONT.BAS and CONTSUB.BAS).
10) CONTSUB.BAS is a module consisting of one subprogram that loads the preselected resistivity contour values.
11) PICKCONT.MAK is a make file created by QuickBASIC 4.0 to call the modules: PICKCONT.BAS, ATOSUB.BAS, CONTSUB.BAS, and DRCT.BAS.
12) CONTVAL.DAT is an example file containing user selected resistivity contour values for program PICKCONT.
13) Directory named WTEST. This directory contains example data files.

GETTING STARTED

Before running the automatic interpretation program, make a backup copy of the disk. If you do not have a hard disk then make sure that you have a formatted floppy disk on which to save files. If you wish to copy the executable program Atowen.exe to a hard disk, make sure to copy the PD program EGADUMPL.COM also.

To run the program, type: ATOWEN and press <ENTER>. In this manual, and on screen displays, we use <ENTER> to indicate the <ENTER> or <RETURN> key, and we use <Esc> to indicate the Esc (or Escape) key.

The program is menu driven and most prompts require a single key stroke: Y, N, a letter, a number, <Esc>, or <ENTER>. Prompts requiring a single key stroke, begin with the word PRESS. Prompts requiring an entry followed by a carriage return, end with the word <ENTER>. Furthermore, almost every menu or prompt includes a way (press <Esc>) or choice (press an appropriate number) to get out of that particular menu or prompt. Pressing <Esc> will return you to the MAIN MENU or to a similar menu.

After the program is loaded into memory, a screen is displayed showing the program title, credits, the program version, and the message "Press any Key to Continue". After you press a key, the program will proceed and test the computer system for the presence of an EGA. If successful, the program will load the PD program "EGADUMPL.COM". If unsuccessful, the program will test the system for the presence of CGA. If successful, it will load "GRAPHICS.COM" from DOS. If unsuccessful it will display an appropriate message and then exit. For the program to find the file "GRAPHICS.COM", the appropriate path should have been set up by the user in the AUTOEXEC.BAT file.

DISK DRIVE MENU

Under MS-DOS, disk drives are designated by the letters A and B (for f8loppy drives) or C (for hard drive). Hard disks may be partitioned by the user and designated by other letters.

The Disk Drive Menu is a single prompt that asks you to press the appropriate letter (A, B, C, etc.) for the disk drive that you wish to save data to or to retrieve data from. If you do not wish to save the sounding data, press <Esc>. NOTE: If you press <Esc> and you change your mind later, you cannot save the sounding data you entered without returning to the DISK DRIVE MENU, designating a drive, and reentering the data.

MAIN MENU

The Main Menu has four options:

1. ENTER new sounding from keyboard.

2. LOAD a sounding from disk.

3. RETURN to DISK MENU.

4. QUIT.

Press the appropriate number to proceed.

Whether you wish to ENTER or LOAD data, the next prompt will request the name of the Survey Area. Soundings are grouped by Survey Area.

Individual files for: field sounding data, digitized sounding data, and interpreted layering are automatically saved under a single directory with file extensions of FLD, DGT, and LYR respectively. This directory is automatically created by the program. Its name is based on the name of the survey area, and it is always placed under the Root Directory of the selected drive.

SURVEY AREA

The name of the survey area can be more than one word, contain a space, and can be up to 25 characters long. For example: LONE MOUNTAIN, Tule Valley, Alamo road, etc., are all valid names. Regardless of how you type the name of the Survey Area, it is automatically capitalized.

In order to minimize the amount of typing required by the user, the name of the Survey Area is entered only once (after selecting ENTER or LOAD from the MAIN MENU). Furthermore, the name of the survey area is automatically used in the TITLE of the sounding, in the name of the DIRECTORY, and in the FILE NAME. The following paragraphs describe how this is done.

Survey Area and Sounding Title:

The complete name of the survey area (including spaces, apostrophes, hyphens, etc) is used as the first part of the title of a sounding. The second part of the sounding title is the sounding number (which is provided by the user at a subsequent prompt).

Example: the TITLE of sounding number 5 obtained in the Lone Mountain area will be automatically generated and saved by the program as: LONE MOUNTAIN 5.

Survey Area and Name of Directory:

The program will capitalize all the letters in the name of the SURVEY AREA, delete the delimiters (spaces, apostrophes, hyphens, commas, etc) and use the first 8 characters as the name of a directory to save data to or to retrieve data from.

Example: the name of the directory for data obtained in the Lone Mountain survey area is LONEMOUN. If a directory with the name LONEMOUN does not exist under the root directory, you will be asked if you wish to create a new directory with that name.

Survey Area and File Name:

In MS-DOS, each file name can be up to 8 characters long plus 3 extension characters (for a total of 11 characters).

The first 4 characters in the file name are taken from the first four characters of the Survey Area. The next four characters in the file name are reserved for the sounding number (e.g. 125). A letter indicating that the sounding was smoothed (S), extended (X), or corrected (C) for the effect of a winding road (Zohdy and Bisdorf, 1982) may be used as part of the sounding number in the file name (e.g. 125X). The three extension characters in a file name are

automatically designated as FLD (for field), DGT (for digitized), or LYR (for layering).

Example: the FIELD data for sounding 5 obtained in the Survey Area named Lone Mountain will be saved in a sequential file named LONE5.FLD. This file will be listed in the LONEMOUN directory, which is listed under the Root Directory.

UNITS

The Wenner current-electrode spacings ("a") may be entered in feet or meters. Thus when the prompt:

"Electrode spacings: in Feet or Meters ? (F/M) "

is displayed, press F for feet or M for meters. If you enter the "a" spacings in feet, you will have the option later to convert them to meters and to display the sounding curve in the units you decided upon. The apparent resistivities are assumed to be in ohm-meters. The units for interpreted layer depths are the same as those used for the electrode spacings.

SOUNDING NUMBER

When prompted to enter a sounding number, enter only the appropriate number. You need not type the name of the survey area repeatedly for every sounding you wish to enter or load. As mentioned earlier, the program will combine the name of the SURVEY AREA and the sounding number to form the sounding title.

ENTERING SOUNDING DATA

Two methods are provided to enter sounding data. You may enter the data as obtained in the field (in electrode spacings "a" and apparent resistivity values), or you may enter a digitized sounding curve. In either method, the "a" units may be in feet or meters, the apparent resistivities are assumed to be in ohm-meters.

Standard Field Data:

For entering field data, enter all the "a" values first, all the apparent resistivities second, and then you may edit any errors using the provided edit submenu.

The "a" values must be entered in ascending order. When done entering all the "a" values, type 0 (zero) and press <ENTER>. Now you may enter the corresponding apparent resistivities.

The edit submenu allows you to Continue, Edit, Delete, Insert, Append, or Escape. You can not Continue unless the number of "a" equals the number of apparent resistivities.

Digitized Data:

You may wish to process the sounding curve and smooth it manually prior to entering it. If you do, you must digitize the sounding curve at the rate of 8 points per logarithmic (log) cycle.

To enter a digitized sounding, the program will prompt you to enter the smallest "a" value, and then all you have to do is to enter the digitized apparent resistivity values, as the program automatically increments the "a" values at the rate of 8 points per log cycle.

GRAPHIC DISPLAY

When the data for a given sounding are entered and you press C (to Continue), the sounding curve will be displayed on a 4 x 3 log-log graph with the message:

" CURVE LOOKS OK ? (Y/N/<Esc>)."

If you press N, the program returns to the edit screen. If you press Y, you will be asked:

" Save this Sounding ? (Y/N/<Esc>) "

if you press Y, then you will be asked if you wish to:

" Enter another sounding (Y/N/<Esc>) ? "

If you press Y, the program returns to SOUNDING NUMBER to enter another sounding. If you press N to either of the above two prompts, the program proceeds with processing the entered sounding data as follows.

If you entered the "a"s in FEET you will be asked if you wish to convert them to METERS. Next you will be asked if you wish to:

" LIST field data and DUMP GRAPHICS ? (Y/N/<Esc>) "

If you press Y you will be prompted with a message to make sure that the printer is turned on, to adjust the paper and to press <ENTER> when ready. Figure 1 shows an example of a field curve named WTEST 1. The "a" values were entered in feet and converted to meters. The listing of the field data, and the EGA screen dump were made on an Epson FX-185 dot matrix printer.

DATA PROCESSING

The field curve is digitized (or sampled), from right to left, at the rate of 8 points per log cycle. The digitized curve is displayed with a prompt to save it. If the digitized curve is saved the file will automatically have the extension DGT. The digitizing

WTEST 1 (FIELD DATA)

A, m	(ft)	App. Res.	A, m	(ft)	App. Res.
0.96	(3.15)	350.00	20.69	(67.86)	47.00
1.41	(4.62)	345.00	30.36	(99.61)	45.00
2.07	(6.79)	340.00	44.56	(146.21)	53.00
3.04	(9.96)	315.00	65.41	(214.61)	67.00
4.46	(14.62)	285.00	96.01	(315.00)	83.00
6.54	(21.46)	220.00	140.94	(462.40)	94.00
9.60	(31.50)	140.00	206.84	(678.60)	100.00
14.09	(46.24)	76.00	303.61	(996.10)	93.00
			445.65	(1462.10)	74.00

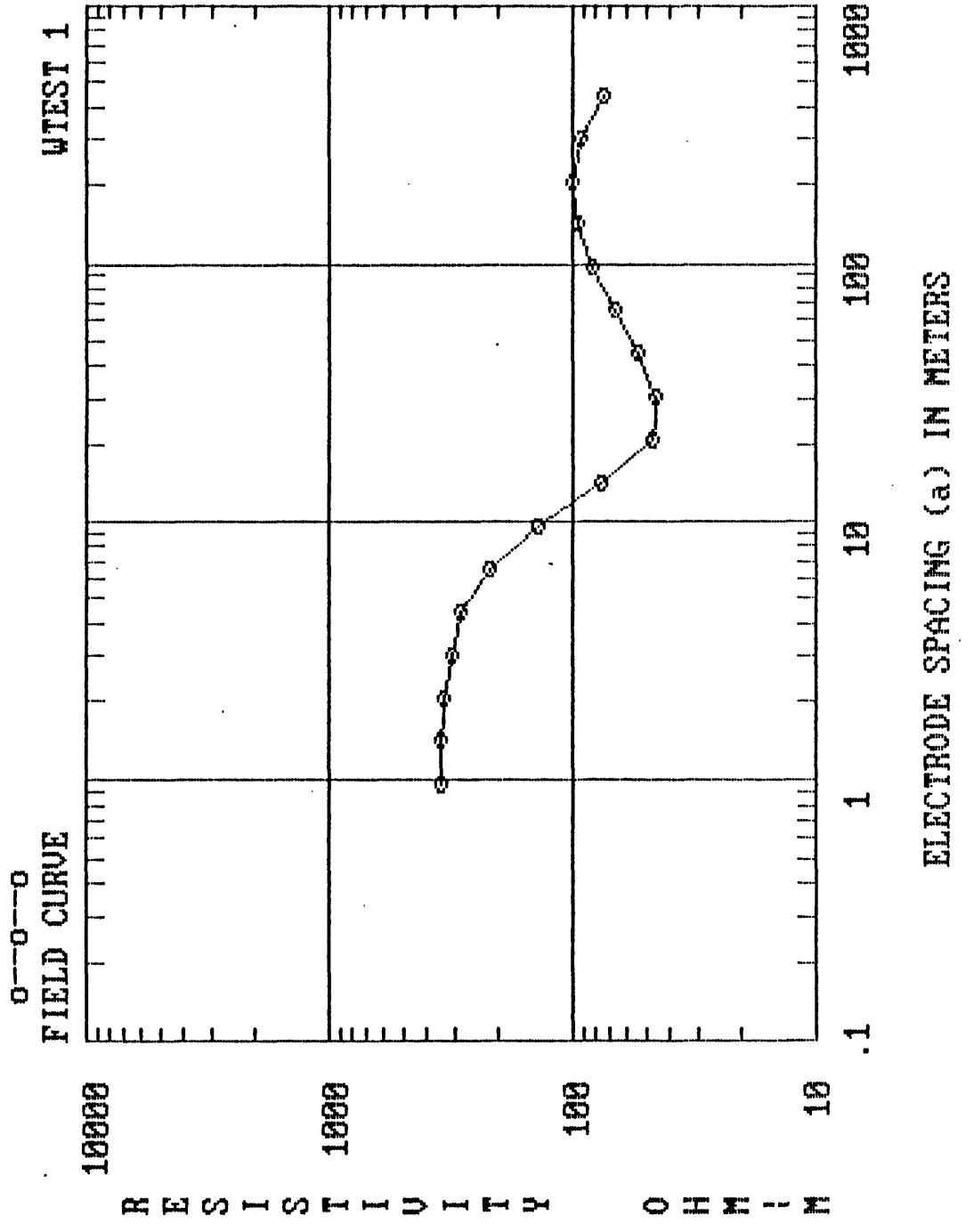

Figure 1. Example of the graphics output for a field curve.

is done using a cubic spline routine (Anderson, 1971) converted to QuickBASIC.

OPTIONS

Three options are offered prior to the automatic interpretation of a sounding curve. These are:

* Fix DEPTH-SHIFT factor at (..., 0.7, 0.8, ...) ?
* Change Number of Layers per cycle from 8 to ...?
* Fix last layer resistivity at ?

It is not recommended that any of these options be used before becoming familiar with the program.

* The DEPTH-SHIFT factor refers to the shifting of the digitized electrode spacings to determine depths (Zohdy, 1989). The program is written so that an optimum depth-shift factor, usually in the range between .35 and .5, is determined. A fixed depth-shift factor may be selected by the user to fix the depths or to help in using layer compression.

* Changing the number of layers per log cycle determines the amount of layer compression or expansion. The default value is 8. For values greater than 8 (9, 10 etc) the number of layers per log cycle is increased, hence the layers are compressed. Conversely, the number of layers per log cycle may be made less than 8 (7, 6, etc) to expand the layer thicknesses. Layer compression/expansion may be used alone or in conjunction with fixing the depth-shift factor. For sounding curves with flat left branches, compression may be applied with a depth-shift factor that is equal to or greater than unity to produce highly compressed layers.

* The resistivity of the last layer may be fixed by the user at any positive value.

NOTE: the use of any or all of the above options does not guarantee a successful fit. Experimentation with these options is useful in imposing some constraints and in generating generally unlikely models.

AUTOMATIC INTERPRETATION

The automatic interpretation is generally done in two sets of iterations (two passes). In the first set of iterations, the best fitting theoretical sounding curve is determined. If the least root mean square misfit (between digitized and calculated apparent resistivities) is greater than 2 percent then this means that the digitized curve is noisy and therefore anomalous layers may have been created. To eliminate these anomalous layers, the best fitting theoretical curve is reinterpreted as if it is a smoothed version of the digitized curve. If the digitized curve, under interpretation, is an exceptionally smooth and complete curve and is fitted within 2 percent in the first pass, then a second set of iterations is not performed (Zohdy, 1989).

