U.S. Department of the Interior
Geological Survey

Mail Stop 964
Box 25046, Federal Center
Denver, Colorado 80225

Program IMSLPW: Marquardt inversion of plane-wave frequency soundings

by

Walter L. Anderson

OPEN-FILE REPORT 79-586

1979

CONTENTS.

DISCLAIMER 3

INTRODUCTION 4

PARAMETERS AND DATA REQUIRED 5

PROGRAM FILES 5

DETAIL PARAMETER AND DATA DEFINITIONS 6

$parms parameters 6

$init parameters 12

DATA MATRIX NOTES 12

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING 13

SPECIAL OBJECT FORMAT PHRASES 14

MULTICS OPERATING INSTRUCTIONS 14

ERROR MESSAGES 15

REFERENCES 16

Appendix 1.-- Source listing 17

Source availability 17

Appendix 2.-- Conversion to other systems 30

Appendix 3.-- Test problem input/output listing 32

DISCLAIMER.

This program was written in Fortran IV for a Honeywell Multics 68/80 system*. Although program tests have been made, no guarantee (expressed or implied) is made by the author regarding accuracy or proper functioning of this program on all computer systems.

---

* Brand or manufacturers' names used in this report are for descriptive purposes only and do not constitute endorsement by the U.S. Geological Survey.

By Walter L. Anderson

INTRODUCTION.

Program IMSLPW is a general purpose program for inversion of magnetotelluric and other plane-wave (Wait, 1962; Vozoff, 1972) frequency sounding data obtained over a horizontally stratified earth for the quasi-static case (i.e., neglecting displacement currents). An IMSL (International Mathematical and Statistical Library, 1977) derivative-free Marquardt (1963) nonlinear least squares subprogram (ZXSSQ) is used for inversion of frequency sounding data. See appendix 2 notes for conversion to other systems where the IMSL library is not available.

The following program options are currently available:

(1) Simultaneous (or joint) inversion of apparent resistivity and/or phase of surface impedance for frequency soundings.
(2) Scaling parameter and observation spaces to constrain the solution space and to reduce round-off effects.
(3) Weighted observations.
(4) Holding certain parameters fixed (constrained).
(5) Object-time format control of reading the observed data matrix.

To provide as much timely computer information as possible, this report is being released without a mathematical formulation section. The interested reader may consult the cited references for more details.

PARAMETERS AND DATA REQUIRED.

Parameters required by program IMSLPW are read using Fortran namelist read statements with specific names: $parms and $init. Default values are used whenever a corresponding parameter is omitted in a namelist. The input data matrix is read from an optional alternate file (unless overridden) using a Fortran object-time format. Preceding the $parms statement is a required 80 (or less) character title.

The general input order read by program IMSLPW is:

1. Title line (always required, max. 80 characters).
2. $parms --non-default parameters--$
   (note $parms may begin in col. 1 on Multics).
3. (Object-time format) statement defining the given format of the input data matrix. The object format begins with "(" placed in col. 1.
4. Optionally, the data matrix read under the object format may be inserted here if the alternate data file is not used (see parameter ialt below).
5. $init --non-default parameters--$
6. Optionally, subsequent runs using the same data matrix but with changed $parms and $init parameters may be given by repeating steps 1,2,3, and 5 (provided parameters istop=0 and ialt is not 5).

The above general input order is required whether the job is being run in time-sharing or batch modes (see job operating instructions below).

PROGRAM FILES.

file05	title, input parameters $parms, object format (for reading data matrix on unit ialt=10--default), and $init parameters.
file06	output on-line printer file (see file16 for more detail output).
file10	default input data matrix file read under the object format given in file05. parameter ialt=10 (default) may be changed to any file number other than 06,13, or 16. Note ialt=05 will mean the data matrix is included immediately after the object-time format on file05.
file13	output scratch disk file used as required during execution of IMSLPW.
file16	output master print-type disk file--contains maximum printable output.

DETAIL PARAMETER AND DATA DEFINITIONS.

$parms parameters (with defaults and cross-references):

[names below prefixed with a "*" are not used by IMSLPW, but are included for conversion compatibility if the CALL IMSLMQ is replaced by CALL MARQRT (see appendix 2 paragraph 6 on this type of conversion)]

n= Number of observed data points y(i),i=1,...,n, where n<=200.

k= Total number of parameters (1<=k<=20, k<=n). The value of k must be equal to 2*mm-1, where $init parameter mm>0 is the number of layers in the model.
(cref: $init parameter mm and $parms n,b).

ip= Number of omitted parameters; i.e., number of parameters held fixed or constrained via array ib() to initial input values given in array b(). Default ip=0 with the restrictions that ip<k and n>=k-ip.
(cref: $parms k,n,ib(), and b).

m= Number of independent variables (m<=2) given in the data matrix (y(i),x(i,j),j=1,m),i=1,n. The value of m must be given as follows:
= 1 when $init parameter iob<=2 (defines specific observation type in y(i));
= 2 when $init parameter iob=5 (defines mixed observation types in y(i) via x(i,2)).
(cref: $parms iwt, $init iob, and DATA MATRIX NOTES below for all definitions of x(i,m) used).

ialt= Input data matrix alternate logical unit number (default 10) for reading the data under the object-time format specified in file05. The value of ialt can be any value the operating system supports, but cannot be equal to 6,13, or 16. If ialt=5 is used, then the data matrix ((y(i),x(i,j),j=1,m),i=1,n) will immediately follow the object format on file05.
(cref: $parms n,m, $init iob).

istop= 0 to continue processing after completion of the current problem (i.e., a total restart) with the same data matrix as last used, but by using a revised title, $parms, object-time format, and $init parameters. Note that istop=0 can only be used whenever ialt is not 5 (since file ialt is rewound and read again). Also, all $parms and

$init parameters previously used will be assumed, with the exception of array b(j)--which must always be given.

= 1 (default) to stop the run after completion of the current problem.
(cref: $parms b,ialt).

iwt= 0 (default) for unweighted observations; i.e., all n observations y(i),i=1,...,n will be weighted unity (with assumed standard deviations equal to 1.0).

= 1 for weighted observations given by the formula wt(i)=1.0/x(i,m+1)**2, where x(i,m+1) is the standard deviation augmented to the data matrix for the given m<=2. Note: wt(i)=1.0 is stored automatically if iwt=0 or when iwt=1 and x(i,m+1)=0.0 (to avoid division by 0).
(cref: $parms n,m, $init iob, and DATA MATRIX NOTES).

* ider= 0 (default) to use analytic derivatives, which calls both forward problem (fcode) and analytic derivative (pcode) subroutines.

= 1 to use estimated derivatives, which calls only subroutine fcode. [if converting to subprogram MARQRT (as in appendix 2), then ider=1 must always be used, since pcode is a dummy routine].
(cref: $parms del).

iprt= 0 (default) for standard abbreviated printout format for each iteration. Note scaled values of parameters b(j) and phi (sum of squares) will be given via parameter scalep.

= 1 for detail printout format for each iteration, which includes the parameter changes from the Marquardt algorithm. [note iprt=1 behaves like iprt=-2, unless converting to subprogram MARQRT].

= -1 (recommended if scalep>0 used) for abbreviated printout format for each iteration with printed unscaled values of b(j) but scaled values of phi.

= -2 same as iprt=-1 but also prints on file06 n-observational lines containing: observed value (obs=y(i)), calculated value (cal), residual (res), and x(i,1). Note file16 will always contain the complete obs-cal-res and x(i,m) data printout. Option iprt=-2 may be useful for time-sharing runs to examine on-line the final solution and residuals.
(cref: $parms iout,sp and DATA MATRIX NOTES).

* niter= Maximum number of iterations allowed before accepting the results as "forced off" (default niter=10). Four different types of convergence

tests are possible--one of which is termed "forced off", which will occur whenever niter has been reached and one of the other convergence criteria has not been achieved. Using a small niter may be useful to monitor the progress for a large problem, and as an aid for achieving a convenient restarting procedure with the last b-vector as a new initial estimate.
(cref: $parms b and Marquardt (1963) for convergence tests used).

* inon= 1 (default) to omit nonlinear confidence region calculations.
= 0 to compute nonlinear confidence regions after the last iteration. This option calls subroutine fcode many times, and is not recommended for general use with program IMSLPW unless one is interested in a detailed nonlinear statistical analysis of the final solution.
(see IBM Share program No. 1428 for more details on this option).

* ff= Variance F-ratio statistic (default 4.0) used to compute linear support-plane confidence limits and nonlinear (if inon=0) confidence limits after convergence or niter iterations. The default value is adequate for most applications.

* t= Student's t-statistic (default 2.0) used to compute one-parameter linear confidence limits after convergence or niter iterations. The default value is adequate for most applications.

e= Convergence criterion test parameter (default 0.5e-4). For example, for 2-figure accuracy, use e=.01; for 3-figure accuracy, use e=.001, etc. [for IMSLPW, e is equivalent to $parms eps (see below)].
(cref: Marquardt, 1963).

* tau= Convergence criterion test parameter (default 1e-3).
(cref: Marquardt, 1963).

* xl= Initial Marquardt's lambda factor (default .01) to be added to the diagonal of the Jacobian transpose times Jacobian matrix. For some very ill-conditioned problems, or for poor initial parameter estimates, a larger xl (e.g., 1.0) may prove to be advantageous.
(cref: Marquardt, 1963 and Share program No. 1428).

* modlam= 1 (default) to use a modified Marquardt lambda method at each iteration as described in Tabata and Ito (1973).
= 0 to use the original Marquardt (1963) lambda method at each iteration.

* gamcr= Marquardt's critical angle between the gradient and adjustment vectors (default 45.0 degrees). The value of gamcr should not be set greater than 90 degrees. The default value is usually adequate for most applications.
(cref: Marquardt, 1963).

* del= Factor used in finite-difference equations (default 1e-5). Note del is used only when ider=1 for estimated partial derivative calculations.
(cref: $parms ider).

* zeta= Singularity criterion for matrix inversion (default 1e-31), which may be selected greater than or equal to the machine smallest exponent range.

