

copy 3
BMR PUBLICATIONS COMPACTUS
(LENDING SECTION)

002779⁺



**DEPARTMENT OF
NATIONAL RESOURCES**

**BUREAU OF MINERAL RESOURCES,
GEOLOGY AND GEOPHYSICS**

Record 1977/8



**SCREENING AND AVERAGING MAGNETO-TELLURIC DATA PRIOR
TO ONE-DIMENSIONAL INVERSION**

by

R.F. Moore

The information contained in this report has been obtained by the Department of National Resources as part of the policy of the Australian Government to assist in the exploration and development of mineral resources. It may not be published in any form or used in a company prospectus or statement without the permission in writing of the Director, Bureau of Mineral Resources, Geology and Geophysics.

**BMR
Record
1977/8
c.3**

Record 1977/8

SCREENING AND AVERAGING MAGNETO-TELLURIC DATA PRIOR
TO ONE-DIMENSIONAL INVERSION

by

R.F. Moore

CONTENTS

	<u>Page</u>
Summary	
1. Introduction	1
2. Program SCREE	1
2.1 Introduction	1
2.2 Program description	2
2.3 Running SCREE	3
2.4 Examples	7
3. Program SCRAV	8
3.1 Introduction	8
3.2 Program description	8
3.3 Running SCRAV	9
3.4 Examples	12
4. References	13
Appendix 1: Program listing - SCREE	14
Appendix 2: Program listing - SCRAV	
Appendix 3: Description of SMW input data	

Figures

1. Eastern Officer Basin 1976 MT Survey:
Site 2 unscreened.
2. Eastern Officer Basin 1976 MT Survey:
Site 2 screened once.
3. Eastern Officer Basin 1976 MT Survey:
Site 2 screened twice.

TABLES

1. A sample print-out from SCRAV.
2. A sample listing of an output file from SCRAV.

Summary

Magneto-telluric rotated tensor apparent resistivity and phase data are often scattered. A process of screening that employs statistical measures of central tendency (for a skewed frequency distribution) has been incorporated in a program that will remove outliers from these data. This program (SCREE) is described here. Examples of screening are given. Experience has shown that the median is the best of the four statistical measures calculated.

Before a screened data set can be interpreted using the 1-dimensional inversion program SMW, it has to be averaged and formatted in a form suitable for input. A computer program (SCRAV) was written to do this, and this program is also described here.

1. Introduction

The processing of magneto-telluric (MT) data acquired in the field yields (see Kerr, in prep.):

- i. Cagniard (scalar) apparent resistivity and phase data (with coherency),
- ii. unrotated impedance tensor data (with predictability and tensor skew),
- iii. rotated impedance tensor data (with rotation angles, rotated coherency and tensor skew), and
- iv. tipping vector data.

Only the rotated tensor data, however, undergo further processing. These data are screened and averaged prior to being input into the 1-dimensional MT inversion program SMW. Two programs have been written to perform screening and averaging; these programs will be described in this report.

The first (SCREE) screens the data using statistical measures of central tendency. The second (SCRAV) screens (using the MT parameters rotated-coherency and tensor skew) and averages those data output from either SCREE or from the MT processing program. The averaged data output can then be input to the 1-dimensional inversion program.

2. Program SCREE

2.1 Introduction

Rotated tensor apparent resistivity and phase data are normally scattered. This is thought to be caused by instrumental noise and unfavourable local source-field conditions at the time the data are acquired. To remove outliers, a screening process using statistical measures of central tendency was developed. This process was incorporated in the program SCREE.

2.2 Program description

"SCREE" is a computer program written for BMR's Hewlett-Packard (HP) 2100/21MX computer system. It was written by the author in Fortran IV and runs under the Real Time Executive (RTE) system as a background, disc-resident program. As the program uses both the system communication routine EXEC and the file manager's FMP routines, it can only be run on an HP machine under a disc-based (DOS or RTE) system with FMGR. A listing of SCREE is given in Appendix 1.

The program is logically split up into two phases. Phase 1 divides the period axis into a user-defined number of bands (default = 10.0 per decade). The rotated tensor data are then read-in and each of the apparent resistivity data values is used to progressively build up a frequency distribution for both the tensor components for each period band. When all the data have been read-in, the following statistical measures are calculated for each period band:

- 1) arithmetic mean,
- 2) geometric mean,
- 3) median, and
- 4) mode.

Using one of these four measures (again user-defined), limits for apparent resistivity are calculated for each period band:

$$\text{Upper Limit} = \text{ANTILOG} (\text{LOG} (\text{central measure}) + \text{PCUT})$$
$$\text{Lower Limit} = \text{ANTILOG} (\text{LOG} (\text{central measure}) - \text{PCUT})$$

where PCUT (user defined) is some multiple of an apparent resistivity decade (default = 0.5).

Phase 2 screens the data previously used to build the frequency distributions. The input data are read again. For each two-component tensor, the period band is determined. If

there are less than a user-defined number of values in this band (default = 3) then the current tensor data are screened out. A tensor value that is screened out has its rotated coherency set to 0.0. The program will now screen the data using coherency and skew if the user specified he wants to (default = this screening not performed); it may be desirable to apply tighter limits on coherency and skew than those used by the processing programs (COH = 0.6, SKEW unbounded). These limits are normally applied, however, at a later time in either the plotting programs (see Moore, 1976) or in program SCRAV.

Finally, the two tensor apparent resistivity values are used for screening. If a value is within its respective limits, then its coherency is not altered. Data are thus read in, screened, and output; the new data set, except for those values whose coherency was set equal to 0.0, is identical to the input data set. This allows the user to screen a data set more than once by using the output from one run as input to a subsequent run. This is desirable because the central measures are generally not meaningful on the first run since outliers cause the frequency distribution to be skewed.

2.3 Running SCREE

SCREE is scheduled with parameters by the "ON" request at the operator's console of the 2100 or 21MX. All of the 5 allowable program parameters are used. The program scheduling request is

ON,SCREE,P1,P2,P3,P4,P5

where P1-5 are the program parameters used for program option control. If the user wishes to schedule SCREE from a terminal, he must type

*LG,SCREE

on the terminal keyboard and type in the five parameters (separated by commas) in response to the message

P1,P2,P3,P4,P5:

typed on the terminal printer. The significance of each of the five parameters is described below:

P1 (default = 1).

All communication between SCREE and the user is via the logical unit number (LU1) specified by P1. If P1 = 0, LU1 is set equal to 1. If P1 is less than 0, the data will be screened using tensor skew; LU1 is set equal to the absolute value of P1.

P2 (default = 2)

The logical unit number (LU2) of the rotated tensor input data is set equal to P2. The data input format is described in Moore (1976). If P2 = 0, LU2 is set equal to 2. If LU2 = 2, input will be from a FMGR disc file; the alternative is normally a magnetic tape unit. If the basic input option (BIO) is true (only if LU1 is the operator's console or a terminal), the user is expected to respond to the message.

NAME OF INPUT DISC FILE:

typed on LU1. If P2 is less than 0, the data will be screened using rotated coherency; LU2 is set equal to the absolute value of P2.

P3 (default = 2)

The logical unit number (LU3) of the screened rotated tensor data to be output is set equal to P3. If P3 = 0, LU3 is set equal to 2. If LU2 = 2, SCREE will create a new disc file to which it can output the screened data. If a file with the same

name already exists, it will be purged (only if its security code is 56). If BIO is true, the user is expected to respond to the message

NAME OF OUTPUT DISC FILE:

typed on LU1. If P3 is less than 0, the program will print (on LU = 6) all the statistical data calculated for each of the period bands together with the frequency distributions; LU3 is set equal to the absolute value of P3.

P4

This parameter specifies which statistical measure is to be used (TYPE) to calculate the apparent resistivity limits:

P4	TYPE	Measure used
0	0	median
1	1	mode
2	2	arithmetic mean
3	3	geometric mean
> 3	0	median

P5 (default = 0)

If $P5 \neq 0$, the user, depending upon what P5 is, can specify one or all of the screening parameters normally set by the program. If BIO is true, the user is expected to respond to the message(s) typed on LU1, otherwise the parameters are read in immediately from LU1. Each of the possible options is discussed below:

If $P5 < 0$, the user may specify the number of bands/decade to be used when dividing up the period and apparent resistivity axes. The maximum allowable values are 50.0 and 18.0 respectively. If P5 was negative, it is made positive.

If P5 = 0, all parameters are set by the program.

If P5 = 1 or if P5 > 6, no further input is expected on LU1.

This condition allows the user to make P5 negative and only have to input the number of bands/decade parameters.

If P5 = 2, coherency and skew limits may be input on LU1. Two coherencies and two skews are input together with a centre period (CT). If the period of a given data point is less than TC, the first coherency and skew limits are used; otherwise, the second pair of values is used. Hence, different coherency and skew cutoffs may be used at short and at long periods; this form of screening must, however, have been specified (see P1 and P2).

If P5 = 3, limits on the range of acceptable period values may be set. The user must specify the log (base 10) of the minimum and maximum periods allowable (integer numbers). For example, if LTMIN = -2, then the minimum period is 0.01 seconds. If the period of any rotated tensor data is outside the specified range, then that data is screened out.

If P5 = 4, PCUT may be specified.

If P5 = 5, NIBSIG may be specified. If when the data is being screened a data point occurs in a band with fewer than NIBSIG values then the point is screened out.

If P5 = 6, all of the above parameters are to be input on LU1. The default values (set internally) for each of these parameters are:

TNPD, RNP	D	= 10.0, 10.0 bands per decade
CT		= 10.0 seconds
COH. CUTOFFS		= 0.70, 0.70

