INSTITUTE
OF
HYDROLOGY

MICROCOMPUTER PROGRAMS TO ASSIST THE ACQUISITION AND INTERPRETATION RESISTIVITY SOUNDING DATA

by

J W FINCH

ABSTRACT

This report describes and lists four programs, designed for use on microcomputers, for the acquisition and interpretation of resistivity sounding data. The programs are designed to be used principally with the BGS Offset Wenner system and are written for the Epson HX-20 and Research Machines 380Z microcomputers. They are user-friendly and actions by the operator are in response to prompts from the program in a question-and-answer form.

REPORT NO 91

September 1984

CONTENTS

INTRODUCTION

This report describes four programs that have been designed to assist the acquisition and interpretation of vertical electrical sounding data. The first three programs are designed for use with data acquired using the BGS Offset Wenner sounding system. The fourth program is more general and can accept any data acquired using either Wenner or Schlumberger data.

The programs are designed to be user-friendly and data entries or decisions are made by the operator in response to prompts using question and answer. All entries are in free format and are echoed by the program to be verified by the operator. The first two programs, used for data acquisition and transfer, are written in Basic for use on the Epson HX-20 portable computer. The other programs, for data acquisition and interpretation are in Microsoft Fortran for the Research Machines 380Z.

HARDWARE REQUIREMENTS

The first two programs require a standard Epson HX-20 portable computer fitted with a microcassette drive, plus a cable with the appropriate connectors to link the Epson HX-20 to the Research Machines 380Z.

The other two programs require a Research Machines 380Z microcomputer with 56K RAM, a high resolution graphics board, twin double density, dual sided, 5.25" floppy disc drives, a colour monitor and an Epson FX-80 dot-matrix printer. In addition Microsoft Fortran is necessary as is some form of software capable of editing text, programs and data.

SYSTEMS DESIGN

The basic concept of the system is to use the HX-20 in the field for data acquisition and the 380Z in the office for interactive, iterative interpretation. Data is entered using the HX-20's keyboard in response to prompts and, after verification, it is stored on a microcassette and listed on the microprinter. This is done using program 1. Program 2 is then used to transfer data from microcassettes to disc files on the 380Z. To do this it is necessary to modify one of the utility routines that run under CP/M, the 380Z's operating system. Program 3 runs on the 380Z and processes the data from the HX-20 to produce Wenner sounding curves which it stores in disc files. These can then be interpreted using Program 4.

The program requirements are:

	Language	Size
1	Basic	4946 bytes
2	Basic	2492 bytes
3	Fortran	22460 bytes
4	Fortran	33843 bytes

The soundings are identified by their reference number. This can have up to eight characters which may be alphanumerics or special characters such as - or /. It is important that the reference number is unique and so it is recommended that a systematic approach to the numbers is adopted. Combinations of map codes and dates are often effective.

On the 3807 the filename qualifier .DAT is used to signify unprocessed BGS readings whilst the qualifier .INT signifies unformatted apparent resistivity curves.

BGS OFFSET WENNER SYSTEM

This field system allows Wenner apparent resistivity curve data to be acquired more accurately and with less man power than conventional systems. The equipment consists of two multicore cables fitted with metallic take-outs at pre-determined distances, a number of electrodes, a switch box and cables to connect the system to a resistivity meter.

The system uses a five electrode array with the electrodes equi-spaced. However, only four electrodes are used at a time. Thus, for each electrode spacing a series of measurements can be made using different electrode combinations. Two of these configurations D1 and D2 are conventional Wenner arrays but shifted by one spacing. Barker (1980) has shown that by taking the mean of these two readings a substantial reduction in the effects of lateral inhomogeneities can be achieved. The three remaining configurations A, B and C do not use the central electrodes but are a tripotential system. They allow interpolation between the points produced by the offset Wenner readings.

The use of a multicore cable reduces the time taken to lay out the cables and move the electrodes. The fixed takeouts for the elctrodes reduce errors due to incorrect spacing distances whilst the system of readings allow further checks on data quality to be carried out.

Three measures of error are possible from the readings. The first, the reading error, is a measure solely of the instrument, operator and electrode array. The sum of readings B and C must equal reading A, irrespective of the subsurface conditions. Thus any significant deviation from this equality indicates a malfunction in the system.

The second measure is that if subsurface conditions are laterally homogeneous, the two readings D1 and D2 will be equal. Hence the degree of difference between the two readings is a measure of subsurface lateral inhomogeneity. This is called the offset error.

The third error measure, the potential ladder error, is generated when the sets of A, B and C readings are used to interpolate between the offset data. This error is also a measure of subsurface lateral inhomgeneity.

Program 1 in this report allows these error measures to be calculated in the field so that quality checks can be made when the data is acquired. It also allows the full Wenner resistivity curve to be generated so that a field interpretation can be carried out. This allows subsequent survey work to be planned on the basis of the data already collected.

PROGRAM 1

This program is designed to be used in the field to acquire BGS Offset Wenner data. It should be loaded and run at the beginning of the day's work and then the HX-20 should be left on until the day's work is completed. This is because when the program is started, it creates a file on the microcassette which it names with the date. Thus, if the program is run on the same day with the same tape it will create two files with the same name.

The program uses the RAM file of the HX-20 to keep a record of the files that occur on the microcassette. Before a microcassette is removed from the HX-20 this list of files is written on to a file named CASO:DIR.DIR at the beginning of the tape. Before a microcassette can be used a blank copy of this file must be written on to it. This is done using a program called CASSFMT. Microcassettes that contain old data that is no longer required can be re-used after the directory file has been rewritten as blank.

For each sounding, the program asks initially for data about the site of the sounding. Unknown information can be entered as blank or zero. The information requested is:

- Site grid reference
- Azimuth of the array
- The V.E.S. (site) number

The operator is then asked to enter readings. For each set of readings he is asked initially for the electrode spacing number, as marked on the BGS switch box. If this spacing has already been used then the operator is asked to verify that it is to be used again. This protects against accidental repetition of readings but allows readings to be repeated if a mistake was made. The operator is then asked to give the configuration he is using, i.e. whether it is A, B, C, D1 or D2, and the reading. If the configuration has already been used then a warning is given and the operator is asked to verify the request.

When the five readings have been given for an electrode spacing the program checks the reading error and, if it exceeds 1%, a warning is displayed and the operator is asked to verify that set of readings. The offset error is also given so that a running check on data quality is made.

When the operator has given the readings for all the electrode spacings he wishes to use the program checks that the spacings used were those intended. If this is verified then the input readings, the calculated error measures and Wenner apparent resistivity curve are listed on the microprinter. The input readings are then written to the microcassette. A listing of the output from this program is given below.

File CASO: 09/27/83.DAT
starting at 2

V.E.S. FP9

Grid Ref. 61472 89812
Azimuth 13

INPUT DATA

A	C	B
15.300	14.700	0.6840
7.550	6.910	0.5220
4.870	4.540	0.3050
2.590	2.410	0.1700
1.090	1.030	0.0580
0.346	0.323	0.0210

D1	D2
11.100	11.600
5.970	5.370
3.510	3.430
1.880	2.120
0.850	0.080
0.275	0.2[illegible]

ERROR MEASURES

Obs.	Offset	Pot. lad
-0.0055	-0.0441	0.0037
0.0156	0.1050	-0.1270
0.0051	0.0231	0.0400
0.0[illegible]9	-0.1200	-0.0220
0.00[illegible]8	-0.0437	0.0025
0.00[illegible]0	-0.0216	0.0000

PROCESSING RESULTS

Spac.	Resis.
0.5	35.7
1.0	35.6
1.5	41.5
2.0	43.6
3.0	47.5
4.0	50.3
6.0	49.0
8.0	43.7
12.0	34.8
16.0	27.9
24.0	22.3
32.0	18.5

