U.S. Department of the Interior
Geological Survey

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

Program MARQLOOPS: Marquardt inversion of loop-loop frequency soundings

by

Walter L. Anderson

OPEN-FILE REPORT 79-240

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 11

DATA MATRIX NOTES 12

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING 14

SPECIAL OBJECT FORMAT PHRASES 15

MULTICS OPERATING INSTRUCTIONS 15

ERROR MESSAGES 16

REFERENCES 17

Appendix 1.-- Source listing 18

Source availability 18

Appendix 2.-- Conversion to other systems 69

Appendix 3.-- Test problem input/output listing 70

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 manufactures' 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 MARQLOOPS is a general-purpose program for inversion of various loop-loop frequency sounding data obtained over a horizontally stratified earth for the quasi-static case (i.e., neglecting displacement currents). A modified Marquardt (1963) nonlinear least squares algorithm (MARQRT) is used for inversion of frequency sounding data. An adaptive digital filtering algorithm (ZHANKS) developed by Anderson (1979) is used for evaluating all Hankel transforms.

The following program options are currently available:

(1) Simultaneous (or joint) inversion of up to five different loop-loop configurations (e.g., see Frischknecht, 1967); both ground and airborne loop cases, and the dipole wire-loop case (same as in Anderson, 1977), are provided.
(2) Simultaneous inversion of loop-loop soundings and Schlumberger soundings.
(3) Mixed frequency (parametric) and/or distance (geometric) sounding inversion. Also, mixed observation types can be used (e.g., amplitude, phase, real or imaginary parts).
(4) Inclusion of an additional amplitude shift parameter in the least squares when the correct primary field normalization factor is unknown.
(5) Scaling parameter and observation spaces to constrain the solution space and to reduce round-off effects.
(6) Weighted observations.
(7) Holding certain parameters fixed (constrained).
(8) 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 MARQLOOPS 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 MARQLOOPS is:

1. Title line (always required, max. 80 characters).
2. $parms --non-default parameters--$
   (note $parms begins 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 MARQLOOPS.
file16	output master print-type disk file--contains maximum printable output (if parameter iout=1).

DETAIL PARAMETER AND DATA DEFINITIONS.

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

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 specified from one of the following options:
- = 2*mm-1, where $init parameter mm>0 is the number of layers in the model, and mm>0 is used to indicate that the amplitude shift option is not selected.
- = 2*|mm|, where |mm| is the number of layers in the model, and mm<0 is used to indicate the amplitude shift option is selected.

(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<=4) 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:
- = 2 when $init parameter iob<=4 (defines specific observation type in y(i));
- = 3 when $init parameter iob=5 (defines mixed observation types in y(i) via x(i,3));
- = 4 when $init parameter iob=6 (defines mixed observation types in y(i) via x(i,3) and distance or elevation types in x(i,4)).

(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<=4. 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. ider=1 option is useful to check the validity of the analytic derivatives, but is not recommended for general use because of accuracy and timing considerations.
(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.

= -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 MARQLOOPS 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.
(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 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 MARQLOOPS, 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).

Note: Due to the possible wide range of numbers commonly encountered in electromagnetic problems, it is recommended that scalep=1 and scaley=2 be generally used for program MARQLOOPS. A special case automatically occurs whenever sy=2 and iob>=5 and both amplitude and phase data are included in the data matrix; in this case, the program will use ln(amplitude) or arcsinh(phase) accordingly.
(cref: $init iob and $parms b,k,n)

b()= Array of initial guesses for all k-parameters. These values must be supplied greater than zero for program MARQLOOPS (i.e., positive conductivities 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 conductivities (in mhos per meter), and

b(|mm|+1),b(|mm|+2),...,b(2*|mm|-1) are the |mm|-1 layer thicknesses (in meters); and if mm<0 (amplitude shift option) include

b(2*|mm|)>0 as the estimated amplitude shift parameter used in the model as b(2*|mm|)*z/z0, where z/z0 is the mutual coupling ratio.

Note: If only phase data (iob=2) or multiple distance soundings (iob=6) are used, then the shift parameter option (mm<0) should not be used--or one should fix b(2*|mm|) using parameters

ip and ib.
(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).

$end [end of $parms namelist]

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

iob= Observation-type defined for y(i): [where we define Z=z/z0 (if mm>0), or Z=b(2*|mm|)*z/z0 (if mm<0)]
= 1 (default) defines y(i) as the amplitude of Z;
= 2 defines y(i) as the phase of Z, expressed in (-180,+180) degrees [mm>0 option is recommended when iob=2];
= 3 defines y(i) as the real-part of Z;
= 4 defines y(i) as the imaginary-part of Z;
(note: for iob<=4, m=2 must also be given in $parms).
= 5 defines mixed observation-type frequency soundings and/or Schlumberger sounding, where the i-th observation type is given by x(i,3)=1.0 for amplitude of Z, =2.0 for phase of Z, =3.0 for real of Z, =4.0 for imaginary of Z, or =5.0 for apparent resistivity of a Schlumberger array;
(note: for iob=5, m=3 must also be given in $parms).
= 6 defines mixed observation-type frequency, distance (or elevation), and/or Schlumberger sounding, where the i-th observation type is given by x(i,3) between 1.0 and 5.0 (same as in iob=5 case), and y0=x(i,4)>0.0 defines the loop-loop separation -or- x(i,4)<=0.0 defines the loops elevation sum as h=z+h'=|x(i,4)|.
(note: for iob=6, m=4 must also be given in $parms; also, mm>0 option is recommended when iob=6).
(cref: $parms m,b(), $init mm, and DATA MATRIX NOTES).

mm= Number of layers in the model (1<=|mm|<=10; default mm=1). Use mm>0 for no amplitude shift option (i.e., Z=z/z0 mutual coupling). Use mm<0 for amplitude shift option (i.e., Z=b(2*|mm|)*z/z0

shifted mutual coupling).
Note: make sure $parms k=2*mm-1 (if mm>0) or k=2*|mm| (if mm<0).
(cref: $parms k,b(), $init iob).

y0= Transmitter-receiver separation, where y0>0 meters. Note y0 must be given, unless iob=6 is used for distance soundings.
(cref: $init iob and DATA MATRIX NOTES).

h= (z+h′) loop elevation sum, where z=receiver loop elevation (meters), and h′=transmitter loop elevation (meters). When h=0.0 (default), the ground case is assumed; when h>0.0, the airborne case is defined. Note h must be given (or assumed 0.0), unless iob=6 is used to vary the loops elevation sum.
(cref: $init iob and DATA MATRIX NOTES).

eps= Requested convolution integration tolerance used to compute all Hankel transforms using subprogram ZHANKS (default .1e-5).

$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<=4. 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 various loop-loop configurations (iob<=4, m=2, and max. 4 items per record):

1. y(i)= i-th observation, where $init iob<=4 defines the particular type.
2. x(i,1)= i-th frequency (x(i,1)>0.0 Hz.).
3. x(i,2)= i-th loop-loop configuration; use x(i,2)=1.0 for horizontal coplanar loops, =2.0 for perpendicular loops, =3.0 for vertical coplanar loops, =4.0 for vertical coaxial loops, or =5.0 for horizontal coplanar loop and wire element (same as the dipole case in Anderson, 1977).

4. x(i,3)= standard deviation of observation i (include only if iwt=1).

(b) Mixed observation types, frequency sounding and/or Schlumberger sounding (iob=5, m=3, and max. 5 items per record):

1. y(i)= i-th observation (where actual type is defined by x(i,3)).
2. x(i,1)= i-th frequency (if x(i,3)<5.0) or AB/2 meter spacing (if x(i,3)=5.0).
3. x(i,2)= i-th loop-loop configuration (must be between 1.0 and 5.0 as defined in (a)3 above). For Schlumberger data (x(i,3)=5.0), x(i,2) must be given between 1.0 and 5.0 (but not used).
4. x(i,3)= observation type in y(i); use x(i,3)= 1.0 for amplitude, =2.0 for phase (degrees), =3.0 for real part, =4.0 for imaginary part, or =5.0 for Schlumberger apparent resistivity (ohm-meters). Note x(i,3)<5.0 defines the observation type of y(i) for the loop-loop configuration defined in x(i,2); however, x(i,2) must be between 1.0 and 5.0 (but not used) when x(i,3)=5.0 for a Schlumberger sounding observation.
5. x(i,4)= standard deviation of observation i (include only if iwt=1). Note: for joint inversion of loop-loop and Schlumberger data, a weighted least squares should be used (iwt=1 option) to produce near-equal magnitudes.

(c) Mixed observation types, both frequency and distance (or elevation) loop-loop soundings, plus joint inversion of Schlumberger and loop-loop soundings (iob=6, m=4, and max. 6 items per record):

1. y(i)= i-th observation (where actual type is defined by x(i,3)).
2. x(i,1)= i-th frequency (if x(i,3)<5.0) or AB/2 meter spacing (if x(i,3)=5.0).
3. x(i,2)= i-th loop-loop configuration (must be between 1.0 and 5.0 as defined in (a)3 above). For Schlumberger data (x(i,3)=5.0), x(i,2) must be given between 1.0 and 5.0 (but not used).
4. x(i,3)= observation type in y(i) (must be between 1.0 and 5.0 as defined in (b)4 above).
5. x(i,4)= distance y0=x(i,4)>0.0 -or- elevation sum h=z+h'=|x(i,4)| if x(i,4)<=0.0. Note: one should preset either y0 or h in $init

parameter input and use x(i,4) to change either distances (for distance soundings) or elevations; both distances and elevations cannot be changed simultaneously.

6. x(i,5)= standard deviation of observation i (include only if iwt=1). Note: for joint inversion of loop-loop and Schlumberger data, a weighted least squares should be used (iwt=1 option) to produce near-equal magnitudes.

For a given loop-loop configuration defined by x(i,2), the data matrix should be grouped or ordered with equal consecutive frequencies (and distances or elevations, if used) with respect to each observation type (for example, see the grouping used in appendix 3). This ordering is not mandatory, but it will significantly reduce the total calculation time when ider=0 (default case).

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING.

1. Mixed observation types (real and imaginary parts), horizontal coplanar loops, ground case (h=0), and amplitude shift option (mm<0):

```
example 1.
$parms n=60,k=6,m=3,iprt=-1,sp=1,sy=2,ialt=5,
 b=.1,.2,.3,10,20,2$
(4f10.0)
1.98        1.          1.          3.
-.027       1.          1.          4.
1.85        1.6         1.          3.
-.034       1.6         1.          4.
--(etc. for 56 more observations)--
$init mm=-3,iob=5,y0=100,h=0$
```

2. Distance soundings (y0=x(i,4)>0.0), horizontal and vertical coplanar loops, airborne case (h>0), no amplitude shift (mm>0), mixed observation types (amplitude and phase):

```
example 2
$parms n=100,k=5,m=4,iprt=-2,sp=1,sy=2,ialt=5,
 b=.1,.2,.3,10,20$
(5f10.0)
1.01        1.          1.          1.          100.
-2.3        1.          1.          2.          100.
0.987       1.          3.          1.          100.
-5.23       1.          3.          2.          100.
--(etc. for rest of y0=x(i,4)=100. soundings)--
```

```
0.79        1.          1.          1.          300.
-2.34       1.          1.          2.          300.
0.867       1.6         3.          1.          300.
-10.23      1.6         3.          2.          300.
--(etc. for rest of y0=x(i,4)=300. soundings)--
$init mm=3,iob=6,h=2$
```

3. Joint loop-loop and Schlumberger soundings, weighted (iwt=1), both airborne and ground loops (x(i,4)<=0.0 as h=|x(i,4)|), and amplitude shift option (mm<0):

```
example 3
$parms n=50,k=6,m=4,iprt=-1,sp=1,sy=2,ialt=5,iwt=1,
 b=.1,.2,.3,10,20,2$
(6f10.0)
1.98        1.          1.          3.          -1.         .02
-.027       1.          1.          4.          -1.         .02
--(etc. for rest of loop-loop sounding at h=|-1.|)--
1.56        1.2         1.          3.          0.          .02
-.034       1.2         1.          4.          0.          .02
--(etc. for rest of loop-loop sounding at h=0)--
9.98        4.          1.          5.          0.          .05
8.23        6.          1.          5.          0.          .05
--(etc. for rest of Schlumberger data)--
$init mm=-3,iob=6,y0=200$
```

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 (/5f10.0). Similarly, if we wanted only amplitudes to be used in example 2, then the format (5f10.0/) 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 and >udd>Emod1_inv>WAnderson>lib_1.