The user is kept aware of what is going on with a display of the root mean square values. When the interpretation is completed, a plot of the digitized data, the calculated sounding curve, and the corresponding layering is displayed. The following prompt is displayed at the top of the screen:

" SAVE LAYERING ? (Y/N/<Esc>)"

If you press Y the layering (depths and resistivities) is automatically saved in a file with the extension LYR. The next prompt is:

" LIST RESULTS and DUMP GRAPHICS ? (Y/N/<Esc>) "

If you press Y then a prompt is displayed to urge you to make sure that the printer is turned on, to adjust the paper, and to press <ENTER> when ready. Figure 2 shows the results of interpreting sounding WTEST 1. The table of depths and resistivities is followed by a screen dump of the graphics.

Following the graphics dump, or if the answer to the above prompt was N, a menu with six options (including Quit) is displayed. The items on this menu are self explanatory.

LOADING AN EXAMPLE

Several test examples, including the sounding named WTEST 1, are given under the directory WTEST. To run one of the test examples type WTEST for the name of the Survey Area then select LOAD from the Main Menu. The WTEST Directory contains only a few files. When prompted for the sounding number you wish to load, you may type FILE to find out what sounding numbers exist under the directory WTEST (which represents the name of a survey area). If you type FILE, the program will display the file names with extensions FLD and DGT. You may use the cursor keys to highlight the file names. Highlight the file name you want and then press <ENTER>, or

WTEST 1 (INTERPRETATION)

DEPTH, m	(ft)	RESIS.	DEPTH, m	(ft)	RESIS.
0.67	(2.18)	344.12	15.79	(51.80)	421.52
0.89	(2.91)	365.58	21.05	(69.07)	411.80
1.18	(3.88)	390.10	28.08	(92.11)	400.00
1.58	(5.18)	340.26	37.44	(122.83)	371.53
2.11	(6.91)	306.73	49.93	(163.80)	298.84
2.81	(9.21)	383.23	66.58	(218.43)	248.24
3.74	(12.28)	375.75	88.78	(291.28)	163.90
4.99	(16.38)	267.29	118.39	(388.43)	183.49
6.66	(21.84)	133.03	157.88	(517.99)	168.14
8.88	(29.13)	51.05	210.54	(690.75)	101.81
11.84	(38.84)	24.28	99999.00	(99999.00)	41.57

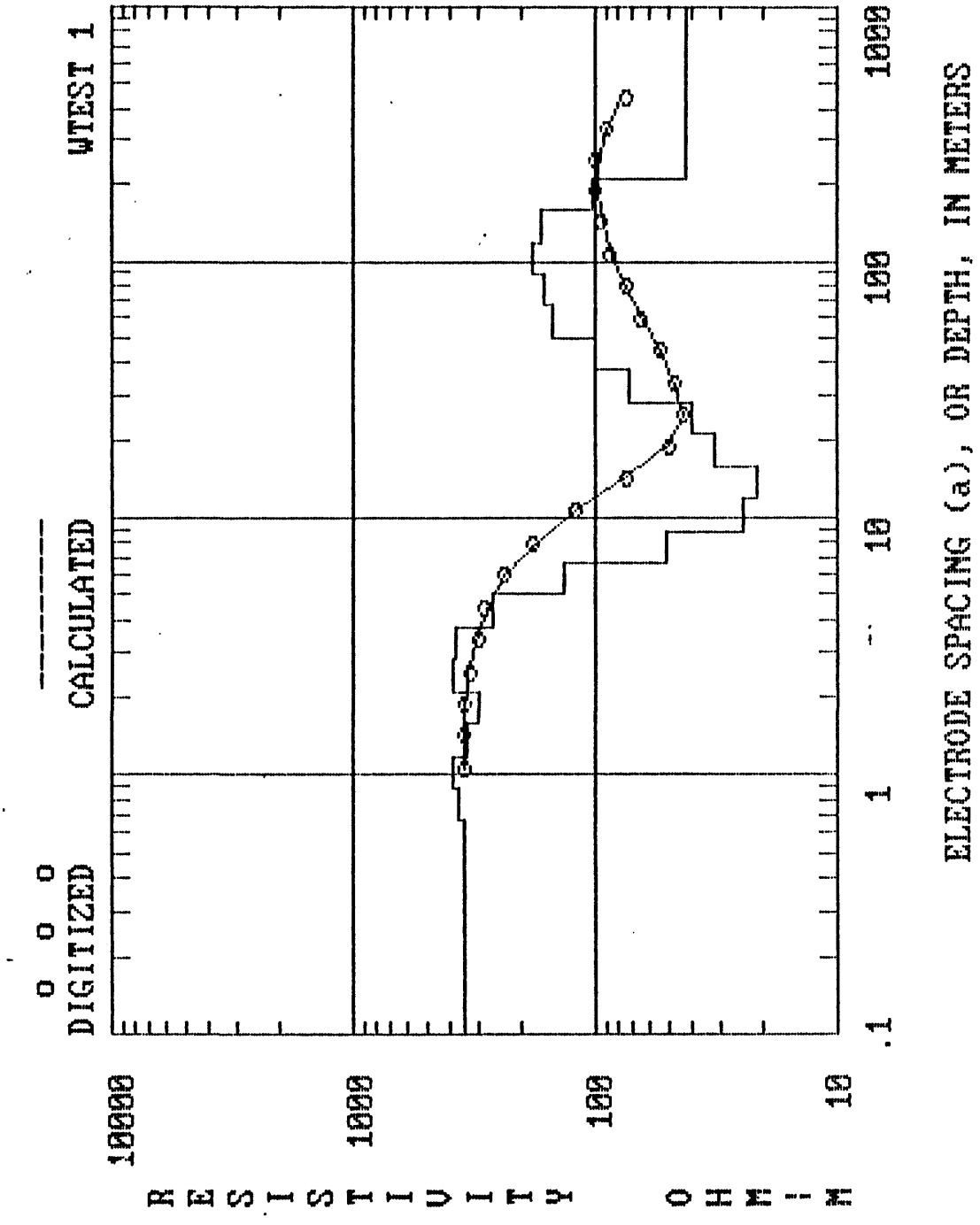

Figure 2. Example of the graphics output for an interpretation.

highlight EXIT to exit without making a selection. If there are a lot of names use Page Down to display more names.

If you know the sounding number you wish to load (for example sounding 1), then type 1 to load sounding WTEST 1. The next prompt is:

"LOAD Field or Digitized sounding ? (F/D)"

Press the appropriate letter.

When a sounding is loaded and graphically displayed you will be prompted:

" Is this the sounding you want ? (Y/N) "

If you press N, you will go back to sounding number. If you press Y, you will be prompted:

" CURVE LOOKS OK ? (Y/N/<Esc>) "

If you press N, you get a chance to edit the sounding, re-save it, print it, etc. From here on, the program follows the same steps and prompts as if you entered a new sounding.

PICK CONTOUR

The program PICKCONT.EXE is a separate program that reads layering files created by ATOWEN.EXE and automatically interpolates the depths at preselected resistivity values. The purpose is to facilitate the manual contouring of resistivity on geoelectric cross sections. The preselected resistivity contour values are: 1, 1.5, 2, 3, 4.5, 7, 10, 15, 20, 30, 45, 70, 100, 140, 200, 300, 450, 700, 1000, 1500, 2000, 3000, 4500, 7000. If a file named CONTVAL.DAT is found the program will attempt to load the resistivity contour values from that file. An example file containing the standard resistivity contour values is included on the diskette. By modifying this file the user can select the resistivity contour values to be used.

To run the PICKCONT.EXE program you must exit the program ATOWEN.EXE and then type PICKCONT. The program will prompt you to select the drive to retrieve data from, then it will display the directories of the various survey areas so that you may select the one you wish by highlighting it. The next screen displays the file names that end with the extension LYR (layering files). Make sure the printer is turned on, select the file you want, and press <ENTER>. A list of the depths and corresponding, preselected, resistivity values is printed.

Table 1 Example of program PICKCONT output.

WTEST 1

DEPTH IN FEET	CONTOUR VALUE
13.	300.0
16.	200.0
18.	150.0
21.	100.0
24.	70.0
28.	45.0
31.	30.0
57.	30.0
84.	45.0
105.	70.0
143.	100.0
195.	150.0
479.	150.0
602.	100.0
675.	70.0
777.	45.0

LAST DEPTH= 798.

Lists of depths and preselected resistivities for soundings along a given cross section are very useful in the manual construction of geoelectric cross sections of contoured interpreted true resistivity. Table 1 shows the results of running PICKCONT for sounding WTEST 1.

REFERENCES

Anderson, W.L., 1971, Application of bicubic spline functions to two dimensional gridded data: NTIS (Natl. Tech. Inf. Service), PB-203579, Springfield, VA.

Koefoed, Otto, 1979, Geosounding Principles, 1 resistivity sounding measurements: Elsevier Scientific Publishing Company, New York, 276 p.

Zohdy, A.A.R., 1989, A new method for the automatic interpretation of Schlumberger and Wenner sounding curves: Geophysics, v. 54, No. 2, p. 245-253

Zohdy, A.A.R., and Bisdorf, R.J., 1982, Schlumberger soundings in the Medicine Lake area, California: U.S. Geol. Survey Open-File Report, 82-887, 162 p.

Zohdy, A.A.R., and Bisdorf, R.J., 1989, Programs for the automatic processing and interpretation of Schlumberger sounding curves in QuickBASIC 4.0: U.S. Geol. Survey Open-File Report, 89-137 A&B, 64 p. + diskette.