Fig. 1

EASTERN OFFICER BASIN '76 SITE 02

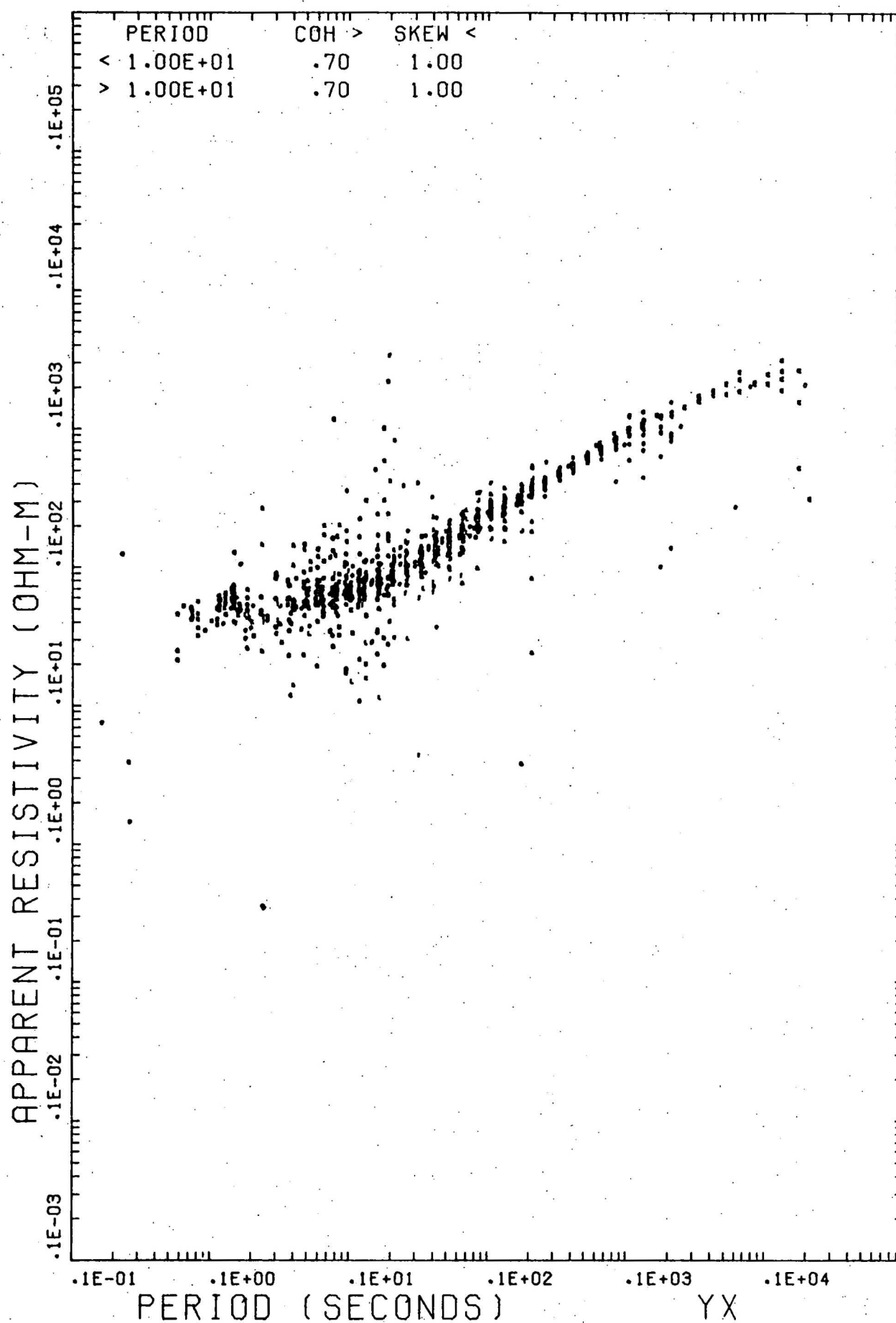


Fig. 2

EASTERN OFFICER BASIN '76 SITE 02

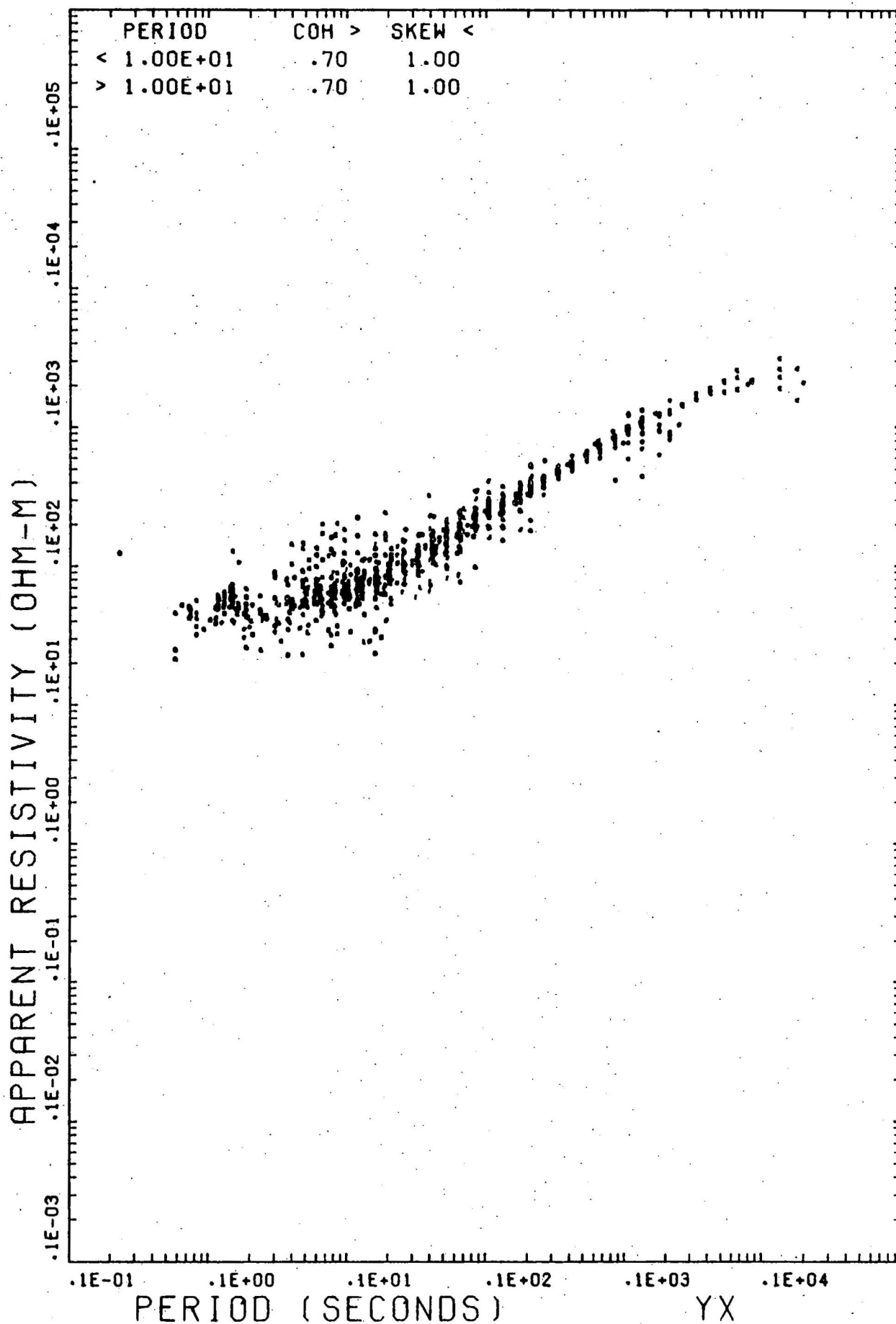
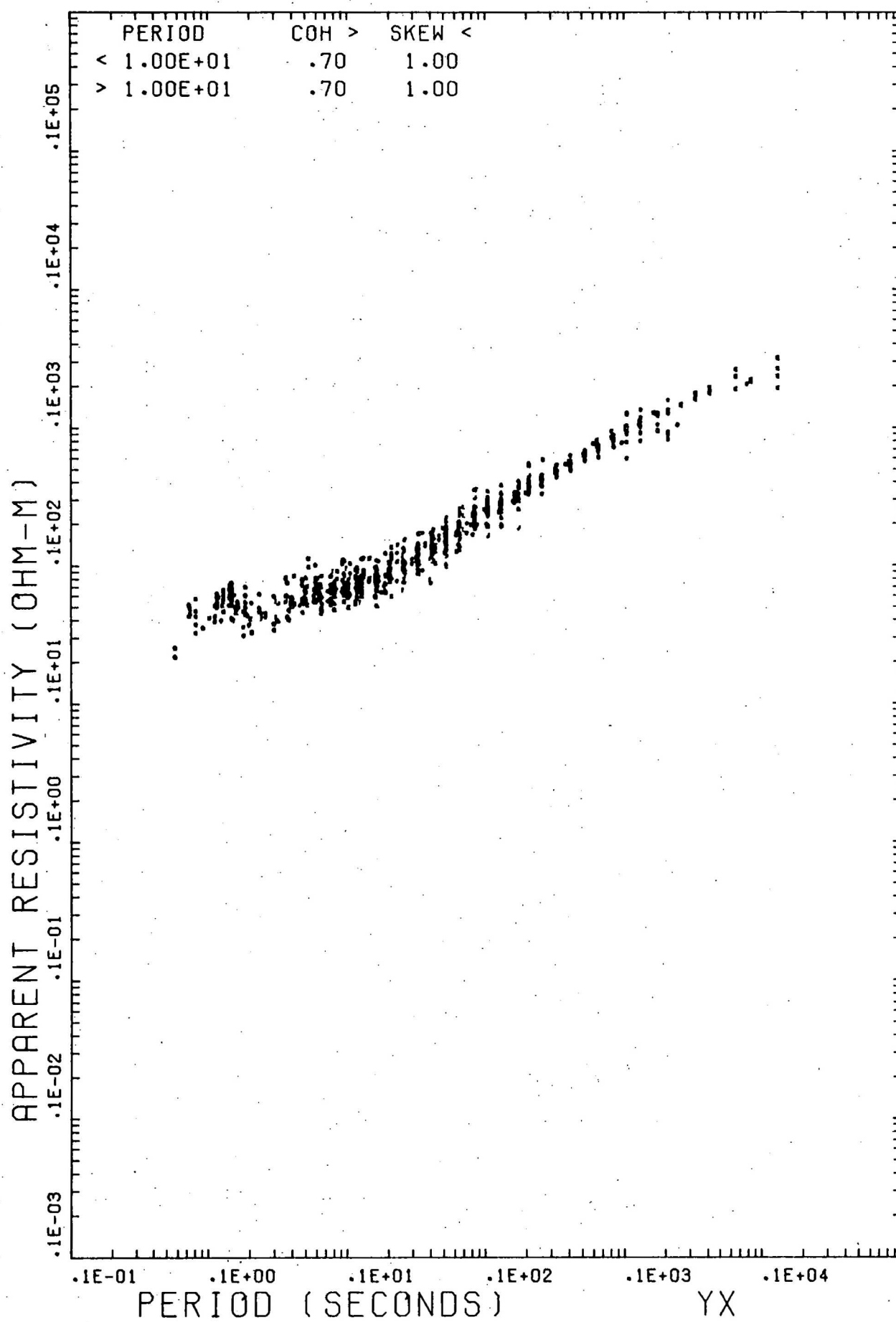


Fig. 3

EASTERN OFFICER BASIN '76 SITE 02



SKEW CUTOFFS = 1.00, 1.00
LTMIN, LTMAX = -2, 4
PCUT = 0.5 decades
NIBSIG = 3 points

2.4 Examples

The rotated tensor data for site 2 of the Eastern Officer Basin magneto-telluric survey, 1976, was screened twice using the median to calculate resistivity limits with no coherency or skew screening. PCUT was set equal to 0.5 and 0.2 respectively and all other parameters were set by the program. Plots of apparent resistivity for the three cases

- 1) before screening,
- 2) after screening once, and
- 3) after screening twice

are shown in Figures 1-3. The effects of screening can be easily seen.

3. Program SCRAV

3.1 Introduction

Before a rotated tensor data set (screened or unscreened) can be interpreted using inversion techniques, in particular the 1-dimensional program SMW, it must be averaged. A program was written to average rotated tensor data and format it for SMW. The program is called SCRAV and it is described here.

3.2 Program description

SCRAV is a computer program written for BMR's Hewlett-Packard (HP) 2100/21MX computer system. It was written by the author in Fortran IV and runs under the RTE system as a background, disc-resident program. As with SCREE, the program uses both the system communication routine EXEC and the file manager's FMP routines and hence can only be run on an HP machine under a disc-based (DOS or RTE) system with FMGR. A listing of SCRAV can be found in Appendix 2.

SCRAV operates in 3 modes. In the first mode, the program will process a rotated tensor data set and output an averaged data set only. In the second mode, a rotated tensor data set is input and an averaged data set is output but this time the data output can be input directly to SMW. In the third mode, the program will convert an averaged data set previously output from SCRAV to an SMW input deck; no averaging is done in this mode. Hence, new SMW input data sets can be readily generated from an averaged data set produced by SCRAV under mode 1 operation.

The actual process of averaging is relatively simple. As with SCREE, period is divided up into a user-defined number of bands. All those data with acceptable coherency and skew values are summed in their respective period bands. The data is then averaged using the geometric mean for apparent resistivity and

the arithmetic mean for phase. Thus, average values for both tensor components for each of the period bands are calculated.

3.3 Running SCRAV

SCRAV is scheduled with parameters by the "ON" request at the operator's console of the 2100 or the 21MX. All of the 5 allowable program parameters are used. The program scheduling request is

ON, SCRAV, P1, P2, P3, P4, P5

where P1-5 are the program parameters used for program option control. If the user wishes to schedule SCRAV from a terminal, he must type

*LG, SCRAV

on the terminal keyboard and type in the five parameters (separated by commas) in response to the message

P1, P2, P3, P4, P5:

typed on the terminal printer. The significance of each of these five parameters is described below;

P1 (default = 1)

All communication between SCRAV and the user is via the logical unit number (LU1) specified by P1. If P1 = 0, LU1 is set equal to 1. If P1 is less than 0, the program will run under mode 3; LU1 is set equal to the absolute value of P1.

P2 (default = 2)

For modes of operation 1 or 2, the logical unit number (LU2) of the rotated tensor data is set equal to P2. The data input format is described in Moore (1976). For mode 3, the

logical unit number (LU2) of the averaged data output from a previous SCRAV run (mode = 3) is set to P2. If P2 = 0, LU2 is set equal to 2. If LU2 = 2, input will be from an FMGR disc file; the alternative is normally a magnetic tape unit. If the basic input option (BIO) is true (only if LU1 is the operator's console or a terminal), the user is expected to respond to the message

NAME OF INPUT DISC FILE 1:

typed on LU1. If P2 is less than 0, the mode of operation is 1. If P1 was also less than 0, the sign of P2 is ignored and the mode of operation remains 3; LU2 is set equal to the absolute value of P2.

P3 (default = 2)

If the mode of the operation is 2 or 3, then SCRAV expects to find SMW input data on the logical unit number (LU3) specified by P3. If the mode of operation is 1, this logical unit is never referenced. Appendix 3 describes the information and formats of information expected in this file. If P3 = 0, LU3 is set equal to 2. If LU3 = 2, input will be from an FMGR disc file; the alternative is normally a paper tape reader. If BIO is true, the user is expected to respond to the message

NAME OF INPUT DISC FILE 2:

typed on LU1. If P3 is less than 0, only the first tensor component is output (if P4 is less than 0 also, both components are output as normal); LU3 is set equal to the absolute value of P3.

P4 (default = 2)

The logical unit number (LU4) of the averaged data to be output is set equal to P4. If P4 = 0, LU4 is set equal to 2. If LU4 = 2, SCRAV will create a new disc file to which it can output the averaged data. If a file with the same name already

exists, it will be purged (only if its security code is 56). If BIO is true, the user is expected to respond to the message

NAME OF OUTPUT DISC FILE:

typed on LU1. If P4 is less than 0, only the second component is output (if P3 is less than 0, both components are output as normal); LU4 is set equal to the absolute value of P4.

P5 (default = 0)

If $P5 \neq 0$, the user, depending upon what P5 is, can specify one or all of the parameters normally set by the program. If BIO is true, the user is expected to respond to the message(s) typed on LU1; otherwise the parameters are read in immediately from LU1. Each of the possible options is discussed below:

If $P5 < 0$, the user may specify the number of bands/decade to be used when dividing up period. The maximum number allowable is 90.0. If P5 was negative, it is made positive.

If $P5 = 0$, all parameters are set by the program.

If $P5 = 1$ or $P5 > 4$, no further input is expected on LU1. This condition allows the user to make P5 negative and only input the number of bands/decade parameter.

If $P5 = 2$, coherency and skew limits may be input on LU1 (see $P5 = 2$ for SCREE); this screening is unavoidable.

If $P5 = 3$, limits on the range of acceptable period values may be set (see $P5 = 3$ for SCREE).

If $P5 = 4$, all of the above parameters are to be input on LU1. The default values (set internally) for each of these parameters are:

NPD	= 3.0 bands per decade
CT	= 10.0 seconds
COH. CUTOFFS	= 0.70, 0.70
SKEW CUTOFFS	= 1.00, 1.00
LTMIN, LTMAX	= -2, 4

3.4 Examples

Those data shown in Figure 3 were averaged using SCRAV. The printer output for this run is shown in Table 1. A listing of the output disc file created (mode of operation 1) is shown in Table 2.