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 1-3 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 MARQLOOPS by typing: marqloops

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 MARQLOOPS. 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 MARQLOOPS.

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., 1977, Marquardt inversion of vertical magnetic field measurements from a grounded wire source: U.S. Geol. Survey Rept. USGS-GD-77-003, 76p. avail. from U.S. Dept. Comm. NTIS, Springfield, Va., 22161 as Rept. PB-263-924/AS.

----------, 1979 (in press), Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics, v. ,no. , p - .

Frischknecht, F.C., 1967, Fields about an oscillating magnetic dipole over a two-layer earth, and applications to ground and airborne electromagnetic surveys: Quarterly of Col. School of Mines, v.62, no. 1, 326 p.

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.

Appendix 1.-- Source listing

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

```
C--MARQLOOPS--MARQUARDT INVERSION OF LOOP-LOOP EM DATA--12/26/78.00000010
      SUBROUTINE MARQRT(FCODE,PCODE,SUBZ,SUBEND)                 00000170
      SUBROUTINE GJR (A,N,EPS,MSING)                             00009930
      SUBROUTINE UNSCAL(BIN,BOUT,SCALEP)                         00010590
      REAL FUNCTION ASINH(X)                                     00010800
      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                            00010880
      SUBROUTINE POLAR2(Z,AMP,PHZ180)                            00011110
      SUBROUTINE RECUR1(G,V1,F1)                                 00011400
      SUBROUTINE RECURF(G,DEL,SIG1,V1,F1,PF1,JJ)                 00011710
      SUBROUTINE KELVIN(X,M,B)                                   00012330
      COMPLEX FUNCTION FVP(X)                                    00014130
      COMPLEX FUNCTION KERN(X)                                   00014580
      COMPLEX FUNCTION FG2(G)                                    00014840
      COMPLEX FUNCTION FG(G)                                     00014980
      COMPLEX FUNCTION FG3(G)                                    00015060
      COMPLEX FUNCTION RG2(G)                                    00015130
      COMPLEX FUNCTION RG(G)                                     00015260
      COMPLEX FUNCTION RG3(G)                                    00015340
      SUBROUTINE IKS2(B8,I0K0,I1K1,IKDIF)                        00015410
      COMPLEX FUNCTION PFBJG(G)                                  00016180
      COMPLEX FUNCTION PFBJG2(G)                                 00016360
      COMPLEX FUNCTION PRBJG(G)                                  00016430
      COMPLEX FUNCTION PRBJG2(G)                                 00016510
      SUBROUTINE MODIFY(N)                                       00016580
      SUBROUTINE SWAP(ICODE)                                     00016800
      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                00017040
      SUBROUTINE FCODE(Y,X,B,PRNT,F,IN,IDER)                     00020460
      SUBROUTINE PCODE(P,X,B,PRNT,F,IN,IP,IB)                    00022290
      SUBROUTINE SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)             00024320
      SUBROUTINE SUBEND(Y,X,B,K,N,TITLE,IOUT)                    00025360
```