Mean obs. err. 0.0077
Mean off. err. 0.0712
Mean pot. err. 0.0002


```
10 CLEAR200,480:DEFFIL16,0:S%=0:F%=0:FD$=SPACE$(12):FORI=0TO29:PUT%I,S%,F%,FD$:N
EXTI:GOTO100
20 IE=0:IFID$="OD"THENGOTO30ELSEIFID$="ID"THENGOTO40ELSEIFID$="CF"THENGOTO80ELSE
IFID$="OF"THENGOTO50ELSEIFID$="IF"THENGOTO90ELSERETURN
30 WIND-4000:TAPCNT=0:OPEN"O",£1,"CASO:DIR.DIR":FORI=0TO29:GET%I,S%,F%,FD$:PRINT
£1,S%,F%,FD$:NEXTI:CLOSE£1:WIND-4000:TAPCNT=0:RETURN
40 WIND-4000:TAPCNT=0:OPEN"I",£1,"CASO:DIR.DIR":FORI=0TO29:IFEOF(1)THENCLOSE£1:W
IND-4000:TAPCNT=0:RETURNELSEINPUT£1,S%,F%,FD$:PUT%I,S%,F%,FD$:NEXTI:CLOSE£1:WIND
-4000:TAPCNT=0:RETURN
50 IFLEN(FD$)<12THENFD$=FD$+SPACE$(8-LEN(FD$))+".DAT"
60 WIND-4000:TAPCNT=0:FORI=0TO29:GET%I,S%,F%,FF$:IFS%=<0THENIFI=0THENS%=200:GOTO
70ELSEGET%(I-1),F%,S%,FF$:GOTO70ELSENEXTI:IE=1:RETURN
70 IT=I:S%=S%+5:WINDS%:OPEN"O",£1,("CASO:"+FD$):RETURN
80 CLOSE£1:F%=TAPCNT:PUT%IT,S%,F%,FD$:WIND-4000:TAPCNT=0:RETURN
90 IFLEN(FD$)<12THENFD$=FD$+SPACE$(8-LEN(FD$))+".DAT":WIND-4000:TAPCNT=0:FORI=0T
O29:GET%I,S%,F%,FF$:IFFF$=FD$THENWIND(S%-5):OPEN"I",£1,("CASO:"+FD$):RETURNELSEI
FS%=0THENIE=1:RETURNELSENEXTI:IE=1:RETURN
100 DIM R(5,11),ES$(5),E1(11),E2(11),E3(11),AA(25),RW(25),D(11),DD(11)
110 PI=8.0*ATN(1.0)
120 PRINT "Cass. ready Y/N"
130 A$=INKEY$:IF A$="Y" THEN 150 ELSE IF A$="N" THEN140 ELSE 130
140 PRINT "Insert cassette": PRINT "and start again":END
150 ID$="ID":GOSUB20:FD$=DATE$+".DAT":ID$="OF":GOSUB20
160 IFIE=1THENPRINT"CASSETTE FULL":GOTO120
170 LPRINT: LPRINT"File ";FD$
180 FOR I=0 TO 24:AA(I)=0.0: NEXT I
190 DATA A,C,D1,D2,B
200 FOR I=0 TO 4: READ ES$(I): NEXT I
210 FOR I=0 TO 4: FOR J=0 TO 9: R(I,J)=0.0: NEXT J: NEXT I
220 CLS:PRINT "Give grid letters":PRINT "of the grid ref.":INPUTSJ$
230 PRINT "Give easting of"
240 INPUT "v.e.s.";G1
250 PRINT "Give northing of"
260 INPUT "v.e.s.";G2
270 CLS:PRINT "Grid reference": PRINT SJ$;G1;G2
280 PRINT "Correct Y/N"
290 A$=INKEY$:IF A$="Y" THEN 300 ELSE IF A$="N" THEN 220 ELSE 290
300 CLS:PRINT "Give azimuth of"
310 INPUT "array";AZ
320 CLS:PRINT "Azimuth =";AZ
330 PRINT "Correct Y/N"
340 A$=INKEY$:IF A$="Y" THEN 350 ELSE IF A$="N" THEN 300 ELSE 340
350 CLS:PRINT "Give v.e.s."
360 INPUT "number";V$
370 CLS:PRINT "V.e.s. =";V$
380 PRINT "Correct Y/N"
390 A$=INKEY$:IF A$="Y" THEN 400 ELSE IF A$="N" THEN 350 ELSE 390
400 CLS:PRINT "Give electrode spa."
410 INPUT "number";N
420 CLS:PRINT "Spacing no. =";N
430 PRINT "Correct Y/N"
440 A$=INKEY$:IF A$="Y" THEN 450 ELSE IF A$="N" THEN 400 ELSE 440
450 FOR K=0 TO 4: IF R(K,N)<>0 THEN 470: ELSE NEXT K
460 GOTO 500
470 CLS:SOUND 4,4: PRINT "Spacing already used"
480 PRINT "replace Y/N"
490 A$=INKEY$:IF A$="Y" THEN 500 ELSE IF A$="N" THEN 400 ELSE 490
500 FOR K=0 TO 4
510 CLS:PRINT "Which config."
520 INPUT "A,C,D1,D2 or B";C$
530 FOR I=0 TO 4: IF ES$(I)=C$ THEN 550 ELSE NEXT I
540 CLS:PRINT "No such config.": GOTO 510
550 IF R(I,N)=0.0 THEN 580 ELSE SOUND 4,4: PRINT "reading already given"
560 PRINT "Replace Y/N"
570 A$=INKEY$:IF A$="Y" THEN 580 ELSE IF A$="N" THEN 510 ELSE 570
580 INPUT "Give reading"; R(I,N)
590 CLS:PRINT "Config. = ";C$
600 PRINT "Reading ="; R(I,N)
610 PRINT "Correct Y/N"
620 A$=INKEY$:IFA$="Y" THEN 630 ELSE IF A$="N" THEN 550 ELSE 620
630 NEXT K
640 E1(N)=(R(0,N)-R(1,N)-R(4,N))/R(0,N)
650 CLS:PRINT "Obs. error =";:PRINT USING"££.££££";E1(N)
660 IF ABS(E1(N))<0.01 THEN 680 ELSE PRINT "Obs. error > 0.01": PRINT "please ve
rify":PRINT "Correct Y/N"
670 A$=INKEY$:IF A$="Y" THEN 680 ELSE IF A$="N" THEN 450 ELSE 670
680 E2(N)=2*(R(2,N)-R(3,N))/(R(2,N)+R(3,N))
```

```
690 PRINT "Offset err =";:PRINT USING"££.££££";E2(N)
700 PRINT "Another spacing":PRINT"Y/N"
710 A$=INKEY$:IF A$="Y" THEN 400 ELSE IF A$="N" THEN 720 ELSE 710
720 NS=-1: NF=0
730 FOR I=0 TO 9
740 FOR J=0 TO 4
750 IF R(J,I)<>0 THEN IF NS=-1 THEN NS=I
760 IF R(J,I)=0 THEN IF NS<>-1 THEN NF=I-1: GOTO 780
770 NEXT J: NEXT I
780 CLS:PRINT "Spacings used":PRINT "from ";NS;:PRINT "to ";NF
790 PRINT "Correct Y/N"
800 A$=INKEY$:IF A$="Y" THEN 810 ELSE IF A$="N" THEN 400 ELSE 800
810 LPRINT:LPRINT:LPRINT "V.E.S. ";V$
820 AA(NS)=0.5*NS
830 FOR I=NS TO NF
840 J=(I-NS+1)*2+NS-1:K=J+1
850 AA(J)=AA(NS)*2^(I-NS+1)
860 AA(K)=3.0*AA(NS)*2^(I-NS)
870 NEXT I
880 LPRINT:LPRINT"Grid Ref. ";:LPRINT USING"£££££";G1;:LPRINT USING" £££££";G2
890 LPRINT " Azimuth ";:LPRINT USING"£££";
900 LPRINT:LPRINT"INPUT DATA"
910 LPRINT:LPRINT"   A       C        B":LPRINT
920 FOR I=NS TO NF
930 LPRINT USING"£££.£££";R(0,I);:LPRINT USING"££££.£££";R(1,I);:LPRINT USING"££
£.££££";R(4,I)
940 NEXT I
950 LPRINT:LPRINT"    D1        D2":LPRINT
960 ES1=0: ES2=0: ES3=0
970 PRINT£1,SJ$,G1,G2,AZ,NS,NF,V$
980 FOR I=NS TO NF
990 PRINT£1,I,R(0,I),R(1,I),R(2,I),R(3,I),R(4,I)
1000 NEXT I
1010 FOR I=NS TO NF
1020 EE=ABS(R(0,I))+ABS(R(1,I))+ABS(R(4,I))
1030 LPRINT USING"££££.£££";R(2,I);:LPRINT USING"  ££££.£££";R(3,I)
1040 R(4,I)=R(4,I)-E1(I)*R(0,I)*R(4,I)/EE
1050 R(1,I)=R(1,I)+E1(I)*R(0,I)*R(1,I)/EE
1060 R(0,I)=R(0,I)+E1(I)*R(0,I)*R(0,I)/EE
1070 D(I)=(R(3,I)+R(2,I))/2
1080 ES1=ES1+E1(I)*E1(I)
1090 ES2=ES2+E2(I)*E2(I)
1100 DD(I)=R(1,I)-D(I)
1110 NEXT I
1120 ES1=SQR(ES1/(NF-NS+1))
1130 ES2=SQR(ES2/(NF-NS+1))
1140 RW(NS)=D(NS)
1150 FOR I=NS+1 TO NF-1
1160 II=I*2-1: IJ=I+1: I1=I-1: I2=I*2-2
1170 RW(I2)=D(I)
1180 RW(II)=D(I)/2+R(4,I)-R(4,I1)+D(IJ)/2
1190 NEXT I
1200 II=NF*2-1: IJ=NF*2: NN=NF*2-2
1210 RW(NN)=D(NF)
1220 RW(II)=D(NF)/2+R(4,NF)-R(4,NF-1)+D(NF)
1230 RW(IJ)=DD(NF)*2
1240 FOR I=NS TO NF-1
1250 II=I+1
1260 E3(I)=(DD(I)-D(II)/2)/D(II)
1270 ES3=ES3+E3(I)*E3(I)
1280 NEXT I
1290 ES3=SQR(ES3/(NF-NS)): E3(NF)=0
1300 LPRINT:LPRINT"ERROR MEASURES":LPRINT:LPRINT"  Obs.   Offset  Pot.lad":LPRIN
T
1310 FOR I=NS TO NF
1320 LPRINT USING" ££.££££";E1(I);:LPRINT USING" ££.££££";E2(I);:LPRINT USING "
££.££££";E3(I)
1330 NEXT I
1340 LPRINT: LPRINT"PROCESSING RESULTS":LPRINT
1350 LPRINT "  Spac.  Resis.":LPRINT
1360 FOR I=NS TO NF*2
1370 RW(I)=RW(I)*PI*AA(I)
1380 LPRINT USING "  £££.£";AA(I);:LPRINT USING"   ££££.£";RW(I)
1390 NEXT I
1400 LPRINT: LPRINT"Mean obs. err.";:LPRINT USING" ££.££££";ES1
1410 LPRINT"Mean off. err.";:LPRINT USING" ££.££££";ES2
1420 LPRINT"Mean pot. err.";:LPRINT USING" ££.££££";ES3
```



```
1430 CLS:PRINT"Data written to tape"
1440 PRINT"[C]ontinue or [S]top"
1450 A$=INKEY$:IF A$="C" THEN 180 ELSE IF A$="S" THEN 1460 ELSE 1450
1460 ID$="CF":GOSUB20:ID$="OD":GOSUB20:END
```

PROGRAM 2

This program allows a data file from the microcassettes to be transferred, via the serial ports to the 380Z. Before it can be used a modified version of PIP, the utility routine that runs under CP/M on the 380Z, must be produced to allow the S104 interface to be used for input. The 380Z manual describes how to do this and this program assumes the modified version of PIP is called JWFPIP.

In addition a cable, part number #714, is needed to connect the HX-20 to the 380Z. This cable should be used to connect the socket on the back of the HX-20, for the serial interface, with the socket on the back of the 380Z.

The program uses prompts to tell the user how to set up the 380Z as well as the HX-20. After establishing that a microcassette has been placed in the micro-cassette drive, the operator is asked for the name of the file for transfer. If this file exists on the microcassette the operator is then told how to set up the 380Z. The bit rate for data transfer is asked for. The value given is checked against those available on the HX-20 and, if it is available, the corresponding 380Z code is displayed on the HX-20 screen. Once JWFPIP has been loaded and run on the 380Z, the data is transferred from the HX-20 to the 380Z.

WARNING The first time the serial interfaces are used there is a tendency for some random characters to be generated. To overcome this the program sends a blank line at the beginning of the file. This line should be edited out of the file on the 380Z before it is used by the subsequent programs.

```
10 CLEAR200,480:DEFFIL16,0:S%=0:F%=0:FD$=SPACE$(12):FORI=0TO29:PUT%I,S%,F%,FD$:N
EXTI:GOTO110
20 IE=0:IFID$="OD"THENGOTO30ELSEIFID$="ID"THENGOTO40ELSEIFID$="CF"THENGOTO80ELSE
IFID$="OF"THENGOTO50ELSEIFID$="IF"THENGOTO90ELSERETURN
30 WIND-4000:TAPCNT=0:OPEN"O",£1,"CAS0:DIR.DIR":FORI=0TO29:GET%I,S%,F%,FD$:PRINT
£1,S%,F%,FD$:NEXTI:CLOSE£1:WIND-4000:TAPCNT=0:RETURN
40 WIND-4000:TAPCNT=0:OPEN"I",£1,"CAS0:DIR.DIR":FORI=0TO29:IFEOF(1)THENCLOSE£1:W
IND-4000:TAPCNT=0:RETURNELSEINPUT£1,S%,F%,FD$:PUT%I,S%,F%,FD$:NEXTI:CLOSE£1:WIND
-4000:TAPCNT=0:RETURN
50 IFLEN(FD$)<12THENFD$=FD$+SPACE$(8-LEN(FD$))+".DAT"
60 WIND-4000:TAPCNT=0:FORI=0TO29:GET%I,S%,F%,FF$:IFS%=<0THENIFI=0THENS%=200:GOTO
70ELSEGET%(I-1),F%,S%,FF$:GOTO70ELSENEXTI:IE=1:RETURN
70 IT=I:S%=S%+5:WINDS%:OPEN"O",£1,("CAS0:"+FD$):RETURN
80 CLOSE£1:F%=TAPCNT:PUT%IT,S%,F%,FD$:WIND-4000:TAPCNT=0:RETURN
90 IFLEN(FD$)<12THENFD$=FD$+SPACE$(8-LEN(FD$))+".DAT"
100 WIND-4000:TAPCNT=0:FORI=0TO29:GET%I,S%,F%,FF$:IFFF$=FD$THENWIND(S%-5):OPEN"I
",£1,("CAS0:"+FD$):RETURNELSEIFS%=0THENIE=1:RETURNELSENEXTI:IE=1:RETURN
110 DIM B(7),R(5,11)
120 DATA 110,150,300,600,1200,2400,4800
130 FOR I=0 TO 6: READ B(I): NEXT I
140 E$=".DAT"
150 PRINT "Cass. ready Y/N"
160 A$=INKEY$:IF A$="Y" THEN 190 ELSE IF A$="N" THEN 170 ELSE 160
170 PRINT "Insert cassette"
180 PRINT "and start again":END
```

```
190 CLS:ID$="ID":GOSUB20
200 PRINT"Give filename":INPUT;B$
210 PRINT "Filename = ";B$
220 PRINT "Correct Y/N"
230 A$=INKEY$:IF A$="Y" THEN 240 ELSE IF A$="N" THEN 200 ELSE 230
240 FD$=B$+E$:ID$="IF":GOSUB20
250 IF IE=0THEN280ELSEPRINT"No such file":PRINT"Directory listing?":PRINT" Y/N"
260 A$=INKEY$:IFA$="Y"THEN270ELSEIFA$="N"THEN200ELSE260
270 FOR I=0TO29:GET%I,S%,F%,FF$:IFS%=0THENIFI=0THENPRINT"DIRECTORY EMPTY":GOTO20
0ELSEGOTO200ELSEPRINTI;FF$:NEXTI:GOTO200
280 CLS:PRINT"Set up 380Z. Call":PRINT"front panel (Ctrl F)"
290 PRINT"option O, printer 4"
300 INPUT "Give bit rate";BR
310 FOR I=0 TO 6
320 IF BR=B(I) THEN 350
330 NEXT I
340 PRINT "No such bit rate";BR: GOTO 300
350 IF I=1 THEN PRINT"not on the 380Z":GOTO 300
360 PRINT "Bit rate =";BR
370 PRINT "Correct Y/N"
380 A$=INKEY$:IF A$="Y" THEN390ELSEIFA$="N" THEN300ELSE380
390 F$="COM0:("+RIGHT$(STR$(I),1)+"8N1F)"
400 IF I>1 THEN I=I-1:CLS:PRINT"380Z code = ";I
410 PRINT"to return to CP/M":PRINT "type K"
420 PRINT"Ready Y/N"
430 A$=INKEY$:IF A$="Y" THEN 440 ELSE IF A$="N" THEN410ELSE 430
440 CLS:PRINT"call JWFPIP"
450 PRINT"filename.DAT=INP:"
460 PRINT "Ready Y/N"
470 A$=INKEY$:IF A$="Y" THEN 480 ELSE IF A$="N" THEN 440 ELSE 470
480 OPEN"O",£2,F$
490 PRINT£2
500 INPUT £1,SJ$,G1,G2,AZ,NS,NF,V$
510 FOR I=NS TO NF
520 INPUT £1,II,R(0,I),R(1,I),R(2,I),R(3,I),R(4,I)
530 NEXT I
540 PRINT£2," ";:PRINT£2,USING"\      \";V$;:PRINT£2,USING"\\";SJ$;
550 PRINT£2,USING"£££££";G1;:PRINT£2,USING"£££££";G2;:PRINT£2,USING"£££";AZ;
560 PRINT£2,USING"££";NS;:PRINT£2,USING"££";NF
570 FOR I=NS TO NF
580 PRINT£2," ";:PRINT£2,USING"£££";I;:PRINT£2,USING"££££.££££";R(0,I);
590 PRINT£2,USING"££££.££££";R(1,I);:PRINT£2,USING"££££.££££";R(2,I);
600 PRINT£2,USING"££££.££££";R(3,I);:PRINT£2,USING"££££.££££";R(4,I)
610 NEXT I
620 PRINT £2,CHR$(&H1A)
630 IF EOF(1) THEN 640 ELSE 500
640 CLOSE£1:CLOSE£2:PRINT"[C]ontinue or [S]top"
650 A$=INKEY$:IF A$="C"THEN150ELSEIFA$="S"THENENDELSE650
```