* iout= Printout file06 and file16 control.
= 1 (default) for print output on both file06 and file16.
= 0 for print output only on file06.
Note: file16 output may be useful for deferred output when running the job from a time-sharing terminal; also, file16 may be used as an input file for other processing programs (e.g., plot routines). For this version, file06 output has been purposely reduced for time-sharing terminal use; however, for iout=1 (default), a complete printable output is always given on file16.
(cref: $parms iprt).

sp= scalep (equivalent names) is a parameter scaling option.
= 0 (default) to ignore parameter scaling (i.e., unscaled parameters).
= 1 (recommended for program IMSLPW) to scale parameters b(j) using ln(b(j)), provided the initial b(j)>0 for all j=1,2,...,k. Note scalep=1 will automatically constrain the final solution space such that b(j)>0 for all j in (1,k).
= 2 to scale parameters b(j) using arcsinh(b(j)). This option allows for log-type parameter scaling whenever b(j) is positive or negative for any j in (1,k). However, for program IMSLPW, the initial parameters b(j)>0 must be given; hence sp=2 should not be used (sp=2 is defined here for possible use

in other applications).
(cref: $parms b,k).

* sy= scaley (equivalent names) is an observation scaling option.
= 0 (default) to ignore observation scaling (i.e., unscaled observations y(i)).
= 1 to scale observations y(i) using ln(y(i)), provided y(i)>0 for all i=1,2,...,n.
= 2 to scale observations y(i) using arcsinh(y(i)). This option allows for log-type observation scaling whenever y(i) is positive, negative, or zero for any i in (1,n). A special case automatically occurs whenever sy=2, iob=5, and both resistivity and phase data are included; in this case, the MARQRT subprogram will use ln(resistivity) or arcsinh(phase) accordingly.
(cref: $init iob, $parms n and DATA MATRIX NOTES)

b()= Array of initial guesses for all k-parameters. These values must be supplied greater than zero for program IMSLPW (i.e., positive resistivities and thicknesses). The default values are set to b(j)=0 for all j=1 to k, and would result in an error condition if any b(j) was not supplied greater than zero.

The parameter order must be given as:

b(1),b(2),...,b(mm) are the mm layer resistivities (in ohm-meters), and

b(mm+1),b(mm+2),...,b(2*mm-1) are the mm-1 layer thicknesses (in meters).
(cref: $parms k,ip,ib and $init mm,iob).

ib()= Array of ip-indicies (in any order) corresponding to any b() parameter to hold fixed to its input value. e.g., ip=2,ib(1)=3,ib(2)=5 will hold fixed b(3), b(5) in the least squares. If ip=0 (default), leave out array ib in the namelist.
(cref: $parms ip,b).

[the following $parms are parameters used only by IMSL subprogram ZXSSQ, and cannot be used if converting to subprogram MARQRT as described in appendix 2].

iopt= 1 (default) implies strict descent of the sum of squares is desired in the derivative-free Marquardt algorithm (ZXSSQ), with default values used in the input array parm().
= 0 implies strict descent is not necessary (i.e., the

"best" or optimum Marquardt parameter used may not yield a strict decreasing sum of squares at each iteration).

= 2 implies strict descent is desired with user parameter choices as given (or assumed) in input array parm().
(cref: $parms parm()).

parm()= array of length 4 required only when iopt=2. The default is parm()=.01,2.,120.,.1, where each element is defined by the corresponding index as follows:

i=1, the initial value of the Marquardt parameter used to scale the diagonal of the approximate Hessian matrix, xjtj, by the factor (1.0+parm(1)). A small value gives a Newton step, while a large value gives a steepest descent step. (default parm(1)=.01).

i=2, the scaling factor used to modify the Marquardt parameter, which is decreased by parm(2) after an immediately successful descent direction, and increased by the square of parm(2) if not. (default parm(2)=2 where parm(2)>1 must be used).

i=3, an upper bound for increasing the Marquardt parameter. The search for a descent point is abandoned if parm(3) is exceeded. parm(3)>100 is recommended. (default parm(3)=120).

i=4, value for indicating when central rather than forward differencing is to be used for calculating the Jacobian (partial derivatives). The switch is made when the norm of the gradient of the sum of squares function becomes smaller than parm(4). Central differencing is good in the vicinity of the solution, so parm(4) should be small. (default parm(4)=.1).
(cref: $parms iopt).

nsig= The first convergence criterion. Convergence is satisfied if on 2 successive iterations, the parameter estimates agree, component by component, to nsig digits. (default nsig=3; using nsig>5 may not converge since single precision is used).

eps= The second convergence criterion. Convergence is statisfied if on 2 successive iterations the residual sum of squares estimates have relative differences <= eps. (default eps=0.0).
(cref: $parms e, which is equivalent to eps).

delta= The third convergence criterion. Convergence is statisfied if the Euclidean norm of the approximate gradient is <= delta. (default

delta=0.0).

Note: The Marquardt iteration is terminated, and convergence is considered achieved, if any one of the three convergence conditions (nsig,eps, or delta) is satisfied.

maxfn= The maximum number of function evaluations (i.e., calls to subroutine FUNC in ZXSSQ) allowed. The actual number of calls to FUNC may exceed maxfn slightly. Note: unless maxfn>0 is given, maxfn=2*k*niter is used as the default value. (cref: $parms k,niter, where default niter=10).

$end [end of $parms namelist]

$init parameters (with defaults and cross-references):

iob= Observation-type defined for y(i):
= 1 (default) defines y(i) as the apparent resistivity.
= 2 defines y(i) as the phase of surface impedance, expressed in (-180,+180) degrees.
(note: for iob<=2, m=1 must also be given in $parms).
= 3 behaves like iob=5 (see below).
= 4 behaves like iob=5 (see below).
= 5 defines mixed observation-type frequency soundings where the i-th observation type is given by x(i,2)=1.0 for apparent resistivity, or =2.0 for phase of surface impedance.
(note: for iob=5, m=2 must also be given in $parms).
(cref: $parms m,b(), $init mm, and DATA MATRIX NOTES).

mm= Number of layers in the model (1<=mm<=10; default mm=1).
Note: make sure $parms k=2*mm-1.
(cref: $parms k,b(), $init iob).

$end [end of $init parameters]

DATA MATRIX NOTES.

The data matrix is defined as the sequence of ordered rows: (y(i),x(i,j),j=1,m*), where i=row number 1,2,...,n, and m*=m+1 if iwt=1, otherwise m*=m<=2. The data matrix is

read on logical unit ialt (default 10) using an object-time format statement (see any Fortran manual). The number of items read depends on $parms m,iwt and $init iob as previously defined. The various data matrix options are summarized as follows:

(a) Specific observation type, frequency sounding for apparent resistivity or phase of surface impedance (iob<=2, m=1, and max. 3 items per record):

1. y(i)= i-th observation, where $init iob<=2 defines the particular type.
2. x(i,1)= i-th frequency (x(i,1)>0.0 Hz.).
3. x(i,2)= standard deviation of observation i (include only if iwt=1).

(b) Mixed observation types: apparent resistivity and/or phase of surface impedance frequency soundings (iob=5, m=2, and max. 4 items per record):

1. y(i)= i-th observation (where actual type is defined by x(i,2)).
2. x(i,1)= i-th frequency (x(i,1)>0.0 Hz.).
3. x(i,2)= observation type in y(i); use x(i,2)= 1.0 for apparent resistity, or =2.0 for phase of surface impedance.
4. x(i,3)= standard deviation of observation i (include only if iwt=1). Note: for joint inversion of apparent resistivity and phase data, a weighted least squares should be used (iwt=1 option) to produce near-equal magnitudes.

The data matrix should be grouped or ordered with equal consecutive frequencies with respect to each observation type.

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING.

1. Apparent resistivity sounding (default iob=1):

```
example 1.
$parms n=60,k=5,m=1,iprt=-1,sp=1,ialt=5,
 b=10,20,30,100,200$
(2f10.0)
1.98        1.
1.85        1.6
--(etc. for 58 more observations)--
$init mm=3$
```

2. Mixed observation types (iob=5), apparent resistivity and phase of surface impedance, and weighted observations (iwt=1):

```
example 2
$parms n=100,k=5,m=2,iprt=-2,sp=1,iwt=1,ialt=5,
 b=10,20,30,100,200$
(4f10.0)
20.1        1.2         1.          .05
52.4        1.2         2.          .08
29.87       4.          1.          .05
55.23       4.          2.          .09
--(etc. for rest of soundings)--
$init mm=3,iob=5$
```

SPECIAL OBJECT FORMAT PHRASES.

One may use special Fortran object formats to skip observations without changing the data matrix. For example, if we wish to use only the phase data in example 2 above, we could set n=50 and use the format (/4f10.0). Similarly, if we wanted only apparent resistivities to be used in example 2, then the format (4f10.0/) would accomplish the desired result.

Also, if an existing data matrix file does not have the proper defined column ordering in the form (y(i),x(i,j),j=1,m)), then the Fortran "tn" format phrase may be used to begin at any column n in the data record. For example, the format (t41,f10.0,t1,3f10.0) will select y(i) using col.41-50 and x(i,1) beginning at col.1.

MULTICS OPERATING INSTRUCTIONS.

1. Initially, one should add the following libraries (via the command "asr") to his search rules after the working directory: >udd>Emod1_inv>WAnderson>lib_em, >udd>Emod1_inv>WAnderson>lib_1, and >iml>imsl.

2. Either attach "file05" to a predetermined ascii (stream) parameter file, or let file05 default to "user_input" (i.e., the user's terminal). The order of parameters and data on file05 must be given as defined in the section PARAMETERS AND DATA REQUIRED above. To attach file05, type:
   io attach file05 vfile_ parameter_file_name

3. Attach "file10" to an input data matrix ascii file if ialt=10 (default) is used. If ialt=5 is selected, then

ignore this step, but include the data matrix following the object-time format on "file05"--see examples 1 and 2 above. In practice, it is usually best to use distinct files file05 and file10 for parameters and data respectively. To attach file10, type:
io attach file10 vfile_ data_file_name

4. Set the underflow condition handler off by typing: set_ufl -off

5. Execute program IMSLPW by typing: imslpw

If file05 was not attached, then the user must anticipate the required title, $parms, object format, and $init to be typed on "user_input". Prompt messages are not printed on the terminal.

Note "file16" is the complete print file (normally disk on Multics), and "file06" is always the on-line terminal print file. File16 should either be deleted or dprinted to a line-printer after running program IMSLPW. Also, file13 (if used) should be deleted after running the program. To submit the job as a batch job (called absentee on Multics), prepare step 1-5 above in a segment with .absin suffix and use the "enter_abs_request" command.

ERROR MESSAGES.

Most parameter and/or data errors are noted by self-explanatory messages appearing in the printed file(s), and the job is terminated. For example, the message "error--some $parms out of range" means that a violation (or omission) of a required parameter range has been committed in the $parms namelist. Check all $parms values, correct, and resubmit the job.