Source Availability

An updated 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.

```
C--MARQLOOPS--MARQUARDT INVERSION OF LOOP-LOOP EM DATA-- 12/26/78.      00000010
C** HONEYWELL MULTICS VERSION **                                        00000020
C                                                                       00000030
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.            00000040
C                                                                       00000050
C  (FOR DETAILS, SEE MULTICS DOCUMENTATION FOR PGM MARQLOOPS)           00000060
C                                                                       00000070
C  SUBROUTINES FCODE,PCODE,SUBZ,AND SUBEND TO LINK WITH PGM MARQRT.     00000080
C  L.SQ.FITTING FUNCTIONS FOR Z/Z0 MUTUAL COUPLING RATIO FOR            00000090
C  UP TO FIVE DIFFERENT LOOP-LOOP ORIENTATIONS AND SCHLUMBERGER         00000100
C  APPARENT RESISTIVITY DATA.                                           00000110
C                                                                       00000120
      EXTERNAL FCODE,PCODE,SUBZ,SUBEND                                  00000130
      CALL MARQRT(FCODE,PCODE,SUBZ,SUBEND)                              00000140
      STOP                                                              00000150
      END                                                               00000160

      SUBROUTINE MARQRT(FCODE,PCODE,SUBZ,SUBEND)                        00000170
C--(MARQRT)-- GENERAL MARQUARDT NONLINEAR LEAST SQUARES-- 7/11/78.      00000180
C** HONEYWELL MULTICS VERSION **                                        00000190
C  SUBPROGRAM MARQRT IS TO BE LINKED/LOADED WITH USER WRITTEN           00000200
C  SUBROUTINES (FCODE,PCODE,SUBZ, AND SUBEND) FOR                       00000210
C  SPECIFIC NONLINEAR PROBLEM TO BE SOLVED.                             00000220
C                                                                       00000230
C--THE USER MUST DECLARE THE CALLING PARAMETERS FCODE,PCODE,            00000240
C  SUBZ,SUBEND (ANY DESIRED NAMES MAY BE USED) AS EXTERNAL IN           00000250
C  MAIN CALLING PROGRAM; E.G.,                                          00000260
C                                                                       00000270
C       EXTERNAL FCODE,PCODE,SUBZ,SUBEND                                00000280
C       CALL MARQRT(FCODE,PCODE,SUBZ,SUBEND)                            00000290
C       STOP                                                            00000300
C       END                                                             00000310
C                                                                       00000320
C--THIS IS A MODIFIED VERSION OF 'IBM SHARE PROGRAM NO. 1428'.          00000330
C *** MODIFIED BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY,DENVER, COLORADO 00000340
C  FOR NAMELIST INPUT, IMPROVED ESTIMATED DERIVATIVES,                  00000350
C  MODIFIED MARQUARDT LAMBDA DETERMINATION,                             00000360
C  DATA AND PARAMETER SCALING, WEIGHTED OBSERVATIONS, AND               00000370
C  OTHER CHANGES--ALL DONE IN SINGLE-PRECISION FOR THE                  00000380
C *** HONEYWELL MULTICS SYSTEM ***                                      00000390
C                                                                       00000400
C--SEE SHARE PROGRAM NO. 1428 AND/OR DOCUMENTATION OF 'MARQHZ',         00000410
C  N.T.I.S REPORT PB-263-924, P33-39, FOR DETAILS ON CODING THE         00000420
C  REQUIRED SUBROUTINES FCODE,PCODE,SUBZ, AND SUBEND.                   00000430
C                                                                       00000440
C--OPERATING NOTE FOR HONEYWELL MULTICS SYSTEM:  $$$$$$$$$$$$$$$$$$$$ 00000450
C  (UNIT 5 USED INSTEAD OF UNIT 1 AS IN DOCUMENTATION OF 'MARQHZ')      00000460
C  TO OBTAIN ON-LINE (INTERACTIVE) PRINTING ON UNIT 6 AND               00000470
C  DEFERRED PRINTING ON UNIT 16, USE MULTICS RUN.EC, I.E.,              00000480
C  'RUN &1' OR 'RUN_EO &1' AND DPRINT '&1.FILE16.LIST' AFTER RUN.       00000490
C                                                                       00000500
C--FOLLOWING CHARACTER STATEMENTS ONLY FOR MULTICS SYSTEM:              00000510
```

```
      CHARACTER*5 TITLE                                                 00000520
      CHARACTER*4 FMT                                                   00000530
      INTEGER SCALEP,SCALEY, SP,SY                                      00000540
      DIMENSION FMT(18),PRNT(5),SPRNT(5),TITLE(16)                      00000550
      DIMENSION BS(20),DB(20),BA(20),G(20),IB(19),SA(20),P(20)          00000560
      DIMENSION A(20,20),B(20), BINV(20)                                00000570
      DIMENSION X(200,5),Y(200),WT(200)                                 00000580
      DIMENSION XNU(5),SS(4)                                            00000590
      EQUIVALENCE (X(1,5),WT(1)),(IOUT,IFSS1),(IDER,IWS2),              00000600
     1 (IPRT,IWS3),(NITER,IWS4),(INON,IWS6),(SP,SCALEP),(SY,SCALEY)     00000610
C=============================================================        00000620
      NAMELIST/PARMS/N,K,IP,M,IALT,IDER,IPRT,NITER,INON,NPRNT,          00000630
     1 IB,FF,T,E,TAU,XL,GAMCR,DEL,ZETA,B,IOUT,IWT,ISTOP,                00000640
     2 SCALEP,SCALEY,MODLAM, SP,SY                                      00000650
C=============================================================        00000660
      DATA XNU/1.33,1.78,3.16,10.,100./                                 00000670
C                                                                       00000680
C     MAX NO OF PARAMETERS IS K=20  (NOTE: K=N IS ALLOWED)              00000690
C     MAX NO OF IND VARS IS M=4                                         00000700
C     MAX NO OF OBSERVATIONS IS N=200                                   00000710
C     INTERNAL #IWHER# SWITCH USAGE--                                   00000720
C                 IWHER =-1 MEANS INITIALIZE VIA SUBROUTINE SUBZ.       00000730
C                 IWHER = 0 MEANS START NEW PROBLEM OR END RUN          00000740
C                 IWHER = 1 MEANS GET P(S) AND F                        00000750
C                 IWHER GREATER THAN 1 MEANS GET F ONLY                 00000760
C--FOLLOWING CALL TO SUPPRESS EXP-UNDERFLOW MESSAGES                    00000770
C  FOR THE DEC-10 AND OTHER SYSTEMS:  $$$$$$$$$$$$$$$$$$$$$            00000780
C//         CALL ERRSET(0)                                             00000790
C** FOR THE HONEYWELL MULTICS SYSTEM, USE (INSTEAD) THE FOLLOWING:      00000800
C    IO DETACH ERROR_OUTPUT                                             00000810
C    IO ATTACH ERROR_OUTPUT DISCARD_                                    00000820
C  (OR-- ON USGS SYS, USE SET_UFL -OFF)                                 00000830
C**                                                                     00000840
C--PRESET GLOBAL PARMS (SOME MAY BE OVERRIDDEN BY $PARMS READ-IN)       00000850
      IP=0                                                              00000860
      N=0                                                               00000870
      K=0                                                               00000880
      M=0                                                               00000890
      NPRNT=0                                                           00000900
      MODLAM=1                                                          00000910
      ISTOP=1                                                           00000920
      IWT=0                                                             00000930
      IALT=10                                                           00000940
      IOUT=1                                                            00000950
      IDER=0                                                            00000960
      IPRT=0                                                            00000970
      MITER=10                                                          00000980
      INON=1                                                            00000990
      LSCALP=0                                                          00001000
      LSCALY=0                                                          00001010
      FF=4.0E0                                                          00001020
      E=.00005E0                                                        00001030
```

```
      TAU=.001E0                                                        00001040
      T=2.0E0                                                           00001050
      DEL=.00001E0                                                      00001060
      ZETA=.1E-30                                                       00001070
      GAMCR=45.0E0                                                      00001080
C                                                                       00001090
   10 GAMMA=0.E0                                                        00001100
      SCALEP=LSCALP                                                     00001110
      SCALEY=LSCALY                                                     00001120
      XLL=0.E0                                                          00001130
      SE=0.0                                                            00001140
      NITER=MITER                                                       00001150
   20 IWHER=0                                                           00001160
      ISS=1                                                             00001170
      INU=4                                                             00001180
      XNUFAC=10.0                                                       00001190
      GO TO 150                                                         00001200
   30 CONTINUE                                                          00001210
      IF (IWHER.GT.0) GO TO 100                                         00001220
      IF (IWHER.EQ.0) GO TO 240                                         00001230
C=======================================================              00001240
C     INITIALIZATION (IWHER=-1, IFSS1=IOUT)                             00001250
      CALL SUBZ (Y,X,BINV,PRNT,NPRNT,N,TITLE,IFSS1)                     00001260
C     ********************************************                      00001270
      IPRNT=NPRNT-1                                                     00001280
      IF(NPRNT.LT.0) IPRNT=IABS(NPRNT)-2                                00001290
C                                                                       00001300
C--NOTE: IPRNT IS A SPECIAL INDEX USED IN SCALEY=2 CASES                00001310
C  TO MIX LOG OR ASINH TYPE SCALING WHEN ABS(X(I,IPRNT))=1. OR NOT 1.   00001320
C  RESPECTIVELY, AND ONLY WHEN IPRNT.GT.1                               00001330
      NPRNT=IABS(NPRNT)                                                 00001340
      IF(SCALEY.EQ.0) GO TO 90                                          00001350
      DO 80 I=1,N                                                       00001360
      IF(SCALEY-1) 90,40,60                                             00001370
   40 IF(Y(I).LE.0.)CALL ERRMSG(30HSOME Y(I).LE.0 AND SCALEY=1...,      00001380
     1 6,6,16)                                                          00001390
   50 Y(I)=ALOG(Y(I))                                                   00001400
      GO TO 80                                                          00001410
   60 IF(IPRNT.LE.1) GO TO 70                                           00001420
      IF(ABS(X(I,IPRNT)).NE.1.0) GO TO 70                               00001430
      IF(Y(I).LE.0.)                                                    00001440
     1CALL ERRMSG(50HSOME Y(I).LE.0 WHEN ABS(X(I,IPRNT))=1 AND SCALEY=2,00001450
     2 10,6,16)                                                         00001460
      GO TO 50                                                          00001470
   70 Y(I)=ASINH(Y(I))                                                  00001480
   80 CONTINUE                                                          00001490
   90 CONTINUE                                                          00001500
      IF (IBOUT.EQ.0) GO TO 150                                         00001510
      GO TO 20                                                          00001520
  100 CONTINUE                                                          00001530
C=======================================================              00001540
C                  COMPUTE F VIA SUBR. FCODE                          00001550
```

```
C                  NPRNT IS THE NO OF OTHER WORDS TO BE PRINTED          00001560
C                  THE WORDS TO BE PRINTED ARE IN PRNT(1)...PRNT(5)      00001570
C--CALL FCODE FOR CURRENT BINV AND I-TH OBSERVATION (IFSS2=IDER)          00001580
      CALL FCODE(Y,X,BINV,PRNT,F,I,IFSS2)                                 00001590
C     ***********************************************                     00001600
      FINV=F                                                              00001610
      IF(SCALEY-1) 140,110,120                                            00001620
  110 F=ALOG(F)                                                           00001630
      GO TO 140                                                           00001640
  120 IF(IPRNT.LE.1) GO TO 130                                            00001650
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 110                                00001660
  130 F=ASINH(F)                                                          00001670
  140 CONTINUE                                                            00001680
      IF (IWHER.NE.1) GO TO 150                                           00001690
      IF (IFSS2.NE.0) GO TO 150                                           00001700
C=====================================================                  00001710
C                  COMPUTE P(J)=DF/DB VIA SUBR PCODE FOR J=1,K.          00001720
C                      USING X(I,L) AND B(J)                             00001730
C--CALL PCODE FOR CURRENT BINV,FINV AND I-TH OBSERVATION                  00001740
      CALL PCODE(P,X,BINV,PRNT,FINV,I,IP,IB)                              00001750
C     ***********************************************                     00001760
C                THIS IS GENERAL #IWHER# SWITCH                           00001770
  150 CONTINUE                                                            00001780
      IF (IWHER.LT.0) GO TO 320                                           00001790
      IF (IWHER.EQ.0) GO TO 160                                           00001800
C               1  2  3  4  5                                             00001810
      GO TO (490,1560,530,580,590), IWHER                                 00001820
C                  READ FIRST CARD OF NEXT CASE                         00001830
  160  ITCT=0                                                             00001840
      IBOUT=0                                                             00001850
C=====================================================                  00001860
C  READ $PARMS --$                                                        00001870
C--ALWAYS PRESET XL=.01 (MAY BE OVERRIDDEN BY $$PARMS READ-IN)           00001880
C  AND CLEAR B(I),I=1,20 TO FORCE INITIALIZATION...                       00001890
      XL=.01                                                              00001900
      DO 170 I=1,20                                                       00001910
  170 B(I)=0.E0                                                           00001920
      READ(5,180) TITLE                                                   00001930
  180 FORMAT(16A5)                                                        00001940
      READ(5,PARMS)                                                       00001950
C--TEST $PARMS                                                            00001960
      IF(N.GT.200.OR.K.GT.20.OR.M.GT.4.OR.IWT.GT.1.OR.IP.GT.19.OR.        00001970
     1 IALT.EQ.6.OR.IALT.EQ.13.OR.IALT.EQ.16.OR.                          00001980
     2 N.LT.1.OR.K.LT.1.OR.M.LT.1.OR.IWT.LT.0.OR.IP.LT.0.OR.              00001990
     3 SCALEY.LT.0.OR.SCALEY.GT.2.OR.SCALEP.LT.0.OR.SCALEP.GT.2.OR.       00002000