SCRAV : EASTERN OFFICER BASIN '76 SITE 02

TMIN = 1.000E-02 TMAX = 1.000E+04

CT = 10.00 CC = .700 .700 SC = 1.00 1.00

NUMBER OF FILES PROCESSED : 125

SOME OR ALL OF THE FOLLOWING DATA HAS BEEN WRITTEN TO LUN : 2

FILENAME : E03Q02

EASTERN OFFICER BASIN '76 SITE 02

C:** 3.00 -2 4 1.0 .70 .70 1.00 1.00

COMPONENT : 1

BAND	POINT	PERIOD	N	AP.RES.	SD	PHASE	SD
3	1	.6213E-01	11	.9947	1.353	-43.00	6.715
4	2	.1463	31	1.050	1.349	-44.39	7.139
5	3	.3162	30	1.055	1.374	-40.63	12.21
6	4	.6313	99	2.183	1.343	-39.34	13.21
7	5	1.463	110	3.417	1.299	-43.44	14.00
8	6	3.162	80	4.366	1.219	-51.91	10.72
9	7	6.313	73	4.236	1.198	-61.97	7.639
10	8	14.63	61	3.340	1.191	-70.75	13.32
11	9	31.62	35	1.800	1.326	-52.77	37.80
12	10	63.13	23	.8115	1.409	-67.13	26.45
13	11	146.8	18	.5622	1.317	-59.00	17.96
15	12	621.3	5	.3386	1.537	-53.40	22.61
16	13	1463	2	.6890	1.276	-49.00	22.00

COMPONENT : 2

BAND	POINT	PERIOD	N	AP.RES.	SD	PHASE	SD
3	1	.6313E-01	13	3.895	1.320	-43.47	4.433
4	2	.1463	67	5.203	1.222	-42.51	4.443
5	3	.3162	32	5.036	1.241	-43.44	8.762
6	4	.6313	132	6.457	1.196	-41.25	9.540
7	5	1.463	129	7.791	1.220	-32.79	12.53
8	6	3.162	92	11.49	1.247	-24.57	13.38
9	7	6.313	93	18.16	1.276	-13.46	6.844
10	8	14.63	83	28.78	1.263	-22.19	13.21
11	9	31.62	38	46.64	1.156	-21.71	10.53
12	10	63.13	32	69.71	1.136	-20.75	2.311
13	11	146.8	31	100.9	1.217	-23.09	6.723
14	12	316.2	8	151.0	1.191	-27.54	3.127
15	13	621.3	7	212.8	1.097	-34.00	5.297
16	14	1463	4	241.1	1.198	-37.03	9.189

Table 1

0001	13												
0002	EASTERN OFFICER BASIN '76 SITE 02				C: 1	3.00	-2	4	1.0	.70	.70	1.00	1.00
0003	.6813E-01	.1468	.3162	.6813	1.468	3.162	6.813	14.68					
0004	31.62	68.13	146.8	681.3	1463								
0005	.9947	1.050	1.055	2.183	3.417	4.256	4.236	3.340					
0006	1.830	.8115	.5622	.3386	.6890								
0007	-43.00	-44.39	-40.63	-39.84	-43.44	-51.91	-61.97	-70.75					
0008	-52.77	-67.13	-59.00	-53.40	-49.00								
0009	11.00	31.00	30.00	99.00	110.0	80.00	73.00	61.00					
0010	35.00	23.00	18.00	5.000	2.000								
0011	11.00	31.00	30.00	99.00	110.0	80.00	73.00	61.00					
0012	35.00	23.00	18.00	5.000	2.000								
0013	14												
0014	EASTERN OFFICER BASIN '76 SITE 02				C: 2	3.00	-2	4	1.0	.70	.70	1.00	1.00
0015	.6813E-01	.1468	.3162	.6813	1.468	3.162	6.813	14.68					
0016	31.62	68.13	146.8	316.2	681.3	1463							
0017	3.895	5.203	5.036	6.457	7.791	11.49	18.16	28.78					
0018	46.64	69.71	100.9	151.0	212.8	241.1							
0019	-43.47	-42.51	-43.44	-41.25	-32.79	-24.57	-18.46	-22.19					
0020	-21.71	-20.75	-23.09	-27.54	-34.00	-37.03							
0021	13.00	67.00	32.00	132.0	129.0	92.00	93.00	88.00					
0022	38.00	32.00	31.00	8.000	7.000	4.300							
0023	13.00	67.00	32.00	132.0	129.0	92.00	93.00	88.00					
0024	38.00	32.00	31.00	8.000	7.000	4.000							
0025	-1	-1											
0026													

4. References

KERR, D.W., in prep. Processing magneto-telluric data. Bur. Miner. Resour. Aust. Record (unpublished).

MOORE, R.F., 1976. Graphic presentation of magneto-telluric data. Bur. Miner. Resour. Aust. Record 1976/97 (unpublished).

Appendix 1

Program Listing - SCREE

FTN4

C

PROGRAM SCREE

C

C

LAST REVISION 14/02/77

C

C

/SCREE/ CALCULATES, FOR A USER-DEFINED NUMBER OF PERIOD BANDS, THE FOLLOWING 4 MEASURES OF CENTRAL TENDENCY FOR A GIVEN SET OF ROTATED TENSOR DATA OUTPUT FROM THE MT PROCESSING PROGRAM 'TENSCL':

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

- 1) ARITHMETIC MEAN;
- 2) GEOMETRIC MEAN;
- 3) MEDIAN AND
- 4) MODE.

SCREE THEN USES 1 OF THESE 4 (USER SPECIFIED) TO SCREEN THE DATA IN AN ATTEMPT TO REMOVE OUTLIERS.

ALL 5 OF THE 'RTE' PARAMETERS ARE USED TO SPECIFY OPTIONS ;THEIR SIGNIFICANCE IS DEFINED BELOW :

P1 = LUN BASIC INPUT (LU1)

- < 0 ALL DATA TO BE SCREENED USING TENSOR SKEW; OTHERWISE NOT.
LU1=-P1
- = 0 LU1=1
- > 0 LU1=P1 (NORMALLY 1,5,TERMINAL LUN, NOT 2)

P2 = LUN OF TENSCL-OUTPUT DATA (LU2)

- < 0 ALL DATA TO BE SCREENED USING COHERENCY; OTHERWISE NOT.
LU2=-P2
- = 0 LU2=2
- > 0 LU2=P2 (NORMALLY 13, 7 OR 2)

P3 = LUN OF SCREENED DATA OUTPUT FROM SCREE (LU3)

- < 0 PRINT AM,GM,MED AND MODE DATA; OTHERWISE SKIP.
LU3=-P3
- = 0 LU3=2
- > 0 LU3=P3 (NORMALLY 5 OR 2)

P4 = THE SCREENING PARAMETER OPTION (TYPE)

- < 0 TYPE=0
- = 0 USE MEDIAN FOR SCREENING
- = 1 USE MODE FOR SCREENING
- = 2 USE ARITHMETIC MEAN FOR SCREENING
- = 3 USE GEOMETRIC MEAN FOR SCREENING
- > 3 TYPE=0

P5 = OPTION FOR SCREENING DATA :

- < 0 INPUT NUMBER OF BANDS / DECADE ON LU1
- = 0 ALL PARAMETERS SET INTERNALLY
- /P5/= 1 OR
- > 6 SKIP FURTHER INPUT

= 2 OR
 = 6 INPUT CENTRE PERIOD, 2 COH. CUTOFFS AND
 2 SKEW CUTOFFS
 = 3 OR
 = 6 INPUT IFIX(ALOGT(TMIN AND TMAX)) ON LUI
 = 4 OR
 = 6 INPUT CUTOFF PARAMETER (PCUT);
 IF TYPE=0, RES DATA WITHIN +/- PCUT DECADES OF MEDIAN
 ARE ACCEPTED.
 = 5 OR
 = 6 INPUT NO. POINTS REQUIRED / T BAND FOR
 STATISTICAL SIGNIFICANCE

DEFAULTS :

NUMBER BANDS/DECADE = 10. FOR PERIOD AND RESISTIVITY.
 CENTRE PERIOD (CT) = 10.0 SECONDS
 COH. CUTOFFS = .70,.70
 SKEW CUTOFFS = 1.00,1.00
 LOG(10) TMIN, TMAX = -2,4
 PCUT = 0.5 DECADES
 NIBSIG = 3

SUBROUTINES :

FMP ROUTINES USED :

OPEN
 READF
 WRITE
 CLOSE
 CREAT
 PURGE

LOGICAL BIO,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG0
 INTEGER FNAME1(3),FNAME2(3),TYPE,STRK,SLASH(2)
 INTEGER DCB(144),DCB1(144),SECTR,TRACK,WORD,BLOCK(256)
 INTEGER OPT,BUFF(40),AFM1(6)
 REAL LM
 DIMENSION LUN(5),ISIZE(2),CC(2),SC(2),NAME(40),RMIN(2,300),
 * T1(99),NAV(2),COH(2),RES(2),PHZ(2),RMAX(2,300),
 * RESB(300),SUM(4),NIB(300),MODEP(126)
 EQUIVALENCE (LUI,LUN(1)),(LU2,LUN(2)),(LU3,LUN(3)),
 * (TYPE,LUN(4)),(OPT,LUN(5)),(NAME21,NAME(21)),
 * (NAME22,NAME(22))
 DATA BIO,FLAG1,FLAG2,FLAG3,FLAG4,FLAG5,FLAG0/7*.FALSE./
 DATA ISIZE,RMIN/32,0,0.01/
 DATA RNPD,TNPD,FLRMIN,RMAX/10.,10.,-2.,100000./
 DATA CT,CC,SC/10.,2*.7,2*1./
 DATA LTMIN,LTMAX/-2,4/
 DATA NIB,PCUT/300*0,0.50/
 DATA LSECTR,FM1,AFM1/-1,-1.,2H.,2H-.,2H10,2H0E,2H+0,2H1 /
 DATA BLOCK/60*0/
 DATA NIBSIG/3/
 DATA SLASH/2H(/,2H) /

```

C
C *** GET PARAMETERS AND SET FLAGS ACCORDINGLY
C
C     CALL RMPAR(LUN)
C
C     IF(TYPE.LT.100B) GO TO 1
C     SCREE INITIATED FROM A TERMINAL -
C     PICK UP NEW PARAMETERS
C     WRITE(LU1,300)
300 FORMAT("P1,P2,P3,P4,P5 :_-")
C     READ(LU1,*) LUN
C     BIO IS SET TRUE IF LU1 IS EITHER THE SYSTEM
C     CONSOLE OR A TERMINAL ON THE 2100
C     IF TRUE, THE USER IS PROMPTED FOR INPUT
C     ON LU1 ONLY; OTHERWISE NO PROMPTING
C     BIO=.TRUE.
C     IF(LU1.EQ.5.OR.LU1.EQ.7.OR.LU1.EQ.13) BIO=.FALSE.
1 IF(LU1.GE.0) GO TO 5
C     SCREENING TO BE DONE USING SKEW
C     FLAG3=.TRUE.
C     LU1=-LU1
5 IF(LU1.EQ.0) LU1=1
C     IF(LU1.EQ.1) BIO=.TRUE.
C     IF(LU2.GT.0) GO TO 15
C     IF(LU2.EQ.0) GO TO 10
C     SCREENING TO BE DONE USING COHERENCY
C     FLAG4=.TRUE.
C     LU2=-LU2
C     GO TO 15
10 LU2=2
15 IF(LU3.GT.0) GO TO 20
C     IF(LU3.EQ.0) GO TO 17
C     SCREE PRINTS T BAND STATISTICS
C     FLAG0=.TRUE.
C     LU3=-LU3
C     GO TO 20
17 LU3=2
C     IF FLAG1 IS TRUE, LU2 IS A DISC FILE
20 IF(LU2.EQ.2) FLAG1=.TRUE.
C     IF FLAG2 IS TRUE, LU3 IS A DISC FILE
C     IF(LU3.EQ.2) FLAG2=.TRUE.
C
C *** GET DISC FILE NAMES IF REQUIRED
C
C     IF(.NOT.FLAG1) GO TO 31
C     GET FILENAME OF TENS (OR SCREENED) DATA
C     IF(BIO) WRITE(LU1,310)
310 FORMAT("NAME OF INPUT DISC FILE :_-")
C     READ(LU1,320) FNAME1
320 FORMAT(40A2)
31 IF(.NOT.FLAG2) GO TO 35
C     GET FILENAME TO BE CREATED FOR OUTPUT
C     IF IT IS A DUPLICATE NAME, THE OLD FILE
C     WILL BE PURGED AND A NEW FILE CREATED IF
C     THE SEC. CODE WAS 56
C     IF(BIO) WRITE(LU1,340)
340 FORMAT("NAME OF OUTPUT DISC FILE :_-")
C     READ(LU1,320) FNAME2
C     CHECK 'TYPE'
35 IF(TYPE.LT.0.OR.TYPE.GT.3) TYPE=0
C     IF OPT=0 , NO SCREENING PARAMETERS ARE INPUT
C     IF(OPT.EQ.0) GO TO 60

```

```

      IF(OPT.GT.0) GO TO 40
C     INPUT THE NUMBER OF BANDS PER DECADE TO BE USED ;
C     IF NPD=3.0 AND TMIN=.01, TMAX=10000. THEN THERE
C     WILL BE 3.0*6 (DECADES) = 18 AVERAGING BANDS
38  IF(BIO) WRITE(LU1,350)
350  FORMAT("NUMBER OF POINTS/DECADE (FOR PERIOD,"
      * "RESISTIVITY (REAL)):_")
      READ(LU1,*) TNPD,RNPD
      OPT=-OPT
      IF(RNPD.LE.18.) GO TO 40
      WRITE(1,352)
352  FORMAT("SCREE - RNPD MUST BE .LE. 18.0")
      IF(.NOT.BIO) GO TO 81
      GO TO 38
C     IF OPT NOT 2,3,4,5 GO TO SCREENING
40  GO TO (60,50,57,58,59,50,60), OPT
C     INPUT CENTRE PERIOD AND COH., SKEW CUTOFFS
50  IF(BIO) WRITE(LU1,355)
355  FORMAT("CT,CC(2),SC(2) :_")
      READ(LU1,*) CT,CC,SC
      IF(OPT.EQ.2) GO TO 60
C     INPUT THE LOG (BASE 10) OF TMIN, TMAX (INTEGERS)
C     ONLY DATA IN (TMIN,TMAX) WILL BE USED FOR SCREENING
C     WHERE TMIN, TMAX ARE MINIMUM AND MAXIMUM PERIOD VALUES
57  IF(BIO) WRITE(LU1,360)
360  FORMAT("LOG TMIN, LOG TMAX (INTEGER) :_")
      READ(LU1,*) LTMIN,LTMAX
      IF(OPT.EQ.3) GO TO 60
C     INPUT SCREENING LIMIT PARAMETER.
C     DATA OUTSIDE +/- PCUT DECADES OF CHOSEN CENTRAL MEASURE
C     WILL BE SCREENED OUT.
58  IF(BIO) WRITE(LU1,362)
362  FORMAT("NO. OF DECADES FOR SCREENING LIMITS :_")
      READ(LU1,*) PCUT
      IF(OPT.EQ.4) GO TO 60
C     INPUT MINIMUM NUMBER OF POINTS / T BAND FOR
C     SIGNIFICANCE - DATA WITHIN A T BAND WITH FEWER
C     THAN NBSIG VALUES WILL BE SCREENED OUT.
59  IF(BIO) WRITE(LU1,363)
363  FORMAT("MINIMUM NO. POINTS/ T BAND FOR SIGNIFICANT STATS. :_")
      READ(LU1,*) NBSIG
60  TMIN=10.**LTMIN
      TMAX=10.**LTMAX
      NTHS=(LTMAX-LTMIN)*IFIX(TNPD+0.001)
      IF(NTHS.LE.300) GO TO 62
      WRITE(1,365) NTHS
365  FORMAT("SCREE - NTHS=",I4," > 300")
      GO TO 81
62  FLTMIN=LTMIN
C
C *** GET DISC SPACE (USED FOR STORING CUMULATIVE FREQ. DIST'NS)
C
      CALL EXEC(4,13,STRK,1,NSPT)
C
C *** ZERO DISC TABLE AREA
C
63  DO 600 I=0,12
      TRACK=STRK+I
      DO 600 J=1,NSPT
      NXST=J-1
      CALL EXEC(2,102B,BLOCK,64,TRACK,NXST)
600  CONTINUE