PROGRAM 3

The purpose of this program is to allow readings from the BGS system to be input, either from a file created by transferring data from the HX-20 or directly entered via the keyboard, and to process the data to produce a Wenner apparent resistivity curve which is then written, unformatted, to a file for interpretation by program 4.

The program assumes that a file containing BGS readings has the qualifer .DAT. If data is being entered via the keyboard then the program creates such a file and writes the unprocessed data to it. The Wenner apparent resistivity curve is written, unformatted, to a file with the same name as the file containing the BGS readings but with the qualifier .INT.

When the program is started the operator is asked whether the data is to be entered from a file. If the answer is yes then the operator is asked for the name of the file and the program checks that the file exists. If the data is to be entered via the keyboard then the operator is asked to give a filename and the program checks that this does not already exist.

If the data is being entered from a file then the program processes all the data on the file sequentially. The only decisions made by the operator are whether the results are listed and a plot of the data is printed. Manually entered data is made in response to prompts very similar to those in program 1. All data entries are echoed by the program on the VDU and must be verified by the operator before they are accepted.

The input data as well as the apparent resistivity curve and the error measures can be listed on the printer. An example of the output from the program is given below.

The subroutine EPSON is a proprietary machine code subroutine. Its function is to dump graphics, in this case plots of the apparent resistivity curve, to the Epson dot-matrix printer.

V.E.S. No. FP9 **Grid Ref: SU 61472 89812** **Azimuth: 13**

INPUT DATA

Electrode Spacing	A	C	D1	D2	B
.5	15.3000	14.7000	11.1000	11.6000	.6840
1.0	7.5500	6.9100	5.5700	5.3700	.5220
2.0	4.8700	4.5400	3.9100	3.4300	.3050
4.0	2.5900	2.4100	1.8800	2.1200	.1700
8.0	1.0900	1.0300	.8500	.8880	.0580
16.0	.3460	.3230	.2750	.2810	.0210

PROCESSING RESULTS

Electrode Spacing	Wenner Resistivity	Observed Error	Offset Error	Lateral Error
.5	33.66	-.0055	.0441	.0837
1.0	35.63	.0158	-.1058	-.1270
1.5	41.60			
2.0	48.61	.0051	-.0231	.0408
3.0	47.40			
4.0	50.27	.0039	.1200	-.0228
6.0	48.97			
8.0	43.68	.0018	.0437	.0825
12.0	34.78			
16.0	27.85	.0058	.0316	0.0000
24.0	22.31			
32.0	18.47			

R.M.S. Observational Error = .0077
R.M.S. Offset Wenner Difference = .0712
R.M.S. Potential Ladder Difference = .0802

The main section of the program handles all data entries and sets up the input and output channels.

```
      INTEGER*4 M1,M2
      DIMENSION FSP(3),FSP2(3),A(25),RWA(25)
      COMMON /A/ RGS(25),RGN(25),PB(25),AA(25),PA(20),PG(20)
      DATA YES,RNO/'Y   ','N   '/
      DATA FSP,FSP2/'    ','    ','DAT ','    ','    ','INT '/
      DATA ALF/Z'0A'/
      WRITE(5,600)
  600 FORMAT(' PROGRAM INPUT'//,
     1' This program is used to input resistivity data collected by'/,
     2' the Offset-Wenner method to a disc file for interpretation'//)
   21 IFLG=1
      WRITE(5,908)
  908 FORMAT(' Do you want to enter data from file? (Y/N)'/)
      READ(1,314) ANS
      IF (ANS.EQ.RNO) GOTO 921
      IF (ANS.NE.YES) GOTO 21
      IFLG=0
      CALL DISKFL(FSP)
      GOTO 32
  921 WRITE(5,402)
  402 FORMAT(' Give the number of the disc drive on which the data file
     1 is to be created'/' 1=A,2=B,3=C,4=D'/)
      READ(1,403,ERR=921)ID
      IF(ID.NE.1.AND.ID.NE.2.AND.ID.NE.3.AND.ID.NE.4) GOTO 921
      WRITE(5,318)
  318 FORMAT(' Enter the primary name of your data file '/)
      READ(1,319)FSP(1),FSP(2)
  319 FORMAT(2A4)
  403 FORMAT(I1)
      CALL OPEN(6,FSP,ID)
      READ(6,ERR=32)DUMMY
  605 WRITE(5,1000)FSP(1),FSP(2),ID
 1000 FORMAT(' File ',2A4,' already exist on drive ',I2/,
     1' Try another name!!'/)
      ENDFILE 6
      GOTO 21
   32 WRITE(5,420)
  420 FORMAT(' Give the number of the disc drive on which the output fi
     1le'/' is to be created'/' 1=A,2=B,3=C,4=D'/)
      READ(1,403,ERR=21)ID1
      IF(ID1.NE.1.AND.ID1.NE.2.AND.ID1.NE.3.AND.ID1.NE.4) GOTO 32
      FSP2(1)=FSP(1)
      FSP2(2)=FSP(2)
      CALL OPEN(7,FSP2,ID1)
      READ(7,ERR=65)DUMMY
  650 WRITE(5,1000)FSP2(1),FSP2(2),ID1
      ENDFILE 7
      GOTO 21
C
C     ENTER SITE DETAILS AND RESISTIVITY READINGS TO DISK FILE
C
   65 IF (IFLG) 965,906,965
  906 READ(6,93,END=82) AIP1,AIP2,GRSJ,M1,M2,NZ,NS,NF
      N=NF-NS+1
      STAT=NS*0.5
      DO 907 I=1,N
      AA(I)=2.0**(I-1)*STAT
      READ(6,108) II,PA(I),PG(I),RGS(I),RGN(I),PB(I)
  907 CONTINUE
      GOTO 952
  965 WRITE(5,301)
  301 FORMAT(1X,'Give the V.E.S. number - (up to 8 characters)'/)
      READ(1,302) AIP1,AIP2
  302 FORMAT(2A4)
      WRITE(5,490) AIP1,AIP2
  490 FORMAT(' You have entered the V.E.S. number as ',2A4)
   28 WRITE(5,313)
      READ(1,314) ANS
      IF (ANS.EQ.RNO) GOTO 965
      IF (ANS.NE.YES) GOTO 28
   60 WRITE(5,303)
  303 FORMAT(1X,'Give the grid letters of the sounding grid reference'/
     1)
```



```
      READ(1,304) GRSJ
   22 WRITE(5,405)
      READ(1,406,ERR=22) M1
   23 WRITE(5,407)
      READ(1,406,ERR=23) M2
  406 FORMAT(I5)
  405 FORMAT(' Give the easting of the soundings 5-figure grid reference
     1'/)
  407 FORMAT(' Give the northing of the soundings 5-figure grid referenc
     1e'/)
      WRITE(5,408) GRSJ,M1,M2
  408 FORMAT(' The grid reference you have entered is ',A2,2I7)
   29 WRITE(5,313)
      READ(1,314) ANS
      IF (ANS.EQ.RNO) GOTO 60
      IF (ANS.NE.YES)GOTO 29
   61 WRITE(5,305)
  304 FORMAT(A2)
  305 FORMAT(1X,'Give the azimuth of the sounding (up to three digits)'/
     1)
      READ(1,306,ERR=61) NZ
  306 FORMAT(I3)
   50 WRITE(5,307)
  307 FORMAT(1X,'Give the first spacing used (up to two digits)'/)
      READ(1,308,ERR=50) NS
  308 FORMAT(I2)
      WRITE(5,320)
  320 FORMAT(1X,'Give the last spacing used (up to two digits)'/)
      READ(1,308,ERR=50) NF
      WRITE(5,500) NZ,NS,NF
  500 FORMAT(' You have entered the azimuth as'I4/' the first spa
     1cings as'I3/' the last spacing as'I3/)
   40 WRITE(5,313)
      READ(1,314) ANS
      IF (ANS.EQ.RNO)GOTO 61
      IF (ANS.NE.YES)GOTO 40
      N=NF-NS+1
   31 WRITE(5,312) N
  312 FORMAT(' Give ',I2,' sets of readings (up to 10 digits including
     1 a decimal point)'/)
      DO 5 I=1,N
   62 WRITE(5,410) I
      READ(1,201,ERR=62) PA(I)
   24 WRITE(5,411) I
      READ(1,201,ERR=24) PG(I)
   25 WRITE(5,412) I
      READ(1,201,ERR=25) RGS(I)
   26 WRITE(5,413) I
      READ(1,201,ERR=26) RGN(I)
   27 WRITE(5,414) I
      READ(1,201,ERR=27) PB(I)
  410 FORMAT(' Give the A reading for spacing',I2/)
  411 FORMAT(' Give the C reading for spacing',I2/)
  412 FORMAT(' Give the D1 reading for spacing',I2/)
  413 FORMAT(' Give the D2 reading for spacing',I2/)
  414 FORMAT(' Give the B reading for spacing',I2/)
  480 FORMAT(5X,'WARNING: The observed error is greater than 1%'/,14X,'T
     1here may be an incorrect input data value'/)
      EE=(PA(I)-PB(I)-PG(I))/PA(I)
      IF (ABS(EE)-0.01) 90,90,91
   91 WRITE(5,480)
   90 WRITE(5,415) I, PA(I),PG(I),RGS(I),RGN(I),PB(I)
  415 FORMAT(' The readings for electrode spacing',I2,' you have entered
     1 are'/5F12.4)
   41 WRITE(5,313)
      READ(1,314) ANS
      IF (ANS.EQ.RNO)GOTO 62
      IF (ANS.NE.YES)GOTO 41
    5 CONTINUE
  313 FORMAT(1X,'Correct?  Y or N ?'/)
  314 FORMAT(A1)
  201 FORMAT(F10.0)
C
C
   52 WRITE (6,93) AIP1,AIP2,GRSJ,M1,M2,NZ,NS,NF
   93 FORMAT(1X,2A4,A2,2I5,I3,2I2)
      STAT=0.5*NS
```

```
      DO 54 I=1,N
      AA(I)=2.0**(I-1)*STAT
      WRITE(6,108) I,PA(I),PC(I),RCS(I),RCN(I),PB(I)
108   FORMAT(I4,5F9.4)
54    CONTINUE
952   CALL DGS(AIP1,AIP2,GRSJ,M1,M2,NZ,N,M,A,RWA)
      CALL LO2LO3
      CALL RESPLT(A,RWA,M,0.0,0.0)
      WRITE(5,555)
555   FORMAT(1H1)
      WRITE(7)AIP1,AIPZ,GRSJ,M1,M2,NZ,M
      DO 70 I=1,M
      WRITE(7) A(I),RWA(I)
70    CONTINUE
901   WRITE(5,900)
900   FORMAT(' Do you want the plot printed? (Y/N)'/)
      READ(1,314) ANS
      IF (ANS.EQ.RNO) COTO 42
      IF (ANS.NE.YES) COTO 901
      WRITE(2,902)
902   FORMAT(/1X,'PLOTTED RESULTS'/)
      CALL EPSON(0,2)
42    IF (IFLG) 942,906,942
942   WRIYE(5,800)
800   FORMAT(' Do you wish to enter more field data? (Y or N)'/)
      READ(1,314)ANS
      IF (ANS.EQ.RNO)COTO 82
      IF (ANS.NE.YES)COTO 42
      GOTO 965
82    WRITE(5,83)FSP(1),FSP(2)
83    FORMAT(' your field data has been written to file ',2A4//)
      ENDFILE 6
      ENDFILE 7
      CALL RESOL(0,2)
      STOP
      END
```