     4 N.LT.K) CALL ERRMSG(30HSOME $PARMS OUT OF RANGE..      ,6,6,16)   00002010
      DO 210 I=1,K                                                        00002020
      IF(B(I).EQ.0.E0) CALL ERRMSG(20HSOME B(I) = 0.0       ,4,6,16)      00002030
      BINV(I)=B(I)                                                        00002040
      IF(SCALEP-1) 210,190,200                                            00002050
  190 IF(B(I).LT.0.0)CALL ERRMSG(30HSOME B(I).LT.0. AND SCALEP=1..,       00002060
     1 6,6,16)                                                            00002070
```

```
      B(I)=ALOG(B(I))                                                   00002080
      GO TO 210                                                         00002090
  200 B(I)=ASINH(B(I))                                                  00002100
  210 CONTINUE                                                          00002110
      MAXITR=IWS4                                                       00002120
      MITER=NITER                                                       00002130
      ITER=1                                                            00002140
      WRITE (6,2730)                                                    00002150
      IF (IFSS1.NE.1) GO TO 250                                         00002160
      WRITE (16,2730)                                                   00002170
      GO TO 250                                                         00002180
C=======================================================              00002190
C                   END OF LAST PROBLEM                                 00002200
  220 CALL SUBEND(Y,X,BINV,K,N,TITLE,IOUT)                              00002210
C     ************************************************                  00002220
  240 IF(ISTOP.EQ.1.OR.IALT.EQ.5) GO TO 241                             00002230
C--INITIALIZE FOR NEXT PROB (SAME IALT DATA), SINCE ISTOP=0             00002240
      GO TO 10                                                          00002250
C--FOLLOWING CLOSE STMT ONLY FOR HONEYWELL MULTICS:                     00002260
  241 CALL CLOSE_FILE('-ALL')                                           00002270
C     STOP                                                              00002280
      RETURN                                                            00002290
  250 CONTINUE                                                          00002300
      IF (IP.LE.0) GO TO 280                                            00002310
      DO 270 I=1,IP                                                     00002320
      IF (IB(I).GT.0) GO TO 270                                         00002330
      CALL ERRMSG(30HIP.GT.1 BUT SOME IB(I).LE.0...,6,6,16)             00002340
  270 CONTINUE                                                          00002350
  280 CONTINUE                                                          00002360
      IF (K.GT.10) GO TO 290                                            00002370
C--IBKT=1 MEANS USE UPPER A MATRIX FOR SCRATCH STORAGE                  00002380
C       =2 MEANS USE FILE 13 FOR SCRATCH STORAGE                        00002390
      IBKT=1                                                            00002400
      GO TO 300                                                         00002410
  290 IBKT=2                                                            00002420
  300 XKDB=1.E0                                                         00002430
C--READ OBJECT TIME FORMAT FOR DATA ON FILE IALT.                       00002440
      READ(5,2480) (FMT(I),I=1,18)                                      00002450
      M1=M+IWT                                                          00002460
      DO 310 I=1,N                                                      00002470
      READ(IALT,FMT) Y(I),(X(I,L),L=1,M1)                               00002480
C--SET UP WTS VIA IWT PARM                                              00002490
      WT(I)=1.0E0                                                       00002500
      IF(IWT.EQ.1.AND.X(I,M1).NE.0.0) WT(I)=1.0E0/X(I,M1)**2            00002510
  310 CONTINUE                                                          00002520
      IF(IALT.NE.5) REWIND IALT                                         00002530
      IWHER=-1                                                          00002540
       GO TO 30                                                         00002550
  320 IBKA=1                                                            00002560
C                                                                       00002570
C     .................................................................00002580
C                   START THE CALCULATION OF THE PTP MATRIX             00002590
```

```
      WRITE(6,2520) TITLE                                                00002600
      WRITE (6,2530) N,K,IP,M,GAMCR,DEL,MODLAM,FF,T,E,TAU,XL,ZETA,       00002610
     1 IALT,ISTOP,IWT,IWS2,IWS3,IWS4,IWS6,IFSS1,NPRNT,SCALEP,SCALEY      00002620
      IF(IP.GT.0) WRITE(6,330) (IB(J),J=1,IP)                            00002630
  330 FORMAT(4H IB=,19I3)                                                00002640
      WRITE(6,340) FMT                                                   00002650
  340 FORMAT(5H FMT=,18A4)                                               00002660
      IF(SCALEP.GT.0.AND.IPRT.GE.0) WRITE(6,350) (BINV(J),J=1,K)         00002670
  350 FORMAT(/30H -INITIAL UNSCALED PARAMETERS-/(12X,4E17.8))            00002680
      IF (IFSS1.NE.1) GO TO 360                                          00002690
      WRITE(16,2520) TITLE                                               00002700
      WRITE (16,2530) N,K,IP,M,GAMCR,DEL,MODLAM,FF,T,E,TAU,XL,ZETA,      00002710
     1 IALT,ISTOP,IWT,IWS2,IWS3,IWS4,IWS6,IFSS1,NPRNT,SCALEP,SCALEY      00002720
      IF(IP.GT.0) WRITE(16,330) (IB(J),J=1,IP)                           00002730
      WRITE(16,340) FMT                                                  00002740
      IF(SCALEP.GT.0.AND.IPRT.GE.0) WRITE(16,350) (BINV(J),J=1,K)        00002750
  360 CONTINUE                                                           00002760
  370 CONTINUE                                                           00002770
      DO 380 I=1,K                                                       00002780
      G(I)=0.E0                                                          00002790
      DO 380 J=1,K                                                       00002800
  380 A(I,J)=0.E0                                                        00002810
      IF(IBKA-2) 390,400,400                                             00002820
  390 IFSS3=IWS3                                                         00002830
      IFSS2=IWS2                                                         00002840
      GO TO 410                                                          00002850
  400 IFSS3=1                                                            00002860
      GO TO 420                                                          00002870
  410 IF(IPRT.GE.0) WRITE (6,2540) (B(J),J=1,K)                          00002880
      IF (IFSS1.NE.1) GO TO 420                                          00002890
      IF(IPRT.GE.0) WRITE (16,2540) (B(J),J=1,K)                         00002900
  420 CONTINUE                                                           00002910
  430 FORMAT(/11H -UNSCALED-)                                            00002920
C--THIS IS I=1 TO N SPECIAL NON-DO LOOP                                  00002930
  450 I=1                                                                00002940
      DO 460 J=1,K                                                       00002950
  460 CALL UNSCAL(B(J),BINV(J),SCALEP)                                   00002960
      IF(IPRT.LT.0) WRITE(6,2540) (BINV(J),J=1,K)                        00002970
      IF(IFSS1.EQ.1.AND.IPRT.LT.0)WRITE(16,2540)(BINV(J),J=1,K)          00002980
      PHI=0.E0                                                           00002990
      IF (IFSS2.EQ.0) GO TO 480                                          00003000
      GO TO 510                                                          00003010
  470 IF (IFSS2.EQ.1) GO TO 520                                          00003020
C     ..................................................                 00003030
C                  THIS IS THE ANALYTICAL P(J) ROUTINE                   00003040
  480 IWHER=1                                                            00003050
C                  GET P(J) AND F                                        00003060
      GO TO 30                                                           00003070
  490 IF (IP.LE.0) GO TO 640                                             00003080
      DO 500 II=1,IP                                                     00003090
      IWS=IB(II)                                                         00003100
  500 P(IWS)=0.E0                                                        00003110
```

```
      GO TO 640                                                    00003120
C      ..................................................          00003130
C                   THIS IS THE ESTIMATED P(J) ROUTINE            00003140
C  (VIA K.M. BROWN S METHOD)                                       00003150
  510 CONTINUE                                                     00003160
      ISW=1                                                        00003170
      IF(XL.LT.0.1E-3) ISW=2                                       00003180
  520 IWHER=3                                                      00003190
      GO TO 30                                                     00003200
  530 FWS=FINV                                                     00003210
      FSAV=F                                                       00003220
      DO 540 II=1,NPRNT                                            00003230
  540 SPRNT(II)=PRNT(II)                                           00003240
      J=1                                                          00003250
  550 IF (IP.LE.0) GO TO 570                                       00003260
      DO 560 II=1,IP                                               00003270
      IF ((J-IB(II)).EQ.0) GO TO 610                               00003280
  560 CONTINUE                                                     00003290
  570 HH=DEL*ABS(BINV(J))                                          00003300
      IF(ISW.EQ.2) HH=1.E3*HH                                      00003310
      IF(HH.LE.5.E-5) HH=5.E-5                                     00003320
      TWS=B(J)                                                     00003330
      TWS1=BINV(J)                                                 00003340
      BINV(J)=TWS1+HH                                              00003350
      IWHER=4                                                      00003360
      GO TO 30                                                     00003370
  580 B(J)=TWS                                                     00003380
      BINV(J)=TWS1                                                 00003390
      IF(ISW.EQ.1) GO TO 600                                       00003400
C--CENTRAL DIFFERENCES (ISW=2--WHEN XL.LT..1E-3)                   00003410
      FHH=FINV                                                     00003420
      BINV(J)=TWS1-HH                                              00003430
      IWHER=5                                                      00003440
      GO TO 30                                                     00003450
  590 B(J)=TWS                                                     00003460
      BINV(J)=TWS1                                                 00003470
      P(J)=.5E0*(FHH-FINV)/HH                                      00003480
      GO TO 620                                                    00003490
C--FORWARD DIFFERENCES (ISW=1--WHEN XL.GE..1E-3)                   00003500
  600 P(J)=(FINV-FWS)/HH                                           00003510
      GO TO 620                                                    00003520
  610 P(J)=0.E0                                                    00003530
  620 J=J+1                                                        00003540
      IF ((J-K).LE.0) GO TO 550                                    00003550
      FINV=FWS                                                     00003560
      F=FSAV                                                       00003570
      DO 630 II=1,NPRNT                                            00003580
  630 PRNT(II)=SPRNT(II)                                           00003590
C                   END OF ESTIMATED P S ROUTINE                   00003600
C      ..................................................          00003610
C                   NOW, USE THE P(J) TO MAKE PARTIALS MATRIX      00003620
C--SET UP FOR SCALING PARTIAL DERIVATIVES AS SELECTED              00003630
```

```
  640 IF(SCALEP-1) 650,710,730                                          00003640
  650 IF(SCALEY-1) 750,660,690                                          00003650
  660 DEN=1.0E0/FINV                                                    00003660
  670 DO 680 JJ=1,K                                                     00003670
  680 P(JJ)=P(JJ)*DEN                                                   00003680
      GO TO 750                                                         00003690
  690 IF(IPRNT.LE.1) GO TO 700                                          00003700
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 660                              00003710
  700 DEN=1.0E0/SQRT(FINV*FINV+1.0E0)                                   00003720
      GO TO 670                                                         00003730
  710 DO 720 JJ=1,K                                                     00003740
  720 P(JJ)=BINV(JJ)*P(JJ)                                              00003750
      GO TO 650                                                         00003760
  730 DO 740 JJ=1,K                                                     00003770
      DEN=BINV(JJ)+SQRT(BINV(JJ)**2+1.0E0)                              00003780
  740 P(JJ)=0.5E0*(DEN+1.0E0/DEN)*P(JJ)                                 00003790
      GO TO 650                                                         00003800
  750 IF(IBKA.EQ.2) WRITE(13) (P(JJ),JJ=1,K)                            00003810
      DO 760 JJ=1,K                                                     00003820
      G(JJ)=G(JJ)+WT(I)*(Y(I)-F)*P(JJ)                                  00003830
      DO 760 II=JJ,K                                                    00003840
      A(II,JJ)=A(II,JJ)+WT(I)*P(II)*P(JJ)                               00003850
  760 A(JJ,II)=A(II,JJ)                                                 00003860
  770 WS=Y(I)-F                                                         00003870
      IF (IFSS3.LE.0) GO TO 810                                         00003880
C--LAST ITERATION RESULTS AND DATA MATRIX FOR PRINTING                  00003890
      IF(I.GT.1) GO TO 771                                              00003900
      IF(IOUT.EQ.0) GO TO 773                                           00003910
      WRITE(16,430)                                                     00003920
      WRITE(16,2550)                                                    00003930
773   IF(IPRT.LT.-1) WRITE(6,772)                                       00003940
772   FORMAT(/11H -UNSCALED-/3X,1HI,4X,3HOBS,11X,3HCAL,11X,3HRES,       00003950
     1 8X,6HX(I,1))                                                     00003960
771   IF(IPRT.LT.-1) WRITE (6,2700) I,Y(I),F,WS,PRNT(1)                 00003970
      IF(NPRNT.GT.0) GO TO 790                                          00003980
      IF (IFSS1.NE.1) GO TO 780                                         00003990
      WRITE (16,2700) I,Y(I),F,WS                                       00004000
  780 CONTINUE                                                          00004010
      GO TO 810                                                         00004020
790   CONTINUE                                                          00004030
      IF (IFSS1.NE.1) GO TO 800                                         00004040