```

```

C
C *** BUILD DIST'N OR SCREEN DATA
C
1234 BUFF=LU2
C   GET THE DATA TITLE
   IF(.NOT.FLAG1) GO TO 65
   CALL OPEN(DCB,IERR,FNAM1,1)
   CALL ERROR(1,IERR,FNAM1,999)
   DO 500 I=1,40
500  BUFF(I)=2H
   CALL READF(DCB,IERR,BUFF,40,LEN)
   CALL ERROR(3,IERR,FNAM1,LEN)
   CALL CODE
65  READ(BUFF,320) NAME
   IF(FLAG5) GO TO 70
C   OUTPUT THE SCREENING PARAMETERS ETC
   WRITE(6,400) (NAME(I),I=1,20),TMIN,TMAX,RMIN(1),RMAX(1),
   *CT,CC,SC,TNPD,RNPD,PCUT,NIBSIG,TYPE
400  FORMAT(1H1//25X"SCREE : ",20A2//
   *10X"TMIN,TMAX =",2F12.4/10X"RMIN,RMAX =",2F12.4/
   *10X"CT,CC,SC :",5F11.4/10X"TNPD,RNPD =",2F11.4/
   *10X"PCUT,NIBSIG,TYPE =",F11.4,2I5/)
70  NBAND=0
   NFILE=0
75  NFILE=NFILE+1
C   GET A NEW TO (NEXT FILE)
   BUFF=LU2
   IF(.NOT.FLAG1) GO TO 80
   CALL READF(DCB,IERR,BUFF,30,LEN)
   CALL ERROR(3,IERR,FNAM1,LEN)
   CALL CODE
80  READ(BUFF,200) TO
200  FORMAT(5(F11.3,1X))
C   IF TO.LT.0 END OF DATA
   IF(TO.GT.0.) GO TO 82
   NFILE=NFILE-1
C   IF FLAG5 .TRUE. PASSES 1-2 COMPLETE
   IF(.NOT.FLAG5) GO TO 125
   IF(FLAG1) CALL CLOSE(DCB)
   IF(LU2.EQ.7.OR.LU2.EQ.13) REWIND LU2
   IF(.NOT.FLAG2) GO TO 77
   CALL WRITE(DCB1,IERR,AFM1,6)
   CALL ERROR(5,IERR,FNAM2,6)
   IF(FLAG2) CALL CLOSE(DCB1)
   GO TO 81
77  WRITE(LU3,200) FM1
81  WRITE(1,370)
370  FORMAT("SCREE OFF")
C   RELEASE TRACKS ASSIGNED TO SCREE
   CALL EXEC(5,-1)
C   END
   CALL EXEC(6)
C   OUTPUT TO IF PASS 2
82  IF(.NOT.FLAG5) GO TO 84
   BUFF=LU3
   IF(FLAG2) CALL CODE
   WRITE(BUFF,200) TO
   IF(.NOT.FLAG2) GO TO 84
   CALL WRITE(DCB1,IERR,BUFF,6)
   CALL ERROR(5,IERR,FNAM2,6)
C   WHICH BAND ?
84  IF(NBAND.NE.0) GO TO 85

```



```

NBAND=1
T1(1)=T0
IBAND=1
GO TO 92
85 DO 510 I=1,NBAND
   IF(T0.EQ.T1(I)) GO TO 90
510 CONTINUE
   NBAND=NBAND+1
   I=NBAND
   T1(NBAND)=T0
90 IBAND=I
92 CONTINUE
C SUM DATA FROM THIS FILE
  N=0
  NOB=0
  NAV(1)=0
  NAV(2)=0
C GET THE T,COH AND SKEW FOR
C THE CURRENT TENSOR VALUE
95 BUFF=L02
   IF(.NOT.FLAG1) GO TO 100
   CALL READF(DCH,IERR,BUFF,30,LEN)
   CALL ERROR(3,IERR,FNAM1,LEN)
   CALL CODE
100 READ(BUFF,200) T,COH,SKEW
   IF(T.GT.0.) GO TO 103
   IF(FLAG5) GO TO 96
C CURRENT FILE EXHAUSTED - OUTPUT OUT-OF-BOUNDS
C DATA IF ANY THEN GET NEW FILE
   IF(NOB.GT.0)
     *WRITE(6,420) NFILE,T0,N,NOB
420 FORMAT(5X," SCREE - FILE=",I4," T0=",G10.4,
     * " N=",I4," NUMBER OUT OF BOUNDS=",I5)
   GO TO 75
C OUTPUT END-OF-FILE MARKER
96 BUFF=L03
   IF(.NOT.FLAG2) GO TO 97
   CALL CODE
97 WRITE(BUFF,200)(FM1,I=1,4)
   IF(.NOT.FLAG2) GO TO 75
   CALL WRITE(DCH1,IERR,BUFF,24)
   CALL ERROR(5,IERR,FNAM2,24)
   GO TO 75
C GET TENSOR APP.RES., PHASE AND ROTATION ANGLE
103 IF(.NOT.FLAG1) GO TO 105
   CALL READF(DCH,IERR,BUFF,30,LEN)
   CALL ERROR(3,IERR,FNAM1,LEN)
   CALL CODE
105 READ(BUFF,200) RES,PHZ,ROT
   N=N+1
   IF(FLAG5) GO TO 111
C PASS 1 -
C CHECK THE CURRENT TENSOR VALUE IS WITHIN LIMITS
   NOB=NOB+1
   IF(T.LT.TMIN.OR.T.GT.TMAX) GO TO 95
   IF(RES(1).LT.RMIN.OR.RES(1).GT.RMAX) GO TO 95
   IF(RES(2).LT.RMIN.OR.RES(2).GT.RMAX) GO TO 95
   NOB=NOB-1
C IF A COH. VALUE = 0, THIS DATA FILE MUST BE OUTPUT FROM
C A PREVIOUS SCREE RUN. ANY POINT WITH COH=0. WILL
C BE EXCLUDED FROM FURTHER AVERAGING .
   IF(COH(1).EQ.0..OR.COH(2).EQ.0.) GO TO 95

```

```

C      INCLUDE CURRENT TENSOR APP.RES. IN APROPRIATE
C      FREQUENCY DIST'N BY UPDATING DISC TABLE
      TLOG=ALOGT(T)
      NTB=IFIX((TLOG-FLTMIN)*TNPD+1.001)
      NIB(NTB)=NIB(NTB)+1
      SECTR=4*NTB-3
      TRACK=STRK+SECTR/NSPT
      SECTR=MOD(SECTR,NSPT)-1
      DO 520 IC=1,2
      TEMP=ALOGT(RES(IC))
      NRB=IFIX((TEMP-FLRMIN)*RNPD+1.001)
      WORD=2*NRB
      IF(IC.EQ.1) WORD=WORD-1
      NX SCT=SECTR+WORD/64
      WORD=MOD(WORD,64)
      IF(WORD.NE.0) GO TO 119
      NX SCT=NX SCT-1
      WORD=64
C      DISC OPERATIONS ARE MINIMISED BY CHECKING WHETHER OR NOT
C      THE CURRENT DISC AREA TO BE UPDATED IS THE SAME AS THE
C      ONE IMMEDIATELY BEFORE AND HENCE PERFORMING DISC I/O
C      ONLY WHEN NECESSARY.
119  IF(IC.EQ.1) GO TO 120
      IF(NX SCT.EQ.LSCTR) GO TO 121
      CALL EXEC(2,102B,BLOCK,64,TRACK,LSCTR)
120  CALL EXEC(1,102B,BLOCK,64,TRACK,NX SCT)
121  BLOCK(WORD)=BLOCK(WORD)+1
      LSCTR=NX SCT
      IF(IC.EQ.2) CALL EXEC(2,102B,BLOCK,64,TRACK,LSCTR)
C
      NAV(IC)=NAV(IC)+1
520  CONTINUE
C      GO GET ANOTHER VALUE
      GO TO 95
C      PASS 2 -
C      N.B. VALUES SCREENED OUT HAVE THEIR COH SET TO 0.00
C
C      WHICH T BAND ?
111  NTH=IFIX((ALOGT(T)-FLTMIN)*TNPD+1.001)
C      IF NUMBER OF VALUES IN THIS BAND < NIBSIG, SCREEN THIS
C      THIS POINT OUT.
      IF(NIB(NTH).LT.NIBSIG) GO TO 116
      I=1
      IF(T.GT.TC) I=2
C      IF OPTED, SCREEN USING SKEW
      IF(.NOT.FLAG3) GO TO 112
      IF(SKEW.LE.SC(1)) GO TO 112
116  COH(1)=0.
      COH(2)=0.
      GO TO 117
C      IF OPTED, SCREEN USING COHERENCY
112  IF(.NOT.FLAG4) GO TO 114
      IF(COH(1).LT.CC(1)) COH(1)=0.
C      SCREEN DATA BY CHECKING APP.RES. VALUES ARE WITHIN
C      THE LIMITS PREVIOUSLY DEFINED
      IF(COH(2).LT.CC(1)) COH(2)=0.
114  DO 525 IC=1,2
      IF(RES(IC).LT.RMIN(IC,NTB).OR.RES(IC).GT.RMAX(IC,NTB))
      *COH(IC)=0.
525  CONTINUE
C
C      OUTPUT DATA

```

```

C
117 BUFF=LU3
   IF(FLAG2) CALL CODE
   WRITE(BUFF,200) T,COH,SKEW
   IF(.NOT.FLAG2) GO TO 113
   CALL WRITE(DCB1,IERR,BUFF,24)
   CALL ERROR(5,IERR,FNAM2,30)
113 IF(FLAG2) CALL CODE
   WRITE(BUFF,200) RES,PHZ,ROT
   IF(.NOT.FLAG2) GO TO 95
   CALL WRITE(DCB1,IERR,BUFF,30)
   CALL ERROR(5,IERR,FNAM2,30)
   GO TO 95
C   ALL INPUT DATA HAS BEEN INCLUDED IN THE FREQ. DIST. TABLES.
C   NOW CALCULATE, USING ONE OF THE CENTRAL MEASURES, FOR EACH
C   T BAND , AN UPPER AND LOWER LIMIT THAT WILL BE USED
C   FOR SCREENING THE DATA IN PASS 2.
C *** CALCULATE BAND PERIODS, MEAN RESISTIVITY
C *** AND PHASE DATA AND STANDARD DEVIATIONS
C
125 IF(LU2.EQ.7.OR.LU2.EQ.13) REWIND LU3
   IF(FLAG1) CALL CLOSE(DCB)
   IF(.NOT.FLAG2) GO TO 128
C   CREAT OUTPUT FILE
126 CALL CREAT(DCB1,IERR,FNAM2,ISIZE,3,56)
   IF(IERR.NE.-2) GO TO 127
   CALL PURGE(DCB1,IERR,FNAM2,56)
   CALL ERROR(0,IERR,FNAM2,999)
   WRITE(1,382) FNAM2
382 FORMAT("SCREE - FILE : ",3A2," PURGED")
   GO TO 126
127 CALL ERROR(2,IERR,FNAM2,888)
128 CT=ALOGT(CT)
C   IF SKEW AND COHERENCY IS NOT USED FOR SCREENING,
C   ASTERISKS WILL APPEAR IN THE HEADER RECORD OF THE
C   OUTPUT FILE INSTEAD OF THE LIMITING VALUES.
   IF(FLAG3) GO TO 129
   SC(1)=1.E+05
   SC(2)=SC(1)
129 IF(FLAG4) GO TO 131
   CC(1)=1.E+05
   CC(2)=CC(1)
C   INCLUDE SCREENING PARAMETERS IN HEADER RECORD
131 CALL CODE
   WRITE(NAM21,425) LTMIN,LTMAX,CT,CC,SC,TNPD,RNPD,
   *PCUT,NIBSIG,TYPE
425 FORMAT(2I3,F4.1,2F3.2,2F4.2,F4.1,F5.1,F3.1,2I2)
   WRITE(6,427) NFILE
427 FORMAT(/10X"NUMBER OF FILES PROCESSED :",I3)
   WRITE(6,430) LU3
430 FORMAT(/10X"THOSE DATA SCREENED HAVE BEEN",
   * " WRITTEN TO LUN :",I3)
   IF(FLAG2) WRITE(6,432) FNAM2
432 FORMAT(41X"FILENAME : ",3A2)
   WRITE(6,433) NAME
433 FORMAT(/10X"HEADER RECORD :",40A2)
   NTBS=(LTMAX-LTMIN)*IFIX(TNPD+0.001)
   NRBS=7*IFIX(RNPD+0.001)
C   SET UP A TABLE OF BAND-CENTRE RESISTIVITIES
   TEMP=1./RNPD
   RESB(1)=FLRMIN+0.5*TEMP

```

```

DO 530 I=2,NRBS
RESB(I)=RESB(I-1)+TEMP
RESB(I-1)=10.**RESB(I-1)
530 CONTINUE
RESB(NRBS)=10.**RESB(NRBS)
IF(FLAGO) WRITE(6,434)
434 FORMAT(1H1///30X"APPARENT RESISTIVITY STATS FOR THIS RUN",
*" FOLLOW :"/
*" PERIOD CENTRE N ARITHMETIC COEFF. OF GEOMETRIC",
*" COEFF. OF MEDIAN MEDIAN MODAL MODE ",
*" RESISTIVITY-LIMITS"/
*" BAND PERIOD MEAN VARIATION MEAN ",
*" VARIATION BAND",13X"BAND",13X"LOWER UPPER"/
*55X"(LOG.MEAN)"/
*1X,126(1H-)/" APPARENT RESISTIVITY FREQUENCY DISTRIBUTION "/)

```

```

C
C CALCULATE FOR EACH T BAND THE 4 CENTRAL MEASURES AND
C ESTABLISH THE UPPER AND LOWER LIMITS FOR EACH OF THESE
C BANDS USING 1 OF THESE MEASURES AND PCUT
C

```

```

DO 560 I=1,NTHS
C IF NO VALUES IN THIS BAND, SKIP
IF(NIB(I).EQ.0) GO TO 560
C GET I'TH FREQ. DIST'N FROM DISC
SECTR=4*I-3
TRACK=STRK+SECTR/NSPT
SECTR=MOD(SECTR,NSPT)-1
DO 535 J=0,3
WORD=64*J+1
CALL EXEC(1,1028,BLOCK(WORD),64,TRACK,SECTR+J)

```

```

535 CONTINUE
DO 550 IC=1,2
C CALCULATE 4 MEASURES AND DATA LIMITS FOR COMPONENT 'IC'
DO 540 J=1,4
SUM(J)=0.