APPENDIX A

****

Program ATOWEN.BAS

****

```
DECLARE FUNCTION CTOFT! (a!)
DECLARE SUB Wconvs (vv!(), ves!(), nrad%, nradii%, Xk!(), nc%)
DECLARE SUB CHKGRAPH (E!, GR$)
DECLARE SUB GRAPHSET (GR$)
DECLARE SUB PRTSCR ()
DECLARE SUB UPSORT (a$(), N%)
DECLARE SUB FILESLCT (nam$, fl$(), N%, k%)
DECLARE SUB DIRECT (a$(), N%, p$, k%)
DECLARE SUB LVES (a!(), B!(), N%, t$, U$, k%)
DECLARE SUB CLR5L ()
DECLARE SUB CLRLNE (i%)
DECLARE SUB PRSSANYKY ()
DECLARE FUNCTION PrssNmbr% (N%)
DECLARE SUB SETCOORD (AX!(), AY!(), X!(), y!(), N%)
DECLARE SUB KOEFOED4 (Xk!(), Nk%, Fshft!, Nr%)
DECLARE SUB setscrn ()
DECLARE SUB Logplt (AX!(), AY!())
DECLARE FUNCTION CAPNWT$ ()
DECLARE SUB cntr (i%, a$)
DECLARE SUB HIGHL (a$)
DECLARE FUNCTION algt! (X!)
DECLARE FUNCTION DROUND! (tx1!, ty1!)
DECLARE SUB Kernel (L%, X!(), t!(), R!(), V!(), N%)
DECLARE SUB conves (vv!(), ves!(), nrad%, Xk!(), Nk%)
DECLARE SUB Spline1 (M%, H!, X!(), y!(), a!(), B!(), C!(), t!, D!(), p!(), S!())
DECLARE SUB Cubic1 (M%, X!(), y!(), a!(), B!(), C!(), Delx!, xx!(), Yy!(), k%)
DECLARE SUB Pltbox (AX!(), AY!())
DECLARE SUB RMS1 (Rhodig!(), ves!(), RMS!, Layers%)
    DEFINT I-N
    CONST FALSE = 0, true = NOT FALSE, con1 = .3048, Xratio = 1.333521
    ON ERROR GOTO ChkError
    DIM ab2(45), psob(45), ABX(45), PSY(45), AX(5), AY(5), FILENAME$(300)
    DIM Abdig(45), Rhodig(45), RhoDig1(45), Rho(45), Thick(45), depth(45)
    DIM ves(45), Rhof(45), Thickf(45), Depthf(45), Vesf(45)
    DIM Xamda(65), vv(65), Xk(20), Pso(45), absh(45), obsh(45), a(45)
    DIM B(45), C(45), D(2), S(45), p(45), Ps(45), It(65), Ss(65)
    DIM F1$(300), F2$(150)
    KOEFOED4 Xk(), Nk, Fshft, Nr
    setscrn
    WIDTH 80
    LOCATE , , 0
    Es$ = CHR$(27)
    cntr 6, " WENNER SOUNDING DATA PROCESSING AND INTERPRETATION PROGRAM "
    cntr 10, " By "
    cntr 14, "   Adel  A.R.  Zohdy   "
    cntr 12, "   Robert J. Bisdorf   "
    cntr 16, "   U.S. GEOLOGICAL SURVEY, DENVER, CO.   "
    cntr 20, " 1989 "
    cntr 22, " Version 1.5 "
    PRSSANYKY
    CRVPLT = true
    CHKGRAPH E, GR$
    CRVPLT = FALSE
    setscrn
    IF GR$ = "N" THEN
        CLS
```

```
        BEEP
        cntr 10, " SORRY, PROPER GRAPHICS INTERFACE NOT AVAILABLE ... "
        cntr 12, " CAN'T USE THIS PROGRAM !. "
        PRSSANYKY
        CLS
        SYSTEM
    END IF
    CLS
    cntr 10, " PLEASE WAIT ..... "
    a$ = " LOADING " + GR$ + "GA GRAPHICS PRINTER DRIVER. "
    cntr 12, a$
    IF GR$ = "E" THEN SHELL "EGADUMPL"
    IF GR$ = "C" THEN SHELL "GRAPHICS"

DrvMnu:
    setscrn
    cntr 23, " * If you DO NOT wish to save or retrieve press <Esc>. "
    cntr 7, " DISK-DRIVE MENU "
    LOCATE 10, 20: PRINT " Please type letter of DISK DRIVE "
    LOCATE 12, 21
    HIGHL " (A, B, C, etc.) "
    LOCATE 14, 20, 1
    PRINT " to SAVE or RETRIEVE DATA from: ";
    a$ = CAPNWT$
    IF a$ = CHR$(13) THEN BEEP: GOTO DrvMnu
    IF a$ = Es$ THEN DD$ = "": GOTO MainMnu
    DD$ = a$
    LOCATE 14, 52: PRINT DD$; : DD$ = DD$ + ":"
    LOCATE 16, 20: PRINT " Please wait ...          ";
    chkdrive$ = "CHKDRIVE"
    OPEN DD$ + chkdrive$ FOR INPUT AS #1
    CLOSE #1
    chkdrive$ = ""

MainMnu:
    setscrn
    LodF = 0
    Edt = 0
    Mnu$ = "ENTRY"
    ABUnit$ = ""
    LOCATE , , 0
    cntr 7, " MAIN MENU "
    LOCATE 10, 26
    PRINT "1. "; : HIGHL " ENTER "
    PRINT " SOUNDING FROM KEYBOARD."
    LOCATE 12, 26
    PRINT "2. "; : HIGHL " LOAD "
    PRINT " SOUNDING FROM DATA DISK."
    LOCATE 14, 26
    PRINT "3. "; : HIGHL " RETURN "
    PRINT " TO DISK-DRIVE MENU."
    LOCATE 16, 26
    PRINT "4. "; : HIGHL " QUIT. "
    k = PrssNmbr(4)
ON k GOTO SrvyArea, LodFile, DrvMnu, Kwit

SrvyArea:
    IF DD$ = "" THEN GOTO ABUnit
    Dir$ = ""
    CLS
    file = FALSE
    LOCATE 10, 16: PRINT "Type name of "; : HIGHL " SURVEY AREA "
    PRINT " <ENTER>: "
    LOCATE 13, 16: PRINT "* If NO NAME press <ENTER>."
    LOCATE 14, 16: PRINT "* You may use SPACES in name of survey area."
    LOCATE 15, 16: PRINT "* Automatic Assignments:"
    LOCATE 16, 16: PRINT "  Sounding Title = Survey Area + Sounding Number."
    LOCATE 17, 16: PRINT "  Directory name = First 8 Letters of Survey Name."
    LOCATE 10, 52
```

```
     LINE INPUT "", SurveyNm$
     IF SurveyNm$ = "" THEN GOTO ABUnit
     SurveyNm$ = UCASE$(SurveyNm$)
     a$ = SurveyNm$
     GOSUB CapNTrim
     IF a$ = "" THEN BEEP: GOTO SrvyArea

chkdir:
     Dir$ = LEFT$(a$, 8)
     LOCATE 10, 52: PRINT Dir$; "                    "
     LOCATE 22, 16: PRINT "Please wait, checking directory ..."
     IF LEN(Dir$) = 4 AND Dir$ = "FILE" THEN
          BEEP
          LOCATE 22, 16
          PRINT "Can't use the name FILE for Survey area !!         "
          PRSSANYKY
          GOTO SrvyArea
     END IF
     chkdir$ = "dir"
     CHDIR DD$ + "\" + Dir$
     IF chkdir$ <> "" THEN GOTO ABUnit
     BEEP
     LOCATE 22, 16
     PRINT "Above directory not found, Wish to create a new one ? (Y/N)"
     GOSUB YsNo
     IF a$ = "Y" THEN
          MKDIR DD$ + "\" + Dir$
          GOTO ABUnit
     END IF
     GOTO SrvyArea

ABUnit:
     CLS
     ABunits$ = ""
     cntr 12, " Electrode spacings: in Feet or Meters ? (F/M) "
     a$ = CAPNWT$
     IF a$ <> "F" AND a$ <> "M" THEN BEEP: GOTO ABUnit
     PRINT a$
     IF a$ = "F" THEN ABUnit$ = "FEET"
     IF a$ = "M" THEN ABUnit$ = "METERS"

VesNmbr:
     setscrn
     IF convert$ = "Y" THEN
          convert$ = "N"
          ABUnit$ = "FEET"
          FOR i = 1 TO Nab2
               ab2(i) = ab2(i) / con1
          NEXT i
     END IF
     cntr 8, " ENTER SOUNDING ... "
     LOCATE 12, 16: PRINT " Type Sounding ";
     HIGHL " NUMBER ": PRINT " <ENTER>: "
     LOCATE 14, 16: PRINT " (4 Digits and/or Characters max)"
     LOCATE 16, 16: PRINT " * If no number type 0 <ENTER>. "
     LOCATE 18, 16: PRINT " * To return to MAIN MENU press <ENTER>."
     LOCATE 12, 48: LINE INPUT " ", VesNum$
     IF VesNum$ = "" THEN GOTO MainMnu
     IF VesNum$ = "0" THEN VesNum$ = "": GOTO DataEntryMnu
     a$ = VesNum$
     GOSUB CapNTrim
     VesNum$ = a$
     LOCATE 14, 23
     PRINT VesNum$; "             "
     TITLE$ = SurveyNm$ + " " + VesNum$

DataEntryMnu:
     setscrn
     cntr 4, " DATA ENTRY MENU "
```

```
    VesTyp$ = ""
    LOCATE 11, 15
    PRINT "1. Enter (a) and "; : HIGHL " FIELD "
    PRINT " APPARENT RESISTIVITIES."
    LOCATE 13, 15
    PRINT "2. Enter SMALLEST (a) and "; : HIGHL " DIGITIZED "
    PRINT " APPARENT RESISTIVITIES."
    LOCATE 15, 15
    PRINT "3. Return to "; : HIGHL " MAIN MENU. "
    k = PrssNmbr(3)
    IF k = 3 THEN GOTO MainMnu
    IF k = 1 THEN
        VesTyp$ = "FLD"
        IF Dir$ = "" THEN GOTO InptFld
    ELSE
        VesTyp$ = "DGT"
        IF Dir$ = "" THEN GOTO InptDgt
    END IF
    FILENAME$ = LEFT$(Dir$, 4) + LEFT$(VesNum$, 4) + "." + VesTyp$
    LOCATE 23, 16: PRINT "Please wait ..."
    CLOSE
    ChkFil = 1
    OPEN DD$ + "\" + Dir$ + "\" + FILENAME$ FOR INPUT AS #1
    CLOSE #1
    IF ChkFil THEN
        BEEP
        CLS
        LOCATE 10, 12
        PRINT "File for: "
        LOCATE 12, 12
        HIGHL " " + Dir$ + " " + VesNum$ + " (" + VesTyp$ + ") already Exists! "
        LOCATE 14, 12
        PRINT "Do you wish to replace it ? (Y/N/<Esc>)"
        GOSUB YsNo
        IF a$ = Es$ THEN GOTO MainMnu
        IF a$ = "N" THEN GOTO VesNmbr
    END IF
    IF VesTyp$ = "DGT" THEN GOTO InptDgt

InptFld:
    COLOR 7, 1: CLS
    LOCATE 21, 22: COLOR 0, 7
    PRINT " ENTER ALL DATA NOW ..... EDIT LATER! "
    LOCATE 22, 22
    PRINT " Type (a) values   &   press   <ENTER>. "
    LOCATE 23, 22
    PRINT " When done, type  0 &   press   <ENTER>. "
    LOCATE 25, 22
    PRINT " TO  RETURN  TO  MENU  TYPE  -1 <ENTER>. "; : COLOR 7, 1
    xx = 5: y = 4: X = 6: i = 1

AB2Hdr1:
    LOCATE 2, xx - 2
    PRINT "#"
    LOCATE 2, xx
    HIGHL " (a) "

AB2Nmbr:
    LOCATE y, X - 4: PRINT i
    LOCATE y, X: INPUT "", ab$
    IF ab$ = "" AND ab2(i) <> 0 THEN ab$ = STR$(ab2(i))
    LOCATE y, X: HIGHL ab$
    ab2(i) = VAL(ab$)
    IF ab2(i) < 0 THEN ab2(i) = 0: GOTO DataEntryMnu
    IF ab$ = "0" AND i = 1 THEN BEEP: GOTO AB2Nmbr
    IF ab$ = "0" THEN GOTO DonAB
    IF ab$ <> "" AND ab2(i) <> 0 AND ab2(i) <> -1 THEN
        LOCATE y, X
        PRINT "      "
```

```
        LOCATE y, X
        HIGHL STR$(ab2(i))
    END IF
    IF ab2(i) = 0 THEN
        BEEP
        LOCATE y, X
        PRINT "     "
        GOTO AB2Nmbr
    END IF
    i = i + 1
    IF y = 18 THEN
        y = 4
        xx = xx + 24
        X = X + 24
    ELSE
        y = y + 1
    END IF
    GOTO AB2Nmbr

DonAB:
    CLR5L
    Nab2 = i - 1
    xx = 15
    y = 4
    X = 16
    i = 1

RoFldPrmpt:
    LOCATE 21, 22: COLOR 0, 7
    PRINT " ENTER ALL DATA NOW ........ EDIT LATER! "
    LOCATE 22, 22
    PRINT " Type  APP. RESISTIVITY & press <ENTER>. "
    LOCATE 23, 22
    PRINT " When  done  type   0   & press <ENTER>. "
    COLOR 7, 1
25 LOCATE 2, xx - 2
    PRINT "# "
    LOCATE 2, xx
    HIGHL " APP.RHO "

InptRo:
    LOCATE y, X - 4: PRINT i
    LOCATE y, X: INPUT "", Ps$
    IF Ps$ = "" AND psob(i) <> 0 THEN Ps$ = STR$(psob(i))
    LOCATE y, X: HIGHL Ps$
    psob(i) = VAL(Ps$)
    IF psob(i) < 0 THEN
        psob(i) = 0
        GOTO DataEntryMnu
    END IF
    IF Ps$ = "0" AND i = 1 THEN
        BEEP
        GOTO InptRo
    END IF
    IF Ps$ = "0" THEN
        NAPR = i - 1
        GOTO ChkNmbr
    END IF
    IF Ps$ <> "" AND psob(i) <> 0 AND psob(i) <> -1 THEN
        LOCATE y, X
        PRINT "     "
        LOCATE y, X
        HIGHL STR$(psob(i))
    END IF
    IF psob(i) = 0 THEN BEEP: GOTO InptRo
    i = i + 1
    IF y = 18 THEN
        y = 4
        xx = xx + 24
```

```
        X = X + 24
        GOTO 25
    END IF
    y = y + 1
    GOTO InptRo

ChkNmbr:
     IF Nab2 <> NAPR THEN GOSUB NtEql
     GOSUB Dsply
     GOTO EdtPrmpt

InptDgt:
    VesTyp$ = "DGT"
    CLS
    LOCATE 12, 12
    PRINT " PLEASE TYPE SMALLEST DIGITIZED (a) <ENTER> "
    LOCATE 16, 20
    HIGHL " TO RETURN TO MENU PRESS <ENTER> "

ABMIN:
    LOCATE 12, 60
    INPUT "", ABMIN$
    IF ABMIN$ = "" THEN GOTO DataEntryMnu
    IF VAL(ABMIN$) = 0 THEN BEEP: GOTO ABMIN
    Abdig(1) = VAL(ABMIN$)
    COLOR 7, 4: CLS
    LOCATE 21, 16: COLOR 0, 7
    PRINT " ENTER ALL DATA NOW ..... EDIT LATER! "
    LOCATE 22, 16
    PRINT " INPUT DIGITIZED APP. RESISTIVITIES & PRESS <ENTER> "
    LOCATE 23, 16
    PRINT " WHEN  DONE,  TYPE  0  AND  PRESS ..... <ENTER> "
    LOCATE 25, 16
    PRINT " TO  RETURN  TO  MENU  TYPE  -1  AND PRESS  <ENTER> ";
    COLOR 7, 4
    xx = 5: y = 4: X = 6: i = 1

DgtHdr:
    LOCATE 2, xx - 2: PRINT "#"
    LOCATE 2, xx
    COLOR 0, 7: PRINT " (a) ": COLOR 7, 4
    GOSUB PrtAbDgt
    xx = xx + 10: y = 4: X = X + 10
    LOCATE 2, xx - 2: PRINT "# "
    LOCATE 2, xx
    COLOR 0, 7: PRINT " APP.RHO ": COLOR 7, 4

InptRoDgt:
    LOCATE y, X - 4: PRINT i
    LOCATE y, X: INPUT "", Rhodig$
    IF Rhodig$ = "" AND Rhodig(i) > 0 THEN Rhodig$ = STR$(Rhodig(i))
    LOCATE y, X
    COLOR 0, 7: PRINT Rhodig$: COLOR 7, 4
    Rhodig(i) = VAL(Rhodig$)