Subroutine BGS

This subroutine processes the input readings from the BGS offset Wenner system to produce a Wenner apparent resistivity curve. The variables used in the call are:

- AIP1,AIP2 - the sounding reference number
- GRSJ - the UK grid reference letters
- M1 - grid reference easting
- M2 - grid reference northing
- NZ - azimuth of the array
- N - the number of electrode spacings used
- M - the number of electrode separations on the processed curve
- A - output vector containing the electrode separations
- RWA - output vector containing the Wenner apparent resistivities

The BGS offset readings are entered to the subroutine using vectors through the common block A. The vectors are:

- RGS - D1 readings
- RGN - D2 readings
- PB - B readings
- PA - A readings
- PG - C readings
- AA - electrode separations


```
      SUBROUTINE BGS(AIP1,AIP2,GRSJ,M1,M2,NZ,N,M,A,RWA)
      INTEGER *4 M1,M2
      DIMENSION RA(25),RB(25),RC(25),E(25),RWB(20),RWC(20),
     1RC1(20),RC2(20),EL(20),H(20),A(25),RWA(25)
      COMMON /A/ RCS(25),RCN(25),PB(25),AA(25),PA(20),PC(20)
      DATA YES,RNO/'Y   ','N   '/
      DATA CR/Z'0D'/
      STAT=AA(1)
      IFLC=0
      PI=4.0*ATAN(1.0)
      A(1)=STAT
      DO 20 I=1,10
      J=I*2
      K=J+1
      A(J)=2.0**I*STAT
      A(K)=3.0*STAT*2.0**(I-1)
20    CONTINUE
      WRITE(5,60)
60    FORMAT(' Do you wish to print processed results? (Y or N)'/)
      READ(1,61) ANS
61    FORMAT(A1)
      IF(ANS.EQ.RNO)GOTO 70
      IF(ANS.NE.YES)GOTO 20
93    FORMAT(/// V.E.S. No. 1X,2A4,13X,'Grid Ref.',1X,A2,1X,I5,1X,I5,4X
     1,9HAzimuth: ,I3///)
104   FORMAT(/22X,'INPUT DATA'//,4X,'Electrode',/5X,'Spacing',5X,1HA,9X,
     11HC,9X,2HD1,8X,2HD2,8X,1HB/)
      WRITE(2,77)
77    FORMAT(1H1)
      WRITE(2,93) AIP1,AIP2,GRSJ,M1,M2,NZ
      WRITE(2,104)
      DO 54 I=1,N
      WRITE(2,108) AA(I),PA(I),PC(I),RCS(I),RCN(I),PB(I)
54    CONTINUE
108   FORMAT(4X,F6.1,5F10.4)
109   FORMAT(/4X,9HElectrode,4X,6HWenner,4X,8hObserved,2X,6HOffset,
     13X,7HLateral/5X,7HSpacing,2X,11HResistivity,3(4X,5HError)/)
113   FORMAT(4X,F6.1,3X,F9.2,3X,3F9.4)
114   FORMAT(4X,F6.1,3X,F9.2)
70    EEE=0.0
      HHH=0.0
      DO 32 I=1,N
      E(I)=PA(I)-PB(I)-PC(I)
      EE=ABS(PA(I))+ABS(PB(I))+ABS(PC(I))
      RA(I)=PA(I)-E(I)*PA(I)/EE
      RB(I)=PB(I)+E(I)*PB(I)/EE
      RC(I)=PC(I)+E(I)*PC(I)/EE
      RC1(I)=(RCN(I)+RCS(I))/2.0
      H(I)=(RCN(I)-RCS(I))/RC1(I)
      RC2(I)=RC(I)-RC1(I)
      E(I)=E(I)/RA(I)
      EEE=EEE+E(I)*E(I)
      HHH=HHH+H(I)*H(I)
32    CONTINUE
      EEE=SQRT(EEE/FLOAT(N))
      HHH=SQRT(HHH/FLOAT(N))
      RWA(1)=RC1(1)
      K=N-1
      DO 3 I=2,K
      II=I*2-1
      IJ=I+1
      I1=I-1
      I2=I*2-2
      RWA(I2)=RC1(I)
      RWA(II)=RC1(I)/2.0+RB(I)-RB(I1)+RC1(IJ)/2.0
3     CONTINUE
      II=N*2-1
      IJ=N*2
      NN=N*2-2
      RWA(NN)=RC1(N)
      RWA(II)=RC1(N)/2.0+PB(N)-RB(K)+RC2(N)
      PWA(IJ)=RC2(N)*2.0
      ELL=0.0
      DO 4 I=1,K
      II=I+1
      EL(I)=(RC2(I)-RC1(II)/2.0)/RC1(II)
      ELL=ELL+EL(I)*EL(I)
```

```
    4 CONTINUE
      ELL=SQRT(ELL/FLOAT(K))
      EL(N)=0.0
      M=N*2
      DO 33 I=1,M
      RWA(I)=2.0*PI*A(I)*RWA(I)
   33 CONTINUE
      IF(ANS.EQ.RNO)GOTO 95
      WRITE(2,115)
  115 FORMAT(///17X,'PROCESSING RESULTS')
      WRITE(2,109)
      WRITE(2,113) A(1),RWA(1),E(1),H(1),EL(1)
      DO 6 I=2,N
      II=I*2-2
      IJ=I*2-1
      WRITE(2,113) A(II),RWA(II),E(I),H(I).EL(I)
      WRITE(2,114) A(IJ),RWA(IJ)
    6 CONTINUE
  480 FORMAT(5X,'WARNING:  The observed error for spacing',I2' is greater
     1 than 1%'/,14X,'There may be an incorrect input data value'/)
      WRITE(2,114) A(M),RWA(M)
      WRITE(2,110) EEE
  110 FORMAT(//13X,29HR.M.S. Observational Error = ,F5.4)
      WRITE(2,111) HHH
  111 FORMAT(8X,34HR.M.S.  Offset Wenner Difference = ,F5.4)
      WRITE(2,112) ELL
  112 FORMAT(5X,37HR.M.S.  Potential Ladder Difference = ,F5.4/)
   95 DO 80 I=1,N
      IF(ABS(E(I))-0.01) 80,80,85
   85 WRITE(5,480)I
   80 CONTINUE
      DO 4539 I=1,M
      IF(RWA(I).LE.0.0)GOTO 4540
 4539 CONTINUE
      GOTO 9999
 4540 M=-1
      WRITE(5,9991)
      IF(ANS.EQ.YES)WRITE(2,9991)
 9991 FORMAT(/' NEGATIVE RESISTIVITIES CANNOT INTERPET'/)
 9999 RETURN
      END
```

Subroutine LG2LG3

This subroutine produces a log/log grid on the 380Z's VDU in high resolution graphics mode. The grid is two cycles vertically by three cycles horizontally and is drawn in red:

```
      SUBROUTINE LG2LG3
      BYTE CLT(16)
      CALL RESOL(0,2)
      CLT(1)=MIX(0,0,0)
      CLT(2)=MIX(7,0,0)
      CLT(3)=MIX(0,7,0)
      CLT(4)=MIX(0,0,2)
      CALL COLOUR(CLT)
      CALL PLOT(79,190,1)
      CALL LINE(79,31,1)
      CALL LINE(318,31,1)
      DO 20 I=1,3
      DO 21 J=1,10
      IX1=ALOG10(J*10.0**(I-1))*80+79
      CALL PLOT(IX1,31,1)
      CALL LINE(IX1,190,1)
   21 CONTINUE
   20 CONTINUE
      DO 22 I=1,2
      DO 23 J=1,10
      IX1=ALOG10(J*10.0**(I-1))*80+31
      CALL PLOT(79,IX1,1)
      CALL LINE(318,IX1,1
   23 CONTINUE
   22 CONTINUE
      RETURN
      END
```


Subroutine RESPLT

This subroutine plots the data points of an apparent resistivity curve on the 380Z's VDU using the high resolution graphics. The points are plotted in blue. The variables used by the subroutine are:

X - vector containing the electrode separations
Y - vector containing the apparent resistivities
M - the number of data points

Values returned by the subroutine are:

XST - the value of the leftmost log cycle
AM - the logarithm of the mean apparent resistivity value

```
      SUBROUTINE RESPLT(X,Y,M,XST,AM)
      DIMENSION X(M),Y(M)
      IF (XST) 32,31,32
   31 IF (AM) 32,30,32
   30 AM=ALOG10(Y(1))
      DO 20 I=2,M
      AM=AM+ALOG10(Y(I))
   20 CONTINUE
      AM=IFIX(AM/FLOAT(M))
      XST=ALOG10(X(1))
      IF (XST) 21,22,22
   21 XST=XST-1.0
   22 XST=AINT(XST)
   32 DO 23 I=1,M
      IX=80*(ALOG10(X(I))-XST)+79
      IY=80*(ALOG10(Y(I))-AM)
   26 IF (IY) 24,25,25
   24 IY=IY+160
   25 IF (IY-160) 27,27,28
   28 IY=IY-160
   27 IY=IY+31
      CALL PLOT(IX,IY,3)
   23 CONTINUE
      RETURN
      END
```

Subroutine DISKFL

The purpose of this subroutine is to open a disc file on a specified drive. Questions are displayed on the VDU and answers given by the operator through the keyboard are used to control the file that is opened. There is only one vector returned, FSP, which contains the name of the file.

```
      SUBROUTINE DISKFL(FSP)
      DIMENSION FSP(3)
      DATA YES,RNO/'Y   ','N   '/
   21 WRITE(5,402)
  402 FORMAT(' Give the number of the disc drive on which the data file
     1 is located'/' 1=A,2=B,3=C,4=D'/)
      READ(1,403)ID
      IF(ID.NE.1.AND.ID.NE.2.AND.ID.NE.3.AND.ID.NE.4) GOTO 21
  403 FORMAT(I1)
   66 WRITE(5,318)
  318 FORMAT(1X,'Enter the primary name of the data file'/)
      READ(1,319)FSP(1),FSP(2)
  319 FORMAT(2A4)
      CALL OPEN(6,FSP,ID)
      READ(6,ERR=33,END=33)DUMMY
      GOTO 99
   33 WRITE(5,44)FSP(1),FSP(2),FSP(3),ID
   44 FORMAT(' File ',2A4,'.',A4,' is empty or does not exist on drive ',
     1I1/' Do you wish to continue? (Y or N)'/)
      READ(1,55)ANS
   55 FORMAT(A1)
      IF(ANS.EQ.RNO)STOP
      IF(ANS.NE.YES)GOTO 33
      GOTO 21
   99 REWIND 6
      RETURN
      END
```

PROGRAM 4

This program is based on the work of Koefoed (1979a), but the algorithm that carries out the interpretation has been modified to improve the speed as recommended by Koefoed (1979b). In addition, routines have been added that allow the program to handle data with irregular electrode spacings and the input and output routines added to make the system user-friendly. Koefoed's program is designed for the Schlumberger array but this program is designed to handle both Wenner and Schlumberger array data. Ideally this should have been done by the application of a linear filter to convert Wenner array data to Schlumberger array data. However, the limit of the computer's memory made this impractical. Instead Wenner array data is merely shifted in the x-axis by the square root of 2. This approximation only produces unsatisfactory results if very large differences in the resistivity of adjacent layers are encountered.