```

```

540 CONTINUE
MODE=0
MODET=0
MED=NIB(I)/2
MEDT=0
MEDG=0
FNIB=0.
DO 545 J=1,NRBS
WORD=J*2
IF(IC.EQ.1) WORD=WORD-1
N=BLOCK(WORD)
IF(N.EQ.0) GO TO 545
IF(N.LT.MODET) GO TO 122
IF(N.GT.MODET) MODEN=0
MODEN=MODEN+1
MODEP(MODEN)=J
MODE=J
MODET=N

```

```

122 IF(MEDG.NE.0) GO TO 123
MEDT=MEDT+N
IF(MEDT.GT.MED) MEDG=J

```

```

123 FN=FLOAT(N)
FNIB=FNIB+FN
SUM(1)=SUM(1)+RESB(J)*FN
SUM(2)=SUM(2)+RESB(J)*RESB(J)*FN
RLOG=ALOGT(RESB(J))
SUM(3)=SUM(3)+RLOG*FN

```

```

SUM(4)=SUM(4)+RLOG*RLOG*FN
545 CONTINUE
C IF THE MODAL FREQUENCY IS SHARED BY 2 OR MORE BANDS THEN
C THE BAND NEAREST BUT GREATER THAN THE MEDIAN WITH THAT
C FREQUENCY IS SET TO BE THE MODAL BAND .
IF(MODEN.EQ.1) GO TO 135
DO 543 J=1,MODEN
IF(MODEP(J).GE.MEDG) GO TO 134
543 CONTINUE
C N.B. STANDARD DEV'NS CAN BE CALCULATED FOR THE ARITH.MEAN AND
C THE LOG. MEAN BUT NOT FOR THE GEOM. MEAN .
J=MODEN
134 MODE=MODEP(J)
135 AM=SUM(1)/FNIB
LM=SUM(3)/FNIB
GM=10.**LM
SDA=SQRT(SUM(2)/FNIB-AM*AM)
SDL=SQRT(SUM(4)/FNIB-LM*LM)
CVA=SDA/AM
CVL=SDL/LM
C DEPENDING ON 'TYPE' CALCULATE THE DATA LIMITS
GO TO (140,145,150,155), TYPE+1
140 RLOG=ALOGT(RESB(MEDG))
GO TO 160
145 RLOG=ALOGT(RESB(MODE))
GO TO 160
150 RLOG=ALOGT(AM)
GO TO 160
155 RLOG=LM
160 RMIN(IC,1)=10.**(RLOG-PCUT)
RMAX(IC,1)=10.**(RLOG+PCUT)
T=10.**((FLTMIN+FLOAT(1)/TNPD-1.)/(2.*TNPD))
C IF OPTED PRINT OUT VALUES
IF(.NOT.FLAG0) GO TO 550
WRITE(6,450)
*1,T,NIB(1),AM,CVA,GM,CVL,MEDG,RESB(MEDG),MODE,RESB(MODE),
*RMIN(IC,1),RMAX(IC,1)
450 FORMAT(15,2XG10.4,14,4(1XG10.4),4X14,3XG10.4,
*14,2XG10.4,2(1XG10.4))
550 CONTINUE
IF(.NOT.FLAG0) GO TO 560
C IF OPTED OUTPUT FREQ. DIST'NS.
N=NRBS*2
WRITE(6,451)
451 FORMAT(1X,126(1H_)/1X,12("1234567890"),"123456")
DO 551 IC=1,2
CALL CODE
WRITE(MODEP,452)(BLOCK(J),J=IC,N,2)
452 FORMAT(126I2)
WRITE(6,453)(MODEP(J),J=1,NRBS)
453 FORMAT(1X,126A1)
WRITE(6,454)(MODEP(J),J=1,NRBS)
454 FORMAT(1X,126R1)
551 CONTINUE
WRITE(6,SLASH)
560 CONTINUE
C SET FLAG5 .TRUE. AND GO TO PASS 2 AFTER OUTPUTTING
C THE NEW DATA FILE HEADER TO LU3.
BUFF=LU3
FLAG5=.TRUE.
IF(FLAG2) CALL CODE
WRITE(BUFF,320) NAME

```

```

IF(.NOT.FLAG2) GO TO 1234
CALL WRITE(DCB1,IERR,NAME,40)
CALL ERROR(5,IERR,FNAM2,40)
GO TO 1234
END
SUBROUTINE ERROR(IC,IERR,FNAME,LEN)

```

```

C
C ERROR DETECTS 'FMP' CALL ERRORS OR EOF/SOF ERRORS
C ON 'READF' CALLS WHERE
C IC IS THE USER DEFINED CODE
C IERR IS THE ERROR CODE RETURNED BY THE FMP CALL,
C FNAME IS THE FILE BEING ACCESSED AND
C LEN IS THE LENGTH OF THE RECORD READ
C (< 0 IMPLIES READ THROUGH EOF)

```

```

C IF IERR<0 OR LEN<0, THEN SCREE IS ABORTED AFTER
C A MESSAGE IS SENT TO LU 1; THE FOLLOWING DATA IS
C OUTPUT :

```

```

C FILENAME
C FMP ERROR CODE
C THE NUMBER OF THE CALL TO ERROR
C THE PROGRAM CODE PASSED TO ERROR AND
C THE LENGTH OF THE DISC RECORD READ (READF ONLY)

```

```

C THE PROGRAM CODES ARE :

```

- 0 - ERROR ON PURGE REQUEST
- 1 - ERROR ON OPEN REQUEST
- 2 - ERROR ON CREAT REQUEST
- 3 - ERROR ON READF REQUEST IN 'SCREE' FROM INPUT FILE
- 4 - UNDEFINED
- 5 - ERROR ON WRITE REQUEST IN 'SCREE'

```

C
C INTEGER FNAME(3)
C DATA ICALL/0/
C ICALL=ICALL+1
C IF(IERR.GE.0.AND.LEN.GE.0) RETURN
C WRITE(1,300) FNAME,IERR,ICALL,IC,LEN
300 FORMAT("SCREE FMP ERROR : FILE '"',3A2,'" ERROR =",16/
* "CALL =",13," CODE =",13," LENGTH =",16/
* "SCREE ABORTED")
C CALL EXEC(6)
C END
C ENDS

```

SYSTEM BULLETIN 23 --- 1/ 6/76. USE -SYSBULL,LFN. OR -SYSBULL. 3333333

HH.MM.SS CPU SECOND ORIGIN

15.57.24	CYL.	ED VERSION 06/07/77		
15.59.57	00000.018 CYB.	-MOB,P3000.		
15.59.57	00000.021 JOB.	-DISPOSE,OUTPUT,ST=RIOCP,*PR=CZR.		
15.59.57	00000.022 JOB.	-COMMENT. BOB MOORE X228		
15.59.57	00000.022 LOD.	-COPY.		
15.59.59	00000.174 USR.	UT031 - EOI ENCOUNTERED		
15.59.59	00000.175 USR.	UT034 - EOR - 664	EOS - 1	EOP - 1
15.59.59	00000.176 CYB.			
15.59.59	00000.176 CYB.	MDMRAPF	8.000	BLOCKS
15.59.59	00000.177 CYB.	DFILEPF	0.143	BLOCKS
15.59.59	00000.177 CYB.	OUTPUT	7.036	BLOCKS
15.59.59	00000.177 CYB.	SC050 - 000007 SC/LC	SWAPS	
15.59.59	00000.177 CYB.			
15.59.59	00000.178 CYB.	JOB	0.179	SEC
15.59.59	00000.178 CYB.	USR	0.058	SEC
15.59.59	00000.178 CYB.	URC	0.893	SS
15.59.59	00000.178 CYB.	SCM	0.324	KWSEC
15.59.59	00000.179 CYB.	RMS	0.002	MWSEC
15.59.59	00000.179 CYB.	I/O	0.007	MW
15.59.59	00000.179 CYB.	TOTAL	1.103	SS
15.59.59	00000.179 CYB.	PRI	3.000	K
15.59.59	00000.179 CYB.	RATE	0.500	\$/SEC
15.59.59	00000.180 CYB.	CHARGE		0.55 \$

Appendix 2

Program Listing SCRAV

C

PROGRAM SCRAV

C

C

LAST REVISION 16/02/77

C

C

'SCRAV' SCREENS AND AVERAGES MAGNETO-TELLURIC APPARENT RESISTIVITY AND PHASE DATA OUTPUT FROM THE MT PROCESSING PROGRAM 'TENSC' OR FROM THE SCREENING PROGRAM 'SCREE' PRIOR TO INPUT TO THE MT INVERSION PROGRAM 'SMW'.

C

C

C

C

C

C

C

C

C

C

ALL 5 OF THE 'RTE' PARAMETERS ARE USED TO SPECIFY OPTIONS : THEIR SIGNIFICANCE IS DEFINED BELOW :

C

P1 = LUN BASIC INPUT (LU1)

C

C

C

C

C

C

C

C

< 0 PRODUCE INVERSION RUN INPUT FROM PREVIOUS 'SCRAV' OUTPUT (ON LU2) AND INVERSION MODEL DATA INPUT (ON LU3); OTHERWISE NORMAL RUN .
LU1=-P1

= 0 LU1=1

> 0 LU1=P1 (NORMALLY 1,5, TERMINAL LUN. NOT 2)

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

C

P2 = LUN OF TENSC-OUTPUT DATA (LU2) (P1,NOT,<0) OR LUN OF PREVIOUSLY SCREENED DATA (P1<0) .

< 0 NO INVERSION DATA TO BE INCLUDED WITH OUTPUT : OTHERWISE INVERSION DATA INCLUDED (N.B. IGNORED IF P1<0)

LU2=-P2

= 0 LU2=2

> 0 LU2=P2 (NORMALLY 13, 7 OR 2 FOR P1,NOT,<0, ONLY 13, 7, 5 OR 2 FOR P1<0)

P3 = LUN OF INVERSION DATA TO BE INCLUDED IN OUTPUT (P2,NOT,LT,0) (LU3)

(THIS FILE MUST CONSIST OF 'SMW' CARD TYPES 2-6 OR MORE)

N.B. AN SMW INPUT DECK CAN BE USED FOR THIS.

< 0 COMPONENT 1 ONLY OUTPUT (P4,NOT,LT,0)
LU3=-P3

= 0 LU3=2

> 0 LU3=P3 (NORMALLY 5 OR 2)

P4 = LUN OF OUTPUT FROM SCRAV (LU4)

< 0 COMPONENT 2 ONLY OUTPUT (P3,NOT,LT,0) (IF BOTH P3,P4 < 0, OUTPUT BOTH COMPONENTS)
LU4=-P4

= 0 LU4=2

> 0 LU4=P4 (NORMALLY 4 OR 2)

P5 = OPTION FOR SCREENING PARAMETER DATA :
(SIGNIFICANT ONLY IF P1,NOT,<0)

< 0 INPUT NUMBER OF BADDS / DECADE ON LU1

= 0 ALL PARAMETERS SET INTERNALLY

```

C /P5/= 1 OR
C   > 4   SKIP FURTHER INPUT
C   = 2 OR
C   = 4   INPUT CENTRE PERIOD, 2 COH. CUTOFFS AND
C           2 SKEW CUTOFFS
C   = 3 OR
C   = 4   INPUT IFIX(ALOGT(TMIN AND TMAX)) ON LUI

```

DEFAULTS :

```

C NUMBER BANDS/DECADE = 3.0
C CENTRE PERIOD (CT) = 10.0 SECONDS
C COH. CUTOFFS = .70..70
C SKEW CUTOFFS = 1.00,1.00
C LOG(10) TMIN, TMAX = -2.4

```

SUBROUTINES :

```

C ERROR - A ROUTINE TO CHECK FOR 'FMP' CALL ERRORS
C DISC - A ROUTINE TO READ 80 CHARACTER, FORMATTED
C         (8G10.3) RECORDS TO OR FROM THE DISC INTO
C         OR OUT OF ARRAYS (8 ELEMENTS PER RECORD)
C GDATA - A ROUTINE FOR GETTING PREVIOUSLY SCREENED
C         DATA FROM A SPECIFIED DISC FILE OR LU