    IF Rhodig(i) < 0 THEN Rhodig(i) = 0: GOTO DataEntryMnu
    IF Rhodig$ = "0" AND i = 1 THEN BEEP: GOTO InptRoDgt
    IF Rhodig$ = "0" THEN NAPR = i - 1: GOTO DnRhoDig
    IF Rhodig$ <> "" AND Rhodig(i) <> 0 AND Rhodig(i) <> -1 THEN
        LOCATE y, X
        PRINT "      "
        LOCATE y, X
        COLOR 0, 7: PRINT Rhodig(i): COLOR 7, 4
    END IF
    IF Rhodig(i) = 0 THEN BEEP: GOTO InptRoDgt
    y = y + 1
    i = i + 1
    Abdig(i) = Abdig(i - 1) * Xratio
    IF y = 19 THEN
        y = 4
```

```
        IF i = 16 THEN
            xx = 30
            X = 30
        ELSEIF i = 31 THEN
            xx = 60
            X = 60
        END IF
        GOTO DgtHdr
    ELSEIF y < 19 AND i < 16 THEN
        xx = 5
        X = 6
    ELSEIF y < 19 AND i > 16 AND i < 31 THEN
        xx = 30
        X = 30
    ELSEIF y < 19 AND i > 31 THEN
        xx = 60
        X = 60
    END IF
        GOSUB PrtAbDgt
        xx = xx + 10
        X = X + 10
        GOTO InptRoDgt

DnRhoDig:
    Nab2 = NAPR
    FOR i = 1 TO NAPR
        ab2(i) = Abdig(i)
        psob(i) = Rhodig(i)
    NEXT i
    GOSUB Dsply

EdtPrmpt:
    Edt = -1
     CLR5L
     LOCATE 21, 11
     PRINT "Press: "; : HIGHL " C ": PRINT " to Continue, ";
     HIGHL " E ": PRINT " to Edit,   ";
     HIGHL " D ": PRINT " to Delete, "
     LOCATE 23, 18
     HIGHL " I ": PRINT " to Insert,   ";
     HIGHL " A ": PRINT " to Append, ";
     HIGHL " <Esc> ": PRINT " to Main Menu. "

WhrTo:
    a$ = CAPNWT$
    IF a$ = "E" THEN B$ = " EDIT ": GOTO EdtDel
    IF a$ = "D" THEN B$ = " DELETE ": GOTO EdtDel
    IF a$ = "I" THEN B$ = " INSERT ": GOTO Insrt
    IF a$ = "A" THEN GOTO Apnd
    IF a$ = Es$ THEN GOTO MainMnu
    IF a$ = "C" THEN
        IF Nab2 <> NAPR THEN
            GOSUB NtEql
            GOTO EdtPrmpt
        ELSEIF VesTyp$ = "DGT" THEN
            FOR i = 1 TO Nab2
                Abdig(i) = ab2(i)
                Rhodig(i) = psob(i)
            NEXT i
        END IF
        IF Nab2 = NAPR THEN GOTO Plt
    END IF
    BEEP
    GOTO WhrTo

EdtDel:
    C$ = " TYPE ENTRY NUMBER TO ": a$ = C$ + B$
    CLR5L
    LOCATE 21, 18
```

```
     COLOR 3, 1: PRINT C$;
     HIGHL B$
     COLOR 3, 1: PRINT " <ENTER> ": COLOR 7, 1
     LOCATE 22, 18
     COLOR 7, 1: PRINT " If NO CHANGE ........ press <ENTER>"
     LOCATE 21, 18 + LEN(a$) + 12
     COLOR 3, 1: INPUT "", N$: COLOR 7, 1
     IF N$ = "" THEN
          GOSUB Dsply
          GOTO EdtPrmpt
     END IF
     IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO EdtDel
     N = VAL(N$)
     IF N = 0 THEN
          BEEP
          GOTO EdtDel
     END IF
     IF N > Q THEN
          GOSUB WrngNmbr
          GOSUB Dsply
          GOTO EdtDel
     END IF
     IF B$ <> " EDIT " THEN GOTO Del
     IF N > Nab2 THEN 1440
     y = 3 + N
     IF N <= 15 THEN LOCATE y, 7
     IF N > 15 AND N <= 30 THEN LOCATE y - 15, 33
     IF N > 30 AND N <= 45 THEN LOCATE y - 30, 59
     AK = ab2(N)
     COLOR 0, 7: INPUT "", ab$
     IF ab$ = "" THEN ab2(N) = AK ELSE ab2(N) = VAL(ab$)
     IF ab2(N) <= 0 THEN BEEP: ab2(N) = AK
     GOSUB Dsply
     GOTO EdtDel

1440    y = 3 + N - Nab2
     IF N <= 15 + Nab2 THEN LOCATE y, 19: GOTO 1480
     IF N <= 30 + Nab2 THEN LOCATE y - 15, 44: GOTO 1480
     IF N <= 45 + Nab2 THEN LOCATE y - 30, 69: GOTO 1480

1480    AKK = psob(N - Nab2)
     COLOR 0, 7
     INPUT "", Ps$
     IF Ps$ = "" THEN psob(N - Nab2) = AKK: GOTO 1500
     psob(N - Nab2) = VAL(Ps$)
     IF psob(N - Nab2) <= 0 THEN BEEP: psob(N - Nab2) = AKK

1500    GOSUB Dsply
     GOTO EdtDel

Del:
     IF B$ <> " DELETE " THEN GOTO Insrt
     IF N > Nab2 THEN GOTO DelRo

DelAB2:
     FOR i = N TO Nab2
          ab2(i) = ab2(i + 1)
     NEXT i
     Nab2 = Nab2 - 1
     GOSUB Dsply
     GOTO EdtDel

DelRo:
     IF N > Q THEN
          GOSUB WrngNmbr
          GOSUB Dsply
     END IF
     FOR i = N TO Q
          psob(i - Nab2) = psob(i - Nab2 + 1)
```

```
    NEXT i
    NAPR = NAPR - 1
    GOSUB Dsply
    GOTO EdtDel

Insrt:
    CLR5L
    LOCATE 21, 6
        COLOR 3, 1: PRINT "Type entry number "; : COLOR 1, 3
        PRINT " ABOVE WHICH ";
        COLOR 3, 1: PRINT " you wish to ";
        COLOR 0, 7: PRINT B$; : COLOR 3, 1
        PRINT " & press <ENTER>  "
    LOCATE 22, 22
        COLOR 7, 1: PRINT "IF NO CHANGE ..... press <ENTER>  "
    LOCATE 21, 75
        INPUT "", N$
    N = VAL(N$)
    IF N$ = "" THEN GOTO EdtPrmpt
    IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO Insrt
    IF N = 0 THEN BEEP: GOTO Insrt
    IF N > Nab2 THEN GOTO InsrtRo

InsrtAB:
    LOCATE 23, 1: PRINT STRING$(79, " ")
    LOCATE 23, 21
    HIGHL " TYPE VALUE OF (a) TO BE INSERTED "
    PRINT "    ";
    INPUT "", N$
    IF N$ = "" THEN GOTO EdtPrmpt
    IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO InsrtAB
    AM = VAL(N$)
    IF AM = 0 THEN BEEP: GOTO InsrtAB
    Nab2 = Nab2 + 1
    FOR i = Nab2 TO N + 1 STEP -1
        ab2(i) = ab2(i - 1)
    NEXT i
    ab2(N) = AM
    GOSUB Dsply
    GOTO Insrt

InsrtRo:
    IF N > Q THEN
        GOSUB WrngNmbr
        GOSUB Dsply
        GOTO Insrt
    END IF
    LOCATE 23, 1: PRINT STRING$(79, " ")
    LOCATE 23, 22
        HIGHL " TYPE VALUE OF APP.RHO TO BE INSERTED "
    PRINT " ";
    INPUT "", N$
    IF N$ = "" THEN GOTO EdtPrmpt
    IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO InsrtRo
    AM = VAL(N$)
    IF AM = 0 THEN BEEP: GOTO InsrtRo
    NAPR = NAPR + 1: Q = Q + 1
    FOR i = Q TO N + 1 STEP -1
        psob(i - Nab2) = psob(i - Nab2 - 1)
    NEXT i
    psob(N - Nab2) = AM
    GOSUB Dsply
    GOTO Insrt

WrngNmbr:
    BEEP
    CLR5L
    cntr 24, " THERE IS NO SUCH NUMBER!. "
    PRSSANYKY
```

```
    CLR5L
    RETURN

Apnd:
    CLR5L
    LOCATE 21, 8
    PRINT " Press ";
    HIGHL "  A  ": PRINT " to append ";
    HIGHL " (a) ": PRINT " , or press ";
    HIGHL "  R  ": PRINT " to append ";
    HIGHL " APP.RHO "
    LOCATE 23, 22
    PRINT " If NO CHANGE  ......  Press <ENTER> "
    a$ = CAPNWT$
    IF a$ = CHR$(13) THEN GOTO EdtPrmpt
    IF a$ = "A" THEN GOTO ApndAB
    IF a$ = "R" THEN GOTO ApndRo
    BEEP: GOTO Apnd

ApndAB:
    CLR5L
    LOCATE 23, 22
    PRINT " If NO CHANGE  ......  Press <ENTER> "
    LOCATE 22, 18
    HIGHL "Type  VALUE OF (a)  to be added <ENTER>"
    INPUT "  ", N$
    IF N$ = "" THEN GOTO EdtPrmpt
    IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO ApndAB
    AN = VAL(N$)
    IF AN <= 0 THEN BEEP: GOTO ApndAB
    Nab2 = Nab2 + 1
    ab2(Nab2) = AN
    GOSUB Dsply
    GOTO Apnd

ApndRo:
    CLR5L
    LOCATE 23, 22
    PRINT " If NO CHANGE  ......  Press <ENTER> "
    LOCATE 22, 18
    HIGHL " Type  VALUE OF APP.RHO  to be added <ENTER>"
    INPUT "  ", N$
    IF N$ = "" THEN GOTO EdtPrmpt
    IF ASC(N$) < 49 OR ASC(N$) > 57 THEN BEEP: GOTO ApndRo
    AN = VAL(N$)
    NAPR = NAPR + 1
    psob(NAPR) = AN
    GOSUB Dsply
    GOTO Apnd

Plt:
    GOSUB FldPlt

CrvOk:
    CLRLNE 1
    LOCATE 1, 24
        PRINT " CURVE LOOKS OK ? (Y/N/<Esc>) "
    GOSUB YsNo
          IF a$ = Es$ THEN GOTO MainMnu
    IF a$ = "N" THEN
        setscrn
        GOSUB Dsply
        GOTO EdtPrmpt
    END IF
    IF DD$ = "" OR Dir$ = "" THEN GOTO FtToM
    CLRLNE 1
    IF Edt = -1 THEN
        LOCATE 1, 20
        PRINT "Save this Sounding ? (Y/N/<Esc>)"
```

```
        GOSUB YsNo
        IF a$ = Es$ THEN GOTO MainMnu
        IF a$ = "N" THEN GOTO FtToM
        GOSUB SaveIt
    END IF
    IF LodF <> -1 THEN
        CLRLNE 1
        LOCATE 1, 15
        PRINT "ENTER another sounding ? (Y/N/<Esc>)"
        GOSUB YsNo
        IF a$ = Es$ THEN GOTO MainMnu
        IF a$ = "Y" THEN GOTO VesNmbr ELSE GOTO FtToM
    END IF

FtToM:
    convert$ = ""
    CLRLNE 1
    LOCATE 1, 15
    IF ABUnit$ = "METERS" THEN
        GOSUB Lbl2
        GOTO DmpGrfx
    END IF
    IF ABUnit$ = "FEET" THEN
        CLRLNE 1
        LOCATE 1, 15
        PRINT " CONVERT (a) from FEET to METERS ? (Y/N/<Esc>)"
        GOSUB YsNo
        IF a$ = Es$ THEN GOTO MainMnu
        IF a$ = "N" THEN
            convert$ = "N"
            GOSUB FldPlt
            GOTO DmpGrfx
        END IF
        convert$ = "Y"
        ABUnit$ = "METERS"
        IF VesTyp$ = "DGT" THEN
            FOR i = 1 TO Nab2
                Abdig(i) = Abdig(i) * con1
                ab2(i) = Abdig(i)
            NEXT i
        ELSE
            FOR i = 1 TO Nab2
                ab2(i) = ab2(i) * con1
            NEXT i
        END IF
        GOSUB FldPlt
        GOTO DmpGrfx
    END IF
    BEEP
    GOTO FtToM

DmpGrfx:
    CLRLNE 1
    LOCATE 1, 10
    PRINT "LIST field data & DUMP GRAPHICS on printer ? (Y/N/<Esc>) "
    GOSUB YsNo
    IF a$ = Es$ THEN GOTO MainMnu
    IF a$ = "Y" THEN
        GOSUB PrntrRdy
        nspc = 14
        IF convert$ = "Y" THEN nspc = 4
        IF VesTyp$ = "FLD" THEN
            LPRINT : LPRINT SPC(nspc); TITLE$; "  (FIELD DATA)": LPRINT
        ELSE
            LPRINT : LPRINT SPC(nspc); TITLE$; "  (DIGITIZED DATA)": LPRINT
        END IF
        XNu = INT(Nab2 / 2)
        XNm = Nab2 / 2
        IF convert$ = "N" OR convert$ = "" THEN
```

```
            form1$ = "            #####.##      #####.##     #####.##    #####.##"
            form2$ = "                                       #####.##    #####.##"
            form$ = "               (a)         App. Res.      (a)       App. Res."
            LPRINT form$
            LPRINT
            FOR i = 1 TO XNu
                LPRINT USING form1$; ab2(i); psob(i); ab2(i + XNu); psob(i + XNu)
            NEXT i
            IF XNm > XNu THEN LPRINT USING form2$; ab2(Nab2); psob(Nab2)
        ELSE
            form3$ = "   #####.## (#####.##)   #####.##     #####.## (#####.##)   #####.##"
            form4$ = "                                      #####.## (#####.##)   #####.##"
            form$ = "     A, m   (   ft   )   App. Res.      A, m   (   ft   )   App. Res."
            LPRINT form$: LPRINT
            FOR i = 1 TO XNu
                LPRINT USING form3$; ab2(i); ab2(i) / con1; psob(i); ab2(i + XNu); ab2(i + XNu) / con1; psob(i
+ XNu)
            NEXT i
            IF XNm > XNu THEN LPRINT USING form4$; ab2(Nab2); ab2(Nab2) / con1; psob(Nab2)
        END IF
        GOSUB DmpGrfxMsg
    END IF
    IF VesTyp$ = "FLD" THEN GOSUB ShftNDig
    GOTO Ato

LodFile:
    setscrn
    IF DD$ = "" THEN
        BEEP
        CLS
        LOCATE 10, 10
        PRINT " You did not select a Drive to load data from !!!"
        PRSSANYKY
        GOTO DrvMnu
    END IF
    file = FALSE
    cntr 8, " LOAD FILE ... "
    LOCATE 12, 16: PRINT "Type name of ";
    HIGHL " SURVEY AREA ": PRINT " <ENTER>: "
    LOCATE 15, 16: PRINT "* You may use SPACES in name of survey area."
    LOCATE 17, 16: PRINT "* To return to MAIN MENU press <ENTER>"
    LOCATE 12, 52
    LINE INPUT "", SurveyNm$
    IF SurveyNm$ = "" THEN GOTO MainMnu
    SurveyNm$ = UCASE$(SurveyNm$)
    a$ = SurveyNm$
    GOSUB CapNTrim
    IF a$ = "" THEN BEEP: GOTO LodFile
    Dir$ = LEFT$(a$, 8)
    LOCATE 12, 52: PRINT Dir$; "                    "
    LOCATE 20, 16: PRINT "Please wait, checking directory ..."