The program uses an iterative interpretation method based on the method of steepest descents. This method, when applied to resistivity sounding interpretation, yields a fast improvement in the layer parameters. The field data are compared with data calculated from a layered model and if the agreement between the two data sets is unsatisfactory, the parameters of the layered model are adjusted in the direction of the "steepest descent" of the difference between the two and the procedure repeated. Unless a starting model is used which is a rough approximation to the final solution, the iterations either fail to converge or do so slowly. However, with experience the operator can provide an adequate starting model merely by examining the field data displayed on the VDU by the program.

The apparent resistivity curve is calculated from the layered model by firstly taking the transform and then applying a linear filter, as described by Ghosh

(1971). This method allows the calculations to be made rapidly with only a small error, generally less than 1%, due to the approximations involved.

The program is designed to be fully interactive and leads the operator through the necessary procedures. All data entries from the keyboard, or decisions for output requirements are made by the operator in response to prompts by the program. In order to minimise operator errors, all data entries are echoed by the program and must be verified by the operator before the program will proceed.

The first section of the program concerns data entry. Data is entered from disc files in an unformatted state. When the program is started the operator is asked to give the name of the data file and which disc drive it is located on. The program checks whether the file exists before proceeding. The data can be interpreted sequentially, i.e. the soundings are dealt with one after another, or soundings specified by the operator.

After a sounding has been selected for interpretation, the field curve is plotted on the VDU and the operator is offered the chance of editing the data. This can consist of deleting, adding or altering points. Experience has shown that editing usually consists of deleting obviously spurious data points.

The operator is asked to input the number of layers for the starting model and then the thicknesses of the layers, followed by their resistivities. The program cannot change the number of layers, only the layer parameter value. In addition the program provides the facility for assigning fixed values for the layer resistivities and/or the depth to the boundary between two layers by adding 100000 to the value. This allows the operator to incorporate knowledge of the local geology, such as the depth to a particular boundary derived from a borehole close to the site of the sounding.

The interpretation algorithm is then called by the program. The iterations are carried out five at a time, after which the operator is presented with the option of terminating the iteration. Otherwise the iteration is terminated either when 30 have been completed or when the difference between the field data and the values calculated from the layered model falls below a level entered earlier by the operator. Iteration also terminates if the adjustments made to the layered model fall below a level set within the program.

On completing the iterations the calculated curve and the field data are displayed on the VDU. The operator is given the option of printing the final layered model and the plot of the curves, before trying another starting model for that sounding, starting to interpret another set of field data, or stopping. An example of the results of an interpretation is shown below.

The data upon which the interpretation is based is listed, as is the final layered model. The R.M.S. relative error is the average difference between the field and calculated apparent resistivity curves and thus is a measure of the quality of the interpretation. The value that can be attained is dependant on the quality of the field data, but, the smaller the value, the better the interpretation. The number of trials is the number of iterations carried out to produce the final model.

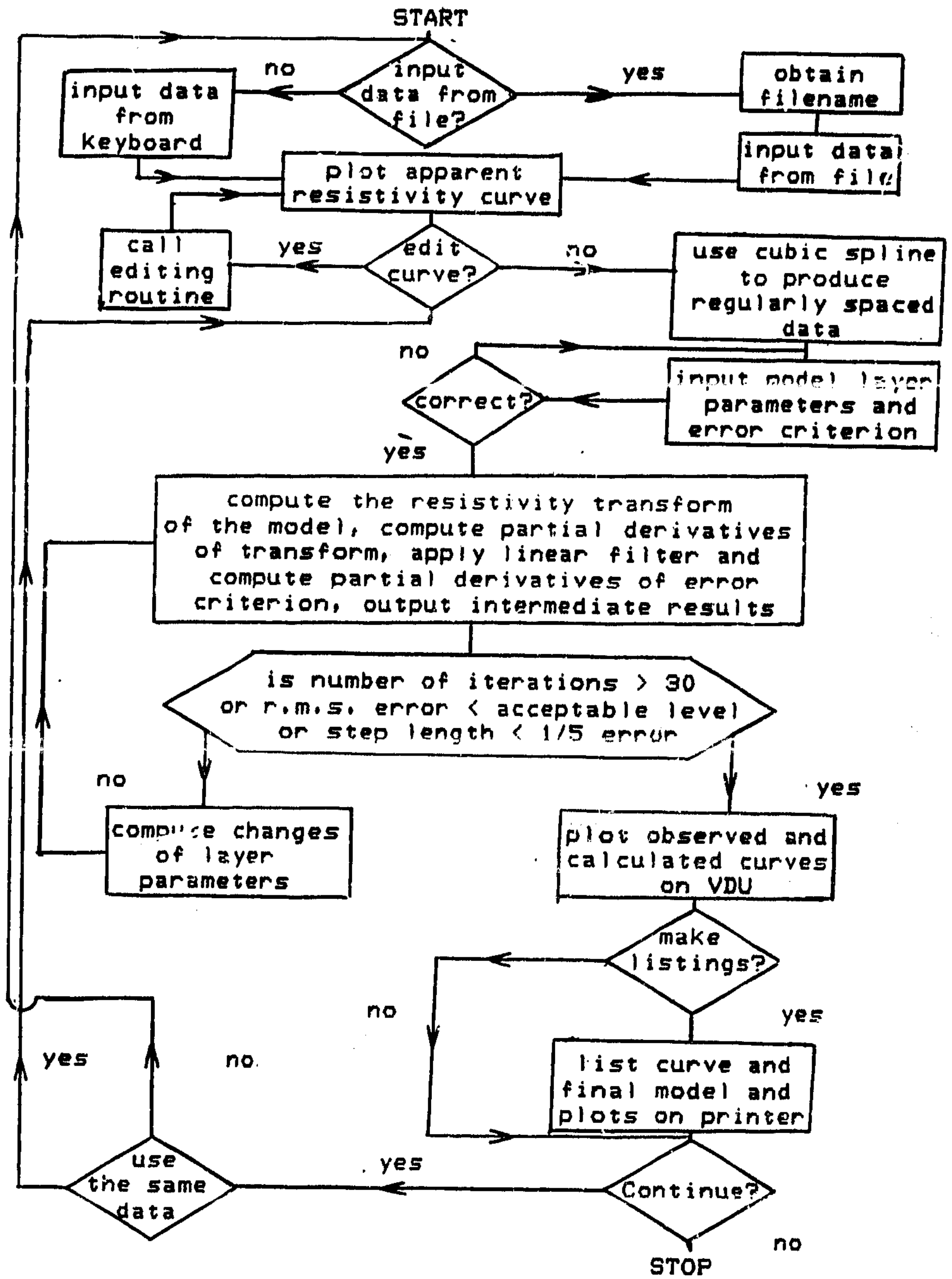
V.E.B. No. FP9

	FIELD CURVE DATA		INTERPRETED MODEL			
	Electrode Separation	Apparent Resistivity	T	Depth	Rho	Reflection Coeffts.
1	.5	35.66	[illegible]		34.1	
2	1.0	35.63				
3	1.5	41.60		1.42	--------	.3779
4	2.0	43.61				
5	3.0	47.40	3.40		75.6	
6	4.0	50.27				
7	6.0	48.97		4.82	--------	-.5338
8	8.0	43.68				
9	12.0	34.78	25.19		23.0	
10	16.0	27.95				
11	24.0	22.31		30.01	--------	-.6264
12	32.0	18.47				
					5.3	
					********	

R.M.S. Relative error = .0313
Maximum rel. error = .0801 at sample 2
Number of trials was 14

The main segment of the program handles setting up the input files and selecting the sounding to be interpreted before handing control to the interpretation algorithm.

```
      LOGICAL FLG
      INTEGER *4 M1,M2
      DIMENSION A(25),RWA(25)
      DATA YES,RNO/'Y   ','N   '/
      CALL RESOL(0,2)
 25   WRITE(5,500)
500   FORMAT(' Which electrode array?  (W)enner or (S)chlumberger'/)
      READ(1,10,ERR=25) WSA
      FLG=.TRUE.
700   CALL DISKFL
 30   WRITE(5,2)
  2   FORMAT(' Do you wish to start interpreting data from the'/,
     1' begining of the data  file? (Y or N)'/)
      READ(1,10)ANS
 10   FORMAT(A1)
      IF(ANS.EQ.RNO)GOTO 130
      IF(ANS.NE.YES)GOTO 30
      GOTO 40
220   FLG=.TRUE.
      WRITE(5,210)
210   FORMAT(' Do you want to interpret the next sounding? (Y or N)'/)
      READ(1,10)ANS
      IF(ANS.EQ.RNO)GOTO 130
      IF(ANS.NE.YES)GOTO 220
      GOTO 40
130   FLG=.FALSE.
 40   CALL SELECT(AIP1,AIP2,CRBJ,M1,M2,NZ,M,FLG,A,RWA)
      IF (N) 90,70,90
 70   WRITE(5,100)
100   FORMAT(' Do you wish to interpret any other soundings? (Y or N)'/)
```

```
      READ(1,10)ANS
      IF(ANS.EQ.RNO)GOTO 99
      IF(ANS.NE.YES)GOTO 70
      IF(N.EQ.0)GOTO 750
      GOTO 220
  750 WRITE(5,760)
  760 FORMAT(' Do you wish to interpret soundings from a different file',
     1'? (Y or N)'/)
      READ(1,10)ANS
      IF(ANS.EQ.RNO)GOTO 220
      IF(ANS.NE.YES)GOTO 750
      ENDFILE 6
      GOTO 700
   90 CALL VESINT(AIP1,AIP2,GRSJ,M1,M2,M,A,RWA,WSA)
      GOTO 70
   99 STOP
      END
```

Subroutine DISKFL

This subroutine opens a disc file on stream 6. The drive number and file name are given from the keyboard by the operator. If the file does not exist the operator is prompted to supply another name.

```
      SUBROUTINE DISKFL
      DIMENSION FSP(3)
      DATA FSP/'    ','    ','INT '/
      DATA YES,RNO/'Y   ','N   '/
   21 WRITE(5,402)
  402 FORMAT(' Give the number of the disc drive on which the data file
     1 is located'/' 1=A,2=B,3=C,4=D'/)
      READ(1,403)ID
      IF(ID.NE.1.AND.ID.NE.2.AND.ID.NE.3.AND.ID.NE.4) GOTO 21
  403 FORMAT(I1)
   66 WRITE(5,318)
  318 FORMAT(1X,'Enter the primary name of the data file'/)
      READ(1,319)FSP(1),FSP(2)
  319 FORMAT(2A4)
      CALL OPEN(6,FSP,ID)
      READ(6,ERR=33,END=33)DUMMY
      GOTO 99
   33 WRITE(5,44)FSP(1),FSP(2),FSP(3),ID
   44 FORMAT(' File ',2A4,'.',A4,' is empty or does not exist on drive ',
     1I1/' Do you wish to continue? (Y or N)'/)
      READ(1,55)ANS
   55 FORMAT(A1)
      IF(ANS.EQ.RNO)STOP
      IF(ANS.NE.YES)GOTO 33
      GOTO 21
   99 REWIND 6
      RETURN
      END
```