      PERR=0.0                                                          00004050
      IF(F.NE.0.0) PERR=100.0*WS/ABS(F)                                 00004060
      WRITE (16,2700) I,Y(I),F,WS,PERR,(PRNT(JJ),JJ=1,NPRNT)            00004070
  800 CONTINUE                                                          00004080
  810 WS=Y(I)-F                                                         00004090
      PHI=PHI+WT(I)*WS*WS                                               00004100
      I=I+1                                                             00004110
      IF (I.LE.N) GO TO 470                                             00004120
C--THIS IN END OF I=1 TO N NON-DO LOOP                                  00004130
      IF(IBKA.NE.2) GO TO 860                                           00004140
C--PRINT UNSCALED PARTIALS SAVED ON FILE 13 (WHEN IBKA=2)                00004150
```

```
  820 FORMAT(/20H -UNSCALED PARTIALS-)                                  00004160
      IF(IOUT.EQ.1) WRITE(16,820)                                       00004170
      REWIND 13                                                         00004180
      DO 850 II=1,N                                                     00004190
      READ(13) (SA(JJ),JJ=1,K)                                          00004200
  830 FORMAT(2X,I3,5E18.8)                                              00004210
  840 FORMAT(2X,I3,5E18.8/(5X,5E18.8))                                  00004220
      IF(IOUT.EQ.1.AND.K.NE.5) WRITE(16,840) II,(SA(JJ),JJ=1,K)         00004230
      IF(IOUT.EQ.1.AND.K.EQ.5) WRITE(16,830) II,(SA(JJ),JJ=1,K)         00004240
  850 CONTINUE                                                          00004250
      REWIND 13                                                         00004260
      WRITE(6,430)                                                      00004270
      IF(IOUT.EQ.1) WRITE(16,430)                                       00004280
  860 CONTINUE                                                          00004290
      IF (IP.LE.0) GO TO 890                                            00004300
      DO 880 JJ=1,IP                                                    00004310
      IWS=IB(JJ)                                                        00004320
      DO 870 II=1,K                                                     00004330
      A(IWS,II)=0.E0                                                    00004340
  870 A(II,IWS)=0.E0                                                    00004350
  880 A(IWS,IWS)=1.E0                                                   00004360
  890 IF(IBKA-2) 900,1770,1780                                          00004370
C                   SAVE SQUARE ROOTS OF DIAGONAL ELEMENTS            00004380
  900 DO 910 I=1,K                                                      00004390
  910 SA(I)=SQRT(A(I,I))                                                00004400
      DO 950 I=1,K                                                      00004410
      DO 930 J=1,K                                                      00004420
      WS=SA(I)*SA(J)                                                    00004430
      IF (WS.GT.0.E0) GO TO 920                                         00004440
      A(I,J)=0.E0                                                       00004450
      GO TO 930                                                         00004460
  920 A(I,J)=A(I,J)/WS                                                  00004470
  930 CONTINUE                                                          00004480
      IF (SA(I).GT.0.E0) GO TO 940                                      00004490
      G(I)=0.E0                                                         00004500
      GO TO 950                                                         00004510
  940 G(I)=G(I)/SA(I)                                                   00004520
  950 CONTINUE                                                          00004530
      DO 960 I=1,K                                                      00004540
  960 A(I,I)=1.E0                                                       00004550
      PHIZ=PHI                                                          00004560
C                   WE NOW HAVE PHI ZERO                              00004570
      IF(IBKT-1) 970,980,970                                            00004580
  970 WRITE (13) A                                                      00004590
      REWIND 13                                                         00004600
      GO TO 1000                                                        00004610
  980 DO 990 II=1,K                                                     00004620
      III=II+10                                                         00004630
      DO 990 JJ=1,K                                                     00004640
  990 A(III,JJ)=A(II,JJ)                                                00004650
C     .....                                                           00004660
 1000 CONTINUE                                                          00004670
```

```
      IF (ITCT.GT.0) GO TO 1030                                          00004680
C                  FIRST ITERATION                                       00004690
      IF (XL.GT.0.E0) GO TO 1010                                         00004700
      XL=0.01E0                                                          00004710
 1010 ITCT=1                                                             00004720
      DO 1020 J=1,K                                                      00004730
 1020 BS(J)=B(J)                                                         00004740
C                  BS(J) CORRESPONDS TO PHIZ                             00004750
 1030 IBK1=1                                                             00004760
      WS=N-K+IP                                                          00004770
      IF(N.GT.K) SE=SQRT(PHIZ/WS)                                        00004780
      IF (IFSS3.GT.0) GO TO 1040                                         00004790
      WRITE (6,2560) ITER,PHIZ,SE,XLL,GAMMA,XL                           00004800
      IF (IFSS1.NE.1) GO TO 1320                                         00004810
      WRITE (16,2560) ITER,PHIZ,SE,XLL,GAMMA,XL                          00004820
      GO TO 1320                                                         00004830
 1040   WRITE(6,2490) PHIZ,SE,XL                                         00004840
      IF (IFSS1.NE.1) GO TO 1320                                         00004850
      WRITE (16,2490) PHIZ,SE,XL                                         00004860
      GO TO 1320                                                         00004870
 1050 PHIL=PHI                                                           00004880
C                  WE NOW HAVE PHI(LAMBDA)                               00004890
      DO 1060 J=1,K                                                      00004900
      IF(ABS(DB(J)/(ABS(B(J))+TAU)).GE.E) GO TO 1080                     00004910
 1060 CONTINUE                                                           00004920
      WRITE (6,2680)                                                     00004930
      IF (IFSS1.NE.1) GO TO 1070                                         00004940
      WRITE (16,2680)                                                    00004950
 1070 CONTINUE                                                           00004960
      GO TO 1670                                                         00004970
 1080 IF (IWS4.EQ.0) GO TO 1110                                          00004980
      IF (IWS4.EQ.1) GO TO 1090                                          00004990
      IWS4=IWS4-1                                                        00005000
      ITER=ITER+1                                                        00005010
      GO TO 1110                                                         00005020
 1090 WRITE (6,2690)                                                     00005030
      IF (IFSS1.NE.1) GO TO 1100                                         00005040
      WRITE (16,2690)                                                    00005050
 1100 CONTINUE                                                           00005060
      GO TO 1670                                                         00005070
 1110 XKDB=1.E0                                                          00005080
      IF (PHIL.GT.PHIZ) GO TO 1190                                       00005090
      XLS=XL                                                             00005100
      DO 1120 J=1,K                                                      00005110
      BA(J)=B(J)                                                         00005120
 1120 B(J)=BS(J)                                                         00005130
      IF (XL.GT..00000001E0) GO TO 1140                                  00005140
      DO 1130 J=1,K                                                      00005150
      B(J)=BA(J)                                                         00005160
 1130 BS(J)=B(J)                                                         00005170
      GO TO 370                                                          00005180
 1140 XL=XL/XNUFAC                                                       00005190
```

```
      IBK1=2                                                          00005200
      GO TO 1320                                                      00005210
 1150 PHL4=PHI                                                        00005220
C                     WE NOW HAVE PHI(LAMBDA/XNUFAC)                 00005230
      IF (PHL4.GT.PHIZ) GO TO 1170                                    00005240
      DO 1160 J=1,K                                                   00005250
 1160 BS(J)=B(J)                                                      00005260
      GO TO 370                                                       00005270
 1170 XL=XLS                                                          00005280
C1170 CONTINUE                                                       00005290
      DO 1180 J=1,K                                                   00005300
      BS(J)=BA(J)                                                     00005310
 1180 B(J)=BA(J)                                                      00005320
      GO TO 370                                                       00005330
 1190 IBK1=4                                                          00005340
      XLS=XL                                                          00005350
      XL=XL/XNUFAC                                                    00005360
      DO 1200 J=1,K                                                   00005370
 1200 B(J)=BS(J)                                                      00005380
      GO TO 1320                                                      00005390
 1210 IF (PHI.LE.PHIZ) GO TO 1260                                     00005400
      XL=XLS                                                          00005410
      IBK1=3                                                          00005420
 1220 XL=XL*XNUFAC                                                    00005430
 1230 DO 1240 J=1,K                                                   00005440
 1240 B(J)=BS(J)                                                      00005450
      GO TO 1320                                                      00005460
 1250 PHIT4=PHI                                                       00005470
C                     WE NOW HAVE PHI(XNUFAC*LAMBDA)                 00005480
      IF (PHIT4.GT.PHIZ) GO TO 1280                                   00005490
 1260 DO 1270 J=1,K                                                   00005500
 1270 BS(J)=B(J)                                                      00005510
      GO TO 370                                                       00005520
 1280 IF (GAMMA.GE.GAMCR) GO TO 1220                                  00005530
      XKDB=XKDB/2.E0                                                  00005540
      DO 1290 J=1,K                                                   00005550
      IF(ABS(DB(J)/(ABS(B(J))+TAU)).GE.E) GO TO 1230                  00005560
 1290 CONTINUE                                                        00005570
      DO 1300 J=1,K                                                   00005580
 1300 B(J)=BS(J)                                                      00005590
      MAXITR=MAXITR-1                                                 00005600
      WRITE (6,2740)                                                  00005610
      IF (IFSS1.NE.1) GO TO 1310                                      00005620
      WRITE (16,2740)                                                 00005630
 1310 CONTINUE                                                        00005640
      GO TO 1670                                                      00005650
C                                                                     00005660
C     .................................................               00005670
C                     SET UP FOR MATRIX INVERSION                     00005680
 1320 IF(IBKT-1) 1330,1340,1330                                       00005690
 1330 READ (13) A                                                     00005700
      REWIND 13                                                       00005710
```

```
      GO TO 1360                                                        00005720
 1340 DO 1350 II=1,K                                                    00005730
      III=II+10                                                         00005740
      DO 1350 JJ=1,K                                                    00005750
 1350 A(II,JJ)=A(III,JJ)                                                00005760
 1360 DO 1370 I=1,K                                                     00005770
 1370 A(I,I)=A(I,I)+XL                                                  00005780
C                  GET INVERSE OF A AND SOLVE FOR DB(J)S                00005790
      IBKM=1                                                            00005800
C      ..................................................               00005810
C                  THIS IS THE MATRIX INVERSION ROUTINE                 00005820
C                  K IS THE SIZE OF THE MATRIX                          00005830
 1380 IF(K.EQ.1) GO TO 1390                                             00005840
      CALL GJR (A,K,ZETA,MSING)                                         00005850
      IF(MSING-1) 1400,1400,1381                                        00005860
1381  CALL ERRMSG(20HSINGULAR MATRIX.....,4,6,16)                       00005870
C--SPECIAL CASE, K=1                                                    00005880
 1390 A(1,1)=1.0/A(1,1)                                                 00005890
 1400 IF(IBKM-1) 1410,1410,1840                                         00005900
C                  END OF MATRIX INVERSION, SOLVE FOR DB(J)             00005910
 1410 DO 1430 I=1,K                                                     00005920
      DB(I)=0.E0                                                        00005930
      DO 1420 J=1,K                                                     00005940
 1420 DB(I)=A(I,J)*G(J)+DB(I)                                           00005950
 1430 DB(I)=XKDB*DB(I)                                                  00005960
      XLL=0.E0                                                          00005970
      DTG=0.E0                                                          00005980
      GTG=0.E0                                                          00005990
      DO 1440 J=1,K                                                     00006000
      DB(J)=DB(J)/SA(J)                                                 00006010
      DTG=DTG+DB(J)*G(J)                                                00006020
      GTG=GTG+G(J)**2                                                   00006030
      B(J)=B(J)+DB(J)                                                   00006040
 1440 XLL=XLL+DB(J)*DB(J)                                               00006050
      KIP=K-IP                                                          00006060