    IF LEN(Dir$) = 4 AND Dir$ = "FILE" THEN
        BEEP
        LOCATE 20, 16
        PRINT "Can't use the name FILE for Survey area !!         "
        PRSSANYKY
        GOTO LodFile
    END IF
    chkdir$ = "LOD"
    CHDIR "\"
    MKDIR DD$ + "\" + Dir$
    IF chkdir$ = "LOD" THEN
        BEEP
        LOCATE 20, 16
        PRINT "Above directory not found !!                    "
        RMDIR DD$ + Dir$
        PRSSANYKY
        GOTO LodFile
    END IF
```

```
LodVes:
     setscrn
     IF DD$ = "" OR Dir$ = "" THEN
          BEEP
          CLS
          LOCATE 10, 10
          PRINT " Can't LOAD, you did not select a Drive and/or a Directory !!"
          PRSSANYKY
          GOTO MainMnu
     END IF

Lod1:
     cntr 8, " LOAD FILE ..... "
     LOCATE 15, 16: PRINT "* To see file names type FILE."
     LOCATE 16, 16: PRINT "* To return to MAIN MENU press <ENTER>."
     LOCATE 12, 16: PRINT " Type Sounding ";
     HIGHL " NUMBER ": PRINT " <ENTER>: ";
     LINE INPUT a$
     IF a$ = "" THEN GOTO MainMnu
     GOSUB CapNTrim
     VesNum$ = a$
     IF LEFT$(VesNum$, 4) = "FILE" THEN
          CLS
          IF file THEN GOTO file1
          cntr 12, " LOADING DISK DIRECTORY: .... PLEASE WAIT. "
          nf1 = 1
          F$ = DD$ + "\" + Dir$ + "\*.FLD"
          DIRECT F1$(), N1, F$, 1
          UPSORT F1$(), N1
          F$ = DD$ + "\" + Dir$ + "\*.DGT"
          DIRECT F2$(), N2, F$, 1
          UPSORT F2$(), N2
          FOR i = N1 + 1 TO N1 + N2
               F1$(i) = F2$(i - N1)
          NEXT i
          NF = N1 + N2 + 1
          F1$(NF) = "EXIT"
          file = true

file1:
     CLS
     cntr 1, " SELECT A FILE OR SELECT EXIT TO RETURN "
     0 < 5 THEN GOTO Replough
     RMSmin = RMS
     IF RMSmin < Ft OR Iterate > Max THEN GOTO Graphics
     Iterate = Iterate + 1
     GOTO FlipFlop          ' FLIP-FLOP SOME MORE

Replough:
2100
     IF Replough = 1 THEN GOTO Graphics
     CLS
     cntr 3, " INTERPRETING SMOOTHED CURVE ..... "
     LOCATE 5, 1
     FOR i = 1 TO Layers
          Rhodig(i) = Vesf(i)
     NEXT i
     Replough = 1
     Ft = 1
     GOTO Init

'-------------------------------------------------------------------------
LCompress:
2200
     depth(1) = Abdig(1) * Shift
     Thick(1) = depth(1)
     FOR i = 2 TO Layers
          depth(i) = Abdig(i) * Shift
          depth(i) = depth(i - 1) * Squish
```

```
        Thick(i) = depth(i) - depth(i - 1)
    NEXT i
    depth(Layers) = 99999
RETURN

'-----------------------------------------------------------------------
LVES:
2300
    IF    COMPUTE KERNEL FUNCTION SPACINGS (XAMDA)
KrnlSpcing:
1700
    Xamda(1) = Fshft * Radmin / Xratio ^ (2 * Nr)
    FOR i = 2 TO nrad
        Xamda(i) = Xratio * Xamda(i - 1)
    NEXT i

Options:
    Compress = 8          ' NUMBER OF LAYERS PER CYCLE
    Shift = 0             ' DEPTH-SHIFT factor
    Iterate = 1           ' NUMBER OF ITERATIONS
    Max = 60              ' MAX NUMBER OF ITERATIONS
    FixRhoN = 0
    CLS
    cntr 6, " OPTIONS "
    LOCATE 10, 10
    PRINT "Do You wish to use any of the following options ? (Y/N)"
    LOCATE 12, 10
    PRINT "* Fix DEPTH-SHIFT factor at (...,0.7, 0.8, ...)  ? "
    LOCATE 14, 10
    PRINT "* Change Number of Layers/Cycle from 8 to (7.5, 8.5, etc.)"
    LOCATE 16, 10
    PRINT "* Fix Last Layer Resistivity at .......? "
    GOSUB YsNo
    IF a$ = Es$ THEN GOTO MainMnu
    IF a$ = "Y" THEN
        LOCATE 20, 10
        COLOR 7,    COMPUTE KERNEL FUNCTION SPACINGS (XAMDA)
KrnlSpcing:
1700
    Xamda(1) = Fshft * Radmin / Xratio ^ (2 * Nr)
    FOR i = 2 TO nrad
        Xamda(i) = Xratio * Xamda(i - 1)
    NEXT i

Options:
    Compress = 8          ' NUMBER OF LAYERS PER CYCLE
    Shift = 0             ' DEPTH-SHIFT factor
    Iterate = 1           ' NUMBER OF ITERATIONS
    Max = 60              ' MAX NUMBER OF ITERATIONS
    FixRhoN = 0
    CLS
    cntr 6, " OPTIONS "
    LOCATE 10, 10
    PRINT "Do You wish to use any of the following options ? (Y/N)"
    LOCATE 12, 10
    PRINT "* Fix DEPTH-SHIFT factor at (...,0.7, 0.8, ...)  ? "
    LOCATE 14, 10
    PRINT "* Change Number of Layers/Cycle from 8 to (7.5, 8.5, etc.)"
    LOCATE 16, 10
    PRINT "* Fix Last Layer Resistivity at .......? "
    GOSUB YsNo
    IF a$ = Es$ THEN GOTO MainMnu
    IF a$ = "Y" THEN
        LOCATE 20, 10
        COLOR 7, 0
        PRINT " IF NO CHANGE press <ENTER>, ELSE type REQUIRED VALUE. "
        COLOR 7, 1
        LOCATE 12, 67
        LINE INPUT a$
```

```
        IF a$ <> "" THEN Shift = VAL(a$)
        LOCATE 14, 72
        LINE INPUT a$
        IF a$ <> "" THEN Compress = VAL(a$)
        LOCATE 16, 60
        LINE INPUT a$
        IF a$ <> "" THEN FixRhoN = VAL(a$)
    END IF

DoAto:
1800
    CLS
    cntr 3, " AUTOMATIC INTERPRETATION WORKING "
    LOCATE 5, 1

Init:
    Squish = 10 ^ (1 / Compress) ' MULTIPLIER FOR GENERATING COMPRESSED DEPTHS
    RMSmin = 99999999   ' RMSmin= SMALLEST RMS PERCENT
    FOR j = 1 TO Layers
        Rho(j) = Rhodig(j)
    NEXT j
    IF Shift = 0 THEN
        Shift = .7
        FOR j = 1 TO 10
            GOSUB LCompress
            GOSUB LVES
            IF RMS < RMSmin THEN
                ShiftMin = Shift
                RMSmin = RMS
                PRINT USING "Shift = ###.###  RMS% = ###.###"; ShiftMin; RMSmin
            ELSE
                EXIT FOR
            END IF
        Shift = Shift * .9
        NEXT j
    Shift = ShiftMin
    END IF
    GOSUB LCompress    'COMPUTE SHIFTED AND COMPRESSED DEPTHS AND THICKNESSES
    GOSUB LVES         'COMPUTE KERNEL & VES & COMPARE Ves TO RhoDig
    GOSUB SaveLyrNVes
    RMSmin = RMS

FlipFlop:
2000
    FOR i = 1 TO Layers
        Rho(i) = Rhodig(i) / ves(i) * Rho(i)
    NEXT i
    GOSUB LVES                                  'COMPUTE AND COMPARE
    PRINT USING " RMS% = ###.###"; RMS
    IF RMS > RMSmin THEN GOTO Replough     'FLIP FAILS TO IMPROME
    GOSUB SaveLyrNVes
    IF (RMSmin / RMS - 1) * 100 < 5 THEN GOTO Replough
    RMSmin = RMS
    IF RMSmin < Ft OR Iterate > Max THEN GOTO Graphics
    Iterate = Iterate + 1
    GOTO FlipFlop          ' FLIP-FLOP SOME MORE

Replough:
2100
    IF Replough = 1 THEN GOTO Graphics
    CLS
    cntr 3, " INTERPRETING SMOOTHED CURVE ..... "
    LOCATE 5, 1
    FOR i = 1 TO Layers
        Rhodig(i) = Vesf(i)
    NEXT i
    Replough = 1
    Ft = 1
    GOTO Init
```

```
'-------------------------------------------------------------------------
LCompress:
2200
    depth(1) = Abdig(1) * Shift
    Thick(1) = depth(1)
    FOR i = 2 TO Layers
        depth(i) = Abdig(i) * Shift
        depth(i) = depth(i - 1) * Squish
        Thick(i) = depth(i) - depth(i - 1)
    NEXT i
    depth(Layers) = 99999
RETURN

'-------------------------------------------------------------------------
LVES:
2300
    IF FixRhoN <> 0 THEN Rho(Layers) = FixRhoN
    CALL Kernel(Layers, Xamda(), Thick(), Rho(), vv(), nrad)
    CALL Wconvs(vv(), ves(), nrad, Layers, Xk(), Nk)
    CALL RMS1(Rhodig(), ves(), RMS, Layers)
RETURN

'-------------------------------------------------------------------------
SaveLyrNVes:
    FOR i = 1 TO Layers
        Vesf(i) = ves(i)
        Rhof(i) = Rho(i)
        Depthf(i) = depth(i)
        Thickf(i) = Thick(i)
    NEXT i
RETURN

'-------------------------------------------------------------------------
Graphics:
    GOSUB IntrPlt
    IF Dir$ = "" THEN GOTO Listem
    CLRLNE 1
    LOCATE 1, 10
    PRINT " SAVE LAYERING ? (Y/N/<Esc>) "
    GOSUB YsNo
    IF a$ = Es$ THEN GOTO MainMnu
    IF a$ = "Y" THEN
        CLRLNE 1
        LOCATE 1, 20
        PRINT "SAVING ......."
        FILENAME$ = LEFT$(FILENAME$, INSTR(FILENAME$, ".")) + "LYR"
        OPEN DD$ + "\" + Dir$ + "\" + FILENAME$ FOR OUTPUT AS #2
        PRINT #2, TITLE$
        PRINT #2, ABUnit$
        PRINT #2, Layers
        FOR i = 1 TO Layers
            PRINT #2, Depthf(i), Rhof(i)
        NEXT i
        CLOSE #2
        CLRLNE 1
        LOCATE 1, 1
            PRINT TITLE$; " saved as: ";
            PRINT " "; FILENAME$; " "; " Press Any Key to Continue"
            WHILE INKEY$ = "": WEND
    END IF
Listem:
    CLRLNE 1
    LOCATE 1, 10
    PRINT " LIST RESULTS & DUMP GRAPHICS on printer ? (Y/N/<Esc>) "
    GOSUB YsNo
    IF a$ = Es$ OR a$ = "N" THEN GOTO Whatnow
    GOSUB PrntrRdy
    nspc = 14
    IF convert$ = "Y" THEN nspc = 6
```

```
    LPRINT : LPRINT SPC(nspc); TITLE$; "  (INTERPRETATION)": LPRINT
    XNu = INT(Layers / 2)
    XNm = Layers / 2
        IF convert$ = "N" OR convert$ = "" THEN
            form5$ = "              #####.##      #####.##      #####.##      #####.##"
            form6$ = "                                          #####.##      #####.##"
            LPRINT "                DEPTH         RESIS.        DEPTH         RESIS."
            LPRINT
            FOR i = 1 TO XNu
                LPRINT USING form5$; Depthf(i); Rhof(i); Depthf(i + XNu); Rhof(i + XNu)
            NEXT i
            IF XNm > XNu THEN LPRINT USING form6$; Depthf(Layers); Rhof(Layers)
        ELSE
            forma$ = "      DEPTH, m (   ft   )      RESIS.   DEPTH, m (   ft   )     RESIS."
            form7$ = "      #####.## (#####.##)    #####.##   #####.## (#####.##)   #####.##"
            form8$ = "                                         #####.## (#####.##)   #####.##"
            LPRINT forma$: LPRINT
            FOR i = 1 TO XNu
                LPRINT USING form7$; Depthf(i); CTOFT(Depthf(i)); Rhof(i); Depthf(i + XNu); CTOFT(Depthf(i
+ XNu)); Rhof(i + XNu)
            NEXT i
            IF XNm > XNu THEN LPRINT USING form8$; Depthf(Layers); Depthf(Layers); Rhof(Layers)
        END IF
    GOSUB DmpGrfxMsg

Whatnow:
        VesTyp$ = "DGT"
        NAPR = Nab2
        Replough = 0
        FOR i = 1 TO Nab2
            Rhodig(i) = RhoDig1(i)
            ab2(i) = Abdig(i)
            psob(i) = RhoDig1(i)
        NEXT i
        setscrn
        LOCATE 6, 12
        PRINT "Do you wish to: "
        LOCATE 8, 12
        PRINT "1. "; : HIGHL " ENTER "
        PRINT " sounding from the ";
        HIGHL " SAME AREA. "
        LOCATE 10, 12
        PRINT "2. "; : HIGHL " LOAD "
        PRINT " sounding from the ";
        HIGHL " SAME AREA. "
        LOCATE 12, 12
        PRINT "3. "; : HIGHL " DISPLAY "
        PRINT " digitized sounding to edit & reinterpret. "
        LOCATE 14, 12
        PRINT "4. "; : HIGHL " REINTERPRET "
        PRINT " same sounding using "; : HIGHL " DIFFERENT OPTIONS. "
        LOCATE 16, 12
        PRINT "5. "; : HIGHL " RETURN TO MAIN MENU. "
        LOCATE 18, 12
        PRINT "6. "; : HIGHL " QUIT. "
        k = PrssNmbr(6)
        ON k GOTO ABUnit, LodVes, Plt, Options, MainMnu, Kwit

'-------------------------------------------------------------------------
ChVsNmbr:
    IF Edt = 0 THEN RETURN
    CLRLNE 1
    LOCATE 1, 10: PRINT "Modify sounding number = "; VesNum$; " ? (Y/N/<Esc>)"
    GOSUB YsNo
    IF a$ = Es$ THEN GOTO MainMnu
    IF a$ = "Y" THEN
        CLRLNE 1
        LOCATE 1, 10
        LINE INPUT "Enter NEW sounding number (4 char. max) ", a$