Exponent underflow may occur when the argument is less than 10**-38 on Multics; this is ok since 0.0 replaces all underflows. To suppress the underflow messages, the command "set_ufl -off" can be used prior to executing IMSLPW.

Exponent overflow and/or arithmetic overflow messages will terminate the run under Multics control. An overflow condition usually means a very poor initial parameter estimate was given in array b() for the model (mm) chosen. First check that all $parms, $init, data matrix values, and object-time format are correct. If no errors are found, then try to revise the model (mm) and/or use better guessed estimates for the starting parameters in array b().

If any parameter begins to approach zero or become unbounded during the least squares iterations, then one may fix (constrain) the parameter to a reasonable value, and restart the program to obtain a constrained least squares solution. This is usually required when the data are not sufficient to resolve all the parameters for the model mm chosen.

REFERENCES.

Anderson, W.L., 1979, Program MARQLOOPS: Marquardt inversion of loop-loop frequency soundings: U.S. Geol. Survey Open-File Rept. 79-240, 75p.

International Mathematical and Statistical Libraries (IMSL), 1977, 7500 Bellaire Blvd., 6th Floor, GNB Bldg., Houston, Texas 77036.

Marquardt, D.W., 1963, An algorithm for least-squares estimation of nonlinear parameters: J. Soc. Indust. Appl. Math, v.11, no. 2, pp. 431-441.

Tabata, T. and Ito,R., 1973, Effective treatment of the interpolation factor in Marquardt's nonlinear least-squares fit algorithm: The Computer Journal, v. 18, no.3, pp. 250-251.

Vozoff, K., 1972, The magnetotelluric method in the exploration of sedimentary basins: Geophysics, v. 37, no. 1, p. 98-141.

Wait, J.R., 1962, Electromagnetic waves in stratified media: The Macmillian Co., N.Y., 372 p.

Appendix 1.-- Source listing

The attached subprograms are listed with beginning line numbers in the following order:

```
C--IMSLPW: PLANE-WAVE INVERSION USING IMSLMQ    (12/14/78)    00000010
      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                         00000080
      SUBROUTINE WARN(MSG,M5,I6,I9,*)                         00000310
      SUBROUTINE POLAR2(Z,AMP,PHZ180)                         00000510
      INTEGER FUNCTION LOC(I,J)                               00000800
      SUBROUTINE IMSLMQ(SUBZ,SUBEND)                          00000910
      SUBROUTINE FPXSSQ(C,N,KIP,F)                            00004080
      SUBROUTINE LNXSSQ(C,N,KIP,F)                            00004410
      SUBROUTINE FCODE(Y,X,B,PRNT,F,I,IDER)                   00004540
      SUBROUTINE IMSLPW_SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)   00005140
      SUBROUTINE IMSLPW_SUBEND(Y,X,B,K,N,TITLE,IOUT)          00005650
```

Source Availability

The current version of the source code may be obtained by writing directly to the author. A magnetic tape copy of the source code will be sent to requestors to be copied and returned to the author. This method of releasing the program was selected in order to satisfy requests for the latest updated version. The magnetic tape will be recorded in the following mode (unless otherwise requested):

> Industry compatible: 9-track, unlabeled, EBCDIC mode, odd-parity, 800 bpi density, 80-character records (unblocked card images), and contained on one file.