Subroutine SELECT

This subroutine selects resistivity sounding data from a sequential file. If the selection fails all the V.E.S. numbers on the file can be listed. The inputs are provided directly from the keyboard. The output variables are:

AIP1,AIP2 - the V.E.S. reference number
GRSJ - the UK grid reference letters
M1 - the grid reference easting
M2 - the grid reference northing
NZ - the azimuth of the array
N - the number of points on the apparent resistivity curve
FLG - if zero the next V.E.S. is taken
A - vector containing the electrode separations
RWA - vector containing the apparent resistivity curve

```
      SUBROUTINE SELECT(AIP1,AIP2,GRSJ,M1,M2,NZ,N,FLG,A,RWA)
      LOGICAL FLG
      INTEGER *4 M1,M2
      DIMENSION A(25),RWA(25)
      DATA BL,YES,RNO/'    ','Y   ','N   '/
      IF(FLG) GOTO 3
   50 REWIND 6
    9 WRITE(5,1)
    1 FORMAT(' Enter V.E.S. number you require'/)
      READ(1,4)RAIP1,RAIP2
    4 FORMAT(2A4)
      IF(RAIP1.EQ.BL.AND.RAIP2.EQ.BL)GOTO 9
    3 READ(6,END=11,ERR=77)AIP1,AIP2,GRSJ,M1,M2,NZ,N
      DO 25 I=1,N
      READ(6,ERR=77) A(I),RWA(I)
   25 CONTINUE
  600 IF(FLG) GOTO 12
      IF(AIP1.EQ.RAIP1.AND.AIP2.EQ.RAIP2) GOTO 12
      GOTO 3
   77 WRITE(5,78)
   78 FORMAT(' READ ERROR: May be corrupt data file!'/)
      STOP
   11 IF(FLG)GOTO 99
      WRITE(5,13)RAIP1,RAIP2
   13 FORMAT(' No such V.E.S. number as ',2A4/,
     1' Do you wish to list V.E.S. numbers on the file? (Y or N)'/)
      READ(1,14,ERR=11)ANS
   14 FORMAT(A1)
      IF(ANS.EQ.RNO)GOTO 20
      IF(ANS.NE.YES)GOTO 11
      REWIND 6
   34 ICOUNT=0
   31 READ(6,END=20)AIP1,AIP2
      IF(AIP1.EQ.BL.AND.AIP2.EQ.BL)GOTO 31
      WRITE(5,6)AIP1,AIP2
    6 FORMAT(1H ,2A4)
      ICOUNT=ICOUNT+1
      IF(ICOUNT.EQ.15)GOTO 33
      GOTO 31
   33 WRITE(5,55)
   55 FORMAT(/' Continue listing? (Y or N)'/)
      READ(1,51)ANS
   51 FORMAT(A1)
      IF(ANS.EQ.RNO)GOTO 20
      IF(ANS.NE.YES)GOTO 33
      GOTO 34
   20 WRITE(5,32)
   32 FORMAT(' Do you wish to continue selection (Y or N)'/)
      READ(1,14)ANS
      IF(ANS.EQ.RNO)STOP
      IF(ANS.NE.YES)GOTO 20
      GOTO 50
   99 N=0
      WRITE(5,333)
  333 FORMAT(' End of file, no more soundings'/)
   12 RETURN
      END
```

Subroutine VESINT

This subroutine is the interpretation algorithm. It also calls the editing and output subroutines. The input variables are:

- AIP1,AIP2 - the V.E.S. reference number
- GRSJ - the UK grid reference letters
- M1 - the grid reference easting
- M2 - the grid reference northing
- JZZ - the number of points on the apparent resistivity curve
- OAB - vector containing the electrode separations
- OAR - vector containing the apparent resistivity curve
- WSA - variable denoting the type of array

```
      SUBROUTINE VESINT(AIP1,AIP2,GRSJ,M1,M2,JZZ,OAB,OAR,WSA)
      INTEGER*4 M1,M2,MI,MJ
      DIMENSION RF(25),P(19),PL(19),G(19),CL(19),T(9),U(19),D(4,19)
     1 ,AB(25),RHO(10),DEPTH(10),RD(10),REFCO(10),W(75),AR(25),C(25)
     2,RMM(25),ZZB(25),OAB(25),OAR(25)
      DATA YES,RNO,CA,WAR,SAR/'Y   ','N   ','C   ','W   ','S   '/
      F=EXP(ALOG(10.)/6.)
      RA=.75
      SQT2=SQRT(2.0)
      RB=.6
      XST=0.0
      AM=0.0
      IFLG=0
      JFLG=0
  500 WRITE(5,2000)
 2000 FORMAT(1H1)
      CALL LG2LG3
      CALL RESPLT(OAB,OAR,JZZ,XST,AM)
C
  800 INUMB=0
      WRITE(5,1007) AIP1,AIP2
      DO 501 J=1,JZZ
      WRITE(5,1006) J,OAB(J),OAR(J)
      AB(J)=OAB(J)
      IF (WSA-SAR) 567,568,567
  567 AB(J)=AB(J)*SQT2
  568 AB(J)=ALOG10(AB(J))
      AR(J)=ALOG10(OAR(J))
  501 CONTINUE
      WRITE(5,1010)
      READ(1,1002) ANS
      IF(ANS.EQ.RNO)GOTO 505
      IF(ANS.NE.YES)GOTO 501
      GOTO 6
  505 IF (IFLG) 168,506,168
    6 CALL EDITXY(OAB,OAR,JZZ)
      IFLG=0
      GOTO 500
  506 XA=10.0**AB(1)
      WRITE(5,2000)
      JZ=JZZ
      JZ3=3*JZ
      CALL CUSP1(JZ,JZ3,AB,AR,C,W)
      IF (W(1)) 166,167,166
  167 IE=-1
      IFLG=1
      DO 160 I=1,25
      ZB=AB(1)+FLOAT(I-1)/6.0
      ZZB(I)=OAB(1)
      IF (I-1) 600,161,600
  600 ZZB(I)=ZZB(I)*10.0**(FLOAT(I-1)/6.0)
  161 RF(I)=CUSP2(IE,JZ,AB,AR,C,ZB)
      IE=1
      RF(I)=10.0**RF(I)
      IF (ZB-AB(JZ)) 160,160,162
  160 CONTINUE
```



```
  162 JZ=I-1
  168 WRITE(5,132)
      READ(1,1001,ERR=168)E
      IF(E.GT.1.0.OR.E.LT.0.0)GOTO 168
C
C     WRITE OUT LAST MODEL PARAMETRS
C
      IF(JFLG.EQ.0)GOTO 633
      WRITE(5,1234)
      WRITE(5,1237)(RD(IKI),RHO(IKI),IKI=1,IK)
      WRITE(5,1238)RHO(IL)
C
  633 WRITE(5,138)
      READ(1,1000,ERR=633)IL
      IF(IL.LE.1.OR.IL.GT.10)GOTO 633
      WRITE(5,3005)
 3005 FORMAT(/' If you do not want a layer parameter to be adjusted, add
     1 100000 to its value'/' When entering a layer parameter use up to
     210 digits including a decimal point'/)
      IK=IL-1
      IT=IK+IL
    8 DO 9 I=1,IK
  634 WRITE(5,139)I
      READ(1,1003,ERR=634)P(I)
      IF(P(I).LE.0.0)GOTO 634
    9 CONTINUE
      DO 10 I=IL,IT
      II=I-IL+1
  635 WRITE(5,140)II
      READ(1,1003,ERR=635)P(I)
      IF(P(I).LE.0.0)GOTO 635
   10 CONTINUE
      WRITE(5,3000)
 3000 FORMAT(/1X,'Layer',4X,'Thickness',2X,'Resistivity')
      DO 96 I=1,IK
      II=I+IK
      WRI[illegible],P(I),P(II)
 3001 FO[illegible],F14.2,F13.2)
   96 CON[illegible]
      WRIT[illegible]) IL,P(IT)
 3002 FORMAT([illegible],17X,F10.2)
  625 WRITE(5,137)
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 8
      IF(ANS.EQ.CA)GOTO 168
      IF(ANS.NE.YES)GOTO 625
   11 WRITE(5,2000)
      QV=99.0
      BT=0.0
      IC=0.0
      IM=0.0
      DO 13 I=1,IT
      U(I)=1.0
   13 CONTINUE
C
C     COMPUTE THE RESISTIVITY TRANSFORM FOR THE CURRENT LAYER
C     MODEL          T(I)
C
   16 XR=XA
      J=1
      Q=0.0
      BM=0.0
      DO 17 I=1,IT
      G(I)=0.0
   17 CONTINUE
   18 X=0.0105*XR
      LB=1
      DO 30 L=1,9
      X=X*F*F
   19 B=P(IT)
      IF (100000.0-B) 20,20,21
   20 B=B-100000.
   21 BC=B
      DO 29 K=1,IK
      ILK=IL-K
      ITK=IT-K
      DW=P(ILK)
```

```
      RW=P(ITK)
      IF (100000.0-DW) 22,22,23
   22 DW=DW-100000.0
   23 IF (100000.0-RW) 24,24,25
   24 RW=RW-100000.8
   25 TH=TANH(DW/X)
   29 B=(B+TH*RW)/(1.+TH*B/RW)
   30 T(L)=B
C
C      COMPUTE THE PARTIAL DERIVATIVES OF THE SAMPLE VALUES OF THE
C      RESISTIVITY TRANSFORM WITH RESPECT TO THE LAYER PARAMETERS
C      D(I,K)
C
   31 X=.1*X
      L=4
      IF (LB-1) 32,32,33
   32 X=.1*X
      DO 41 L=LB,4
   33 D(L,IT)=1.0
      B=PC
      DO 40 K=1,IK
      I=IL-K
      IW=IT-K
      IY=IW+1
      DW=P(I)
      RW=P(IW)
      IF (100000.0-DW) 34,34,35
   34 DW=DW-100000.0
   35 IF (100000.0-RW) 36,36,37
   36 RW=RW-100000.0
   37 TH=TANH(DW/X)
      BA=1.+B*TH/RW
      PA=(1.-TH*TH)/(BA*BA)
      D(L,I)=(RW-B*B/RW)/X
      D(L,IW)=TH*(1.+B*B/(RW*RW)+2.*TH*B/RW)/(BA*BA)
      DO 38 IZ=I,IK
      D(L,IZ)=PA*D(L,IZ)
   38 CONTINUE
      DO 39 IZ=IY,IT
      D(L,IZ)=PA*D(L,IZ)
   39 CONTINUE
      B=(B+TH*RW)/(1.+TH*B/RW)
   40 CONTINUE
      X=X*F*F
   41 CONTINUE
C
C     APPLY THE LINEAR FILTERS AND COMPUTE THE PARTIAL DERIVATIVES
C     OF THE ERROR CRITERION            G(I)
C
   43 RM=0.0148*T(1)-0.0814*T(2)+0.4018*T(3)-1.5716*T(4)+1.972*T(5)+0.18
     1 54*T(6)+0.1064*T(7)-0.0499*T(8)+0.0225*T(9)
      RMS(J)=RM
      DO 46 L=1,8
      T(L)=T(L+1)
   46 CONTINUE
      BA=1.0-RM/RF(J)
   47 DO 50 I=1,IT
      IF (P(I)-100000.0) 48,50,50
   48 B=0.402*D(1,I)-1.571*D(2,I)+1.972*D(3,I)+0.186*D(4,I)
      DO 49 L=1,3
      L1=L+1
      D(L,I)=D(L1,I)
   49 CONTINUE
      G(I)=G(I)+2.0*BA*B*P(I)/RM
   50 CONTINUE
C
C     BRANCHING PROGRAM
C
      Q=Q+BA*BA
      IF (BM*BM-  BA) 52,55,55
   52 BM=-BA
      JM=J
   55 J=J+2
      L=9
      LB=4
      XR=XR*F*F
      X=10.5*XR
```



```
      IF (J-JZ) 19,19,56
   56 IF ((J/2)*2-J) 57,58,57
   57 XR=XA*F
      J=2
      GOTO 18
   58 Q=SQRT(Q/FLOAT(JZ))
C
C      OUTPUT OF INTERMEDIATE RESULTS
C
      WRITE(5,142)IC,ST,Q
      INUMB=INUMB+1
      IF(INUMB.EQ.5)GOTO 960
      GOTO 1100
  960 INUMB=0
  990 WRITE(5,970)
  970 FORMAT(' Continue Iterations? (Y or N)'/)
      READ(1,1002)ANS
      WRITE(5,2000)
      IF(ANS.EQ.RNO)GOTO 115
      IF(ANS.NE.YES)GOTO 990
 1100 CONTINUE
C
      IF (IM) 60,63,60
   60 IF (E-Q) 61,115,115
   61 IF (IC-30) 62,115,115
   62 IF (E/5.0-ST) 63,115,115
   63 IC=IC+1
   68 B=0.0
C
C      COMPUTE THE CHANGES OF THE VALUES OF THE LAYER PARAMETERS
C      IN THE FIRST PHASE
C
      DO 70 I=1,IT
      G(I)=G(I)/FLOAT(JZ)
      B=B+G(I)*G(I)
   70 CONTINUE
      GR=SQRT(B)
      IF (IM) 89,75,89
   75 IF (Q-QV) 81,88,88
   81 ST=(Q*Q-0.9*E*E)/GR
      IF (0.5-ST) 83,84,84
   83 ST=0.5
   84 DO 85 I=1,IT
      GL(I)=G(I)
      PL(I)=P(I)
      PC=ST*G(I)/GR
      P(I)=P(I)*(1.0+PC)
   85 CONTINUE
      GOTO 100
   88 SR=ST
C
C      COMPUTE THE CHANGES OF THE VALUES OF THE LAYER PARAMETERS
C      IN THE SECOND PHASE
C
      GOTO 91
   89 IF (QV-Q) 90,90,91
   90 SR=ST
   91 B=0.0
      DO 98 I=1,IT
      A=P(I)
      IF (A-100000.0) 92,97,97
   92 IF (G(I)/GL(I)) 94,97,93
   93 PC=SR*U(I)*G(I)/GR
      P(I)=A*(1.0+PC)
      GOTO 95
   94 BA=GL(I)/(GL(I)-G(I))
      P(I)=PL(I)+BA*(A-PL(I))
      U(I)=RB*U(I)
      PC=P(I)/A-1.0
   95 B=B+PC*PC
      GL(I)=G(I)
   97 PL(I)=A
   98 CONTINUE
      ST=SQRT(B)
      IF (IM) 99,99,100
   99 IM=1
      SR=RA*SR
```

```
  100 DO 103 I=1,IK
C
C     ADJUST THE VALUES OF LAYER THICKNESSES
C
      IF (100000.0-P(I)) 101,101,103
  101 P(I)=0.0
      DW=0.0
      DO 102 K=1,I
      DW=DW+PL(K)-P(K)
  102 CONTINUE
      P(I)=DW
  103 CONTINUE
      QV=Q
      GOTO 16
  115 CONTINUE
C
C     OUTPUT THE FINAL RESULTS
C
      DO 120 I=1,IK
      IIK=I+IK
      DW=P(I)
      RW=P(IIK)
      IF (100000.0-DW) 116,116,117
  116 DW=DW-100000.0
  117 IF (100000.0-RW) 118,118,180
  118 RW=RW-100000.0
  180 RHO(I)=RW
  120 RD(I)=DW
      HW=P(IT)
      IF (100000.0-RW) 121,121,122
  121 RW=RW-100000.0
  122 RHO(IL)=RW
      DEPTH(1)=RD(1)
      DO 170 I=2,IK
      DEPTH(I)=DEPTH(I-1)+RD(I)
  170 CONTINUE
C
C     PLOT GRAPH
C
      WRITE(5,2000)
      CALL LOGLOG
      CALL MODPLT(ZZB,RMM,JZ,XST,AM)
      CALL RESPLT(OAB,OAR,JZZ,XST,AM)
C
C     WRITE MINI VERSION OF INTERPRETATION
C
      WRITE(5,1234)
 1234 FORMAT(' INTERPRETATION'//'    Thick     Rho')
      RD(1)=DEPTH(1)
      IF(IK.LT.2)GOTO 1235
      DO 1236 IKI=2,IK
      IK1=IKI-1
      RD(IKI)=DEPTH(IKI)-DEPTH(IK1)
 1236 CONTINUE
 1235 WRITE(5,1237)(RD(IKI),RHO(IKI),IKI=1,IK)
 1237 FORMAT(2X,F7.2,F8.1)
      WRITE(5,1238)RHO(IL)
 1238 FORMAT(9X,F8.1)
C
      WRITE(5,148) Q
      WRITE(5,149) BM,JM
      WRITE(5,150) IC
  403 WRITE(5,400)
  400 FORMAT(' Do you wish to print the interpretation (Y or N)'/)
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 402
      IF(ANS.NE.YES)GOTO 403
C
      WRITE(2,77)
   77 FORMAT(1H1)
      WRITE(2,144) AIP1,AIP2
      CALL PMOD(RHO,RD,DEPTH,REFCO,IK,IL,2,OAB,OAR,JZZ)
      WRITE(2,148)Q
      WRITE(2,149)BM,JM
      WRITE(2,150)IC
 8881 WRITE(5,4444)
 4444 FORMAT(' Do you wish a print of the model plot? (Y or N)'/)
```



```
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 402
      IF(ANS.NE.YES)GOTO 8881
      WRITE(2,6575)
 6575 FORMAT(/1X' PLOTTED RESULTS'/)
      CALL EPSON(0,2)
  402 CONTINUE
      JFLG=1
C
  185 WRITE(5,151)
  165 READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 152
      IF(ANS.NE.YES)GOTO 185
      GOTO 800
  152 CALL RESOL(0,2)
      RETURN
 1000 FORMAT(I2)
 1002 FORMAT(A1)
 1001 FORMAT(F10.0)
 1010 FORMAT(' Do you want to alter this sounding data? (Y or N)'/)
  132 FORMAT(1X,'Give acceptable fractional R.M.S. error'/'(up to 10 dig
     1its including a decimal point)'/)
  137 FORMAT(1X,'Correct? (Y or N)'/' (C) will cancel the
     1current instruction'/)
 1006 FORMAT(5X,I3,3X,F6.1,4X,F10.2)
 1007 FORMAT(/' V.E.S. No. ',2A4/)
 1003 FORMAT(F10.0)
  138 FORMAT(1X,'Give number of layers (up to 10):  ')
  139 FORMAT(1X,'Give the thickness of layer',I2':  ')
  140 FORMAT(1X,'Give the resistivity of layer',I2':  ')
  142 FORMAT(1X,I3,' Iterations,   Steplength =',F7.4,'   R.M.S. error =
     1',F7.4)
  144 FORMAT(/' V.E.S. No. ',2A4/)
  145 FORMAT(1X,'Resistivity  Thickness')
  146 FORMAT(2F10.2)
  147 FORMAT(F10.2)
  148 FORMAT(/,1X,'R.M.S. Relative error =',F8.4)
  149 FORMAT(1X,'Maximum rel. error =',F8.4,' at sample',I4)
  150 FORMAT(1X,'Number of trials was',I6/)
  151 FORMAT(1X,'Do you want to continue interpreting this sounding?'
     1' (Y or N)'/)
      END
```