```

```
        GOSUB CapNTrim
        IF a$ = "" THEN BEEP: GOTO ChVsNmbr
        VesNum$ = LEFT$(a$, 4)
        FOR i = LEN(TITLE$) TO 1 STEP -1
            IF MID$(TITLE$, i, 1) = " " THEN
                TITLE$ = LEFT$(TITLE$, i) + VesNum$
                i = 1
            END IF
        NEXT i
        FILENAME$ = LEFT$(Dir$, 4) + VesNum$ + "." + VesTyp$
    END IF
RETURN

'-----------------------------------------------------------------
SaveIt:
    IF DD$ = "" OR Dir$ = "" THEN
        CLRLNE 1
        BEEP
        LOCATE 1, 10
        PRINT " Can't Save! Drive or Directory not specified."
        RETURN
    END IF
    GOSUB ChVsNmbr
    CLRLNE 1
    LOCATE 1, 10: PRINT " SAVING ......"
    OPEN DD$ + "\" + Dir$ + "\" + FILENAME$ FOR OUTPUT AS #1
    PRINT #1, TITLE$
    PRINT #1, ABUnit$
    FOR i = 1 TO Nab2
        PRINT #1, ab2(i), psob(i)
    NEXT i
    CLOSE #1
    CLRLNE 1
    LOCATE 1, 1
        PRINT TITLE$; " saved as: ";
        PRINT " "; FILENAME$; " "; " Press Any Key to Continue"
        WHILE INKEY$ = "": WEND
RETURN

'-------------------------------------------------------------------------
FldPlt:
    CRVPLT = true
    GRAPHSET GR$
    SETCOORD AX(), AY(), ab2(), psob(), Nab2
    CALL Logplt(AX(), AY())
    ABX(1) = algt(ab2(1)): PSY(1) = algt(psob(1))

PltCrv:
    FOR i = 2 TO Nab2
        IF ab2(i - 1) < ab2(i) THEN
            ABX(i) = algt(ab2(i))
            PSY(i) = algt(psob(i))
            LINE (ABX(i - 1), PSY(i - 1))-(ABX(i), PSY(i))
        ELSEIF ab2(i - 1) >= ab2(i) THEN
            ABX(i) = algt(ab2(i))
            PSY(i) = algt(psob(i))
        END IF
    NEXT i
    GOSUB PltCrcl
    LOCATE 2, 28
    PRINT "o--o--o"
    LOCATE 3, 26
    IF VesTyp$ = "DGT" THEN PRINT "DIGITIZED CURVE" ELSE PRINT "FIELD CURVE"
    GOSUB Lbl
    CRVPLT = FALSE
RETURN

'-------------------------------------------------------------------------
Wtr:    a$ = INKEY$: IF a$ = "" THEN GOTO Wtr
```

```
    RETURN

'-------------------------------------------------------------------------
CapNTrim:
    a$ = UCASE$(a$)
    TEST$ = ""           'remove delimiters
    FOR i = 1 TO LEN(a$)
        Z% = ASC(MID$(a$, i, 1))
        IF (Z% >= 48 AND Z% <= 57) OR (Z% >= 65 AND Z% <= 90) THEN TEST$ = TEST$ + CHR$(Z%)
    NEXT i
     a$ = TEST$
RETURN

'-------------------------------------------------------------------------
YsNo:
    a$ = CAPNWT$
    IF a$ = "Y" OR a$ = "N" OR a$ = Es$ THEN RETURN
    BEEP
    GOTO YsNo

'-------------------------------------------------------------------------
NtEql:
    BEEP
    CLR5L
    cntr 24, " Number of (a) is NOT EQUAL to number of APP.RHO!! "
    PRSSANYKY
    CLR5L
    RETURN

'-------------------------------------------------------------------
PrntrRdy:
    CLRLNE 1
    LOCATE 1, 5: PRINT "Make sure printer is on, Adjust Paper, press <ENTER> when ready"
    WHILE INKEY$ <> CHR$(13): WEND
RETURN

'-------------------------------------------------------------------------
DmpGrfxMsg:
    CLRLNE 1
    PRTSCR
RETURN

'-------------------------------------------------------------------------
IntrPlt:
    Intrp = 1
    GRAPHSET GR$
    SETCOORD AX(), AY(), Abdig(), Vesf(), Layers
    CALL Logplt(AX(), AY())
    ABX(1) = algt(Abdig(1)): PSY(1) = algt(Vesf(1))
    FOR i = 2 TO Layers
        ABX(i) = algt(Abdig(i))
        PSY(i) = algt(Vesf(i))
        LINE (ABX(i - 1), PSY(i - 1))-(ABX(i), PSY(i))
    NEXT i
    RAD = .03
    FOR i = 1 TO Layers
        CIRCLE (ABX(i), algt(RhoDig1(i))), RAD
    NEXT i
    LINE (algt(Depthf(1) / 10), algt(Rhof(1)))-(algt(Depthf(1)), algt(Rhof(1)))
    FOR i = 2 TO Layers
        LINE -(algt(Depthf(i - 1)), algt(Rhof(i)))
        LINE -(algt(Depthf(i)), algt(Rhof(i)))
    NEXT i
    LOCATE 2, 28
    PRINT "o  o  o", " ---------"
    LOCATE 3, 26
    PRINT "DIGITIZED", "CALCULATED"
    GOSUB Lbl
    Intrp = 0
```

```
RETURN

'-------------------------------------------------------------------------
PltCrcl:
    RAD = .03
    FOR i = 1 TO Nab2
        CIRCLE (ABX(i), PSY(i)), RAD
    NEXT i
RETURN

'-------------------------------------------------------------------------
Lbl:
    LOCATE 3, (79 - LEN(TITLE$)): PRINT TITLE$
    LOCATE 4, 24 - LEN(STR$(AY(4))): PRINT AY(4)
    LOCATE 10, 24 - LEN(STR$(AY(3))): PRINT AY(3)
    LOCATE 16, 24 - LEN(STR$(AY(2))): PRINT AY(2)
    LOCATE 22, 24 - LEN(STR$(AY(1))): PRINT AY(1)
    LOCATE 23, 24: PRINT AX(1)
    LOCATE 23, 37: PRINT AX(2)
    LOCATE 23, 50: PRINT AX(3)
    LOCATE 23, 63: PRINT AX(4)
    LOCATE 23, (78 - INT(algt(AX(5)))): PRINT AX(5);

Lbl2:
    LOCATE 3, (79 - LEN(TITLE$)): PRINT TITLE$       ' modified title
    LOCATE 25, 35: PRINT "ELECTRODE SPACING (a)";
    IF Intrp = 1 THEN PRINT ", OR DEPTH,";
    PRINT " IN " + ABUnit$;
    LOCATE 5, 17: PRINT "R"
    LOCATE 6, 17: PRINT "E"
    LOCATE 7, 17: PRINT "S"
    LOCATE 8, 17: PRINT "I"
    LOCATE 9, 17: PRINT "S"
    LOCATE 10, 17: PRINT "T"
    LOCATE 11, 17: PRINT "I"
    LOCATE 12, 17: PRINT "V"
    LOCATE 13, 17: PRINT "I"
    LOCATE 14, 17: PRINT "T"
    LOCATE 15, 17: PRINT "Y"
    LOCATE 18, 17: PRINT "O"
    LOCATE 19, 17: PRINT "H"
    LOCATE 20, 17: PRINT "M"
    LOCATE 21, 17: PRINT "¦"
    LOCATE 22, 17: PRINT "M"
RETURN

'-------------------------------------------------------------------------
Dsply:
    setscrn
    xx = 5: y = 4: X = 6: i = 1: Q = 1

DsplABHdr:
    LOCATE 2, xx - 2: PRINT "#"
    LOCATE 2, xx: HIGHL " (a) "

DsplABNmbr:
    LOCATE y, X - 4: COLOR 3, 1: PRINT Q;
    LOCATE y, X: HIGHL STR$(ab2(i))
    Q = Q + 1
    IF i = Nab2 THEN
        y = 4
        xx = 18
        X = 18
        i = 1
        GOTO DsplRoHdr
    END IF
    i = i + 1
    IF y = 18 THEN
        y = 4
```

```
        xx = xx + 26
        X = X + 26
    ELSE
        y = y + 1
    END IF
    GOTO DsplABNmbr

DsplRoHdr:
        LOCATE 2, xx - 2: PRINT "#"
        LOCATE 2, xx
        HIGHL " APP.RHO "

DsplRoNmbr:
        LOCATE y, X - 4
        COLOR 3, 1: PRINT Q
        LOCATE y, X
        HIGHL STR$(psob(i))
    IF i = NAPR THEN RETURN
    i = i + 1
    Q = Q + 1
    IF y = 18 THEN
        y = 4
        xx = xx + 25
        X = X + 25
    ELSE
    y = y + 1
    END IF
    GOTO DsplRoNmbr

'-------------------------------------------------------------------------
Kwit:
    setscrn
    cntr 12, "Are you sure you want to QUIT ? (Y/N/<Esc>)"
    GOSUB YsNo
    IF a$ = Es$ OR a$ = "N" THEN GOTO MainMnu
    IF a$ = "Y" THEN CLS : SYSTEM

'-------------------------------------------------------------------------
ChkError:
    IF ERR = 5 AND CRVPLT THEN
        E = true
        RESUME NEXT
    END IF
    IF ERR = 11 THEN
        BEEP
        setscrn
        LOCATE 16, 21: PRINT " Division by Zero !"
        PRSSANYKY
        RESUME MainMnu
    END IF
    IF ERR = 25 THEN
        BEEP
        setscrn
        LOCATE 10, 10
        PRINT " PRINTER trouble! ....."
        PRSSANYKY
        RESUME MainMnu
    END IF
    IF ERR = 57 THEN
        CLS
        BEEP
        LOCATE 16, 21
        PRINT "Unformatted disk in selected drive!"
        PRSSANYKY
        RESUME DrvMnu
    END IF
    IF ERR = 64 THEN BEEP: RESUME LodFile
    IF ERR = 68 AND chkdrive$ = "CHKDRIVE" THEN
        BEEP
```

```
        LOCATE 16, 21
        PRINT "Drive ("; DD$; ") not found"
        CLOSE #1
        PRSSANYKY
        RESUME DrvMnu
    END IF
    IF ERR = 75 AND chkdir$ = "dir" OR ERR = 76 AND chkdir$ = "dir" THEN 'dir exists
        chkdir$ = ""
        RESUME NEXT
    END IF
    IF ERR = 75 AND chkdir$ = "LOD" OR ERR = 76 AND chkdir$ = "LOD" THEN 'dir exists
        chkdir$ = ""
        RESUME LodVes
    END IF
    IF ERR = 70 THEN
        CLS
        BEEP
        LOCATE 10, 20
        PRINT " DISK WRITE PROTECTED ?"
        PRSSANYKY
        RESUME MainMnu
    END IF
    IF ERR = 71 THEN
        BEEP
        LOCATE 16, 21
        PRINT "Drive ("; DD$; ") NOT READY, Door Open or No Disk in Drive!"
        CLOSE #1
        PRSSANYKY
        RESUME DrvMnu
    END IF
    IF ERR = 53 AND chkdrive$ = "CHKDRIVE" THEN RESUME NEXT
    IF ERR = 53 AND ChkFil = 1 THEN ChkFil = 0: RESUME NEXT
    IF ERR = 53 THEN
        BEEP
        CLS
        LOCATE 10, 16: PRINT "File NOT FOUND ...."
        PRSSANYKY
        RESUME LodVes
    END IF
    setscrn
    PRINT "Can't do what you want. Error number = "; ERR, ERL
    PRSSANYKY
    RESUME MainMnu

'---------------------------------------------------------------------------
PrtAbDgt:
    LOCATE y, X - 4: PRINT i
    LOCATE y, X: PRINT Abdig(i)
    RETURN

'---------------------------------------------------------------------------
ShftNDig:
      Npts = Nab2
    FOR i = 1 TO Npts
        absh(i) = algt(ab2(i))
        obsh(i) = algt(psob(i))
    NEXT i
    temp = 0
    CALL Spline1(Npts, temp, absh(), obsh(), a(), B(), C(), temp, D(), p(), S())
    Delx = 1 / 8
    CALL Cubic1(Npts, absh(), obsh(), a(), B(), C(), Delx, Abdig(), Ps(), Nab2)
    FOR i = 1 TO Nab2 - 1
        LINE (Abdig(i), Ps(i))-(Abdig(i + 1), Ps(i + 1))
    NEXT i
    FOR i = 1 TO Nab2
        Abdig(i) = 10 ^ Abdig(i)
        Rhodig(i) = 10 ^ Ps(i)
    NEXT i
    Radmin = Abdig(1)
```

```
    Layers = Npts
    IF DD$ = "" OR Dir$ = "" THEN
        CLRLNE 1
        LOCATE 1, 1: PRINT " Press any key to continue ..."
        WHILE INKEY$ = "": WEND
        RETURN
    END IF
    CLRLNE 1
    LOCATE 1, 10: PRINT "Save DIGITIZED  curve ? (Y/N/<Esc>"
    GOSUB YsNo
    IF a$ = Es$ THEN GOTO MainMnu
    IF a$ = "Y" THEN
        GOSUB ChVsNmbr
        FILENAME$ = LEFT$(FILENAME$, INSTR(FILENAME$, ".")) + "DGT"
        CLRLNE 1
        LOCATE 1, 10: PRINT " SAVING ..... "
        OPEN DD$ + "\" + Dir$ + "\" + FILENAME$ FOR OUTPUT AS #1
        PRINT #1, TITLE$
        PRINT #1, ABUnit$
        FOR i = 1 TO Nab2
            PRINT #1, Abdig(i), Rhodig(i)
        NEXT i
        CLOSE #1
        CLRLNE 1
        LOCATE 1, 1
        PRINT TITLE$ + " saved as: " + " " + FILENAME$ + " " + " Press Any Key To Continue"
        WHILE INKEY$ = "": WEND
    END IF
RETURN

SUB CHKGRAPH (E, GR$) STATIC
    E = FALSE
    SCREEN 9
    IF NOT E THEN GR$ = "E": EXIT SUB
    E = FALSE
    SCREEN 10
    IF NOT E THEN GR$ = "M": EXIT SUB
    E = FALSE
    SCREEN 2
    IF NOT E THEN GR$ = "C": EXIT SUB
    GR$ = "N"
END SUB

SUB GRAPHSET (GR$) STATIC
IF GR$ = "C" THEN
    CLS
    SCREEN 2
    VIEW (200, 45)-(630, 170)
ELSE
    CLS
    IF GR$ = "E" THEN SCREEN 9:                                    ELSE SCREEN 10
    VIEW (200, 45)-(630, 300)
END IF
END SUB

SUB UPSORT (a$(), N) STATIC
            k = 2: M = N - 1
SORT1:   IF M < INT(N / 2) THEN EXIT SUB
            IF a$(k - 1) > a$(M + 1) THEN SWAP a$(k - 1), a$(M + 1)
            FOR i = k TO M
                IF a$(i) > a$(M + 1) THEN
                    SWAP a$(i), a$(M + 1)
                ELSEIF a$(i) < a$(k - 1) THEN
                    SWAP a$(i), a$(k - 1)
                END IF
            NEXT i
            k = k + 1
            M = M - 1
            GOTO SORT1
```

```
END SUB
```