Note: The source code for all IMSL Library routines used (ZXSSQ, LEQT1P, LUDECP, LUELMP, LINV1P, and UERTST) are only available from International Mathematical and Statistical Libraries (1977), whose address is given in the reference.

```
C--IMSLPW: PLANE-WAVE INVERSION USING IMSLMQ   (12/14/78)                00000010
C** HONEYWELL MULTICS VERSION **                                          00000020
C                                                                         00000030
      EXTERNAL IMSLPW_SUBZ,IMSLPW_SUBEND                                  00000040
      CALL IMSLMQ(IMSLPW_SUBZ,IMSLPW_SUBEND)                              00000050
      STOP                                                                00000060
      END                                                                 00000070

      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                     00000080
C--ERROR MESSAGE WRITE ROUTINE AND STOP, WHERE--                          00000090
C                                                                         00000100
C       MSG=    ANY MULTIPLE OF 5 CHARACTERS--MAX. OF 120                 00000110
C               (USE NH--------- FORM FOR ANSI COMPATABILITY)             00000120
C       M5=     NO.CHARS IN MSG/5 (REMAINDER MUST BE 0) 1.LE.M5.LE.24     00000130
C       I6=     1ST UNIT  FOR WRITE(I6, ) MSG -- USUALLY I6=6 FOR LPT.    00000140
C               IF I6.LE.0 UNIT I6 IGNORED.                               00000150
C       I9=     2ND UNIT  FOR WRITE(I9, ) MSG --                          00000160
C               IF I9.LE.0, UNIT I9 IGNORED.                              00000170
C--MESSAGE WRITTEN IN FORM--                                              00000180
C       /ERROR--MSG HERE                                                  00000190
C                                                                         00000200
      DIMENSION MSG(30)                                                   00000210
      J=5*M5                                                              00000220
      K=J/4+MOD(J,4)                                                      00000230
      IF(I6.GT.0) WRITE(I6,10) (MSG(I),I=1,K)                             00000240
   10 FORMAT(/8H ERROR--,30A4)                                            00000250
      IF(I9.GT.0) WRITE(I9,10) (MSG(I),I=1,K)                             00000260
      CALL CLOSE_FILE('-ALL')                                             00000270
C                                                                         00000280
      STOP                                                                00000290
      END                                                                 00000300

      SUBROUTINE WARN(MSG,M5,I6,I9,*)                                     00000310
C--WARNING MESSAGE WRITE ROUTINE WHERE:                                   00000320
C                                                                         00000330
C     MSG=      'ANY MULTIPLE OF 5 CHARACTERS--MAX. OF 120    '           00000340
C     M5=       NO.CHAR'S IN MSG/5 (REMAINDER MUST BE 0) 1<=M5<=24        00000350
C     I6=       1ST UNIT # FOR WRITE(I6, ) MSG -- USUALLY I6=6 FOR LPT.   00000360
C               IF I6<=0 UNIT I6 IGNORED.                                 00000370
C     I9=       2ND UNIT # FOR WRITE(I9, ) MSG -- IF I9<=0, UNIT I9 IGNO00000380
C     *=        $LABEL NO. TO RETURN AFTER ERROR CALLED ($NO. MUST BE GI00000390
C--MESSAGE WRITTEN IN FORM:                                               00000400
C     '0WARNING--'MSG HERE''                                              00000410
C                                                                         00000420
      DIMENSION MSG(30)                                                   00000430
      J=5*M5                                                              00000440
      K=J/4+MOD(J,4)                                                      00000450
      IF(I6.GT.0) WRITE(I6,6) (MSG(I),I=1,K)                              00000460
6     FORMAT('0WARNING--',30A4)                                           00000470
      IF(I9.GT.0) WRITE(I9,6) (MSG(I),I=1,K)                              00000480
      RETURN (1)                                                          00000490
      END                                                                 00000500
```

```
      SUBROUTINE POLAR2(Z,AMP,PHZ180)                                   00000510
C        PARMS  Z   = GIVEN COMPLEX COORDS Z=(X,Y)                      00000520
C               AMP= COMPUTED AMPLITUDE.                                00000530
C               PHZ180 = COMPUTED PHASE IN (-180.0,180.0) DEGREES.      00000540
C                                                                       00000550
      COMPLEX Z                                                         00000560
      DATA PI,PI2/3.1415927,6.2831853/                                  00000570
      ZR=REAL(Z)                                                        00000580
      ZI=AIMAG(Z)                                                       00000590
      IF(ZR.EQ.0.AND.ZI.EQ.0) GO TO 9                                   00000600
      PV=ATAN2(ABS(ZI),ABS(ZR))                                         00000610
      IF(ZI.GE.0.AND.ZR.GE.0) GO TO 10                                  00000620
      IF(ZI.GE.0.AND.ZR.LT.0) GO TO 20                                  00000630
      IF(ZI.LT.0.AND.ZR.LE.0) GO TO 30                                  00000640
      RAD=PI2-PV                                                        00000650
      GO TO 40                                                          00000660
    9 PHZ180=0.                                                         00000670
      AMP=0.                                                            00000680
      RETURN                                                            00000690
   10 RAD=PV                                                            00000700
      GO TO 40                                                          00000710
   20 RAD=PI-PV                                                         00000720
      GO TO 40                                                          00000730
   30 RAD=PI+PV                                                         00000740
   40 AMP=SQRT(ZR*ZR+ZI*ZI)                                             00000750
      PHZ180=57.29577951*RAD                                            00000760
      IF(PHZ180.GT.180.0) PHZ180=PHZ180-360.0                           00000770
      RETURN                                                            00000780
      END                                                               00000790

      INTEGER FUNCTION LOC(I,J)                                         00000800
C--GETS ACTUAL ADDR OF A(I,J)=A(J,I) SYMMETRIC MATRIX                   00000810
C  STORED AS THE VECTOR A(LOC(I,J)) OF N*(N+1)/2 ELEMENTS--             00000820
C  WHERE ANY I,J.LE.N MAY BE USED (N NOT EXPLICITLY NEEDED)...          00000830
C                                                                       00000840
      IF(I-J) 10,20,20                                                  00000850
   10 LOC=I+(J*J-J)/2                                                   00000860
      RETURN                                                            00000870
   20 LOC=J+(I*I-I)/2                                                   00000880
      RETURN                                                            00000890
      END                                                               00000900

      SUBROUTINE IMSLMQ(SUBZ,SUBEND)                                    00000910
C--IMSLMQ-- DERIVATIVE-FREE 'IMSL' MARQUARDT INVERSION--10/5/78         00000920
C  FOR SOLVING GENERAL NONLINEAR LEAST SQUARES PROBLEMS. USER NEED ONLY 00000930
C  WRITE SUBROUTINES 'FCODE, SUBZ, AND SUBEND' EXACTLY AS USED          00000940
C  IN PROGRAM 'MARQRT'.  ALSO, THE SAME PARAMETER FILE05 AND DATA       00000950
C  FILE10 MAY BE USED BY 'IMSLMQ' AS IN 'MARQRT'.                       00000960
C                                                                       00000970
C--NOTE: 'FCODE' CANNOT BE PASSED AS EXTERNAL DUE TO THE                00000980
C  'BLACK-BOX' NATURE OF IMSL ROUTINE 'ZXSSQ' (SEE IMSL DOC.).          00000990
C  THUS, ONE SHOULD RENAME ACTUAL NAME TO 'FCODE' FOR USE HERE.         00001000
```

```
C  (I.E., SEE CALL FCODE IN 'FPXSSQ'--EXTERNAL FUNCTION FOR ZXSSQ).      00001010
C                                                                        00001020
C--THE USER MUST DECLARE THE CALLING PARAMETERS                          00001030
C  SUBZ,SUBEND (ANY DESIRED NAMES MAY BE USED) AS EXTERNAL IN            00001040
C  MAIN CALLING PROGRAM; E.G.,                                           00001050
C                                                                        00001060
C       EXTERNAL SUBZ,SUBEND                                             00001070
C       CALL IMSLMQ(SUBZ,SUBEND)                                         00001080
C       STOP                                                             00001090
C       END                                                              00001100
C                                                                        00001110
C--THIS INTERFACE BETWEEN 'MARQRT' AND 'IMSLMQ' WAS WRITTEN BY           00001120
C  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.               00001130
C                                                                        00001140
C--SEE DOCUMENTATION OF 'MARQLOOPS', USGS OPEN-FILE REPT 79-240 (1979),  00001150
C  FOR DETAILS ON CODING THE REQUIRED SUBROUTINES FCODE,SUBZ, AND        00001160
C  SUBEND. ALSO SEE IMSL DOCUMENTATION FOR 'ZXSSQ'.                      00001170
C                                                                        00001180
C--THE INPUT ORDER ON FILE05 (PARAMETER FILE) IS:                        00001190
C                                                                        00001200
C  1.  TITLE  (MAX. 80-CHARACTER TITLE--ALWAYS REQUIRED).                00001210
C  2.  $PARMS (SAME PARMS DEFINITIONS FOR PGM MARQRT FOR PARAMETERS:     00001220
C         N,K,IP,M,IALT,ISTOP,IWT,NITER,E,SCALEP,B(),IPRT, AND IB()).    00001230
C         PLUS, ADDITIONAL PARMS FOR 'ZXSSQ' (SEE IMSL DOC):             00001240
C         IOPT,NSIG,MAXFN,EPS,DELTA,PARM(4).                             00001250
C  3.  (OBJECT-TIME FORMAT STATEMENT) FOR READING THE DATA MATRIX        00001260
C       (Y(I),X(I,J),J=1,M*) ON FILE IALT (DEFAULT 10), WHERE M*=M+IWT.  00001270
C  (3A.  INSERT DATA MATRIX HERE ONLY IF IALT=5)                         00001280
C  4.  $INIT OPTIONAL NAMELIST FOR READING ADDITIONAL PARMS IN           00001290
C           SUBROUTINE SUBZ  (WHICH MAY BE A DUMMY SUBROUTINE).          00001300
C  5.  OPTIONALLY, REPEAT STEPS 2-4, IF ISTOP=0 WAS USED IN STEP 2.      00001310
C                                                                        00001320
C--OUTPUT IS GIVEN ON FILE06 (ON-LINE USUALLY), AND                      00001330
C  ON FILE16 (CONTAINS ALL PRINTABLE OUTPUT); FILE06 CONTAINS ONLY       00001340
C  OUTPUT VIA PARM IPRT (0--ABBREVIATED, 1 OR -2 --DETAIL).              00001350
C                                                                        00001360
C--------------------------------------------------------------------    00001370
C--THE USER SHOULD ADD >IML>IMSL TO THE SEARCH RULES ON MULTICS.         00001380
C  (IMSL ROUTINES USED: ZXSSQ,LEQT1P,LUDECP,LUELMP,UERTST,LINV1P)        00001390
C--------------------------------------------------------------------    00001400
C                                                                        00001410
      DIMENSION B(20),GRAD(20),TITLE(16),H(270),COV(20,20),SE(20)        00001420
      DIMENSION SQWT(200),IB(20),PRNT(5),C(20),INDEX(20),                00001430
     & XJAC(200,20),XJTJ(210),WORK(710),PARM(4),F(200)                   00001440
      INTEGER SP,SCALEP,SY,SCALEY                                        00001450
C--THE FOLLOWING CHARACTER STATEMENTS ONLY FOR HONEYWELL MULTICS:         00001460
      CHARACTER*4 FMT(18)                                                00001470
      CHARACTER*5 TITLE                                                  00001480
      COMMON/FIXDAT/Y(200),X(200,5),BFIX(20),YMAX,IIB(20),IIP,NOBS,K     00001490
      COMMON/PRT/IPRT                                                    00001500
      EXTERNAL FPXSSQ,LNXSSQ                                             00001510
      EQUIVALENCE (SQWT(1),X(1,5)),(SP,SCALEP),(N,NOBS),                 00001520
```

```
     & (M,NVARS),(NITER,LIMIT),(SY,SCALEY),(E,EPS),(SS,SSQ),                00001530
     & (IB(1),IIB(1)),(IP,IIP),(B(1),BFIX(1))                               00001540
      NAMELIST/PARMS/N,K,IP,M,IALT,NITER,IB,E,B,IWT,ISTOP,SP,SY,            00001550
     & SCALEP,SCALEY, IDER,IPRT,INON,FF,T,TAU,XL,MODLAM,GAMCR,              00001560
     & DEL,ZETA,IOUT,IOPT,NSIG,MAXFN,EPS,DELTA,PARM                         00001570
C-- NOTE NAMELIST PARMS INCLUDED (BUT IGNORED) FOR COMPATIBILITY           00001580
C  ARE: IDER,INON,FF,T,TAU,XL,MODLAM,GAMCR,DEL,ZETA,IOUT.                  00001590
C  ALSO, SP=SCALEP WILL BE CONSIDERED MODE=1 (ALOG OPTION) ONLY            00001600
C  WHEN SP=2 OR 1 (AND MODE=0 LINEAR WHEN SP=0).  SY=SCALEY IS             00001610
C  IGNORED FOR THIS VERSION OF 'IMSLMQ'.                                   00001620
C                                                                          00001630
C--READ IMSLMQ TITLE LINE                                                  00001640
      READ(5,4) TITLE                                                      00001650
4     FORMAT(16A5)                                                         00001660
C--PRESET DEFAULTS                                                         00001670
      N=0                                                                  00001680
      K=0                                                                  00001690
      M=0                                                                  00001700
      IP=0                                                                 00001710
      IPRT=0                                                               00001720
      ISTOP=1                                                              00001730
      IWT=0                                                                00001740
      IALT=10                                                              00001750
      NITER=10                                                             00001760
      SP=0                                                                 00001770
      MODE=0                                                               00001780
      FMIN=0.0                                                             00001790
      E=1.E-3                                                              00001800
      IOPT=1                                                               00001810
      NSIG=3                                                               00001820
      MAXFN=0                                                              00001830
      EPS=0.0                                                              00001840
      DELTA=0.0                                                            00001850
      PARM(1)=.01                                                          00001860
      PARM(2)=2.                                                           00001870
      PARM(3)=120.                                                         00001880
      PARM(4)=.1                                                           00001890
      DO 5 I=1,20                                                          00001900
      IB(I)=0                                                              00001910
      B(I)=0.0                                                             00001920
5     GRAD(I)=0.0                                                          00001930
C--READ $PARMS                                                             00001940
6     READ(5,PARMS)                                                        00001950