```

FMP ROUTINES USED :

```

C OPEN
C READF
C WRITE
C CLOSE
C CREAT
C PURGE

```

```

C LOGICAL BIO,FLAG1,FLAG2,FLAG3,FLAG4,FLAGS
C INTEGER FNAM1(3),FNAM2(3),FNAM3(3)
C INTEGER DCB(144)
C INTEGER OPT,FLAG,BUFF(40),FLAGG
C REAL NPD
C DIMENSION LUN(5),ISIZE(2),CC(2),SC(2),NAME(40),
C *          T1(99),NAVF(2),COH(2),RES(2),PHZ(2),
C *          NGRP(2,90),SGRP(2,90,2),SGRPS(2,90,2),
C *          PER(90,2),RHO(90,2),PHI(90,2),AWTS(90,2),NGRO(40),
C *          NP(2),H(10),RO(11),HOLD(21),NEG1(5)
C EQUIVALENCE (LUI,LUN(1)),(LUI2,LUN(2)),(LUI3,LUN(3)),
C *          (LUI4,LUN(4)),(OPT,LUN(5)),(NAM21,NAME(21)),
C *          (NGRP(1,1),NGRO(1)),(NAM22,NAME(22))
C DATA BIO,FLAG1,FLAG2,FLAG3,FLAGS,FLAG4/5*,FALSE...TRUE./
C DATA ISIZE,NPD,FLAGG,IPHAZ/32.0,3.0,0.1/
C DATA IOPW,IWTS/2.0/
C DATA CT,CC,SC/10.,.2*,.7,2*1./
C DATA LTMIN,LTMAX/-2.4/
C DATA NGRP,SGRP,SGRPS/180*0.,720*0./
C DATA NEG1/2H .2H -.2H1 .2H .2H-1/

```

```

C *** GET PARAMETERS AND SET FLAGS ACCORDINGLY

```

```

C CALL RMPAR(LUN)

```

```

C      IF(LU4.LT.1000) GO TO 1
C      SCRAV INITIATED FROM A TERMINAL -
C      PICK UP NEW PARAMETERS
      WRITE(LU1,300)
300  FORMAT("P1,P2,P3,P4,P5 :_")
      READ(LU1,*) LUN
C      BIO IS SET TRUE IF LU1 IS EITHER THE SYSTEM
C      CONSOLE OR A TERMINAL ON THE 2100
C      IF TRUE, THE USER IS PROMPTED FOR INPUT
C      ON LU1 AND LU3 ONLY; OTHERWISE NO PROMPTING
      BIO=.TRUE.
      IF(LU1.EQ.5.OR.LU1.EQ.7.OR.LU1.EQ.13) BIO=.FALSE.
1  IF(LU1.GE.0) GO TO 5
C      THIS RUN TO CONCATENATE PREVIOUSLY SCREENED
C      DATA AND A STARTING MODEL
      FLAGS=.TRUE.
      LU1=-LU1
5  IF(LU1.EQ.0) LU1=1
      IF(LU1.EQ.1) BIO=.TRUE.
      IF(LU2.GT.0) GO TO 15
      IF(LU2.EQ.0) GO TO 10
C      IF NORMAL RUN, NO INVERSION DATA TO BE
C      OUTPUT WITH SCREENED DATA
      IF(.NOT.FLAG5) FLAG4=.FALSE.
      LU2=-LU2
      GO TO 15
10  LU2=2
15  IF(LU3.GT.0) GO TO 20
      IF(LU3.EQ.0) GO TO 17
C      OUTPUT COMPONENT 1 DATA ONLY
      FLAGG=1
      LU3=-LU3
17  LU3=2
20  IF(LU4.GT.0) GO TO 30
      IF(LU4.EQ.0) GO TO 25
C      OUTPUT COMPONENT 2 DATA ONLY
      IF(FLAGG.EQ.0) FLAGG=2
      IF(FLAGG.EQ.1) FLAGG=0
      LU4=-LU4
      GO TO 30
25  LU4=2
C      IF FLAG1 IS TRUE, LU2 IS A DISC FILE
30  IF(LU2.EQ.2) FLAG1=.TRUE.
C      IF FLAG2 IS TRUE, INVERSION DATA IS TO BE INCLUDED
C      WITH THE SCREENED OUTPUT AND LU3 IS A DISC FILE
      IF(LU3.EQ.2.AND.FLAG4) FLAG2=.TRUE.
C      IF FLAG3 IS TRUE, LU4 IS A DISC FILE
      IF(LU4.EQ.2) FLAG3=.TRUE.
C
C *** GET DISC FILE NAMES IF REQUIRED AND
C *** OPEN THE FIRST FILE
C
      IF(.NOT.FLAG1) GO TO 35
C      GET FILENAME OF FENSC (OR SCREENED) DATA
      IF(BIO) WRITE(LU1,310)
310  FORMAT("NAME OF INPUT DISC FILE 1 :_")
      READ(LU1,320) FNAM1
320  FORMAT(40A2)
      CALL OPEN(DCB,IERR,FNAM1,1)
      CALL ERROR(1,IERR,FNAM1,999)
35  IF(.NOT.FLAG2) GO TO 40

```

```

C      GET FILENAME OF INVERSION DATA
      IF(BIO) WRITE(LU1,330)
330  FORMAT("NAME OF INPUT DISC FILE 2 :_")
      READ(LU1,320) FNAME2
40  IF(.NOT.FLAG3) GO TO 45
C      GET FILENAME TO BE CREATED FOR OUTPUT
C      IF IT IS A DUPLICATE NAME, THE OLD FILE
C      WILL BE PURGED AND A NEW FILE CREATED IF
C      THE SEC. CODE WAS 56
      IF(BIO) WRITE(LU1,340)
340  FORMAT("NAME OF OUTPUT DISC FILE :_")
      READ(LU1,320) FNAME3
C      IF FLAG5 IS TRUE, GO GET SCREENED DATA
45  IF(FLAG5) GO TO 140
C      IF OPT=0 , NO SCREENING PARAMETER INPUT
      IF(OPT.EQ.0) GO TO 60
      IF(OPT.GT.0) GO TO 50
C      INPUT THE NUMBER OF BANDS PER DECADE TO BE USED :
C      IF NPD=3.0 AND TMIN=.01, TMAX=10000, THEN THERE
C      WILL BE 3.0*6 (DECADES) = 18 AVERAGING BANDS
      IF(BIO) WRITE(LU1,350)
350  FORMAT("NUMBER OF POINTS/DECADE (REAL) :_")
      READ(LU1,*) NPD
      OPT=-OPT
C      IF OPT NOT 2,3,4 GO TO SCREENING
      IF(OPT.EQ.1.OR.OPT.GT.4) GO TO 60
50  IF(OPT.NE.2.AND.OPT.NE.4) GO TO 57
C      INPUT CENTRE PERIOD AND COB., SKEW CUTOFFS
      IF(BIO) WRITE(LU1,355)
355  FORMAT("CT,CC(2),SC(2) :_")
      READ(LU1,*) CT,CC,SC
55  IF(OPT.EQ.2) GO TO 60
C      INPUT THE LOG (BASE 10) OF TMIN, TMAX
C      ONLY DATA IN (TMIN,TMAX) WILL BE USED FOR SCREENING
C      WHERE TMIN, TMAX ARE MINIMUM AND MAXIMUM PERIOD VALUES
57  IF(BIO) WRITE(LU1,360)
360  FORMAT("LOG TMIN, LOG TMAX (INTEGER) :_")
      READ(LU1,*) LTMIN,LTMAX
60  TMIN=10.**LTMIN
      TMAX=10.**LTMAX
C
C *** COMMENCE SCREENING AND AVERAGING
C
      BUFF=LU2
C      GET THE DATA TITLE
      IF(.NOT.FLAG1) GO TO 65
      DO 500 I=1,40
500  BUFF(I)=2H
      CALL READF(DCB,IERR,BUFF,40,LEN)
      CALL ERROR(3,IERR,FNAME1,LEN)
      CALL CODE
65  READ(BUFF,320) NAME
C      OUTPUT THE SCREENING PARAMETERS ETC
      WRITE(6,400) (NAME(I),I=1,20),TMIN,TMAX,CT,CC,SC
400  FORMAT(1H1/5X"SCRAV : ".20A2/5X"TMIN =".G10.3,
      * " TMAX =".G10.3/5X"CT =".G10.3." CC =".2G10.3." SC =".2G10.3)
70  NBAND=0
      NFILE=0
75  NFILE=NFILE+1
C      GET A NEW TO (NEXT FILE)
      IF(.NOT.FLAG1) GO TO 80
      CALL READF(DCB,IERR,BUFF,30,LEN)

```

```

      CALL ERROR(3,IERR,FNAM1,LEN)
      CALL CODE
      80 READ(BUFF,200) TO
      200 FORMAT(5(E11.3,1X))
      C IF TO.LT.0 END OF DATA
      IF(TO.LE.0.) GO TO 125
      C WHICH BAND IS THIS
      IF(NBAND.NE.0) GO TO 85
      NBAND=1
      T1(1)=TO
      IBAND=1
      GO TO 92
      85 DO 510 I=1,NBAND
      IF(TO.EQ.T1(I)) GO TO 90
      510 CONTINUE
      NBAND=NBAND+1
      I=NBAND
      T1(NBAND)=TO
      90 IBAND=I
      92 CONTINUE
      C SUM DATA FROM THIS FILE
      N=0
      NOB=0
      NAV(1)=0
      NAV(2)=0
      C TCO IS A CUTOFF THAT EXCLUDES THE FIRST FEW POINTS OF A FILE
      TCO=TO/2.1
      C GET THE T. COH. (2 COMPON.) AND SKEW FOR
      C THE CURRENT TENSOR VALUE
      95 IF(.NOT.FLAG1) GO TO 100
      CALL READ(DCB,IERR,BUFF,30,LEN)
      CALL ERROR(3,IERR,FNAM1,LEN)
      CALL CODE
      100 READ(BUFF,200) T,COH,SKEW
      IF(T.GT.0.) GO TO 103
      C CURRENT FILE EXHAUSTED - OUTPUT OUT-OF-BOUNDS
      C DATA IF ANY. THEN GO GET NEW FILE
      IF(NOB.GT.0)
      *WRITE(6,420) NFILE,TO,N,NOB
      420 FORMAT(" SCRAV - FILE=",14," TO=",G10.4,
      * " N=",14," NUMBER OUT OF BOUNDS=",15)
      GO TO 75
      C GET TENSOR APP.PES. , PHASE AND ROT*N ANGLE
      103 IF(.NOT.FLAG1) GO TO 105
      CALL READ(DCB,IERR,BUFF,30,LEN)
      CALL ERROR(3,IERR,FNAM1,LEN)
      CALL CODE
      105 READ(BUFF,200) RES,PHZ,ROT
      N=N+1
      C SCREEN THE CURRENT TENSOR VALUE
      IF(T.GT.TCO) GO TO 95
      FLAG=1
      IF(T.GT.CT) FLAG=2
      IF(SKEW.GE.SC(FLAG).OR.
      *(COH(1).LT.CC(FLAG).AND.COH(2).LT.CC(FLAG))) GO TO 95
      DO 520 IC=1,2
      IF(COH(IC).LE.CC(IC)) GO TO 520
      NOB=NOB+1
      IF(T.LT.TMIN.OR.T.GT.TMAX) GO TO 520
      NOB=NOB-1
      C SUM THE RESISTIVITY VALUE
      I=IFIX((ALOG(T)-FLOAT(LPMIN))*NPD+1.001)

```

```

NGRP(IC,1)=NGRP(IC,1)+1
TEMP=ALOGT(RES(IC))
SGRP(IC,1,1)=SGRP(IC,1,1)+TEMP
SGRPS(IC,1,1)=SGRPS(IC,1,1)+TEMP*TEMP
C   FORCE THE PHASE INTO THE RANGE -90. TO 0.0
TEMP=PHZ(IC)
110 IF(TEMP.LT.0.0) GO TO 115
TEMP=TEMP-90.
GO TO 110
C   SUM THE PHASE
115 TEMP=-(TEMP+90.)
SGRP(IC,1,2)=SGRP(IC,1,2)+TEMP
SGRPS(IC,1,2)=SGRPS(IC,1,2)+TEMP*TEMP
NAV(IC)=NAV(IC)+1
520 CONTINUE
C   GO GET ANOTHER VALUE
GO TO 95
C
C *** CALCULATE BAND PERIODS, MEAN RESISTIVITY
C *** AND PHASE DATA AND STANDARD DEVIATIONS
C
125 IF(LU2.EQ.7.OR.LU2.EQ.13) REWIND LU3
IF(FLAG1) CALL CLOSE(DCB)
IF(FLAG4.OR.(.NOT.FLAG3)) GO TO 128
C   CREAT OUTPUT FILE
126 CALL CREAT(DCB,IERR,FNAM3,1SIZE,3,56)
IF(IERR.NE.-2) GO TO 127
CALL PURGE(DCB,IERR,FNAM3,56)
CALL ERROR(0,IERR,FNAM3,999)
WRITE(1,382) FNAM3
GO TO 126
127 CALL ERROR(2,IERR,FNAM3,888)
128 CT=ALOGT(CT)
C   INCLUDE SCREENING PARAMETERS IN HEADER RECORD
CALL CODE
WRITE(NAM21,425) NPD,LTMIN,LTMAX,CT,CC,SC
425 FORMAT("C:***",F5.2,13,1X12,F5.1,4F5.2)
WRITE(6,427) NFILE
427 FORMAT(5X"NUMBER OF FILES PROCESSED :",13)
WRITE(6,430) LU4
430 FORMAT(// " SOME OR ALL OF THE FOLLOWING DATA HAS BEEN".
* " WRITTEN TO LUN :",13)
IF(FLAG3) WRITE(6,432) FNAM3
432 FORMAT(50X"FILENAME : ",3A2)
WRITE(6,433) NAME
433 FORMAT(//1X40A2)
NBAND=(LTMAX-LTMIN)*1FIX(NPD+0.001)+1
FACTR=1./(2.*NPD)
C   DECIDE WHICH COMPONENTS TO BE OUTPUT
N1=1
N2=2
IF(FLAGG.EQ.0) GO TO 130
IF(FLAGG.EQ.1) N2=1
IF(FLAGG.EQ.2) N1=2
130 DO 540 IC=N1,N2
WRITE(6,435) IC
435 FORMAT(// " COMPONENT :",13//
*3X"BAND POINT PERIOD N AP.RES. SDEM PHASE SD")
J=0
DO 530 I=1,NBAND
C   IF(NGRP(IC,1)=0) NO POINTS WERE SUMMED IN THIS BAND
IF(NGRP(IC,1).EQ.0) GO TO 530