****

Module ATOSUB.BAS

****

```
DECLARE FUNCTION CTOFT! (a!)
DECLARE SUB HIGHL (a$)
DECLARE FUNCTION LOGAV! (a!, B!)
DECLARE SUB CLRLNE (i%)
DECLARE SUB cntr (i%, a$)
DECLARE FUNCTION PrssNmbr% (N%)
DECLARE FUNCTION CAPNWT$ ()
DECLARE FUNCTION algt! (X!)
DECLARE FUNCTION DROUND! (tx1!, ty1!)
CONST al10 = 2.302585

DEFINT I-N
FUNCTION algt (X)
   algt = LOG(X) / al10
END FUNCTION

FUNCTION CAPNWT$ STATIC
Cap:
   a$ = INKEY$: IF a$ = "" THEN GOTO Cap
   IF a$ = CHR$(27) THEN GOTO cdone
   IF a$ = CHR$(13) THEN GOTO cdone
   a$ = UCASE$(a$)
   IF a$ < "A" OR a$ > "Z" THEN BEEP: GOTO Cap
cdone: CAPNWT$ = a$
END FUNCTION

SUB CLR5L STATIC
    COLOR 7, 1
    CLRLNE 21
    CLRLNE 22
    CLRLNE 23
    CLRLNE 24
    CLRLNE 25
END SUB

SUB CLRLNE (i) STATIC
LOCATE i, 1: PRINT STRING$(79, " ");
END SUB

SUB cntr (i, a$)
    j = (80 - LEN(a$)) / 2
    LOCATE i, j: HIGHL a$
END SUB

SUB conves (Vv(), Ves(), Nrad, Xk(), Nc) STATIC
   M = 0
   L = 1
   Ll = Nc
Cnvlv:
   Vs = 0
   FOR j = L TO Ll
      Vs = Vs + Vv(j) * Xk(j - M)
   NEXT j
   Ves(L) = Vs
   L = L + 1
   Ll = Ll + 1
```

```
   M = M + 1
   IF L <= Nrad GOTO Cnvlv
END SUB

FUNCTION CTOFT (a) STATIC
B = a / .3048
IF B > 99999! THEN B = 99999!
CTOFT = B
END FUNCTION

SUB Cubic1 (M, X(), Y(), a(), B(), C(), Delx, xx(), Yy(), K) STATIC
   Xmax = X(M)
   Xmin = X(1)
   K = INT(DROUND((Xmax - Xmin) / Delx, 5)) + 1
   IF K < 1 THEN GOTO L9
   i = M
   xx(K) = Xmax
   Yy(K) = Y(M)
   FOR L = 1 TO K - 1
      j = K - L
      xx(j) = Xmax - (K - j) * Delx
L7:
      IF DROUND(xx(j), 5) >= DROUND(X(i - 1), 5) THEN GOTO L8
      i = i - 1
      GOTO L7
L8:
      Z = xx(j) - X(i - 1)
      Yy(j) = Y(i - 1) + ((C(i - 1) * Z + B(i - 1)) * Z + a(i - 1)) * Z
   NEXT L
   EXIT SUB
L9:
   PRINT "ERROR IN CUBIC1"

END SUB

SUB DELAY (B) STATIC
a = TIMER
WHILE TIMER - a < B: WEND
END SUB

FUNCTION DROUND (tx1, ty1) STATIC
DROUND = INT(.5 + (tx1 * (10 ^ ty1))) / (10 ^ ty1)
END FUNCTION

SUB HIGHL (a$) STATIC
COLOR 0, 7
PRINT a$;
COLOR 7, 1
END SUB

SUB Kernel (L, X(), t(), R(), V(), N) STATIC
   FOR j = 1 TO N
       B = 1
       a = -2 / X(j)
       FOR i = L TO 2 STEP -1
          C = B * R(i)
          D = a * t(i - 1)
          IF D < -10 THEN B = 1: GOTO nxt
          Q = (R(i - 1) - C) / (R(i - 1) + C) * EXP(D)
          B = (1 - Q) / (1 + Q)
nxt:     NEXT i
       V(j) = B * R(1)
   NEXT j
END SUB

DEFSNG I-N
FUNCTION LOGAV (a, B) STATIC
LOGAV = 10 ^ ((algt(a) + algt(B)) / 2)
END FUNCTION
```

```
SUB Logplt (AX(), AY()) STATIC
 x0 = algt(AX(1)): xm = algt(AX(5))
 y0 = algt(AY(1)): ym = algt(AY(4))
 WINDOW (x0, y0)-(xm, ym)
 t = .065
FOR i = 1 TO 5
   X = x0 + i - 1
   x1 = 10 ^ X
   LINE (X, y0)-(X, ym)
   FOR j = 2 TO 9
   x2 = algt(x1 * j)
   LINE (x2, y0)-(x2, y0 + t)
   LINE (x2, ym - t)-(x2, ym)
   NEXT j
NEXT i
t = .05
FOR i = 1 TO 4
   Y = y0 + i - 1
   y1 = 10 ^ Y
   LINE (x0, Y)-(xm, Y)
   FOR j = 2 TO 9
      y2 = algt(y1 * j)
      LINE (x0, y2)-(x0 + t, y2)
      LINE (xm - t, y2)-(xm, y2)
   NEXT j
NEXT i
END SUB

DEFINT I-N
SUB LVES (a(), B(), N, t$, U$, K) STATIC
   INPUT #1, t$
   INPUT #1, U$
      FOR i = 1 TO 45
         IF EOF(K) THEN EXIT FOR
         INPUT #K, a(i), B(i)
      NEXT i
      N = i - 1
   CLOSE #1
END SUB

SUB ONEILL (Xk(), N) STATIC
   Xk(1) = -.000318
   Xk(2) = .002072
   Xk(3) = -.004978
   Xk(4) = .01125
   Xk(5) = -.02521
   Xk(6) = .05812
   Xk(7) = -.1436
   Xk(8) = .393
   Xk(9) = -1.1324
   Xk(10) = 2.7044
   Xk(11) = -3.4507
   Xk(12) = .4248
   Xk(13) = 1.1817
   Xk(14) = .6194
   Xk(15) = .2374
   Xk(16) = .08688
   Xk(17) = .0235
   Xk(18) = .01284
   Xk(19) = -.001198
   Xk(20) = .003042
   N = 20
END SUB

DEFSNG I-N
SUB PRSSANYKY
   LOCATE 25, 1: PRINT STRING$(78, " ");
   cntr 25, " Press Any Key to Continue "
   WHILE INKEY$ = "": WEND
```

```
END SUB

DEFINT I-N
FUNCTION PrssNmbr% (N) STATIC
Prss:
    LOCATE 25, 26
    COLOR 0, 7: PRINT " Please Press Appropriate Number "; : COLOR 7, 1
    a$ = INKEY$
    WHILE a$ = "": a$ = INKEY$: WEND
    K = VAL(a$)
    IF K < 1 OR K > N THEN BEEP: GOTO Prss
   PrssNmbr% = K
END FUNCTION

SUB RMS1 (RhoDig(), Ves(), RMS, Layers) STATIC
   RMS = 0
   FOR i = 1 TO Layers
   RMS = ((RhoDig(i) - Ves(i)) / RhoDig(i)) ^ 2 + RMS
   NEXT i
   RMS = (RMS / Layers) ^ .5 * 100
END SUB

SUB SETCOORD (AX(), AY(), X(), Y(), N) STATIC
   AX(1) = 10 ^ INT(algt(X(1)))
   FOR i = 2 TO 5
      AX(i) = AX(i - 1) * 10
   NEXT i
   IF AX(4) > X(N) THEN
      FOR i = 1 TO 5
         AX(i) = AX(i) / 10
      NEXT i
   END IF
   RhoMin = Y(1)
   RhoMax = Y(1)
   FOR i = 1 TO N
      IF Y(i) < RhoMin THEN RhoMin = Y(i)
      IF Y(i) > RhoMax THEN RhoMax = Y(i)
   NEXT i
   AVRHO = EXP((LOG(RhoMin) + LOG(RhoMax)) / 2)
   AY(1) = 10 ^ INT(algt(AVRHO)) / 10
   FOR i = 2 TO 4
      AY(i) = AY(i - 1) * 10
   NEXT i
END SUB

SUB setscrn STATIC
   SCREEN 0
   COLOR 7, 1
   CLS
END SUB

SUB Shft (ab(), ob(), npts, absh(), obsh(), nsh, iseg) STATIC
   DIM sgmnt(15)
   j = 1
   i = 1
   absh(1) = ab(1)
   obsh(1) = ob(1)
   nsgmnt = 1
   amult = 1
   sgmnt(1) = 1
   s1 = ab(1)
lshift1:
   i = i + 1
   IF i > npts THEN GOTO doneshift
   IF ab(i) > s1 THEN
      j = j + 1
      absh(j) = ab(i)
      obsh(j) = ob(i) * amult
      s1 = ab(i)
```

```
   ELSEIF ab(i) = s1 THEN
      amult = obsh(j) / ob(i)
      nsgmnt = nsgmnt + 1
      sgmnt(nsgmnt) = amult
   ELSE
   END IF
   GOTO lshift1
doneshift:
   nsh = j
   IF iseg >= nsgmnt OR iseg < 1 THEN EXIT SUB
   i = nsgmnt - iseg + 1
   amult = sgmnt(i)
   FOR j = 1 TO nsh
      obsh(j) = obsh(j) / amult
   NEXT j
END SUB

SUB Spline1 (M, H, X(), Y(), a(), B(), C(), t, D(), P(), S()) STATIC
   IF (t < 0) OR (t > 1) OR (H < 0) OR (M < 3) THEN GOTO L999
   N = M - 1
   IF t >= 0 THEN GOTO L20
   Ne = N - 1
   IF H = 0 THEN GOTO L11
L1:
   Hh = 3 / H
   FOR i = 1 TO Ne
      B(i) = 4
      a(i) = 1
      C(i) = 1
      P(i) = Hh * (Y(i + 2) - Y(i))
   NEXT i
   P(1) = P(1) - D(1)
   P(Ne) = P(Ne) - D(2)
L3:
   C(1) = C(1) / B(1)
   P(1) = P(1) / B(1)
   FOR i = 2 TO Ne
      Mul = 1 / (B(i) - a(i) * C(i - 1))
      C(i) = Mul * C(i)
      P(i) = Mul * (P(i) - a(i) * P(i - 1))
   NEXT i
   a(Ne + t) = P(Ne)
   i = Ne - 1
L5:
   a(i + t) = P(i) - C(i) * a(i + t + 1)
   i = i - 1
   IF i >= 1 THEN GOTO L5
   IF t = 0 THEN GOTO L6
   a(1) = D(1)
   a(M) = D(2)
L6:
   IF H = 0 THEN GOTO L14
   Hh = 1 / H
   FOR i = 1 TO N
      Mul = Hh * (Y(i + 1) - Y(i))
      B(i) = Hh * (3 * Mul - (a(i + 1) + 2 * a(i)))
      C(i) = Hh * Hh * (-2 * Mul + a(i + 1) + a(i))
   NEXT i
   GOTO Rtn
L11:
   FOR i = 1 TO N
      S(i + 1) = X(i + 1) - X(i)
   NEXT i
   FOR i = 1 TO Ne
      B(i) = 2 * (S(i + 1) + S(i + 2))
      C(i) = S(i + 1)
      a(i) = S(i + 2)
      P(i) = 3 * (S(i + 1) ^ 2 * (Y(i + 2) - Y(i + 1)) + S(i + 2) ^ 2 * (Y(i + 1) - Y(i))) / (S(i + 1) * S(i
+ 2))
```

```
   NEXT i
   P(1) = P(1) - S(3) * D(1)
   P(Ne) = P(Ne) - S(N) * D(2)
   GOTO L3
L14:
   FOR i = 1 TO N
      Hh = 1 / S(i + 1)
      Mul = (Y(i + 1) - Y(i)) * Hh ^ 2
      B(i) = 3 * Mul - (a(i + 1) + 2 * a(i)) * Hh
      C(i) = -2 * Mul * Hh + (a(i + 1) + a(i)) * Hh ^ 2
   NEXT i
   GOTO Rtn
L20:
   Ne = N + 1
   IF H = 0 THEN GOTO L31
   Hh = 3 / H
   FOR i = 2 TO N
      P(i) = Hh * (Y(i + 1) - Y(i - 1))
      B(i) = 4
      C(i) = 1
      a(i) = 1
   NEXT i
   B(1) = 2
   B(Ne) = 2
   C(Ne) = 1
   a(Ne) = 1
   P(1) = Hh * (Y(2) - Y(1)) - .5 * H * D(1)
   P(Ne) = Hh * (Y(M) - Y(N)) + .5 * H * D(2)
   GOTO L3
L31:
   FOR i = 1 TO N
      S(i + 1) = X(i + 1) - X(i)
   NEXT i
   N1 = N - 1
   FOR i = 1 TO N1
      B(i + 1) = 2 * (S(i + 1) + S(i + 2))
      C(i + 1) = S(i + 1)
      a(i + 1) = S(i + 2)
      P(i + 1) = 3 * (S(i + 1) ^ 2 * (Y(i + 2) - Y(i + 1)) + S(i + 2) ^ 2 * (Y(i + 1) - Y(i))) / (S(i + 1) *
S(i + 2))
   NEXT i
   B(1) = 2
   B(Ne) = 2
   C(1) = 1
   C(Ne) = 1
   a(Ne) = 1
   P(1) = 3 * (Y(2) - Y(1)) / S(2) - .5 * S(2) * D(1)
   P(Ne) = 3 * (Y(M) - Y(N)) / S(M) + .5 * S(M) * D(2)
   GOTO L3
L999: PRINT "ERROR IN SPLIN1"

Rtn: END SUB
```

****

Module DRCT.BAS

****

```
DECLARE SUB CNTR (i%, A$)
DECLARE SUB HIGHL (A$)
DECLARE FUNCTION trim$ (A$)
DECLARE SUB LCATE (i%, iof%, m%)
TYPE regtype
     AX    AS INTEGER
```

```
     bx    AS INTEGER
     cx    AS INTEGER
     dx    AS INTEGER
     bp    AS INTEGER
     si    AS INTEGER
     di    AS INTEGER
     flags AS INTEGER
END TYPE
DECLARE SUB interrupt (intnum AS INTEGER, INREG AS regtype, outreg AS regtype)

DEFINT I-N
SUB DIRECT (A$(), n, p$, k) STATIC
DIM INREG AS regtype, outreg AS regtype, f AS STRING * 128
i = 0
INREG.dx = VARPTR(f)
INREG.AX = &H1A00
CALL interrupt(&H21, INREG, outreg)
INREG.AX = &H4E00
p1$ = p$ + CHR$(0)
INREG.dx = SADD(p1$)
INREG.cx = &H255
CALL interrupt(&H21, INREG, outreg)
lupe:
IF outreg.AX AND &H18 THEN GOTO done
   file$ = f$
   at$ = MID$(file$, 22, 1)
   m = ASC(at$)
   f$ = LEFT$(f$, 30) + "                    "
   IF k <> 1 AND m <> 16 THEN GOTO c1
      i = i + 1
      A$(i) = trim$(MID$(file$, 31, 13))
c1:   INREG.AX = &H4F00
      CALL interrupt(&H21, INREG, outreg)
GOTO lupe
done:
   n = i
END SUB

SUB FILESLCT (nam$, fl$(), nf, k) STATIC
IF nf = 0 THEN
   PRINT " no files exist"
   nam$ = ""
   EXIT SUB
ELSE
   n = nf
   IF n > 126 THEN n = 126
END IF
IF k > 126 THEN
   k = k - ist + 1
   iend = ist + 125
   IF iend > nf THEN iend = nf
   n = iend - ist + 1
   m = 1 + INT(n / 6)
   IF m > 21 THEN m = 21
ELSE
   m = 1 + INT(n / 6): nof = 2: j = 1
   IF m > 21 THEN m = 21
   ist = 1
   iend = n
END IF
GOSUB lst
lupp: A$ = INKEY$
   WHILE A$ = "": A$ = INKEY$: WEND
    L = LEN(A$): b = ASC(RIGHT$(A$, 1))
   IF L = 2 AND b = 72 GOTO up
   IF L = 2 AND b = 80 GOTO dwn
   IF L = 2 AND b = 75 GOTO lft
   IF L = 2 AND b = 77 GOTO rgt
   IF L = 2 AND b = 71 GOTO hme
```

```
   IF L = 2 AND b = 79 GOTO enn
   IF L = 2 AND b = 73 GOTO pup
   IF L = 2 AND b = 81 GOTO pdn
   IF b <> 13 GOTO lupp
k = ist + i - 1
nam$ = fl$(k)
EXIT SUB
up:   LCATE i, nof, m: PRINT fl$(ist + i - 1): i = i - 1: IF i < 1 THEN i = n
      LCATE i, nof, m: HIGHL fl$(ist + i - 1): GOTO lupp
dwn:  LCATE i, nof, m: PRINT fl$(ist + i - 1): i = i + 1: IF i > n THEN i = 1
      LCATE i, nof, m: HIGHL fl$(ist + i - 1): GOTO lupp
lft:  LCATE i, nof, m: PRINT fl$(ist + i - 1): i = i - m: IF i < 1 THEN i = 1
      LCATE i, nof, m: HIGHL fl$(ist + i - 1): GOTO lupp
rgt:  LCATE i, nof, m: PRINT fl$(ist + i - 1): i = i + m: IF i > n THEN i = n
      LCATE i, nof, m: HIGHL fl$(ist + i - 1): GOTO lupp
hme:  LCATE i, nof, m: PRINT fl$(ist + i - 1): i = 1
      LCATE i, nof, m: HIGHL fl$(ist + i - 1): GOTO lupp
enn:  LCATE i, nof, m: PRINT fl$(ist + i - 1): i = n
      LCATE i, nof, m: HIGHL fl$(ist + i - 1): GOTO lupp
pdn: LOCATE 1, 1
      IF iend = nf THEN GOTO lupp
      ist = ist + 126
      k = 1
      iend = ist + 125
      IF iend > nf THEN iend = nf
      n = iend - ist + 1
      m = 1 + INT(n / 6)
      IF m > 21 THEN m = 21
      GOSUB lst
      GOTO lupp
pup:  LOCATE 1, 1
      IF ist = 1 THEN GOTO lupp
      ist = ist - 126: IF ist < 1 THEN ist = 1
      k = 1
      iend = ist + 125
      IF iend > nf THEN iend = nf
      n = iend - ist + 1
      m = 1 + INT(n / 6)
      IF m > 21 THEN m = 21
      GOSUB lst
      GOTO lupp
lst:  CLS
      CNTR 1, " SELECT A FILE OR SELECT EXIT TO RETURN "
      FOR i = ist TO iend
         LCATE i - ist + 1, nof, m
         PRINT fl$(i);
      NEXT i
      i = k
      LCATE i, nof, m: HIGHL fl$(ist + i - 1)
      RETURN
END SUB

SUB LCATE (i, iof, m) STATIC
   n = INT((i - 1) / m)
   j = n * 13 + 1
   k = iof + i - n * m
   LOCATE k, j
END SUB

DEFSNG I-N
SUB PRTSCR STATIC
DIM INREG AS regtype, outreg AS regtype
INREG.AX = &H500
CALL interrupt(&H5, INREG, outreg)
END SUB

FUNCTION trim$ (A$) STATIC
capn:
   A$ = UCASE$(A$)
```

```
   T$ = ""'remove spaces
   FOR i = 1 TO LEN(A$)
      j = ASC(MID$(A$, i, 1))
      IF j >= 45 AND j <= 57 OR j >= 65 AND j <= 90 THEN T$ = T$ + CHR$(j)
   NEXT i
   trim$ = T$
END FUNCTION
```