C--TEST $PARMS BEFORE PROCEEDING                                           00001960
      IF(N.GT.200.OR.K.GT.20.OR.M.GT.4.OR.IWT.GT.1.OR.IP.GT.19.OR.         00001970
     & N.LT.1.OR.K.LT.1.OR.M.LT.1.OR.IWT.LT.0.OR.IP.LT.0.OR.               00001980
     & N.LT.K-IP.OR.IALT.EQ.6.OR.IALT.EQ.16)                               00001990
     &CALL ERRMSG('SOME $PARMS OUT OF RANGE.',5,6,16)                      00002000
      DO 7 I=1,K                                                           00002010
      IF(B(I).EQ.0.0)CALL ERRMSG('SOME B(I)=0.0  ',3,6,16)                 00002020
7     CONTINUE                                                             00002030
      IF(MAXFN.EQ.0) MAXFN=2*K*NITER                                       00002040
```

```
      IF(IP.LE.0) GO TO 9                                                00002050
      DO 8 I=1,IP                                                        00002060
      IF(IB(I).GT.0) GO TO 8                                             00002070
      CALL ERRMSG('IP.GT.1 BUT SOME IB(I).LE.0   ',6,6,16)               00002080
8     CONTINUE                                                           00002090
9     IF(SP.NE.0) MODE=1                                                 00002100
      IF(IPRT.EQ.1) IPRT=-2                                              00002110
C--READ OBJECT FORMAT FOR DATA MATRIX ON FILE IALT.                      00002120
      READ(5,10) (FMT(I),I=1,18)                                         00002130
10    FORMAT(18A4)                                                       00002140
      M1=M+IWT                                                           00002150
      YMAX=0.0                                                           00002160
      DO 11 I=1,N                                                        00002170
      READ(IALT,FMT) Y(I),(X(I,J),J=1,M1)                                00002180
      SQWT(I)=1.0                                                        00002190
      IF(IWT.EQ.1.AND.X(I,M1).NE.0.0) SQWT(I)=1.0/X(I,M1)                00002200
      IF(ABS(Y(I)).GT.YMAX) YMAX=ABS(Y(I))                               00002210
11    CONTINUE                                                           00002220
C--INITIALIZATION VIA CALL SUBZ (READ $INIT, TEST B,X,Y, ETC)            00002230
      CALL SUBZ(Y,X,B,PRNT,NDUM,N,TITLE,1)                               00002240
C--WRITE $PARMS ON UNIT 6 AND 16                                         00002250
      WRITE(6,60) TITLE,N,K,IP,M,E,IALT,ISTOP,IWT,NITER,SP,IPRT,         00002260
     & IOPT,NSIG,MAXFN,EPS,DELTA,PARM                                    00002270
60    FORMAT('1I M S L M Q  --     ',16A5//' N=',I5,9X,'K=',I4,10X,'IP='00002280
     & ,I4,9X,'M=',I3,11X,'E=',E10.3/' IALT=',I3,8X,'ISTOP=',I2,8X,     00002290
     & 'IWT=',I2,10X,'NITER=',I6,4X,'SCALEP=',I2/' IPRT=',I3,           00002300
     & 8X,'IOPT=',I2,9X,'NSIG=',I3,8X,'MAXFN=',I6,4X,'EPS=',E10.3/      00002310
     & ' DELTA=',E10.3/' PARM=',4E10.3)                                  00002320
      WRITE(16,60) TITLE,N,K,IP,M,E,IALT,ISTOP,IWT,NITER,SP,IPRT,        00002330
     & IOPT,NSIG,MAXFN,EPS,DELTA,PARM                                    00002340
      IF(IP.EQ.0) GO TO 661                                              00002350
      WRITE(6,660) (IB(I),I=1,IP)                                        00002360
660   FORMAT(/' IB=',19I3)                                               00002370
      WRITE(16,660) (IB(I),I=1,IP)                                       00002380
661   WRITE(6,662) FMT                                                   00002390
662   FORMAT(/' FMT=',18A4/)                                             00002400
      WRITE(16,662) FMT                                                  00002410
      WRITE(6,6000) (B(I),I=1,K)                                         00002420
6000  FORMAT(/' INITIAL PARAMETERS'//(5E16.8))                           00002430
      WRITE(16,6000) (B(I),I=1,K)                                        00002440
C--INTERFACE WITH ZXSSQ USING MARQRT FCODE                               00002450
      DO 1 I=1,20                                                        00002460
1     INDEX(I)=I                                                         00002470
      KIP=K-IP                                                           00002480
      IF(IP.EQ.0) GO TO 400                                              00002490
C--REORDER B TO C WHEN IP>0                                              00002500
      IM=0                                                               00002510
      DO 202 I=1,K                                                       00002520
      DO 201 J=1,IP                                                      00002530
      IF(I.EQ.IB(J)) GO TO 202                                           00002540
201   CONTINUE                                                           00002550
      IM=IM+1                                                            00002560
```

```
      C(IM)=B(I)                                                        00002570
      INDEX(IM)=I                                                       00002580
202   CONTINUE                                                          00002590
      WRITE(6,203) (I,I=1,K)                                            00002600
203   FORMAT(/' PARAMETER INDEX:',20I3)                                 00002610
      WRITE(16,203) (I,I=1,K)                                           00002620
      WRITE(6,204) (INDEX(I),I=1,KIP)                                   00002630
204   FORMAT(' REORDERED AS...:',20I3)                                  00002640
      WRITE(16,204) (INDEX(I),I=1,KIP)                                  00002650
      WRITE(6,206) (C(I),I=1,KIP)                                       00002660
206   FORMAT(/' REORDERED PARAMETERS'//(5E16.8))                        00002670
      WRITE(16,206) (C(I),I=1,KIP)                                      00002680
      GO TO 500                                                         00002690
400   DO 401 I=1,K                                                      00002700
401   C(I)=B(I)                                                         00002710
500   CONTINUE                                                          00002720
      IF(MODE.EQ.0) GO TO 12                                            00002730
C--LOG PARAMETERS CHOSEN (MODE=1 OR SP.NE.0)                            00002740
      DO 111 I=1,KIP                                                    00002750
      IF(C(I).LE.0.0)CALL ERRMSG('SP.NE.0 & SOME B(I).LE.0.',5,6,16)    00002760
111   C(I)=ALOG(C(I))                                                   00002770
      CALL ZXSSQ(LNXSSQ,N,KIP,NSIG,EPS,DELTA,MAXFN,IOPT,PARM,           00002780
     & C,SSQ,F,XJAC,200,XJTJ,WORK,INFER,IER)                            00002790
      DO 1111 I=1,KIP                                                   00002800
1111  C(I)=EXP(C(I))                                                    00002810
      GO TO 21                                                          00002820
C--LINEAR PARAMETERS CHOSEN (MODE=0 OR SP=0)                            00002830
12    CALL ZXSSQ(FPXSSQ,N,KIP,NSIG,EPS,DELTA,MAXFN,IOPT,PARM,           00002840
     & C,SSQ,F,XJAC,200,XJTJ,WORK,INFER,IER)                            00002850
C--ZXSSQ ERRMSG CODE HANDLERS                                           00002860
21    IF(IER.EQ.0) GO TO 100                                            00002870
      IF(IER.EQ.129)                                                    00002880
     &CALL ERRMSG('SINGULARITY DETECTED IN JACOBIAN & RECOVERY FAILED',  00002890
     & 10,6,16)                                                         00002900
      IF(IER.EQ.130)                                                    00002910
     &CALL ERRMSG('N,K-IP,IOPT,PARM(1) OR PARM(2) INCORRECT',8,6,16)    00002920
      IF(IER.EQ.131)                                                    00002930
     &CALL WARN('MARQUARDT PARAMETER EXCEEDED PARM(3)    ',8,6,16,$100) 00002940
      IF(IER.EQ.132) CALL ERRMSG(                                       00002950
     &'AFTER RECOVERY FROM SINGULAR JACOBIAN, B CYCLED BACK AGAIN..',    00002960
     & 12,6,16)                                                         00002970
      IF(IER.EQ.133)CALL WARN('MAXFN EXCEEDED.',3,6,16,$100)            00002980
      IF(IER.EQ.38)CALL WARN('JACOBIAN=0. SOLUTION IS STATIONARY POINT',00002990
     & 8,6,16,$100)                                                     00003000
100   WRITE(6,603) (WORK(I),I=1,5),INFER,IER,SSQ,(C(I),I=1,KIP)         00003010
      WRITE(16,603) (WORK(I),I=1,5),INFER,IER,SSQ,(C(I),I=1,KIP)        00003020
603   FORMAT(//' $$$$  IMSLMQ CONVERGENCE INFORMATION:'//               00003030
     & ' NORM OF GRADIENT',T32,E16.8/' FUNCTION EVALUATIONS',T32,E16.8/ 00003040
     & ' EST. SIGN. DIGITS',T32,E16.8/' MARQUARDT PARAMETER',T32,       00003050
     & E16.8/' NO. ITERATIONS',T32,E16.8/' TYPE CONVERGENCE (INFER)',   00003060
     & T32,I3/' ERROR CODE (IER)',T32,I5/                               00003070
     & ' RESIDUAL SUM-OF-SQUARES (SSQ)=',E16.8//                        00003080
```

```
     &' **** FINAL UNSCALED PARAMETERS'//                                    00003090
     & (5E16.8))                                                            00003100
99    KK=MAX0((KIP+1)*KIP/2,5)                                              00003110
      DO 80 I=1,KIP                                                         00003120
80    GRAD(I)=2.*WORK(KK+I)                                                 00003130
      WRITE(6,82) (GRAD(I),I=1,KIP)                                         00003140
82    FORMAT(/' SCALED GRADIENT'//(5E16.8))                                 00003150
      WRITE(16,82) (GRAD(I),I=1,KIP)                                        00003160
      IF(IPRT.EQ.-2) WRITE(6,699)                                           00003170
699   FORMAT(/3X,'I',4X,'OBS.Y(I)',6X,'CAL',11X,'RES',8X,'X(I,1)',8X,       00003180
     & 'WT(I)')                                                             00003190
      WRITE(16,1699)                                                        00003200
1699  FORMAT(/3X,'I',4X,'OBS.Y(I)',6X,'CAL',11X,'RES',8X,'X(I,1)',8X,       00003210
     & 'X(I,2)',8X,'X(I,3)',8X,'X(I,4)',8X,'WT(I)')                         00003220
      SUMF2=0.0                                                             00003230
      DO 110 I=1,NOBS                                                       00003240
      RES=F(I)/SQWT(I)                                                      00003250
      YCAL=Y(I)-RES                                                         00003260
      WT=SQWT(I)*SQWT(I)                                                    00003270
      SUMF2=SUMF2+F(I)*F(I)                                                 00003280
      IF(IPRT.EQ.-2) WRITE(6,210) I,Y(I),YCAL,RES,X(I,1),WT                 00003290
210   FORMAT(1X,I3,2E14.6,E11.3,2E14.6)                                     00003300
      WRITE(16,211) I,Y(I),YCAL,RES,(X(I,J),J=1,4),WT                       00003310
211   FORMAT(1X,I3,2E14.6,E11.3,5E14.6)                                     00003320
110   CONTINUE                                                              00003330
      IF(N.EQ.KIP) RMSERR=0.0                                               00003340
      IF(N.GT.KIP) RMSERR=SQRT(SUMF2/(N-KIP))                               00003350
      WRITE(6,604) RMSERR                                                   00003360
604   FORMAT(/' **** RMSERR=',E16.8)                                        00003370
      WRITE(16,604) RMSERR                                                  00003380
C--PRINT ON FILE16 (ONLY) THE FINAL SCALED PARTIALS (JACOBIAN)              00003390
      WRITE(16,605)                                                         00003400
605   FORMAT(/' FINAL SCALED PARTIALS (JACOBIAN)')                          00003410
      DO 112 I=1,NOBS                                                       00003420
      WRITE(16,606) I,(XJAC(I,J),J=1,KIP)                                   00003430
606   FORMAT(1X,I3,5E16.8/(4X,5E16.8))                                      00003440
112   CONTINUE                                                              00003450
C--GET INVERSE JACOBIAN TRANSPOSE*JACOBIAN (FROM XJTJ SYMMETRIC MATRIX)     00003460
      CALL LINV1P(XJTJ,KIP,H,5,D1,D2,IER)                                   00003470
      IF(IER.GT.128) CALL ERRMSG('IN LINV1P CALL.',3,6,16)                  00003480
C--FINAL STATISTICS                                                         00003490
      DO 301 I=1,KIP                                                        00003500
      DO 301 J=1,KIP                                                        00003510
301   COV(I,J)=H(LOC(I,J))                                                  00003520
      IF(IPRT.EQ.-2) WRITE(6,120)                                           00003530
120   FORMAT(/' SCALED COVARIANCE MATRIX (INVERSE OF XJTJ)')                00003540
      WRITE(16,120)                                                         00003550
      DO 122 I=1,KIP                                                        00003560
      SE(I)=SQRT(ABS(COV(I,I)))                                             00003570
      IF(IPRT.EQ.-2) WRITE(6,300) INDEX(I),(COV(I,J),J=1,KIP)               00003580
300   FORMAT(1X,I2,10E12.4/(3X,10E12.4))                                    00003590
      WRITE(16,300) INDEX(I),(COV(I,J),J=1,KIP)                             00003600
```

```
122   CONTINUE                                                          00003610
      IF(IPRT.EQ.-2) WRITE(6,304)                                       00003620
304   FORMAT(/' CORRELATION MATRIX')                                    00003630
      WRITE(16,304)                                                     00003640
      DO 131 I=1,KIP                                                    00003650
      IF(SE(I).EQ.0.0) GO TO 132                                        00003660
      DO 129 J=1,KIP                                                    00003670
      IF(SE(J).EQ.0.0) GO TO 129                                        00003680
      COV(I,J)=COV(I,J)/(SE(I)*SE(J))                                   00003690
129   CONTINUE                                                          00003700
133   IF(IPRT.EQ.-2) WRITE(6,300) INDEX(I),(COV(I,J),J=1,KIP)           00003710
      WRITE(16,300) INDEX(I),(COV(I,J),J=1,KIP)                         00003720
      GO TO 131                                                         00003730
132   COV(I,I)=1.0                                                      00003740
      GO TO 133                                                         00003750
131   CONTINUE                                                          00003760
  125 WRITE(6,303)                                                      00003770
  303 FORMAT(/15H   ** PARAMETER,3X,9HSTD ERROR,3X,                     00003780
     & 31HSTD ERROR/PARAMETER  (UNSCALED))                              00003790
      WRITE(16,303)                                                     00003800
      DO 126 I=1,KIP                                                    00003810
      SE(I)=RMSERR*SE(I)                                                00003820
      IF(SP.GT.0) SE(I)=C(I)*SE(I)                                      00003830
      SEC=SE(I)/C(I)                                                    00003840
      WRITE(6,300) INDEX(I),C(I),SE(I),SEC                              00003850
      WRITE(16,300) INDEX(I),C(I),SE(I),SEC                             00003860
126   CONTINUE                                                          00003870
      DO 600 I=1,KIP                                                    00003880
600   H(I)=C(I)                                                         00003890
      IF(IP.EQ.0) GO TO 601                                             00003900
C--PUT SOL C AND BFIX TOGETHER FOR SUBEND USE.                          00003910
      IM=0                                                              00003920
      DO 127 I=1,K                                                      00003930
      H(I)=B(I)                                                         00003940