      IF (KIP.EQ.1) GO TO 1480                                          00006070
      CGAM=DTG/SQRT(XLL*GTG)                                            00006080
      JGAM=1                                                            00006090
      IF (CGAM.GT.0.E0) GO TO 1450                                      00006100
      CGAM=ABS(CGAM)                                                    00006110
      JGAM=2                                                            00006120
 1450 GAMMA=57.2957795E0*(1.5707288E0+CGAM*(-0.2121144E0                00006130
     1+CGAM*(0.074261E0-CGAM*                                           00006140
     2.0187293E0)))*SQRT(1.0E0-CGAM)                                    00006150
      IF(JGAM-1) 1460,1490,1460                                         00006160
 1460 GAMMA=180.E0-GAMMA                                                00006170
      IF (XL.LT.1.0E0) GO TO 1490                                       00006180
      WRITE (6,2670) XL,GAMMA                                           00006190
      IF (IFSS1.NE.1) GO TO 1470                                        00006200
      WRITE (16,2670) XL,GAMMA                                          00006210
 1470 CONTINUE                                                          00006220
      GO TO 1670                                                        00006230
```

```
 1480 GAMMA=0.E0                                                        00006240
 1490 XLL=SQRT(XLL)                                                     00006250
      IBK2=1                                                            00006260
      GO TO 1540                                                        00006270
 1500 IF (IFSS3.LE.0) GO TO 1530                                        00006280
      WRITE (6,2500) (DB(J),J=1,K)                                      00006290
      IF (IFSS1.NE.1) GO TO 1510                                        00006300
      WRITE (16,2500) (DB(J),J=1,K)                                     00006310
 1510 CONTINUE                                                          00006320
      WRITE (6,2510) PHI,XL,GAMMA,XLL                                   00006330
      IF (IFSS1.NE.1) GO TO 1520                                        00006340
      WRITE (16,2510) PHI,XL,GAMMA,XLL                                  00006350
 1520 CONTINUE                                                          00006360
C--PRESET XNUFAC--(IF MODLAM=1)                                         00006370
 1530 GO TO (1570,1150,1250,1210),IBK1                                  00006380
C                                                                       00006390
C     ..................................................................00006400
C                   CALCULATE PHI                                       00006410
 1540 I=1                                                               00006420
      DO 1550 JJ=1,K                                                    00006430
 1550 CALL UNSCAL(B(JJ),BINV(JJ),SCALEP)                                00006440
      PHI=0.E0                                                          00006450
      IWHER=2                                                           00006460
      GO TO 30                                                          00006470
 1560 PHI=PHI+WT(I)*(Y(I)-F)**2                                         00006480
      I=I+1                                                             00006490
      IF (I.LE.N) GO TO 30                                              00006500
      GO TO (1500,2290,1770,2200,2220,2240),IBK2                        00006510
C=================================================                      00006520
C--DETERMINE AN EFFECTIVE MARQUARDT LAMBDA FACTOR (XNUFAC)              00006530
C  BASED ON HISTORY OF SUM OF SQUARES STORED IN LATEST SS(4)--          00006540
 1570 IF(MODLAM.EQ.0) GO TO 1050                                        00006550
      SS(ISS)=PHI                                                       00006560
      INU0=INU                                                          00006570
      GO TO (1590,1580,1600,1610),ISS                                   00006580
C--MACHINE FAILURE IF ISS.GT.4 OR ISS.LT.1                              00006590
C--    STOP 4                                                           00006600
 1580 IS1=0                                                             00006610
      IF(SS(2).GT.SS(1)) IS1=1                                          00006620
 1590 ISS=ISS+1                                                         00006630
      GO TO 1660                                                        00006640
 1600 IS2=0                                                             00006650
      IF(SS(3).GT.SS(2)) IS2=1                                          00006660
      IF(IS1.EQ.IS2) GO TO 1590                                         00006670
      INU=INU0-1                                                        00006680
      GO TO 1590                                                        00006690
 1610 IS3=0                                                             00006700
      IF(SS(4).GT.SS(3)) IS3=1                                          00006710
      IF(IS1.EQ.IS2.AND.IS3.EQ.IS2) GO TO 1620                          00006720
      IF(IS1.EQ.0.AND.IS2.EQ.0.AND.IS3.EQ.1) GO TO 1640                 00006730
      IF(IS1.EQ.1.AND.IS2.EQ.0.AND.IS3.EQ.1) GO TO 1640                 00006740
      IF(IS1.EQ.1.AND.IS2.EQ.1.AND.IS3.EQ.0) GO TO 1640                 00006750
```

```
      GO TO 1650                                                        00006760
 1620 IF(IS1.EQ.0) GO TO 1630                                           00006770
      IF(INU0.GE.3) GO TO 1650                                          00006780
      INU=3                                                             00006790
      GO TO 1650                                                        00006800
 1630 IF(INU0.GE.5) GO TO 1650                                          00006810
      INU=INU0+1                                                        00006820
      GO TO 1650                                                        00006830
 1640 IF(INU0.LE.1) GO TO 1650                                          00006840
      INU=INU0-1                                                        00006850
 1650 IS1=IS2                                                           00006860
      IS2=IS3                                                           00006870
      SS(3)=SS(4)                                                       00006880
 1660 XNUFAC=XNU(INU)                                                   00006890
      GO TO 1050                                                        00006900
C                                                                       00006910
C                                                                       00006920
C     ..................................................................00006930
C                  THIS IS THE CONFIDENCE LIMIT CALCULATION             00006940
 1670 ITR=MAXITR-IWS4+1                                                 00006950
      WRITE(6,1680) ITR                                                 00006960
 1680 FORMAT(1X,I4,11H ITERATIONS)                                      00006970
      IF(IFSS1.EQ.1) WRITE(16,1680) ITR                                 00006980
      DO 1690 J=1,K                                                     00006990
      CALL UNSCAL(BS(J),BINV(J),SCALEP)                                 00007000
      BS(J)=BINV(J)                                                     00007010
 1690 B(J)=BS(J)                                                        00007020
      WRITE(6,2520) TITLE                                               00007030
      IF (IFSS1.NE.1) GO TO 1700                                        00007040
      WRITE(16,2520) TITLE                                              00007050
 1700 CONTINUE                                                          00007060
      IBKA=2                                                            00007070
C--UNSCALE BOTH PARAMETER AND OBSERVATION SPACES PRIOR                  00007080
C  TO FINAL STATISTICS ON LAST INTERATION--AND WHERE                    00007090
C  IBKA=2, IFSS3=0..                                                    00007100
C               THIS WILL PRINT OBS,CAL,RES,ETC.                        00007110
C               AND SAVE UNSCALED PARTIALS ON FILE FILE13..             00007120
      IF(IPRT.GE.0) WRITE(6,1710) (BINV(J),J=1,K)                       00007130
 1710 FORMAT(/28H -FINAL UNSCALED PARAMETERS-/(12X,4E17.8))             00007140
      IF(IFSS1.EQ.1.AND.IPRT.GE.0) WRITE(16,1710) (BINV(J),J=1,K)       00007150
      IF(SCALEY.EQ.0) GO TO 1760                                        00007160
      DO 1750 I=1,N                                                     00007170
      IF(SCALEY.NE.1) GO TO 1730                                        00007180
 1720 Y(I)=EXP(Y(I))                                                    00007190
      GO TO 1750                                                        00007200
 1730 IF(IPRNT.LE.1) GO TO 1740                                         00007210
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 1720                             00007220
 1740 Y(I)=SINH(Y(I))                                                   00007230
 1750 CONTINUE                                                          00007240
 1760 LSCALP=SCALEP                                                     00007250
      LSCALY=SCALEY                                                     00007260
      SCALEP=0                                                          00007270
```

```
      SCALEY=0                                                          00007280
      GO TO 370                                                         00007290
 1770 CONTINUE                                                          00007300
 1780 WS=N-K+IP                                                         00007310
      IF(N.GT.K) SE=SQRT(PHI/WS)                                        00007320
      PHIZ=PHI                                                          00007330
      WRITE (6,2490) PHIZ,SE,XL                                         00007340
      IF (IFSS1.NE.1) GO TO 1790                                        00007350
      WRITE (16,2490) PHIZ,SE,XL                                        00007360
C                                                                       00007370
C                   WE NOW HAVE MATRIX A                                00007380
 1790 IF(IBKT-1) 1800,1810,1800                                         00007390
 1800 WRITE (13) A                                                      00007400
      REWIND 13                                                         00007410
      GO TO 1830                                                        00007420
 1810 DO 1820 II=1,K                                                    00007430
      III=II+10                                                         00007440
      DO 1820 JJ=1,K                                                    00007450
 1820 A(III,JJ)=A(II,JJ)                                                00007460
 1830 IBKM=2                                                            00007470
      GO TO 1380                                                        00007480
C                                                                       00007490
C                   WE NOW HAVE C = A INVERSE                           00007500
 1840 DO 1850 J=1,K                                                     00007510
      IF (A(J,J).LT.0.E0) GO TO 1860                                    00007520
 1850 SA(J)=SQRT(A(J,J))                                                00007530
      GO TO 1870                                                        00007540
 1860 IBOUT=1                                                           00007550
 1870 KST=-4                                                            00007560
      IF (IFSS1.NE.1) GO TO 1880                                        00007570
      WRITE (16,2600)                                                   00007580
 1880 KST=KST+5                                                         00007590
      KEND=KST+4                                                        00007600
      IF (KEND.LT.K) GO TO 1890                                         00007610
      KEND=K                                                            00007620
 1890 DO 1910 I=1,K                                                     00007630
      IF (IFSS1.NE.1) GO TO 1900                                        00007640
      WRITE (16,2620) I,(A(I,J),J=KST,KEND)                             00007650
 1900 CONTINUE                                                          00007660
 1910 CONTINUE                                                          00007670
      IF (KEND.LT.K) GO TO 1880                                         00007680
      IF (IBOUT.EQ.0) GO TO 1920                                        00007690
      WRITE (6,2760)                                                    00007700
      IF (IFSS1.NE.1) GO TO 220                                         00007710
      WRITE (16,2760)                                                   00007720
      GO TO 220                                                         00007730
 1920 DO 1940 I=1,K                                                     00007740
      DO 1940 J=1,K                                                     00007750
      WS=SA(I)*SA(J)                                                    00007760
      IF (WS.GT.0.E0) GO TO 1930                                        00007770
      A(I,J)=0.E0                                                       00007780
      GO TO 1940                                                        00007790
```

```
 1930 A(I,J)=A(I,J)/WS                                          00007800
 1940 CONTINUE                                                  00007810
      DO 1950 J=1,K                                             00007820
 1950 A(J,J)=1.E0                                               00007830
      IF (IFSS1.NE.1) GO TO 1960                                00007840
      WRITE (16,2610)                                           00007850
 1960 CONTINUE                                                  00007860
      KST=-9                                                    00007870
 1970 KST=KST+10                                                00007880
      KEND=KST+9                                                00007890
      IF (KEND.LT.K) GO TO 1980                                 00007900
      KEND=K                                                    00007910
 1980 DO 2000 I=1,K                                             00007920
      IF (IFSS1.NE.1) GO TO 1990                                00007930
      WRITE (16,2750) I,(A(I,J),J=KST,KEND)                     00007940
 1990 CONTINUE                                                  00007950
 2000 CONTINUE                                                  00007960
      IF (KEND.LT.K) GO TO 1970                                 00007970
C                  GET T*SE*SQRT(C(I,I))                       00007980
      DO 2010 J=1,K                                             00007990
 2010 SA(J)=SE*SA(J)                                            00008000
      IF(IBKT-1) 2020,2030,2020                                 00008010
 2020 READ (13) A                                               00008020
      REWIND 13                                                 00008030
      GO TO 2050                                                00008040
 2030 DO 2040 II=1,K                                            00008050
      III=II+10                                                 00008060
      DO 2040 JJ=1,K                                            00008070
 2040 A(II,JJ)=A(III,JJ)                                        00008080
 2050 CONTINUE                                                  00008090
      WRITE (6,2640)                                            00008100
      IF (IFSS1.NE.1) GO TO 2060                                00008110