Subroutine EDITXY

The purpose of this subroutine is to edit pairs of data points. They may be deleted, added or altered. The editing is done interactively by the operator through the keyboard using a question and answer procedure. The variables are:

- **X** – vector containing the x values of the data
- **Y** – vector containing the y values of the data
- **N** – the number of data pairs

```
      SUBROUTINE EDITXY(X,Y,N)
      DIMENSION X(25),Y(25)
      DATA YES,RNO,CA/'Y   ','N   ','C   '/
   22 WRITE(5,100)
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 20
      IF(ANS.NE.YES)GOTO 22
   21 WRITE(5,102)
      READ(1,105,ERR=21) J
      IF  (J) 22,23,24
   23 WRITE(5,108)
      READ(1,109) XP
      WRITE(5,111)
```

```
      READ(1,109) YP
      DO 25 J=1,N
      IF (X(J)-XP) 25,26,25
   26 IF (Y(J)-YP) 25,24,25
   25 CONTINUE
      WRITE(5,103) XP,YP
      GOTO 21
   24 WRITE(5,104)J,X(J),Y(J)
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 21
      IF(ANS.EQ.CA)GOTO 22
      IF(ANS.NE.YES)GOTO 24
   27 N1=N-1
      IF (N-J) 21,60,29
   29 DO 28 I=J,N1
      II=I+1
      X(I)=X(II)
   28 Y(I)=Y(II)
   60 CONTINUE
      N=N1
      GOTO 22
   20 WRITE(5,106)
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 30
      IF(ANS.NE.YES)GOTO 20
   31 WRITE(5,107)
      READ(1,105,ERR=31) J
      IF (J) 20,32,33
   32 WRITE(5,108)
      READ(1,109,ERR=32) XP
  636 WRITE(5,111)
      READ(1,109,ERR=636) YP
      DO 34 J=1,N
      IF (X(J)-XP) 34,35,34
   35 IF (Y(J)-YP) 34,33,34
   34 CONTINUE
      WRITE(5,103) XP,YP
      GOTO 31
   33 WRITE(5,104) J,X(J),Y(J)
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 31
      IF(ANS.EQ.CA)GOTO 20
      IF(ANS.NE.YES)GOTO 33
   36 IF (N-J) 31,62,38
   38 J1=N-J
      DO 37 I=1,J1
      NI=N-I+1
      NJ=NI+1
      X(NJ)=X(NI)
   37 Y(NJ)=Y(NI)
   62 CONTINUE
      N=N+1
      J1=J+1
  637 WRITE(5,108)
      READ(1,109,ERR=637) X(J1)
  638 WRITE(5,111)
      READ(1,109,ERR=638) Y(J1)
      GOTO 20
   30 WRITE(5,110)
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 40
      IF(ANS.NE.YES)GOTO 30
   41 WRITE(5,102)
      READ(1,105,ERR=41) J
      IF (J) 30,43,44
   43 WRITE(5,108)
      READ(1,109,ERR=43) XP
  639 WRITE(5,111)
      READ(1,109,ERR=639) YP
      DO 45 J=1,N
      IF (X(J)-XP) 45,46,45
   46 IF (Y(J)-YP) 45,44,45
   45 CONTINUE
      WRITE(5,103) XP,YP
      GOTO 41
   44 WRITE(5,104) J,X(J),Y(J)
```



```
      READ(1,1002)ANS
      IF(ANS.EQ.RNO)GOTO 41
      IF(ANS.EQ.CA)GOTO 30
      IF(ANS.NE.YES)GOTO 44
   47 WRITE(5,108)
      READ(1,109,ERR=47) X(J)
  640 WRITE(5,111)
      READ(1,109,ERR=640) Y(J)
      GOTO 30
   40 RETURN
  100 FORMAT(' Do you want to remove a point? (Y or N)'/)
 1002 FORMAT(A1)
  102 FORMAT(' Give the number of the point '/ ' if you do not Know the
     1number answer 0'/)
  105 FORMAT(I2)
  103 FORMAT(' No point with these values has been found ',2F7.2)
  104 FORMAT(' Is this the correct point? ',I3,2F8.2/' (Y or N)'/
     1 ' (C) will cancel the current instruction'/)
  106 FORMAT(' Do you want to add a point? (Y or N)'/)
  107 FORMAT(' Give the number of the preceeding point'/' if you do not
     1Know the number answer 0'/)
  108 FORMAT(' Give the electrode spacing of the point'/)
  109 FORMAT(F10.0)
  110 FORMAT(' Do you want to change a point? (Y or N)'/)
  111 FORMAT(' Give the apparent resistivity of the point'/)
      END
```