      DO 128 J=1,IP                                                     00003950
      IF(I.EQ.IB(J)) GO TO 127                                          00003960
128   CONTINUE                                                          00003970
      IM=IM+1                                                           00003980
      H(I)=C(IM)                                                        00003990
127   CONTINUE                                                          00004000
601   CALL SUBEND(Y,X,H,K,N,TITLE,1)                                    00004010
      IF(ISTOP.NE.1) GO TO 6                                            00004020
C--FOLLOWING CALL ONLY FOR HONEYWELL MULTICS SYSTEM:                    00004030
999   CALL CLOSE_FILE('-ALL')                                           00004040
C     STOP                                                              00004050
      RETURN                                                            00004060
      END                                                               00004070

      SUBROUTINE FPXSSQ(C,N,KIP,F)                                      00004080
C--CALCULATES RESIDUAL VECTOR F(N) FOR 'ZXSSQ' (EXTERNAL FPXSSQ) FOR    00004090
C  UNSCALED C(KIP) PARAMETER VECTOR AND DATA X(200,5),Y(200) IN FIXDAT. 00004100
C                                                                       00004110
```

```
C       C= INPUT VECTOR OF PARAMETERS (LENGTH KIP)                      00004120
C       N= NO. OBS. <= 200                                              00004130
C     KIP= NO. PARAMETERS =K-IP (IP>=0)                                 00004140
C       F= OUTPUT VECTOR OF (WEIGHTED) FUNCTION RESIDUALS (LENGTH N)    00004150
C                                                                       00004160
C--CALLS 'FCODE' AS CODED FOR 'MARQRT' WITH FIXED DATA IN COMMON/FIXDAT/00004170
C                                                                       00004180
      DIMENSION C(1),F(1),PRNT(5),SQWT(200),BIP(20)                     00004190
      COMMON/FIXDAT/Y(200),X(200,5),BFIX(20),YMAX,IIB(20),IIP,NOBS,K    00004200
      EQUIVALENCE (SQWT(1),X(1,5))                                      00004210
      IF(IIP.GT.0) GO TO 2                                              00004220
      DO 1 I=1,N                                                        00004230
      CALL FCODE(Y,X,C,PRNT,FF,I,1)                                     00004240
1     F(I)=SQWT(I)*(Y(I)-FF)                                            00004250
      RETURN                                                            00004260
2     IM=0                                                              00004270
      DO 4 I=1,K                                                        00004280
      BIP(I)=BFIX(I)                                                    00004290
      DO 3 J=1,IIP                                                      00004300
      IF(I.EQ.IIB(J)) GO TO 4                                           00004310
3     CONTINUE                                                          00004320
      IM=IM+1                                                           00004330
      BIP(I)=C(IM)                                                      00004340
4     CONTINUE                                                          00004350
      DO 5 I=1,N                                                        00004360
      CALL FCODE(Y,X,BIP,PRNT,FF,I,1)                                   00004370
5     F(I)=SQWT(I)*(Y(I)-FF)                                            00004380
      RETURN                                                            00004390
      END                                                               00004400

      SUBROUTINE LNXSSQ(C,N,KIP,F)                                      00004410
C--INTERFACES TO 'FPXSSQ' TO ALLOW PARMS TO BE IN LOG OR LINEAR         00004420
C  SPACE OUTSIDE, BUT ALWAYS LINEAR WITHIN FPXSSQ.                      00004430
C                                                                       00004440
C--CALLS 'FPSXXQ' (AND IN TURN 'FCODE')                                 00004450
C                                                                       00004460
      DIMENSION C(1),F(1),CTEM(20)                                      00004470
      COMMON/PRT/IPRT                                                   00004480
      DO 1 I=1,KIP                                                      00004490
1     CTEM(I)=EXP_(C(I))                                                00004500
      CALL FPXSSQ(CTEM,N,KIP,F)                                         00004510
      RETURN                                                            00004520
      END                                                               00004530

      SUBROUTINE FCODE(Y,X,B,PRNT,F,I,IDER)                             00004540
C--FUNC FOR IMSLMQ USE                                                  00004550
C--FUNCTION EVALUATION FOR RHOA--APPARENT RESISTIVITY                   00004560
C  AND/OR PHASE OF SURFACE IMPEDANCE                                    00004570
C  FOR PLANE-WAVE (PW) MT OR AMT DATA.                                  00004580
      REAL Y(1),X(200,5),B(1),PRNT(5),F,EPS                             00004590
      REAL RHO(10),H(9),DD(9),SIG(10),KROOT(10),K(10),MU0               00004600
      COMPLEX Z1,ONE,GM,MN,CARG,EVD,T1,T2                               00004610
```

```
      COMMON/CTL/MN,MM,M1,M21,IOB                                          00004620
      DATA MU0/1.25663706E-6/,TWOPI/6.28318531/,ONE/(1.0,0.0)/             00004630
      DATA FREQL/0.0/                                                      00004640
      IF(I.GT.1.OR.MM.EQ.1) GO TO 20                                       00004650
      DO 10 J=2,MM                                                         00004660
      IF(B(J).EQ.B(J-1)) CALL ERRMSG(20HSOME RHO(J)=RHO(J-1),4,6,16)       00004670
   10 CONTINUE                                                             00004680
   20 DO 33 J=1,5                                                          00004690
   33 PRNT(J)=X(I,J)                                                       00004700
      FREQ=PRNT(1)                                                         00004710
      IF(I.EQ.1.OR.IDER.NE.0.OR.FREQ.NE.FREQL) GO TO 35                    00004720
      IF(IOB.EQ.5) GO TO 90                                                00004730
   35 RHO(MM)=B(MM)                                                        00004740
      IF(MM.EQ.1) GO TO 50                                                 00004750
      DO 40 J=1,M1                                                         00004760
      RHO(J)=B(J)                                                          00004770
   40 H(J)=B(J+MM)                                                         00004780
   50 CONTINUE                                                             00004790
C--GET RHOA FUNCTION                                                       00004800
      DO 11 II=1,MM                                                        00004810
11    SIG(II)=1.0/RHO(II)                                                  00004820
      SIG1=SIG(1)                                                          00004830
      DO 2 II=1,MM                                                         00004840
      K(II)=SIG(II)/SIG1                                                   00004850
2     KROOT(II)=SQRT(K(II))                                                00004860
      DEL=SQRT(1.0/(3.947841762E-6*SIG1*FREQ))                             00004870
      OMEGA=TWOPI*FREQ                                                     00004880
      IF(MM.EQ.1) GO TO 30                                                 00004890
      DO 3 II=1,M1                                                         00004900
3     DD(II)=2.828427124*H(II)/DEL                                         00004910
C--RECUR FOR GM                                                            00004920
30    GM=ONE                                                               00004930
      IF(MM.EQ.1) GO TO 5                                                  00004940
      J=M1                                                                 00004950
4     EVD=CEXP(-MN*CMPLX(KROOT(J)*DD(J),0.0))                              00004960
      CARG=(ONE-EVD)/(ONE+EVD)                                             00004970
      T1=CMPLX(KROOT(J),0.0)*GM                                            00004980
      T2=KROOT(J+1)                                                        00004990
      GM=(T1+T2*CARG)/(T2+T1*CARG)                                         00005000
      IF(J.EQ.1) GO TO 5                                                   00005010
      J=J-1                                                                00005020
      GO TO 4                                                              00005030
5     Z1=SQRT(MU0*OMEGA/SIG1)*MN*GM                                        00005040
      GO TO (111,222,90,90,90),IOB                                         00005050
111   F=REAL(Z1*CONJG(Z1))/(7.895683523E-6*FREQ)                           00005060
999   RETURN                                                               00005070
222   CALL POLAR2(Z1,AMP,F)                                                00005080
      GO TO 999                                                            00005090
90    IOBS=PRNT(2)                                                         00005100
      FREQL=FREQ                                                           00005110
      GO TO (111,222),IOBS                                                 00005120
      END                                                                  00005130
```

```
      SUBROUTINE IMSLPW_SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)                00005140
C--INITIALIZATION ROUTINE                                                  00005150
C--FOLLOWING CHARACTER STMT. ONLY FOR HONEYWELL MULTICS SYS:               00005160
      CHARACTER*5 TITLE(16)                                                00005170
      COMPLEX MN                                                           00005180
      REAL    Y(1),X(200,5),B(1),PRNT(1),EPS                               00005190
      REAL    RHO(10),H(9)                                                 00005200
      COMMON/CTL/MN,MM,M1,M21,IOB                                          00005210
      NAMELIST/INIT/MM,IOB                                                 00005220
        DATA ISUBZ/0/                                                      00005230
        IF(ISUBZ.NE.0) GO TO 10                                            00005240
C--PRESET                                                                  00005250
      MN=CSQRT(CMPLX(0.0,1.0))                                             00005260
        ISUBZ=1                                                            00005270
      MM=1                                                                 00005280
      IOB=1                                                                00005290
   10   READ(5,INIT)                                                       00005300
      WRITE(6,20) TITLE                                                    00005310
   20   FORMAT(15H1I M S L P W --,5X,16A5/)                                 00005320
      IF(IOUT.EQ.1) WRITE(16,20) TITLE                                     00005330
      WRITE(6,30) MM,IOB                                                   00005340
   30 FORMAT(4H0MM=,I3,6H  IOB=,I3)                                        00005350
      IF(IOUT.EQ.1) WRITE(16,30) MM,IOB                                    00005360
      IF(IOB.LT.1.OR.IOB.GT.5)CALL ERRMSG('IOB<1 OR >5    ',3,6,16)       00005370
C--TEST $INIT PARMS                                                        00005380
      IF(MM.LT.1.OR.MM.GT.10) CALL ERRMSG(                                 00005390
     120HMM.LT.1.OR.MM.GT.10 ,4,6,16)                                      00005400
C--TEST X(I, ) DATA BEFORE PROCEEDING--                                    00005410
   40 DO 50 I=1,N                                                          00005420
      IF(X(I,1).LE.0.0) CALL ERRMSG(                                       00005430
     120HSOME F=X(I,1).LE.0  ,4,6,16)                                      00005440
      IF(IOB.LE.2) GO TO 50                                                00005450
      IF(X(I,2).LT.1.0.OR.X(I,2).GT.2.0)                                   00005460
     & CALL ERRMSG('SOME X(I,2) OUT OF RANGE WHEN IOB>2',7,6,16)           00005470
   50 CONTINUE                                                             00005480
C--PRESET SOME GLOBAL CONSTANTS                                            00005490
      M1=MM-1                                                              00005500
      M21=2*MM-1                                                           00005510
      WRITE(6,60) (I,I,I=1,MM)                                             00005520
      IF(IOUT.EQ.1) WRITE(16,60) (I,I,I=1,MM)                              00005530
   60 FORMAT(///18H PARAMETER ORDER--//(5X,I3,6X,6H  RHO(,I3,1H)))         00005540
      IF(MM.EQ.1) GO TO 90                                                 00005550
      DO 70 I=1,M1                                                         00005560
      J=MM+I                                                               00005570
      IF(IOUT.EQ.1) WRITE(16,80) J,I                                       00005580
   70 WRITE(6,80) J,I                                                      00005590
   80 FORMAT(5X,I3,6X,6HTHICK(,I3,1H))                                     00005600
   90 NPRNT=2                                                              00005610
      IF(IOB.GT.2) NPRNT=3                                                 00005620
      RETURN                                                               00005630
      END                                                                  00005640
```

```
      SUBROUTINE IMSLPW_SUBEND(Y,X,B,K,N,TITLE,IOUT)                    00005650
C--TERMINATION ROUTINE                                                  00005660
C--FOLLOWING CHARACTER STMT. ONLY FOR HONEYWELL MULTICS SYS:            00005670
      CHARACTER*5 TITLE(16)                                             00005680
      REAL Y(1),X(200,5),B(1)                                           00005690
      WRITE(6,10) TITLE                                                 00005700
   10 FORMAT(15H1I M S L P W --,5X,16A5//                               00005710
     1 28H FINAL UNSCALED PARAMETERS--,10X,11HRESISTIVITY,11X,5HDEPTH/) 00005720
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                                  00005730
      MM=(K+1)/2                                                        00005740
      DO 30 I=1,MM                                                      00005750
      WRITE(6,20) I,B(I),I,B(I)                                         00005760
   20 FORMAT(5X,I3,4X,E16.8,2X,I3,1X,E16.8)                             00005770
      IF(IOUT.EQ.1) WRITE(16,20) I,B(I),I,B(I)                          00005780
   30 CONTINUE                                                          00005790
      IF(K.LE.1) GO TO 60                                               00005800
      M2=MM+1                                                           00005810
      D=0.0                                                             00005820
      DO 50 I=M2,K                                                      00005830
      D=D+B(I)                                                          00005840
      L=I-MM                                                            00005850
      WRITE(6,40) I,B(I),L,D                                            00005860
   40 FORMAT(5X,I3,4X,E16.8,24X,I3,1X,E16.8)                            00005870
      IF(IOUT.EQ.1) WRITE(16,40) I,B(I),L,D                             00005880
   50 CONTINUE                                                          00005890
   60 RETURN                                                            00005900
      END                                                               00005910
```