      WRITE (16,2630)                                           00008120
 2060 CONTINUE                                                  00008130
      WS=K-IP                                                   00008140
      DO 2120 J=1,K                                             00008150
      IF (IP.LE.0) GO TO 2080                                   00008160
      DO 2070 I=1,IP                                            00008170
      IF (J.EQ.IB(I)) GO TO 2100                                00008180
 2070 CONTINUE                                                  00008190
C                                                               00008200
C--COMPUTE STD.ERR, CONF. LIMITS, AND STD.ERR/PARM.             00008210
C                                                               00008220
 2080 HJTD=SQRT(WS*FF)*SA(J)                                    00008230
      STE=SA(J)                                                 00008240
      TWS=STE*T                                                 00008250
      OPL=BINV(J)-TWS                                           00008260
      OPU=BINV(J)+TWS                                           00008270
      SPL=BINV(J)-HJTD                                          00008280
      SPU=BINV(J)+HJTD                                          00008290
      HJTD=0.0                                                  00008300
      IF(BINV(J).NE.0.0) HJTD=STE/BINV(J)                       00008310
```

```
      WRITE (6,2720) J,STE,OPL,OPU,HJTD                                 00008320
      IF (IFSS1.NE.1) GO TO 2090                                        00008330
      WRITE (16,2720) J,STE,OPL,OPU,SPL,SPU,HJTD                        00008340
 2090 CONTINUE                                                          00008350
      GO TO 2120                                                        00008360
 2100 WRITE (6,2570) J                                                  00008370
      IF (IFSS1.NE.1) GO TO 2110                                        00008380
      WRITE (16,2570) J                                                 00008390
 2110 CONTINUE                                                          00008400
 2120 CONTINUE                                                          00008410
C                  NONLINEAR CONFIDENCE LIMIT                          00008420
      IF (IWS6.EQ.1.OR.N.EQ.K) GO TO 220                                00008430
      WS=K-IP                                                           00008440
      WS1=N-K+IP                                                        00008450
      PKN=WS/WS1                                                        00008460
      PC=PHIZ*(1.E0+FF*PKN)                                             00008470
      WRITE (6,2650) PC                                                 00008480
      IF (IFSS1.NE.1) GO TO 2130                                        00008490
      WRITE (16,2650) PC                                                00008500
 2130 CONTINUE                                                          00008510
      WRITE (6,2660)                                                    00008520
      IF (IFSS1.NE.1) GO TO 2140                                        00008530
      WRITE (16,2660)                                                   00008540
 2140 CONTINUE                                                          00008550
      IFSS3=1                                                           00008560
C--      NON- DO LOOP J=1,K                                             00008570
C  (SINCE CONTROL JUMPS OUT AND BACK INSIDE LOOP)                       00008580
      J=1                                                               00008590
 2150 IBKP=1                                                            00008600
      DO 2160 JJ=1,K                                                    00008610
 2160 B(JJ)=BS(JJ)                                                      00008620
      IF (IP.LE.0) GO TO 2180                                           00008630
      DO 2170 JJ=1,IP                                                   00008640
      IF (J.EQ.IB(JJ)) GO TO 2380                                       00008650
 2170 CONTINUE                                                          00008660
 2180 DD=-1.E0                                                          00008670
      IBKN=1                                                            00008680
 2190 D=DD                                                              00008690
      B(J)=BS(J)+D*SA(J)                                                00008700
      IBK2=4                                                            00008710
      GO TO 1540                                                        00008720
 2200 PHI1=PHI                                                          00008730
      IF (PHI1.GE.PC) GO TO 2230                                        00008740
 2210 D=D+DD                                                            00008750
      IF (D/DD.GE.5.E0) GO TO 2420                                      00008760
      B(J)=BS(J)+D*SA(J)                                                00008770
      IBK2=5                                                            00008780
      GO TO 1540                                                        00008790
 2220 PHID=PHI                                                          00008800
      IF (PHID.LT.PC) GO TO 2210                                        00008810
      IF (PHID.GE.PC) GO TO 2250                                        00008820
 2230 D=D/2.E0                                                          00008830
```

```
      IF (D/DD.LE..001E0) GO TO 2420                                    00008840
      B(J)=BS(J)+D*SA(J)                                                00008850
      IBK2=6                                                            00008860
      GO TO 1540                                                        00008870
 2240 PHID=PHI                                                          00008880
      IF (PHID.GT.PC) GO TO 2230                                        00008890
 2250 XK1=PHIZ/D+PHI1/(1.E0-D)+PHID/(D*(D-1.E0))                        00008900
      XK2=-(PHIZ*(1.E0+D)/D+D/(1.E0-D)*PHI1+PHID/(D*(D-1.E0)))          00008910
      XK3=PHIZ-PC                                                       00008920
      BC=(SQRT(XK2*XK2-4.E0*XK1*XK3)-XK2)/(2.E0*XK1)                    00008930
      IF(IBKN-1) 2260,2260,2270                                         00008940
 2260 B(J)=BS(J)-SA(J)*BC                                               00008950
      GO TO 2280                                                        00008960
 2270 B(J)=BS(J)+SA(J)*BC                                               00008970
 2280 IBK2=2                                                            00008980
      GO TO 1540                                                        00008990
 2290 IF(IBKN-1) 2300,2300,2310                                         00009000
 2300 IBKN=2                                                            00009010
      DD=1.E0                                                           00009020
      BL=B(J)                                                           00009030
      PL=PHI                                                            00009040
      GO TO 2190                                                        00009050
 2310 BU=B(J)                                                           00009060
      PU=PHI                                                            00009070
      GO TO (2320,2340,2360,2400), IBKP                                 00009080
 2320 WRITE (6,2620) J,BL,PL,BU,PU                                      00009090
      IF (IFSS1.NE.1) GO TO 2330                                        00009100
      WRITE (16,2620) J,BL,PL,BU,PU                                     00009110
 2330 CONTINUE                                                          00009120
      GO TO 2470                                                        00009130
 2340 WRITE (6,2590) J,BU,PU                                            00009140
      IF (IFSS1.NE.1) GO TO 2350                                        00009150
      WRITE (16,2590) J,BU,PU                                           00009160
 2350 CONTINUE                                                          00009170
      GO TO 2470                                                        00009180
 2360 WRITE (6,2620) J,BL,PL                                            00009190
      IF (IFSS1.NE.1) GO TO 2370                                        00009200
      WRITE (16,2620) J,BL,PL                                           00009210
 2370 CONTINUE                                                          00009220
      GO TO 2470                                                        00009230
 2380 WRITE (6,2570) J                                                  00009240
      IF (IFSS1.NE.1) GO TO 2390                                        00009250
      WRITE (16,2570) J                                                 00009260
 2390 CONTINUE                                                          00009270
      GO TO 2470                                                        00009280
 2400 WRITE (6,2580) J                                                  00009290
      IF (IFSS1.NE.1) GO TO 2410                                        00009300
      WRITE (16,2580) J                                                 00009310
 2410 CONTINUE                                                          00009320
      GO TO 2470                                                        00009330
 2420 IF(IBKN-1) 2430,2430,2440                                         00009340
C                  DELETE LOWER PRINT                                  00009350
```

```
 2430 IBKP=2                                                            00009360
      GO TO 2290                                                        00009370
 2440 IF(IBKP-1) 2450,2450,2460                                         00009380
C                  DELETE UPPER PRINT                                   00009390
 2450 IBKP=3                                                            00009400
      GO TO 2290                                                        00009410
C                  LOWER IS ALREADY DELETED, SO DELETE BOTH             00009420
 2460 IBKP=4                                                            00009430
      GO TO 2290                                                        00009440
C--END OF NON- DO LOOP J=1,K                                            00009450
 2470 J=J+1                                                             00009460
      IF(J.LE.K) GO TO 2150                                             00009470
      GO TO 220                                                         00009480
C     ..................................................................00009490
 2480 FORMAT(18A4)                                                      00009500
 2490   FORMAT(/13X,4H PHI,14X,4H S E,9X,7H LAMBDA/5X,2E18.8,E13.3)     00009510
 2500 FORMAT (/12H INCREMENTS ,4E17.8/(12X,4E17.8))                     00009520
 2510 FORMAT (13X,4H PHI10X,7H LAMBDA6X,7H GAMMA 6X,7H LENGTH/5X,E18.8,300009530
     1E13.3)                                                            00009540
 2520 FORMAT(16H1M A R Q R T  --,5X,16A5)                               00009550
 2530   FORMAT(/5H N = ,I4,8X,4HK = ,I3,9X,5HIP = ,I3,8X,4HM = ,I2,10X, 00009560
     1 6HGAMCR=,E9.3/5H DEL=,E10.3,2X,9HMODLAM = ,I1,6X,3HFF=,E10.3,3X, 00009570
     2 2HT=,E10.3,4X,2HE=,E10.3/5H TAU=,E10.3,2X,3HXL=,E10.3,3X,        00009580
     3 5HZETA=,E10.3,8H IALT = ,I2,7X,8HISTOP = ,I1/7H IWT = ,I1,9X,    00009590
     4 7HIDER = ,I1,8X,7HIPRT = ,I2,7X,8HNITER = ,I4,4X,7HINON = ,I1/   00009600
     5 8H IOUT = ,I2,7X,                                                00009610
     6 8HNPRNT = ,I1,7X,9HSCALEP = ,I1,6X,9HSCALEY = ,I1/)              00009620
 2540 FORMAT (/12H PARAMETERS ,4E17.8/(12X,4E17.8))                     00009630
 2550   FORMAT(3X,1HI,4X,8HOBS.Y(I),6X,3HCAL,11X,3HRES,8X,8H%RES.ERR,6X,00009640
     1 6HX(I,1),8X,6HX(I,2),8X,6HX(I,3),8X,6HX(I,4),8X,6HX(I,5))        00009650
 2560   FORMAT(/1X,4HITER,8X,4H PHI,14X,4H S E,11X,7H LENGTH,6X,        00009660
     1 7H GAMMA ,6X,7H LAMBDA/1X,I4,2E18.8,3E13.3)                      00009670
 2570 FORMAT (2X,I3,20H PARAMETER NOT USED )                            00009680
 2580 FORMAT (2X,I3,12H NONE FOUND )                                    00009690
 2590 FORMAT (2X,I3,36X,2E18.8)                                         00009700
 2600 FORMAT (1H /13H PTP INVERSE )                                     00009710
 2610 FORMAT (1H /30H PARAMETER CORRELATION MATRIX )                    00009720
 2620 FORMAT (2X,I3,5E18.8)                                             00009730
 2630   FORMAT(//4X,13HPARAMETER STD,17X,15HONE - PARAMETER,21X,        00009740
     1 14H SUPPORT PLANE/11X,6H ERROR,12X,6H LOWER,12X,6H UPPER,12X,    00009750
     2 6H LOWER,12X,6H UPPER,10X,14HSTD.ERROR/PARM)                     00009760
 2640   FORMAT(//4X,13HPARAMETER STD,17X,15HONE - PARAMETER/11X,        00009770
     1 6H ERROR,12X,6H LOWER,12X,6H UPPER,10X,14HSTD.ERROR/PARM)        00009780
 2650 FORMAT (//30H  NONLINEAR CONFIDENCE LIMITS //13H PHI CRITICAL,    00009790
     1 E15.8)                                                           00009800
 2660 FORMAT (1H /6H  PARA6X,8H LOWER B8X,10H LOWER PHI10X,8H UPPER B8X,00009810
     110H UPPER PHI)                                                    00009820
 2670 FORMAT (/19H -GAMMA LAMBDA TEST,5X,2E13.3)                        00009830
 2680 FORMAT (/15H -EPSILON TEST )                                      00009840
 2690 FORMAT (/12H -FORCE OFF )                                         00009850
 2700 FORMAT(1X,I3,2E14.6,E11.3,6E14.6)                                 00009860
 2720 FORMAT (2X,I3,6E18.8)                                             00009870
```

```
 2730 FORMAT (1H )                                                      00009880
 2740 FORMAT (/20H -GAMMA EPSILON TEST)                                 00009890
 2750 FORMAT (3X,I5,2X,10F10.4)                                         00009900
 2760 FORMAT (/27H  NEGATIVE DIAGONAL ELEMENT)                          00009910
      END                                                               00009920

      SUBROUTINE GJR (A,N,EPS,MSING)                                    00009930
C     GAUSS-JORDAN-RUTISHAUSER MATRIX INVERSION WITH DOUBLE PIVOTING.   00009940
      DIMENSION A(20,20),B(20),C(20),P(20),Q(20)                        00009950
      INTEGER P,Q                                                       00009960
      MSING=1                                                           00009970
      DO 140 K=1,N                                                      00009980
C     DETERMINATION OF THE PIVOT ELEMENT                                00009990
      PIVOT=0.E0                                                        00010000
      DO 20 I=K,N                                                       00010010
      DO 20 J=K,N                                                       00010020
      IF(ABS(A(I,J))-ABS(PIVOT)) 20,20,10                               00010030
   10 PIVOT=A(I,J)                                                      00010040
      P(K)=I                                                            00010050
      Q(K)=J                                                            00010060
   20 CONTINUE                                                          00010070
      IF(ABS(PIVOT)-EPS) 220,220,30                                     00010080
C     EXCHANGE OF THE PIVOTAL ROW WITH THE KTH ROW                      00010090
   30 IF (P(K)-K) 40,60,40                                              00010100
   40 DO 50 J=1,N                                                       00010110
      L=P(K)                                                            00010120
      Z=A(L,J)                                                          00010130
      A(L,J)=A(K,J)                                                     00010140
   50 A(K,J)=Z                                                          00010150