```

```

J=J+1
N=NGRP(IC,1)
FN=FLOAT(N)
C   CALCULATE AVERAGE APP. RES.
RMEAN=SGRP(IC,1,1)/FN
C   CALCULATE THE STAND. DEV'N
C   N.B. SD IS NOT DEFINED FOR GM. ONLY FOR LOG.MEAN
SDR=SQRT((SGRPS(IC,1,1)/FN-(SGRP(IC,1,1)*
*SGRP(IC,1,1))/(FN*FN))
C   CALCULATE THE PERIOD AT THE CENTRE OF THE BAND
PER(J,IC)=10.**(FLOAT(LTMIN)+FLOAT(2*1-1)*FACTR)
RHO(J,IC)=10.**(RMEAN)
C   CALCULATE THE AVERAGE PHASE
RMEAN=SGRP(IC,1,2)/FN
C   CALCULATE THE STAND. DEV'N
SDP=SQRT((SGRPS(IC,1,2)/FN-(SGRP(IC,1,2)*
*SGRP(IC,1,2))/(FN*FN))
C   WEIGHT = NUMBER POINTS IN BAND
AWTS(J,IC)=FN
PHI(J,IC)=RMEAN
C   PRINT THE AVERAGED DATA
WRITE(6,440) 1,J,PER(J,IC),N,RHO(J,IC),SDR,PHI(J,IC),SDP
440  FORMAT(1X2I5,2XG10.4,13.3X4G10.4)
530  CONTINUE
NP(IC)=J
IF(FLAG4) GO TO 540
C
C *** OUTPUT THE SCREENED AND AVERAGED DATA WITHOUT
C *** ANY INVERSION DATA
C
C   WRITE THE COMPONENT NUMBER INTO THE HEADER
CALL CODE
WRITE(NAM22,445) IC
445  FORMAT(12)
IF(.NOT.FLAG3) GO TO 135
C   WRITE OUTPUT TO DISC FILE
CALL CODE
WRITE(BUFF,447) J
447  FORMAT(5X15)
CALL WRITE(DCB,IERR,BUFF,5)
CALL ERROR(5,IERR,FNAM3,40)
CALL WRITE(DCB,IERR,NAME,40)
CALL ERROR(5,IERR,FNAM3,40)
CALL DISC (,FALSE,,PER(1,IC),J,FNAM3,DCB)
CALL DISC (,FALSE,,RHO(1,IC),J,FNAM3,DCB)
CALL DISC (,FALSE,,PHI(1,IC),J,FNAM3,DCB)
CALL DISC (,FALSE,,AWTS(1,IC),J,FNAM3,DCB)
CALL DISC (,FALSE,,AWTS(1,IC),J,FNAM3,DCB)
GO TO 540
C   WRITE OUTPUT ON LU4
135  WRITE(LU4,447) J
WRITE(LU4,320) NAME
WRITE(LU4,450)(PER(1,IC),I=1,J)
450  FORMAT(8G10.4)
WRITE(LU4,450)(RHO(1,IC),I=1,J)
WRITE(LU4,450)(PHI(1,IC),I=1,J)
WRITE(LU4,450)(AWTS(1,IC),I=1,J)
WRITE(LU4,450)(AWTS(1,IC),I=1,J)
540  CONTINUE
C   IF FLAG4 FALSE, END
IF(.NOT.FLAG4) GO TO 999
C   IF FLAG5 TRUE, GET PREVIOUSLY SCREENED DATA

```

```

140 IF(FLAG5) CALL GDAT(N1,N2,NP,NAME,PER,RHO,PHI,AWTS,BUFF,FLAG1,
    *LU2,FNAM1,DCB)
C
C *** INPUT INVERSION DATA FROM LU3 OR DISC
C
C RESET BIO FOR LU3 INPUT
    BIO=.TRUE.
    IF(LU3.EQ.2.OR.LU3.EQ.5.OR.LU3.EQ.7.OR.LU3.EQ.13) BIO=.FALSE.
    BUFF=LU3
    IF(FLAG2) GO TO 145
C
C LU3 IS NOT A DISC
    IF(BIO) WRITE(BUFF,365)
365 FORMAT("NL,IFLAG,IWTS,IPHAZ,IHOLD (15.5X15.5X315):")
    GO TO 147
C
C OPEN 'FNAM2' AND READ 1ST RECORD
145 CALL OPEN(DCB,IERR,FNAM2,1)
    CALL ERROR(1,IERR,FNAM2,999)
    CALL READF(DCB,IERR,BUFF,18,LEN)
    CALL ERROR(4,IERR,FNAM2,LEN)
    CALL CODE
C
C GET NL,IFLAG,IWTS,IPHAZ,IHOLD
147 READ(BUFF,210) NL,IFLAG,IWTS,IPHAZ,IHOLD
210 FORMAT(15.5X15.5X315)
    NLP1=NL+1
C
C CORRECT IF NECESSARY IFLAG, IWTS, IPHAZ, AND IHOLD
    IF(IFLAG.NE.1.AND.IFLAG.NE.2) IFLAG=2
    IF(IWTS.NE.0.AND.IWTS.NE.1) IWTS=1
    IF(IPHAZ.NE.0.AND.IPHAZ.NE.1) IPHAZ=1
    IF(IHOLD.NE.0.AND.IHOLD.NE.1) IHOLD=0
    IF(.NOT.FLAG2) GO TO 150
C
C GET H,RO DATA FROM DISC AFTER SKIPPING NAME RECORD
    CALL READF(DCB,IERR,BUFF,1,LEN)
    CALL ERROR(4,IERR,FNAM2,LEN)
    CALL DISC (.TRUE.,H,NL,FNAM2,DCB)
    CALL DISC (.TRUE.,RO,NLP1,FNAM2,DCB)
    GO TO 160
C
C SKIP HEADER RECORD BY DUMMY READ
150 IF(BIO) WRITE(BUFF,367)
367 FORMAT("HEADER RECORD (NOT USED) :")
    READ(BUFF,320) I
C
C GET H,RO DATA FROM LU3
    IF(BIO) WRITE(BUFF,370) NL
370 FORMAT("H(1),I=1, ".12." (8G10.4) :")
    READ(BUFF,220)(H(I),I=1,NL)
220 FORMAT(8G10.4)
    IF(BIO) WRITE(BUFF,372) NLP1
372 FORMAT("RO(1),I=1, ".12." (8G10.4) :")
    READ(BUFF,220)(RO(I),I=1,NLP1)
C
C GET INVERSION PARAMETER RECORD (MAXITS ETC)
160 IF(.NOT.FLAG2) GO TO 165
C
C FROM DISC
    CALL READF(DCB,IERR,NGRO,13,LEN)
    CALL ERROR(4,IERR,FNAM2,LEN)
    GO TO 168
C
C FROM LU3
165 IF(BIO) WRITE(BUFF,375)
375 FORMAT("ITS,IPRNT,IWL,IWZ,IPLOT (515) :")
    READ(BUFF,320)(NGRO(I),I=1,13)
168 IF(IHOLD.EQ.0) GO TO 175
    N=NLP1
    IF(IFLAG.EQ.2) N=N+NL
C
C GET HOLD DATA

```



```

      IF(.NOT.FLAG2) GO TO 170
C     FROM DISC
      CALL DISC (.TRUE.,HOLD,N,FNAM2,DCB)
      GO TO 175
C     FROM LU3
170  IF(BIO) WRITE(BUFF,380) N
380  FORMAT("HOLD(1),I=1,".12," (8G10.3) :")
      READ(BUFF,220)(HOLD(1),I=1,N)
175  IF(FLAG2) CALL CLOSE(DCB)
      IF(LU3.EQ.7.OR.LU3.EQ.13) REWIND LU3
C
C *** OUTPUT INVERSION AND AVERAGED DATA FOR SMW
C
      IF(.NOT.FLAG3) GO TO 178
C     OUTPUT TO DISC - CREAT 'FNAM3'
176  CALL CREAT(DCB,IERR,FNAM3,ISIZE,3,56)
      IF(IERR.NE.-2) GO TO 177
      CALL PURGE(DCB,IERR,FNAM3,56)
      CALL ERROR(0,IERR,FNAM3,999)
      WRITE(1,382) FNAM3
382  FORMAT("SCRAV - FILE : ".3A2," PURGED")
      GO TO 176
177  CALL ERROR(2,IERR,FNAM3,888)
C     OUTPUT ONLY COMPONENTS OPTED
178  DO 550 IC=N1,N2
      M=NP(IC)
      BUFF=LU4
      IF(FLAG3) CALL CODE
      WRITE(BUFF,230) NL,M,IFLAG,LUPW,IWTS,IPHAZ,IHOLD
230  FORMAT(16I5)
      CALL CODE
      WRITE(NAM22,445) IC
      IF(.NOT.FLAG3) GO TO 180
C     OUTPUT TO DISC FILE
      CALL WRITE(DCB,IERR,BUFF,18)
      CALL ERROR(6,IERR,FNAM3,16)
      CALL WRITE(DCB,IERR,NAME,40)
      CALL ERROR(6,IERR,FNAM3,40)
      CALL DISC (.FALSE.,N,NL,FNAM3,DCB)
      CALL DISC (.FALSE.,RO,NLP1,FNAM3,DCB)
      CALL WRITE(DCB,IERR,NGRO,13)
      CALL ERROR(6,IERR,FNAM3,13)
      CALL DISC (.FALSE.,PER(1,IC),M,FNAM3,DCB)
      CALL DISC (.FALSE.,RHO(1,IC),M,FNAM3,DCB)
      IF(IPHAZ.EQ.1)
      *CALL DISC (.FALSE.,PHI(1,IC),M,FNAM3,DCB)
      IF(IWTS.EQ.0) GO TO 179
      CALL DISC (.FALSE.,AWTS(1,IC),M,FNAM3,DCB)
      IF(IPHAZ.EQ.1) CALL DISC (.FALSE.,AWTS(1,IC),M,FNAM3,DCB)
179  IF(IHOLD.EQ.1)
      *CALL DISC (.FALSE.,HOLD,N,FNAM3,DCB)
      GO TO 550
C     OUTPUT TO LU4
180  WRITE(BUFF,320) NAME
      WRITE(BUFF,220)(H(1),I=1,NL)
      WRITE(BUFF,220)(RO(1),I=1,NLP1)
      WRITE(BUFF,320)(NGRO(1),I=1,13)
      WRITE(BUFF,220)(PER(1,IC),I=1,M)
      WRITE(BUFF,220)(RHO(1,IC),I=1,M)
      IF(IPHAZ.EQ.1)
      *WRITE(BUFF,220)(PHI(1,IC),I=1,M)
      IF(IWTS.EQ.0) GO TO 181

```

```

        WRITE(BUFF,220)(AWTS(1,IC),I=1,M)
        IF(IPHAZ.EQ.1) WRITE(BUFF,220)(AWTS(1,IC),I=1,M)
181 IF(IHOLD.EQ.1)
      *WRITE(BUFF,220)(HOLD(I),I=1,N)
550 CONTINUE
C      END NOW
999 IF(.NOT.FLAG3) GO TO 1000
C      WRITE END-OF-DATA MARKER ON DISC AND CLOSE
      CALL WRITE(DCB,IERR,NEGI,5)
      CALL ERROR(6,IERR,FNAME,3)
      CALL CLOSE(DCB)
      GO TO 1001
C      WRITE END-OF-DATA MARKER ON LU4
C      FINISH
1000 WRITE(LU4,320) NEGI
1001 WRITE(1,385)
385 FORMAT("SCRAV OFF")
      CALL EXEC(6)
      END
      SUBROUTINE ERROR(IC,IERR,FNAME,LEN)

```

```

C      ERROR DETECTS 'FMP' CALL ERRORS OR EOF/SOF ERRORS
C      ON 'READF' CALLS WHERE
C      IC IS THE USER DEFINED CODE
C      IERR IS THE ERROR CODE RETURNED BY THE FMP CALL.
C      FNAME IS THE FILE BEING ACCESSED AND
C      LEN IS THE LENGTH OF THE RECORD READ
C      (< 0 IMPLIES READ THROUGH EOF)
C
C      IF IERR<0 OR LEN<0, THEN SCRAV IS ABORTED AFTER
C      A MESSAGE IS SENT TO LU 1; THE FOLLOWING DATA IS
C      OUTPUT :
C          FILENAME
C          FMP ERROR CODE
C          THE NUMBER OF THE CALL TO ERROR
C          THE PROGRAM CODE PASSED TO ERROR AND
C          THE LENGTH OF THE DISC RECORD READ (READF ONLY)

```

```

C      THE PROGRAM CODES ARE :

```

```

C      0 - ERROR ON PURGE REQUEST
C      1 - ERROR ON OPEN REQUEST
C      2 - ERROR ON CREAT REQUEST
C      3 - ERROR ON READF REQUEST IN 'SCRAV' FROM INPUT FILE 1
C      4 - ERROR ON READF REQUEST IN 'SCRAV' FROM INPUT FILE 2
C      5 - ERROR ON WRITE REQUEST IN 'SCRAV' - NO INVERSION OUTPUT
C      6 - ERROR ON WRITE REQUEST IN 'SCRAV' - INVERSION DATA IS OUTPUT
C      7 - ERROR ON READF REQUEST IN 'DISC'
C      8 - ERROR ON WRITE REQUEST IN 'DISC'

```

```

C      INTEGER FNAME(3)
C      DATA ICALL/0/
C      ICALL=ICALL+1
C      IF(IERR.GE.0.AND.LEN.GE.0) RETURN
C      WRITE(1,300) FNAME,IERR,ICALL,IC,LEN
300 FORMAT("SCRAV FMP ERROR : FILE '"',3A2,"' ERROR ='",I6/
      *      "CALL ='",I3,"' CODE ='",I3,"' LENGTH ='",I6/
      *      "SCRAV ABORTED")
      CALL EXEC(6)
      END
      SUBROUTINE DISC(READ,A,N,FNAME,DCB)