****

Module WENN.BAS

****

```
DECLARE SUB convES (vv!(), ves!(), nrad%, Xk!(), nc%)
DEFINT I-N

SUB KOEFOED4 (c(), n, s, nr) STATIC
c(1) = .0006
c(2) = -.0042
c(3) = .0193
c(4) = -.1029
c(5) = .567
c(6) = -1.7514
c(7) = 1.2348
c(8) = .8415
c(9) = .138
c(10) = .0589
c(11) = -.0086
c(12) = .0091
c(13) = -.0031
c(14) = .0008
n = 14
s = 1.6258
nr = 8
END SUB

SUB Wconvs (vv(), ves(), nrad, nradii, Xk(), nc) STATIC
   DIM vv1(50), vv2(50), vs(50)
   j1 = 1: j2 = 1
   FOR i = 1 TO nrad STEP 2
      vv1(j1) = vv(i)
      j1 = j1 + 1
      IF i + 1 > nrad THEN EXIT FOR
      vv2(j2) = vv(i + 1)
      j2 = j2 + 1
   NEXT i
   nvs = INT((1 + nradii) / 2)
   convES vv1(), vs(), nvs, Xk(), nc
   j = 1
   FOR i = 1 TO nvs
      ves(j) = vs(i)
      j = j + 2
   NEXT i
   nvs = INT(nradii / 2)
   convES vv2(), vs(), nvs, Xk(), nc
   j = 2
   FOR i = 1 TO nvs
      ves(j) = vs(i)
      j = j + 2
   NEXT i
   FOR i = 1 TO nradii
'   LPRINT i; ves(i)
   NEXT i
END SUB
```

```
SUB wgosh (Xk(), n) STATIC
Xk(1) = -.0067
Xk(2) = .0179
Xk(3) = -.0253
Xk(4) = .0416
Xk(5) = -.0935
Xk(6) = .3473
Xk(7) = -1.3341
Xk(8) = 1.5662
Xk(9) = .4582
Xk(10) = .0282
n = 10
END SUB
```

APPENDIX B

****

Program PICKCONT.BAS

****

```
DECLARE SUB DELAY (B!)
DECLARE SUB UPSORT (A$(), N%)
DECLARE SUB ReadLYR (k%, D!(), R!(), N%, T$, u$)
DECLARE SUB LLYR (D!(), R!(), N%, T$, u$)
DECLARE SUB CNTRES (D!(), R!(), Layers%, DP!(), RS!(), M%)
DECLARE SUB SLCTSOUND (D$, p$, S$(), M%, N$, k%)
DECLARE SUB FILESLCT (NAM$, fl$(), N%, k%)
DECLARE SUB DIRECT (A$(), N%, p$, k%)
DECLARE SUB PRSSANYKY ()
DECLARE SUB CLRLNE (i%)
DECLARE SUB SETSCRN ()
DECLARE SUB CNTR (j%, A$)
DECLARE SUB HIGHL (A$)
DECLARE SUB SETCNT (c!(), nc%)
DECLARE SUB SETPATH (D$, p$, ierr%)
DECLARE FUNCTION ALGT! (A!)
DECLARE FUNCTION LOGAV! (A!, B!)
DECLARE FUNCTION CAPNWT$ ()
DECLARE FUNCTION trim$ (A$)
DECLARE SUB pause ()
DEFINT I-N
CONST false = 0, true = NOT false, ndim = 90
DIM depth(ndim), resist(ndim), CONT(35), DP(100), RS(100)
DIM CNT(100), DPC(100), FILES$(200)
F$ = "   #####.#     #####.#"
G$ = "   LAST DEPTH = #####.#"
ierr = false
ON ERROR GOTO Errorproc
M0: SETSCRN
   CNTR 12, " PRINTER OUTPUT? (Y/N) "
   ans$ = CAPNWT
   SELECT CASE ans$
      CASE "Y"
         printer = true
      CASE "N"
         printer = false
      CASE ELSE
         BEEP
         DELAY 100
         GOTO M0
   END SELECT
SETCNT CONT(), ncnt
ierr = false
M1:   SETPATH Dd$, dir$, ierr
      IFRST = true
M2:   SLCTSOUND Dd$, dir$, FILES$(), NF, NAM$, IFRST
      IFRST = false
      IF NAM$ = "DONE" THEN
         SETSCRN
         END
      END IF
      IF NAM$ = "CHANGE DRIVE" THEN GOTO M1
      inn = FREEFILE
      OPEN Dd$ + ":\" + dir$ + "\" + NAM$ FOR INPUT AS inn
      ReadLYR inn, depth(), resist(), Layers, T$, unit$
      CNTRES depth(), resist(), Layers, DP(), RS(), M
      L = 0: i = 1: j = 1
```

```
M3:   IF i >= M THEN GOTO m6
      SLP = (ALGT(RS(i + 1)) - ALGT(RS(i))) / (ALGT(DP(i + 1)) - ALGT(DP(i)))
      IF SLP > 0 THEN
M4:      IF CONT(j) > RS(i + 1) THEN
            i = i + 1
            GOTO M3
         ELSEIF CONT(j) > RS(i) AND CONT(j) <= RS(i + 1) THEN
            L = L + 1
            CNT(L) = CONT(j)
            DPC(L) = 10! ^ (ALGT(DP(i)) + (ALGT(CONT(j)) - ALGT(RS(i))) / SLP)
         END IF
            j = j + 1
            IF j > ncnt THEN
               j = ncnt
               i = i + 1
               GOTO M3
            ELSE
               GOTO M4
            END IF
      ELSE
m4a:   IF i = 1 THEN j = ncnt
M5:      IF CONT(j) < RS(i + 1) THEN
            i = i + 1
            GOTO M3
         ELSEIF CONT(j) < RS(i) AND CONT(j) >= RS(i + 1) THEN
            L = L + 1
            CNT(L) = CONT(j)
            DPC(L) = 10! ^ (ALGT(DP(i)) + (ALGT(CONT(j)) - ALGT(RS(i))) / SLP)
         END IF
            j = j - 1
            IF j < 1 THEN
               j = 1
               i = i + 1
               GOTO M3
            ELSE
               GOTO M5
            END IF
      END IF
m6:   SETSCRN
      PRINT "    " + T$
      PRINT : PRINT "    DEPTH IN    CONTOUR"
      PRINT "     " + unit$ + "      VALUE": PRINT
      IF L > 18 THEN
      FOR i = 1 TO 18
         PRINT USING F$; DPC(i); CNT(i)
      NEXT i
      LOCATE 3, 40: PRINT "    DEPTH IN    CONTOUR"
      LOCATE 4, 40: PRINT "     " + unit$ + "      VALUE": PRINT
      FOR i = 19 TO L
         LOCATE 5 + i - 18, 40
         PRINT USING F$; DPC(i); CNT(i)
      NEXT i
      ELSE
      FOR i = 1 TO L
         PRINT USING F$; DPC(i); CNT(i)
      NEXT i
      END IF
      IF printer THEN
         LPRINT : LPRINT USING G$; DP(M)
         LPRINT : LPRINT : LPRINT "    " + T$
         LPRINT : LPRINT "    DEPTH IN    CONTOUR"
         LPRINT "     " + unit$ + "      VALUE": LPRINT
         FOR i = 1 TO L
            LPRINT USING F$; DPC(i); CNT(i)
         NEXT i
         LPRINT : LPRINT USING G$; DP(M)
      END IF
      PRSSANYKY
      GOTO M2
END
```

```
Errorproc:
   ierr = true
   SELECT CASE ERR
   CASE 24, 25, 27
      SETSCRN
      CNTR 12, " PRINTER OFF, OFFLINE OR OUT OF PAPER. "
      CNTR 14, " PLEASE FIX "
      PRSSANYKY
      RESUME
   CASE 71
         BEEP
         CNTR 16, "    DRIVE " + Dd$ + " not ready or door open    "
e1:      CNTR 17, "press R to retry or S to select another drive"
         k = INSTR("RS", UCASE$(INPUT$(1)))
         ON k + 1 GOTO e1, e2, e3
e2:      CLRLNE 16
         CLRLNE 17
         RESUME
e3:      RESUME M1
   CASE 76
         ierr = ERR
         RESUME NEXT
   CASE 53
         ierr = ERR
         RESUME
   CASE ELSE
         PRINT "ERROR NUMBER "; ERR
         END
END SELECT
RESUME

SUB CNTRES (D(), R(), N, DP(), RS(), M) STATIC
   j = 1
   DP(j) = 10! ^ (ALGT(D(1)) - .5 * (ALGT(D(2)) - ALGT(D(1))))
   RS(j) = R(1)
   FOR i = 1 TO N - 1
      j = j + 1
      DP(j) = D(i)
      RS(j) = LOGAV(R(i), R(i + 1))
      j = j + 1
      DP(j) = LOGAV(D(i), D(i + 1))
      RS(j) = R(i + 1)
   NEXT i
   M = j
   DP(M) = 10! ^ (ALGT(D(N - 1)) + .5 * (ALGT(D(N - 1)) - ALGT(D(N - 2))))
END SUB

SUB LLYR (D(), R(), N, T$, u$) STATIC
SETSCRN
LOCATE 5, 5: HIGHL T$: PRINT "     UNIT= " + u$
PRINT
PRINT "  DEPTH       RESISTIVITY": PRINT
FOR i = 1 TO N
   PRINT D(i), R(i)
NEXT i
PRSSANYKY
END SUB

SUB ReadLYR (k, D(), R(), N, T$, u$) STATIC
INPUT #k, T$
INPUT #k, u$
INPUT #k, N
FOR i = 1 TO N
   INPUT #k, D(i), R(i)
NEXT i
CLOSE #k
END SUB

SUB SETPATH (D$, dr$, ierr) STATIC
DIM dir$(ndim)
```

```
set1: SETSCRN
      LOCATE 10, 10
      PRINT "type letter of DISK DRIVE "; : HIGHL " (A,B,C...etc) ": PRINT " ";
      D$ = CAPNWT$
      PRINT D$
      CHDIR D$ + ":\"
      IF ierr > 0 THEN
         BEEP
         CNTR 15, "DRIVE " + D$ + " not found"
         PRSSANYKY
         ierr = true
         GOTO set1
      END IF
set2: SETSCRN
      CNTR 1, "  SELECT THE DATA SUB-DIRECTORY  "
      p$ = D$ + ":\*.*"
      DIRECT dir$(), ndir, p$, 0
      ndir = ndir + 1
      dir$(ndir) = "CHANGE DRIVE"
      FILESLCT dr$, dir$(), ndir, 1
      IF dr$ = "CHANGE DRIVE" THEN GOTO set1
      dr$ = trim$(dr$)
END SUB

SUB SLCTSOUND (D$, p$, S$(), NS, N$, k) STATIC
   IF NOT k THEN GOTO sl1
   A$ = D$ + ":\" + RTRIM$(p$) + "\*.lyr"
   DIRECT S$(), NS, A$, 1
   UPSORT S$(), NS
   NS = NS + 1
   S$(NS) = "CHANGE DRIVE"
   NS = NS + 1
   S$(NS) = "DONE"
   ik1 = 1
sl1: SETSCRN
   CNTR 1, "  SELECT A SOUNDING  "
   FILESLCT N$, S$(), NS, ik1
END SUB

SUB UPSORT (A$(), N) STATIC
        k = 2: M = N - 1
SORT1:  IF M < INT(N / 2) THEN EXIT SUB
        IF A$(k - 1) > A$(M + 1) THEN SWAP A$(k - 1), A$(M + 1)
        FOR i = k TO M
           IF A$(i) > A$(M + 1) THEN
              SWAP A$(i), A$(M + 1)
           ELSEIF A$(i) < A$(k - 1) THEN
              SWAP A$(i), A$(k - 1)
           END IF
        NEXT i
        k = k + 1
        M = M - 1
        GOTO SORT1
END SUB
```

****

Module CONTSUB.BAS

****

```
DECLARE SUB PRSSANYKY ()
DECLARE SUB SETSCRN ()
DECLARE SUB CNTR (j%, A$)
DECLARE SUB HIGHL (A$)
DECLARE FUNCTION CAPNWT$ ()
DECLARE SUB pause ()
```

```
DEFINT I-N
errhand:
ierr = ERR
SELECT CASE ierr
   CASE 52, 53
      RESUME NEXT
   CASE ELSE
      PRINT "error ", ERR
      STOP
END SELECT

SUB SETCNT (c(), nc) STATIC
SHARED ierr
SETSCRN
CNTR 12, " LOADING CONTOUR VALUES "
ON ERROR GOTO errhand
in = FREEFILE
OPEN "contval.dat" FOR INPUT AS in
IF ierr = 53 OR ierr = 52 THEN
CNTR 15, " FILE OF CONTOUR VALUES NOT FOUND "
CNTR 17, " PRESET VALUES WILL BE USED "
PRSSANYKY
c(1) = 1!
c(2) = 1.5
c(3) = 2!
c(4) = 3!
c(5) = 4.5
c(6) = 7!
c(7) = 10!
c(8) = 15!
c(9) = 20!
c(10) = 30!
c(11) = 45!
c(12) = 70!
c(13) = 100!
c(14) = 150!
c(15) = 200!
c(16) = 300!
c(17) = 450!
c(18) = 700!
c(19) = 1000!
c(20) = 1500!
c(21) = 2000!
c(22) = 3000!
c(23) = 4500!
c(24) = 7000!
nc = 24
ELSE
i = 1
   DO UNTIL EOF(in) OR i > UBOUND(c)
      INPUT #in, c(i)
      i = i + 1
   LOOP
   nc = i-1
   FOR i = 2 TO nc
      IF c(i) < c(i - 1) THEN
         nc = i - 1
         CNTR 15, "ERROR VALUE OUT OF ORDER"
         CNTR 17, " USING THE FIRST " + STR$(nc) + " VALUES "
         PRSSANYKY
         EXIT FOR
      END IF
   NEXT i
END IF
ON ERROR GOTO 0
END SUB
```