Appendix 2.-- Conversion to other systems

1. All lower-case letters used for parameters and Fortran names in this report should be changed to upper-case letters for most other systems.

2. Any of the following Multics statements and/or calls should be deleted or replaced if converting to another system:

```
character*n          (replace by logical*n or delete)
call open_           (delete)
call close_          (delete)
exp_                 (replace by exp)
dexp_                (replace by dexp)
cexp_                (replace by cexp)
```

3. All Multics exp-underflow messages are suppressed and the result set to 0.0. An equivalent method should be used for other systems.

4. Subprograms ERRMSG and WARN should be changed according to the number of characters per word of the target machine (note that 4 char/word uses format A4 on the Honeywell Multics system; however, 5 char/word is assumed in the input parameter array MSG). Similar changes should be made, if necessary, to other character arrays and format statements (e.g., see subroutine IMSLMQ, arrays TITLE and FMT).

5. Multics names greater than 6-characters (e.g. IMSLPW_SUBZ, IMSLPW_SUBEND, etc) should be renamed to 6 or less characters for most other systems.

6. To replace the IMSL interface routines (IMSLMQ, FPXSSQ, LNXSSQ, and all calls to the IMSL Library) with the nonlinear least squares subprogram MARQRT (available in Anderson, 1979), replace the main program (lines 00000010-00000070) with the following code:

```
C--NON-IMSL MAIN PROGRAM USING MARQRT (ANDERSON, 1979)
      EXTERNAL FCODE,DUMMY_PCODE,IMSLPW_SUBZ,IMSLPW_SUBEND
      CALL MARQRT(FCODE,DUMMY_PCODE,IMSLPW_SUBZ,IMSLPW_SUBEND)
      STOP
      END
```

Note: Subprogram DUMMY_PCODE is referenced as the second external parameter in the CALL MARQRT, but DUMMY_PCODE will never be called since $parms ider=1 should be used (because analytic derivatives are not used in IMSLMQ, the pcode subprogram was not needed). For systems requiring all external references to be available (even if not called),

the following subprogram could be used:

```
SUBROUTINE DUMMY_PCODE(A,B,C,D,E,I,J,K)
CALL ERRMSG('USE IDER=1',2,6,16)
END
```

If converting to subprogram MARQRT, all parameters prefixed by an "*" apply; however, parameters iopt,parm(),nsig, eps,delta, and maxfn cannot be used in this case.

Appendix 3.-- Test problem input/output listing

The following input files (file05 and file10) were used to run a test problem on a Honeywell Multics system. The output listing (file16) follows beginning on the next page.

file05

```
test22 mt model
$parms n=26,k=7,m=2,iprt=-1,sp=1,
 b=300,20,1000,10,500,2500,10000$
(2e16.8,f10.0)
$init mm=4,iob=5$
```

file10

```
 0.11210782e+02  0.10000000e-02        1.
 0.58457666e+02  0.10000000e-02        2.
 0.14660560e+02  0.21544347e-02        1.
 0.61346634e+02  0.21544347e-02        2.
 0.20535064e+02  0.46415888e-02        1.
 0.63284025e+02  0.46415888e-02        2.
 0.30283839e+02  0.99999998e-02        1.
 0.62884422e+02  0.99999998e-02        2.
 0.44306706e+02  0.21544346e-01        1.
 0.58264533e+02  0.21544346e-01        2.
 0.55818147e+02  0.46415886e-01        1.
 0.48278191e+02  0.46415886e-01        2.
 0.50458804e+02  0.99999995e-01        1.
 0.36433882e+02  0.99999995e-01        2.
 0.33573573e+02  0.21544346e+00        1.
 0.31009460e+02  0.21544346e+00        2.
 0.22154600e+02  0.46415885e+00        1.
 0.36808321e+02  0.46415885e+00        2.
 0.20711063e+02  0.99999993e+00        1.
 0.49599791e+02  0.99999993e+00        2.
 0.27481594e+02  0.21544345e+01        1.
 0.58048237e+02  0.21544345e+01        2.
 0.37091171e+02  0.46415884e+01        1.
 0.60917913e+02  0.46415884e+01        2.
 0.49826824e+02  0.99999989e+01        1.
 0.64281853e+02  0.99999989e+01        2.
```

```
i m s l p w --      test22 mt model .

mm=  4  iob=  5


parameter order--

      1         rho(  1)
      2         rho(  2)
      3         rho(  3)
      4         rho(  4)
      5       thick(  1)
      6       thick(  2)
      7       thick(  3)
```

```
i m s l m q  --      test22 mt model

n=   26          k=   7          ip=   0          m=  2           e= 0.000e+00
ialt= 10         istop= 1         iwt= 0          niter=    10    scalep= 1
iprt= -1         iopt= 1          nsig=  3        maxfn=   140    eps= 0.000e+00
delta= 0.000e+00
parm= 0.100e-01 0.200e+01 0.120e+03 0.100e+00

fmt=(2e16.8,f10.0)


initial parameters

 0.30000000e+03  0.20000000e+02  0.10000000e+04  0.10000000e+02  0.50000000e+03
 0.25000000e+04  0.10000000e+05


$$$$  imslmq convergence information:

norm of gradient                0.53204267e-02
function evaluations            0.31000000e+02
est. sign. digits               0.33444838e+01
marquardt parameter             0.31588589e-06
no. iterations                  0.90000000e+01
type convergence (infer)        1
error code (ier)                  0
residual sum-of-squares (ssq)=  0.10631567e-08

**** final unscaled parameters

 0.35999977e+03  0.17000001e+02  0.59998811e+03  0.57000002e+01  0.41999985e+03
 0.23999977e+04  0.11800002e+05

scaled gradient

-0.18019955e-03 -0.30452882e-02 -0.14292333e-03 -0.30049772e-03 -0.24510621e-02
 0.98399918e-03 -0.34516666e-02
```