```

'DISC' READS FROM OR WRITES TO THE DISC
40 WORD RECORDS IN (8G10.4) FORMAT
INTO OR OUT OF A SPECIFIED ARRAY

THE FOLLOWING VARIABLES ARE USED :

READ = .TRUE. THEN OPERATION IS READ; OTHERWISE WRITE
A - ARRAY FOR I/O
N - NUMBER OF ELEMENTS OF A
FNAME - NAME OF FMGR FILE TO BE READ FROM OR WRITTEN TO
DCB - DATA CONTROL BLOCK OF DISC FILE

```

LOGICAL READ
INTEGER FNAME(1),DCB(1),BUFF(40)
DIMENSION A(N)
M=N/8
IF(M.EQ.0) GO TO 10
DO 100 I=1,M
M2=8*I
M1=M2-7
IF(READ) GO TO 50
CALL CODE
WRITE(BUFF,300)(A(J),J=M1,M2)
300 FORMAT(8G10.4)
CALL WRITE(DCB,IERR,BUFF,40)
CALL ERROR(8,IERR,FNAME,40)
GO TO 100
50 CALL READF(DCB,IERR,BUFF,40,LEN)
CALL ERROR(7,IERR,FNAME,40)
CALL CODE
READ(BUFF,300)(A(J),J=M1,M2)
100 CONTINUE
10 M2=N-8*M
IF(M2.EQ.0) RETURN
LEN=M2*5
M1=8*M+1
M2=M1+M2-1
IF(READ) GO TO 60
CALL CODE
WRITE(BUFF,300)(A(J),J=M1,M2)
CALL WRITE(DCB,IERR,BUFF,LEN)
CALL ERROR(8,IERR,FNAME,LEN)
RETURN
60 CALL READF(DCB,IERR,BUFF,40,LEN)
CALL ERROR(7,IERR,FNAME,40)
CALL CODE
READ(BUFF,300)(A(J),J=M1,M2)
RETURN
END
SUBROUTINE GDAT(N1,N2,NP,NAME,PER,RHO,PHI,AWTS,BUFF,
*FLAG,LU,FNAME,DCB)

```

'GDAT' GETS PREVIOUSLY SCREENED MT DATA AND PASSES IT
BACK TO 'SCRAV'. INPUT IS FROM A DISC FILE OR FROM
A LU. THE FOLLOWING VARIABLES ARE USED:

N1,N2 - COMPONENT NUMBERS
NP - 2 M VALUES FOR SMW
NAME - HEADER RECORD
PER - PERIOD DATA
RHO - RESISTIVITY DATA
PHI - PHASE DATA
BUFF - BUFFER AREA

C
C
C
C
C
C

FLAG - TRUE, INPUT FROM DISC
LU - LOGICAL UNIT FOR INPUT
FNAME - DISC FILENAME IF USED
DCB - DATA CONTROL BLOCK FOR DISC FILE

```

LOGICAL FLAG
INTEGER BUFF(40), FNAME(3), DCB(144)
DIMENSION NP(2), NAME(40), PER(90,2), RHO(90,2), PHI(90,2),
*AWTS(90,2), N(2)
I=0
BUFF=LU
5 I=I+1
IF(.NOT.FLAG) GO TO 10
CALL READF(DCB, IERR, BUFF, 5, LEN)
CALL ERROR(9, IERR, FNAME, LEN)
CALL CODE
10 READ(BUFF, 200) M
200 FORMAT(5X15)
IF(M.LE.0) GO TO 50
NP(1)=M
IF(.NOT.FLAG) GO TO 20
CALL READF(DCB, IERR, NAME, 40, LEN)
CALL ERROR(9, IERR, FNAME, LEN)
GO TO 30
20 READ(BUFF, 210) NAME
210 FORMAT(40A2)
30 NAM22=NAME(22)
CALL CODE
READ(NAM22, 220) N(1)
220 FORMAT(12)
K=N(1)
IF(.NOT.FLAG) GO TO 40
CALL DISC(.TRUE., PER(1,K), M, FNAME, DCB)
CALL DISC(.TRUE., RHO(1,K), M, FNAME, DCB)
CALL DISC(.TRUE., PHI(1,K), M, FNAME, DCB)
CALL DISC(.TRUE., AWTS(1,K), M, FNAME, DCB)
CALL DISC(.TRUE., AWTS(1,K), M, FNAME, DCB)
GO TO 5
40 READ(BUFF, 230) (PER(J,K), J=1,M)
230 FORMAT(8G10.3)
READ(BUFF, 230) (RHO(J,K), J=1,M)
READ(BUFF, 230) (PHI(J,K), J=1,M)
READ(BUFF, 230) (AWTS(J,K), J=1,M)
READ(BUFF, 230) (AWTS(J,K), J=1,M)
GO TO 5
50 N1=N(1)
N2=N(2)
IF(1.LE.2) N2=N(1)
IF(FLAG) CALL CLOSE(DCB)
IF(LU.EQ.7.OR.LU.EQ.13) REWIND LU
RETURN
END
ENDS

```

SYSTEM BULLETIN 23 --- 1/ 6/76. USE -SYSBULL.LFN. OR -SYSBULL. 333333

HH.MM.SS CPU SECOND ORIGIN

12.08.47	00000.009	CYB.	ED VERSION 06/07/77				
12.08.47	00000.012	JOB.	-MUSH.P3000.				
12.08.47	00000.013	LOD.	-DISPOSE.OUTPUT.ST=RIOCB.*PR=CZR.				
12.08.47	00000.017	USR.	-BANNER.S S.				
12.08.47	00000.033	LOD.	FORTTRAN LIBRARY 401	18/12/75			
12.08.47	00000.037	USR.	-BANNER.S BOBS/S BMRs/AIRBORNE.				
12.08.47	00000.060	JOB.	FORTTRAN LIBRARY 401	18/12/75			
12.08.47	00000.061	JOB.	-COMMENT. BOB MOORE BMR AIRBORNE X228				
12.08.47	00000.062	CYB.	-ATTACH.F.SCRAPS.ID=DMRAPF.				
12.08.47	00000.064	CYB.	DEFAULT MR=1. READ ACCESS ONLY				
12.08.47	00000.065	LOD.	PF254 - CYCLE 3 ATTACHED FROM SW=SYSTEM				
12.08.48	00000.182	USR.	-COPY.F.OUTPUT.				
12.08.48	00000.182	USR.	UT031 - EOI ENCOUNTERED				
12.08.48	00000.184	CYB.	UT034 - EOP - 736				
12.08.48	00000.184	CYB.	MDMRAPD	5.120	KCHAR		
12.08.48	00000.184	CYB.	DFILEPD	1.330	KCHAR		
12.08.48	00000.185	CYB.	OUTPUT	55.570	KCHAR		
12.08.48	00000.185	CYB.	SC050 - 000000 SC/LC SWAPS				
12.08.48	00000.185	CYB.	CPU-TIME	0.187	SEC	0.098	S
12.08.48	00000.185	CYB.	USR-TIME	0.035	SEC		
12.08.48	00000.186	CYB.	FILE-ID	84.180	KCHAR	0.018	S
12.08.48	00000.186	CYB.	SIATN-ID	62.020	KCHAR	0.195	S
12.08.48	00000.186	CYB.	SCM-TIME	0.383	KWSEC	0.004	S
12.08.48	00000.186	CYB.	RMS-TIME	0.002	MWSEC	0.000	S
12.08.48	00000.187	CYB.	TOTAL	0.600	RU	0.315	S
12.08.48	00000.187	CYB.	PRIORITY	3.000	K		
12.08.48	00000.187	CYB.	RATE	0.525	S/RU		
12.08.48	00000.187	CYB.	CHARGE			0.47	S

Appendix 3

Table A1 shows those data records that must be included in an SMW input file. If the user specified modes of operation 2 or 3, SCRAV will produce an SMW input file; he must therefore supply on LU3 records 1-5 (and 11 if required) of Table A1. A pre-existing SMW input file may be used but only if record 11 is not required (HOLD = 0). All the information given in Table A1 will be read but the following data will be changed by SCRAV:

M	(set equal to number of averaged data)
IFLAG	(only if equal to 0, then set equal to 2)
IOPW	(always set equal to 2)
TITLE	(taken from LU2)

Note that IWTS, IPHAZ and IHOLD affect the execution of SCRAV. If IPHAZ = 1, phase data is also output (with apparent resistivity values). If IWTS = 1, weights for apparent resistivity and phase (if included in output) will be set equal to the number of points in the period band in which the averaged apparent resistivity and phase values were calculated. If IHOLD = 1, parameters fixing information is read in also (to be included with output); records 6-10 must NOT appear on LU3 if IHOLD = 1.

```

*          ***** HP 2100/21MX *****
*          ***** PROGRAM SMW *****
*          *** USER DEFINED DATA RECORDS ***
*
* RECORD(1)      NL,M,IFLAG,IOPW,IWTS,IPHAZ,IHOLD      (715)
*                NL= NO. OF LAYERS ABOVE 1/2 SPACE
*                THAT IS, THE NUMBER OF LAYERS IS (NL+1)
*                ZERO.LT.NL.LE. 7 (LE.0 PROGRAM EXITS)
*                M= NO. OF DATA VALUES (0<M<22)
*                IFLAG=0      FORWARD CASE ONLY
*                =1          RESISTIVITIES ONLY TO CHANGE
*                =2          RESISTIVITIES AND LAYER THICKNESSES
*                IOPW=0      FREQ. SCALE SET INTERNALLY (SEE RECORD(5))
*                =1          FREQUENCIES READ IN
*                =2          PERIOD VALUES READ IN
*                =3          SPECIAL CODE FOR DATA OFF TAPE1
*                IWTS=0      WEIGHTS SET INTERNALLY TO 1.0
*                =1          WEIGHTS READ IN
*                IPHAZ=0     INVERT ON APP. RES. ALONE
*                =1          PHASE DATA USED IN INVERSION
*                        NOTE ... IF IOPW.NE.3 AND IPHAZ=0 RECORDS
*                        8 AND 10 ARE ABSENT (I.E. NO PHASE READ IN)
*                IHOLD=0     NO PARAMETERS FIXED
*                =1          PARAMETERS IND. ON RECORD 10 FIXED
*
* RECORD(2)      (TITLE(I),I=1,40)                      (40A2)
*                TITLE RECORD - FULL 80 COLUMNS AVAILABLE.
*
* RECORD(3)      (H(I),I=1,NL)                          (8F10.0)
*                LAYER THICKNESSES READ IN
*
* RECORD(4)      (RO(I),I=1,NL+1)                      (8F10.0)
*                LAYER RESISTIVITIES READ IN
*
* RECORD(5)      ITS,IPRNT,IWH,IWZ,IFLOT                (515)
*                ITS=MAXIMUM NO. OF ITERATIONS FOR INVERSION
*                (USUALLY ITS=10,11,12)
*                IPRNT=0     NO PRINT OUT DURING INVERSION
*                =1,2       MORE PRINT OUT
*                (USUALLY IPRNT=1)
*                IWH,IWZ INDICATE THICKNESS AND/OR DEPTH ANALYSIS
*                IWH/Z=0     NO STATS OUTPUT
*                =1         SINGULAR VALUE ANALYSIS
*                =2         INFLUENCE MATRIX AS WELL
*                =3         DATA SPACE EIGENVECTORS
*                (USUALLY IWH/Z=1)

```

```

*          IPLOT=0 NO PLOT
*          =1 APP. RES. DATA AND MODEL PLOTTED
*          =2 PHASE DATA AND MODEL AS WELL
*          BOTH DATA AND MODELS PLOTTED
*
* RECORD(6)  IF(IOPW.EQ.0) (F(1),F(M)) (SF10.0)
*             IF(IOPW.EQ.1) (F(I),I=1,M)
*             IF(IOPW.GE.2) (T(I),I=1,M)
*             DATA POINTS READ IN EITHER AS FREQUENCIES
*             (IOPW.EQ.1) OR PERIODS(IOPW.EQ.2). IF IOPW.EQ.0,
*             F(I) ARE SET AS M LOG EQUALLY SPACED POINTS
*             BETWEEN F(1) AND F(M).
*
* RECORD(7)  (ONLY IF IFLAG.GT.0)
*             (Y(I),I=1,M) (8F10.0)
*             APPARENT RESISTIVITY DATA AT PREVIOUSLY READ POINTS
*             NOTE ... Y(I).GT.0.0
*             IF(Y(I).LE.0.0) DATA VALUE IGNORED
*
* RECORD(8)  (ONLY IF IFLAG.GT.0 AND IPHAZ.EQ.1) (8F10.0)
*             (P(I),I=1,M) PHASE IN DEGREES
*             NOTE ... -90.LT.P(I).LT.0
*             IF OUTSIDE RANGE DATA IGNORED
*
* RECORD(9)  (ONLY IF IFLAG.GT.0 AND IWTS.EQ.1)
*             (AWTS(I),I=1,M) (8F10.0)
*             WEIGHTS FOR APPARENT RESISTIVITY DATA
*             AWTS(I).GE.0.0
*             IF AWTS(I)=0.0 DATA POINT IS ELIMINATED
*
* RECORD(10) (ONLY IF IFLAG.GT.0 AND IWTS.EQ.1 AND IPHAZ.GT.0)
*             (PWTS(I),I=1,M) (8F10.0)
*             WEIGHTS FOR PHASE DATA
*             PWTS(I).GE.0.0
*             IF PWTS(I)=0.0 DATA POINT IS ELIMINATED
*
* RECORD(11) (ONLY IF IHOLD.GT.0)
*             (HOLD(I),I=1,NN1) (8F10.0)
*             (NN1=NL+1 IF IFLAG=1
*             =2*NL+1 OTHERWISE)
*             IF HOLD(I)=0.0 PARAMETER IS FIXED AT INITIAL VALUE
*             IF HOLD(I)=1.0 PARAMETER IS FREE
*             PARAMETERS ARE ORDERED IN THE FOLLOWING WAY
*             (RO(1),RO(2),...,H(1),H(2),...,H(NL))

```