Subroutine CUBSPl

This subroutine calculates the values of a cubic spline for a function. The variables are:

- N - the number of points on the function
- N3 - the size of the working area vector, at least 3 times N
- X - vector containing the x values of the function
- Y - vector containing the y values of the function
- D - vector returned with the values of the spline at the x values of the function
- A - vector used as working space

```
      SUBROUTINE CUSP1(N,N3,X,F,D,A)
      DIMENSION X(N),F(N),D(N),A(N3)
      DO 5 I=2,N
      IF (X(I)-X(I-1)) 1,1,5
    1 WRITE(5,3) I
    3 FORMAT(' RETURN FROM CUSP1 BECAUSE X(',I3,') OUT OF ORDEF
      A(1)=1.0
      RETURN
    5 CONTINUE
      I=0
   30 I=I+1
      J=2
      IF (I-1) 6,10,6
    6 J=N-1
      IF (I.EQ.N) GOTO 10
      I1=I-1
      H1=1.0/(X(I)-X(I1))
      II=I+1
      H2=1.0/(X(II)-X(I))
      ITEMP=3*I-2
      A(ITEMP)=H1
      ITEMP=3*I-1
      A(ITEMP)=2.0*(H1+H2)
      ITEMP=3*I
      A(ITEMP)=H2
```

```
      TEMP=(H1*H1-H2*H2)
      II=I+1
      I1=I-1
      D(I)=3.0*(F(II)*H2*H2+F(I)*TEMP-F(I1)*H1*H1)
      GOTO 30
   10 J1=J-1
      H1=1.0/(X(J)-X(J1))
      JJ=J+1
      H2=1.0/(X(JJ)-X(J))
      ITEMP=3*I-2
      A(ITEMP)=H1*H1
      ITEMP=3*I-1
      A(ITEMP)=H1*H1-H2*H2
      ITEMP=3*I
      A(ITEMP)=-H2*H2
      TEMP=H1*H1*H1
      TEMP1=H2*H2*H2
      D(I)=2.0*(F(J)*(TEMP1+TEMP)-F(JJ)*TEMP1-F(J1)*TEMP)
      IF (I.LT.N) GOTO 30
      P=A(4)/A(1)
      A(5)=A(5)-P*A(2)
      A(6)=A(6)-P*A(3)
      D(2)=D(2)-P*D(1)
      I=2
   50 I=I+1
      K=3*I-4
      ITEMP=K+2
      P=A(ITEMP)/A(K)
      ITEMP=K+3
      ITEMP1=K+1
      A(ITEMP)=A(ITEMP)-P*A(ITEMP1)
      ITEMP=I-1
      D(I)=D(I)-P*D(ITEMP)
      IF (I.NE.N-1) GOTO 150
      ITEMP=K+5
      P=A(ITEMP)/A(K)
      ITEMP1=K+6
      ITEMP2=K+1
      A(ITEMP)=A(ITEMP1)-P*A(ITEMP2)
      ITEMP=K+7
      A(ITEMP)=A(ITEMP)
      ITEMP=N-2
      D(N)=D(N)-P*D(ITEMP)
  150 IF (I.LT.N) GOTO 50
      ITEMP=3*N-1
      D(N)=D(N)/A(ITEMP)
      I=2
   60 I=I+1
      J=N+2-I
      ITEMP=3*J
      ITEMP1=J+1
      TEMP=(D(J)-A(ITEMP)*D(ITEMP1))
      ITEMP=3*J-1
      D(J)=TEMP/A(ITEMP)
      IF (I.LT.N) GOTO 60
      D(1)=(D(1)-D(2)*A(2)-D(3)*A(3))/A(1)
      A(1)=0.0
      RETURN
      END
```

Function CUSP2

The purpose of this function is to evaluate a cubic spline. The input and output variables are:

- IX - a flag set to 0 on first entry of the function and -1 subsequently
- N - the number of points on the function
- U - vector containing the x values of the function
- S - vector containing the y values of the function

- D - vector returned with the values of the spline at the x values of the function
- X - x value of the point at which the spline is to be evaluated

```
      FUNCTION CUSP2(IX,N,U,S,D,X)
      DIMENSION U(N),S(N),D(N)
      IFLG=0
      IEPS=-19
      IF (X.LT.U(1)) GOTO 990
      IF (X.GT.U(N)) GOTO 991
      IF (IX.LT.0.OR.IFLG.EQ.0) GOTO 12
      J1=J+1
      IF (X.LE.U(J1)) GOTO 8
    1 J=J+1
      IF (J-25) 11,30,30
   30 WRITE(5,80)
   80 FORMAT(' J=25')
      GOTO 7
   11 J1=J+1
      IF (X-U(J1)) 31,31,1
   31 GOTO 7
   12 J=ABS(X-U(1))/(U(N)-U(1))*(N-1)+1
      ITEMP=N-1
      J=MIN0(J,ITEMP)
      IFLG=1
      IF (X.GE.U(J)) GOTO 11
    2 J=J-1
      IF (J.GT.0) GOTO 20
      WRITE(5,81)
   81 FORMAT(' J=0')
      GOTO 7
   20 IF (X.LT.U(J)) GOTO 2
    7 J1=J+1
      H=U(J1)-U(J)
      Q1=H*D(J)
      Q2=H*D(J1)
      SS=S(J1)-S(J)
      B=3.0*SS-2.0*Q1-Q2
      A=Q1+Q2-2.0*SS
    8 Z=(X-U(J))/H
      CUSP2=((A*Z+B)*Z+Q1)*Z+S(J)
      RETURN
  990 TEMP=AMAX1(ABS(U(1)),ABS(U(N)))
      IF (X-(U(1)-2.0**IEPS*TEMP)) 99,99,33
   33 J=1
      GOTO 7
  991 TEMP=AMAX1(ABS(U(1)),ABS(U(N)))
      IF (X-(U(N)+2.0**IEPS*TEMP)) 34,99,99
   34 J=N-1
      GOTO 7
   99 IFLG=0
      CUSP2=0.0
      RETURN
      END
```

Subroutine PMOD

The purpose of this subroutine is to list a geoelectrical layered model and an apparent resistivity curve on the printer. The input variables are:

- RHO - vector containing the layer resistivities
- RD - vector containing the layer thicknesses
- DEPTH - vector containing the layer boundary depths
- REFCO - vector containing the reflection coefficients
- NMI - the number of layers -1
- NLAY - the number of layers

NC - the output channel number
OAB - vector containing the apparent resistivity curve electrode spacings
OAR - vector containing the apparent resistivity curve resistivity values
JZZ - the number of points on the apparent resistivity curve

```
      SUBROUTINE FMOD(RHO,RD,DEPTH,REFCO,NMI,NLAY,NC,OAB,OAR,JZZ)
      DIMENSION RHO(NLAY),REFCO(NMI),RD(NMI),DEPTH(NMI)
      DIMENSION OAB(25),OAR(25)
      DO 3 I=1,NMI
      I1=I+1
      REFCO(I)=(RHO(I1)-RHO(I))/(RHO(I1)+RHO(I))
    3 CONTINUE
      RD(1)=DEPTH(1)
      IF (NMI-2) 2,6,6
    6 DO 4 I=2,NMI
      I1=I-1
      RD(I)=DEPTH(I)-DEPTH(I1)
    4 CONTINUE
    2 WRITE(NC,1040)
      NLAY4=NLAY*4
      DO 5 I=1,NLAY
      J=I*4-3
      J1=J+1
      J2=J+2
      J3=J+3
      IF(J.LE.JZZ)WRITE(NC,1050)J,OAB(J),OAR(J)
      IF(J.GT.JZZ)WRITE(NC,1080)
      IF(I.LT.NLAY)WRITE(NC,1060)RD(I),RHO(I)
      IF(I.EQ.NLAY)WRITE(NC,1055)RHO(I)
      IF(J1.LE.JZZ)WRITE(NC,1050)J1,OAB(J1),OAR(J1)
      IF(J1.GT.JZZ)WRITE(NC,1080)
      IF(J2.LE.JZZ)WRITE(NC,1050)J2,OAB(J2),OAR(J2)
      IF(J2.GT.JZZ)WRITE(NC,1080)
      IF(I.LT.NLAY)WRITE(NC,1070)DEPTH(I),REFCO(I)
      IF(I.EQ.NLAY)WRITE(NC,1065)
      IF(J3.LE.JZZ)WRITE(NC,1050)J3,OAB(J3),OAR(J3)
      IF(J3.GT.JZZ)WRITE(NC,1080)
    5 CONTINUE
      IF(JZZ.LE.NLAY4)GOTO 99
      JAA=NLAY4+1
      WRITE(NC,1050)(I,OAB(I),OAR(I),I=JAA,JZZ)
 1080 FORMAT(1H )
 1050 FORMAT(5X,I3,3X,F6.1,4X,F10.2)
 1060 FORMAT(1H+,39X,F7.2,12X,F8.1)
 1070 FORMAT(1H+,47X,F8.2,3X,8(1H-),3X,F7.4)
 1055 FORMAT(1H+,58X,F8.1)
 1065 FORMAT(1H+,58X,8(1H*))
 1040 FORMAT(//13X,'FIELD CURVE DATA',23X,'INTERPRETED MODEL'//,
     211X,'Electrode',4X,'Apparent'39X,'Reflection',
     3/10X,'Separation',3X,'Resistivity',6X,'Thickness',2X,
     4'Depth',6X,'Rho',7X,'Coeffts.'/)
   99 RETURN
      END
```

Subroutine MODPLT

This subroutine plots a continuous apparent resistivity curve that has been calculated from the layered model. The curve values are interpolated from the input values using the spline routines. The input variables are:

XX - vector containing the electrode separations
Y - vector containing the apparent resistivity values

N - the number of points on the apparent resistivity curve
XST - the value of the leftmost log cycle in the x-axis for plotting the curve on
AM - the logarithm of the mean apparent resistivity value

```
      SUBROUTINE MODPLT(XX,Y,N,XST,AM)
      DIMENSION XX(N),Y(N),C(25),W(75),X(25)
      DO 20 I=1,N
      X(I)=80.0*(ALOG10(XX(I))-XST)+79
      Y(I)=80.0*(ALOG10(Y(I))-AM)
   20 CONTINUE
      N3=N*3
      CALL CUBP1(N,N3,X,Y,C,W)
      IF (W(1)) 21,22,21
   22 IE=-1
      IX=INT(X(1))
      IXX=INT(X(N))
      DO 23 J=IX,IXX
      AXX=FLOAT(J)
      IY=INT(CUBP2(IE,N,X,Y,C,AXX))
      IF (IY) 24,25,25
   24 IY=IY+160
   25 IF (IY-160) 27,27,28
   28 IY=IY+160
   27 IY=IY+31
      IE=1
      CALL PLOT(J,IY,2)
   23 CONTINUE
      RETURN
   21 STOP
      END
```

REFERENCES

Barker, R.D., 1981. "The Offset System of Electrical Resistivity Sounding and its Use with a Multicore Cable", Geophysical Prospecting vol. 29, pp 128-144.

Ghosh, D.P., 1971. "Inverse Filter Coefficients for the Computation of Apparent Resistivity Standard Curves for a Horizontally Stratified Earth", Geophysical Prospecting vol. 19, pp 769-775.

Koefoed, O., 1979a. "Geosounding Principles, 1, Resistivity Sounding Measurements", publ. Elsevier, Netherlands.

Koefoed, O., 1979b. "Adaption of the Method of Steepest Descent to Resistivity Sounding Interpretation" paper presented at 41st meeting of EAEG at Hamburg.