i	obs.y(i)	cal	res	x(i,1)	x(i,2)	x(i,3)	x(i,4)	wt(i)
1	0.112108e+02	0.112108e+02	0.131e-05	0.100000e-02	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
2	0.584577e+02	0.584577e+02	0.143e-05	0.100000e-02	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
3	0.146606e+02	0.146606e+02	0.191e-05	0.215443e-02	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
4	0.613466e+02	0.613466e+02	0.286e-05	0.215443e-02	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
5	0.205351e+02	0.205351e+02	0.286e-05	0.464159e-02	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
6	0.632840e+02	0.632840e+02	0.286e-05	0.464159e-02	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
7	0.302838e+02	0.302838e+02	0.620e-05	0.100000e-01	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
8	0.628844e+02	0.628844e+02	0.143e-05	0.100000e-01	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
9	0.443067e+02	0.443067e+02	0.119e-04	0.215443e-01	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
10	0.582645e+02	0.582645e+02	0.000e+00	0.215443e-01	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
11	0.558181e+02	0.558181e+02	0.129e-04	0.464159e-01	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
12	0.482782e+02	0.482782e+02	-0.381e-05	0.464159e-01	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
13	0.504588e+02	0.504588e+02	0.906e-05	0.100000e+00	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
14	0.364339e+02	0.364339e+02	-0.525e-05	0.100000e+00	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
15	0.335736e+02	0.335736e+02	-0.238e-05	0.215443e+00	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
16	0.310095e+02	0.310095e+02	-0.525e-05	0.215443e+00	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
17	0.221546e+02	0.221546e+02	-0.477e-05	0.464159e+00	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
18	0.368083e+02	0.368083e+02	0.000e+00	0.464159e+00	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
19	0.207111e+02	0.207111e+02	-0.143e-05	0.100000e+01	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
20	0.495998e+02	0.495998e+02	0.620e-05	0.100000e+01	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
21	0.274816e+02	0.274816e+02	0.691e-05	0.215443e+01	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01

22	0.580482e+02	0.580482e+02	0.811e-05	0.215443e+01	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
23	0.370912e+02	0.370912e+02	0.906e-05	0.464159e+01	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
24	0.609179e+02	0.609179e+02	0.334e-05	0.464159e+01	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01
25	0.498268e+02	0.498268e+02	0.153e-04	0.100000e+02	0.100000e+01	0.000000e+00	0.000000e+00	0.100000e+01
26	0.642819e+02	0.642818e+02	0.477e-05	0.100000e+02	0.200000e+01	0.000000e+00	0.000000e+00	0.100000e+01

**** rmserr= 0.74803485e-05

final scaled partials (jacobian)

1	-0.22723867e-02	-0.49145251e+00	-0.63146143e-01	-0.72392808e+01	-0.20771015e+00
	-0.72452144e+00	-0.58963826e+01			
2	-0.16361184e-01	-0.18487301e+01	-0.15138347e+00	0.59789389e+01	-0.31710121e+00
	0.92111833e-01	-0.77024457e+01			
3	-0.93167853e-02	-0.98384921e+00	-0.11834674e+00	-0.77518710e+01	-0.36006045e+00
	-0.11129034e+01	-0.10124368e+02			
4	-0.299955e4e-01	-0.31158372e+01	-0.21912966e+00	0.67412921e+01	-0.36404636e+00
	0.11898924e+01	-0.75917423e+01			
5	-0.163611b4e-01	-0.22367935e+01	-0.24254810e+00	-0.81983702e+01	-0.63375954e+00
	-0.14600413e+01	-0.17612683e+02			
6	-0.51810415e-01	-0.53516865e+01	-0.32200351e+00	0.70948026e+01	-0.40390546e+00
	0.34232607e+01	-0.57491584e+01			
7	-0.50901461e-01	-0.58823230e+01	-0.54238773e+00	-0.80200779e+01	-0.11085256e+01
	-0.58085448e+00	-0.29881643e+02			
8	-0.89077557e-01	-0.92056827e+01	-0.46836874e+00	0.69257323e+01	-0.46768000e+00
	0.74280632e+01	-0.12862137e+01			
9	-0.14815961e+00	-0.17390148e+02	-0.12880140e+01	-0.59727909e+01	-0.18166888e+01
	0.68492709e+01	-0.44051393e+02			
10	-0.15361334e+00	-0.14880483e+02	-0.63229780e+00	0.61018991e+01	-0.62977363e+00
	0.13319409e+02	0.64236503e+01			
11	-0.38630573e+00	-0.44911489e+02	-0.26504653e+01	-0.14478562e+01	-0.24473482e+01
	0.31182605e+02	-0.41542306e+02			
12	-0.21178644e+00	-0.19253986e+02	-0.62727956e+00	0.43343465e+01	-0.10717567e+01
	0.17552804e+02	0.15049385e+02			
13	-0.60172798e+00	-0.66731791e+02	-0.32551627e+01	0.12757119e+01	-0.25492103e+01
	0.54134262e+02	-0.13890994e+02			
14	-0.22905657e+00	-0.14657654e+02	-0.14971072e+00	0.20011770e+01	-0.21718776e+01
	0.11457612e+02	0.16761864e+02			
15	-0.50265192e+00	-0.52517909e+02	-0.20884228e+01	0.78079716e+00	-0.27662209e+01
	0.42145288e+02	0.21290432e+01			
16	-0.21360434e+00	-0.94796987e+00	0.70255312e+00	0.46263769e+00	-0.43827285e+01
	-0.57511467e+01	0.10115095e+02			
17	-0.36721768e+00	-0.29652384e+02	-0.66951639e+00	0.18597728e+00	-0.41630606e+01
	0.17906472e+02	0.29399171e+01			
18	-0.26268790e+00	0.13201708e+02	0.14000880e+01	-0.52258078e-01	-0.74757939e+01
	-0.23528387e+02	0.24229208e+01			
19	-0.39357737e+00	-0.16381750e+02	0.29733051e+00	0.46110069e-02	-0.77835947e+01
	-0.15146897e+01	0.79204287e+00			
20	-0.44447883e+00	0.15503651e+02	0.95262857e+00	-0.52258078e-01	-0.89815818e+01
	-0.22070408e+02	-0.86451365e+00			
21	-0.70625777e+00	-0.15165630e+02	0.37302225e+00	-0.10759016e-01	-0.14684532e+02
	-0.90956999e+01	-0.20314644e+00			
22	-0.79442637e+00	0.67755297e+01	-0.92837376e-01	0.61480092e-02	-0.87938012e+01
	-0.28245338e+01	-0.14094711e+00			
23	-0.14534185e+01	-0.22539778e+02	-0.54364229e-01	0.00000000e+00	-0.25526648e+02
	-0.59116550e+00	0.17119082e-01			
24	-0.12561753e+01	0.47776170e+01	-0 58546093e-01	0.00000000e+00	-0.94731772e+01
	0.25351376e+01	0.14265902e-01			
25	-0.28259400e+01	-0.25058885e+02	0.00000000e+00	0.00000000e+00	-0.44322423e+02
	0.41519065e+00	0.00000000e+00			
26	-0.19342555e+01	0.67113245e+01	0.50182365e-02	0.00000000e+00	-0.93677721e+01

-0.19384729e+00 0.00000000e+00

scaled covariance matrix (inverse of xjtj)

1	0.8416e+00	0.2719e-01	0.3971e+00	0.3278e-02	-0.7622e-01	0.5818e-01	-0.1340e-01
2	0.2719e-01	0.1844e-02	0.3029e-01	0.2499e-03	-0.3147e-02	0.3875e-02	-0.1004e-02
3	0.3971e+00	0.3029e-01	0.4105e+01	0.3225e-01	-0.4997e-01	0.2507e+00	-0.1020e+00
4	0.3278e-02	0.2499e-03	0.3225e-01	0.2923e-02	-0.3916e-03	0.1786e-02	-0.1219e-02
5	-0.7622e-01	-0.3147e-02	-0.4997e-01	-0.3916e-03	0.7689e-02	-0.6776e-02	0.1646e-02
6	0.5818e-01	0.3875e-02	0.2507e+00	0.1786e-02	-0.6776e-02	0.1802e-01	-0.6468e-02
7	-0.1340e-01	-0.1004e-02	-0.1020e+00	-0.1219e-02	0.1646e-02	-0.6468e-02	0.2814e-02

correlation matrix

1	0.1000e+01	0.6901e+00	0.2136e+00	0.6609e-01	-0.9475e+00	0.4725e+00	-0.2754e+00
2	0.6901e+00	0.1000e+01	0.3481e+00	0.1076e+00	-0.8358e+00	0.6723e+00	-0.4406e+00
3	0.2136e+00	0.3481e+00	0.1000e+01	0.2944e+00	-0.2813e+00	0.9217e+00	-0.9487e+00
4	0.6609e-01	0.1076e+00	0.2944e+00	0.1000e+01	-0.8259e-01	0.2461e+00	-0.4250e+00
5	-0.9475e+00	-0.8358e+00	-0.2813e+00	-0.8259e-01	0.1000e+01	-0.5756e+00	0.3538e+00
6	0.4725e+00	0.6723e+00	0.9217e+00	0.2461e+00	-0.5756e+00	0.1000e+01	-0.9084e+00
7	-0.2754e+00	-0.4406e+00	-0.9487e+00	-0.4250e+00	0.3538e+00	-0.9084e+00	0.1000e+01

** parameter	std error	std error/parameter	(unscaled)
1	0.3600e+03	0.2471e-02	0.6863e-05
2	0.1700e+02	0.5461e-05	0.3212e-06
3	0.6000e+03	0.9093e-02	0.1516e-04
4	0.5700e+01	0.2305e-05	0.4044e-06
5	0.4200e+03	0.2755e-03	0.6559e-06
6	0.2400e+04	0.2410e-02	0.1004e-05
7	0.1180e+05	0.4682e-02	0.3968e-06

i m s l p w -- test22 mt model

final unscaled parameters-- resistivity depth

1	0.35999977e+03	1	0.35999977e+03		
2	0.17000001e+02	2	0.17000001e+02		
3	0.59998811e+03	3	0.59998811e+03		
4	0.57000002e+01	4	0.57000002e+01		
5	0.41999985e+03			1	0.41999985e+03
6	0.23999977e+04			2	0.28199975e+04
7	0.11800002e+05			3	0.14619999e+05