C     EXCHANGE OF THE PIVOTAL COLUMN WITH THE KTH COLUMN                00010160
   60 IF (Q(K)-K) 70,90,70                                              00010170
   70 DO 80 I=1,N                                                       00010180
      L=Q(K)                                                            00010190
      Z=A(I,L)                                                          00010200
      A(I,L)=A(I,K)                                                     00010210
   80 A(I,K)=Z                                                          00010220
   90 CONTINUE                                                          00010230
C     JORDAN STEP                                                       00010240
      DO 130 J=1,N                                                      00010250
      IF (J-K) 110,100,110                                              00010260
  100 B(J)=1.0E0/PIVOT                                                  00010270
      C(J)=1.0E0                                                        00010280
      GO TO 120                                                         00010290
  110 B(J)=-A(K,J)/PIVOT                                                00010300
      C(J)=A(J,K)                                                       00010310
  120 A(K,J)=0.0E0                                                      00010320
  130 A(J,K)=0.0E0                                                      00010330
      DO 140 I=1,N                                                      00010340
      DO 140 J=1,N                                                      00010350
  140 A(I,J)=A(I,J)+C(I)*B(J)                                           00010360
C     REORDERING THE MATRIX                                             00010370
      DO 200 M=1,N                                                      00010380
```

```
      K=N-M+1                                                          00010390
      IF (P(K)-K) 150,170,150                                          00010400
  150 DO 160 I=1,N                                                     00010410
      L=P(K)                                                           00010420
      Z=A(I,L)                                                         00010430
      A(I,L)=A(I,K)                                                    00010440
  160 A(I,K)=Z                                                         00010450
  170 IF (Q(K)-K) 180,200,180                                          00010460
  180 DO 190 J=1,N                                                     00010470
      L=Q(K)                                                           00010480
      Z=A(L,J)                                                         00010490
      A(L,J)=A(K,J)                                                    00010500
  190 A(K,J)=Z                                                         00010510
  200 CONTINUE                                                         00010520
  210 RETURN                                                           00010530
  220 PRINT 230, P(K),Q(K),PIVOT                                       00010540
  230 FORMAT (/16H SINGULAR MATRIX3H I=I3,3H J=I3,7H PIVOT=E16.8/)     00010550
      MSING=2                                                          00010560
      GO TO 210                                                        00010570
      END                                                              00010580

      SUBROUTINE UNSCAL(BIN,BOUT,SCALEP)                               00010590
C// MODIFIED TO TRAP ERRORS >10**38 ON MULTICS                         00010600
C--UNSCALE PARMETER BIN TO BOUT VIA SCALEP                            00010610
      INTEGER SCALEP                                                   00010620
      IF(SCALEP-1) 10,20,30                                            00010630
   10 BOUT=BIN                                                         00010640
      GO TO 40                                                         00010650
   20 IF(BIN.GT.88.028) GO TO 99                                       00010660
      BOUT= EXP_(BIN)                                                  00010670
      GO TO 40                                                         00010680
   30 BOUT= SINH(BIN)                                                  00010690
   40 RETURN                                                           00010700
   99 WRITE(6,699) BIN                                                 00010710
      WRITE(16,699) BIN                                                00010720
699   FORMAT('0"UNSCAL" ARG=',E16.8,' >88.028 FOR EXP_() ON MULTICS'/  00010730
     & ' --CHECK ALL $PARMS AND DATA --IF OK, THEN--'/                  00010740
     & ' --TRY RESTARTING WITH DIFFERENT SCALING OPTION(S) --OR--'/     00010750
     & ' --RESTART WITH BETTER "GUESSED" STARTING PARAMETERS.')        00010760
      CALL CLOSE_FILE('-ALL')                                          00010770
      STOP                                                             00010780
      END                                                              00010790

      REAL FUNCTION ASINH(X)                                           00010800
C--INVERSE HYPERBOLIC SIN FUNCTION                                     00010810
C                                                                      00010820
      REAL*8 X2                                                        00010830
      X2=X                                                             00010840
      ASINH=DLOG(X2+DSQRT(X2*X2+1.0D0))                                00010850
      RETURN                                                           00010860
      END                                                              00010870
```

```
      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                   00010880
C--ERROR MESSAGE WRITE ROUTINE AND STOP, WHERE--                        00010890
C                                                                       00010900
C       MSG=    ANY MULTIPLE OF 5 CHARACTERS--MAX. OF 120               00010910
C               (USE NH-------- FORM FOR ANSI COMPATABILITY)            00010920
C       M5=     NO.CHARS IN MSG/5 (REMAINDER MUST BE 0) 1.LE.M5.LE.24   00010930
C       I6=     1ST UNIT  FOR WRITE(I6, ) MSG -- USUALLY I6=6 FOR LPT.  00010940
C               IF I6.LE.0 UNIT I6 IGNORED.                             00010950
C       I9=     2ND UNIT  FOR WRITE(I9, ) MSG --                        00010960
C               IF I9.LE.0, UNIT I9 IGNORED.                            00010970
C--MESSAGE WRITTEN IN FORM--                                            00010980
C       /ERROR--MSG HERE                                                00010990
C                                                                       00011000
      DIMENSION MSG(30)                                                 00011010
      J=5*M5                                                            00011020
      K=J/4+MOD(J,4)                                                    00011030
      IF(I6.GT.0) WRITE(I6,10) (MSG(I),I=1,K)                           00011040
   10 FORMAT(/8H ERROR--,30A4)                                          00011050
      IF(I9.GT.0) WRITE(I9,10) (MSG(I),I=1,K)                           00011060
      CALL CLOSE_FILE('-ALL')                                           00011070
C                                                                       00011080
      STOP                                                              00011090
      END                                                               00011100

      SUBROUTINE POLAR2(Z,AMP,PHZ180)                                   00011110
C       PARMS  Z  = GIVEN COMPLEX COORDS Z=(X,Y)                        00011120
C              AMP= COMPUTED AMPLITUDE.                                 00011130
C              PHZ180 = COMPUTED PHASE IN (-180.0,180.0) DEGREES.       00011140
C                                                                       00011150
      COMPLEX Z                                                         00011160
      DATA PI,PI2/3.1415927,6.2831853/                                  00011170
      ZR=REAL(Z)                                                        00011180
      ZI=AIMAG(Z)                                                       00011190
      IF(ZR.EQ.0.AND.ZI.EQ.0) GO TO 9                                   00011200
      PV=ATAN2(ABS(ZI),ABS(ZR))                                         00011210
      IF(ZI.GE.0.AND.ZR.GE.0) GO TO 10                                  00011220
      IF(ZI.GE.0.AND.ZR.LT.0) GO TO 20                                  00011230
      IF(ZI.LT.0.AND.ZR.LE.0) GO TO 30                                  00011240
      RAD=PI2-PV                                                        00011250
      GO TO 40                                                          00011260
    9 PHZ180=0.                                                         00011270
      AMP=0.                                                            00011280
      RETURN                                                            00011290
   10 RAD=PV                                                            00011300
      GO TO 40                                                          00011310
   20 RAD=PI-PV                                                         00011320
      GO TO 40                                                          00011330
   30 RAD=PI+PV                                                         00011340
   40 AMP=SQRT(ZR*ZR+ZI*ZI)                                             00011350
      PHZ180=57.29577951*RAD                                            00011360
      IF(PHZ180.GT.180.0) PHZ180=PHZ180-360.0                           00011370
      RETURN                                                            00011380
```

```
      END                                                                00011390

      SUBROUTINE RECUR1(G,V1,F1)                                         00011400
C--BACKWARD RECURRENCE FOR COMPLEX V1,F1 GIVEN REAL*4 ARGUMENT G AND:    00011410
COMMON/MODEL/ PARAMETERS:                                                00011420
C     K(10) = NORMALIZED CONDUCTIVITY ARRAY (M VALUES,WHERE K(1)=1.0).   00011430
C     D(9)  = LAYER THICKNESS ARRAY (M-1 VALUES) D=2*THICKNESS/DEL.      00011440
C     M     = NUMBER LAYERS (M.GE.1.AND.M.LE.10)                         00011450
C             SPECIAL CASE WHEN M=1 (HOMOGENEOUS--D IGNORED)             00011460
C                                                                        00011470
C--NOTE: G,K,D ARE REAL*4                                                00011480
C                                                                        00011490
C                                                                        00011500
      COMMON/MODEL/K,D,M                                                 00011510
      REAL*4 K(10),D(9)                                                  00011520
      COMPLEX C,VM,V1,F1,EVD,ONE                                         00011530
      DATA ONE/(1.0,0.0)/                                                00011540
      F1=ONE                                                             00011550
      G2=G*G                                                             00011560
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                       00011570
      IF(M.EQ.1) GO TO 2                                                 00011580
      J=M-1                                                              00011590
    1 V1=CSQRT(CMPLX(G2,2.0*K(J)))                                       00011600
      EVD=CEXP(-V1*D(J))                                                 00011610
      C=(ONE-EVD)/(ONE+EVD)                                              00011620
      F1=(VM*F1+V1*C)/(V1+VM*F1*C)                                       00011630
      IF(J.EQ.1) GO TO 3                                                 00011640
      J=J-1                                                              00011650
      VM=V1                                                              00011660
      GO TO 1                                                            00011670
    2 V1=VM                                                              00011680
    3 RETURN                                                             00011690
      END                                                                00011700

      SUBROUTINE RECURF(G,DEL,SIG1,V1,F1,PF1,JJ)                         00011710
C--GET PF1=PARTIAL OF F1 W/R PARM. JJ, EVALUATED AT                      00011720
C  THE GIVEN G,DEL, AND SIG1 (OTHER MODEL PARMS IN COMMON/MODEL/)        00011730
C  ALSO GIVEN ARE V1,F1 AS IN RECUR1.                                    00011740
C                                                                        00011750
      IMPLICIT COMPLEX (A-H,O-Z)                                         00011760
      REAL   K,D,G,G2,DEL,SIG1                                           00011770
      COMMON/MODEL/K,D,M                                                 00011780
      DIMENSION K(10),D(9)                                               00011790
      DATA ONE,ZERO,CI/(1.0,0.0),(0.0,0.0),(0.0,1.0)/                    00011800
      TWODEL=CMPLX(2.0/DEL,0.0)                                          00011810
      JJM=JJ-M                                                           00011820
      FM=ONE                                                             00011830
      PF1=ZERO                                                           00011840
   30 G2=G*G                                                             00011850
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                       00011860
   50 IF(M.EQ.1) GO TO 150                                               00011870
C--INITIALIZE PARTIAL INDEX J=M-1 (NUM. INDEX)                           00011880
```

```
      J=M-1                                                              00011890
C--LOOP ON J INDEX                                                       00011900
   70 V1=CSQRT(CMPLX(C2,2.0*K(J)))                                       00011910
      EVD=CEXP(-V1*D(J))                                                 00011920
   90 EVD1=ONE+EVD                                                       00011930
      E1=(ONE-EVD)/EVD1                                                  00011940
      E11=ONE+E1                                                         00011950
      T=VM*FM                                                            00011960
      DEN=V1+T*E1                                                        00011970
      F1=(T+V1*E1)/DEN                                                   00011980
      IF(JJ.LE.M) GO TO 100                                              00011990
C--RECUR FOR PF1 W/R DIST                                                00012000
      EMD1=ZERO                                                          00012010
      IF(JJM.EQ.J) EMD1=(TWODEL*V1*EVD*E11)/EVD1                         00012020
      PF1=((VM*PF1+V1*EMD1)-F1*VM*(FM*EMD1+E1*PF1))/DEN                  00012030
      GO TO 140                                                          00012040
C--RECUR FOR PF1 W/R SIGMA                                               00012050
  100 VMS=ZERO                                                           00012060
      VMS1=ZERO                                                          00012070
      EMS1=ZERO                                                          00012080
      IF(JJ.EQ.1) GO TO 110                                              00012090
      IF(J+1.EQ.JJ) VMS=CI/(SIG1*VM)                                     00012100
      IF(J.EQ.JJ) VMS1=CI/(SIG1*V1)                                      00012110
      GO TO 120                                                          00012120
  110 IF(M.GT.1) VMS=-CI*K(J+1)/(SIG1*VM)                                00012130
      IF(J.GT.1) VMS1=-CI*K(J)/(SIG1*V1)                                 00012140
  120 IF(JJ.NE.J) GO TO 130                                              00012150
      IF(J.EQ.1) EMS1=(EVD*V1*D(1)*E11)/(2.0*SIG1*EVD1)                  00012160
      IF(J.GT.1) EMS1=(D(J)*EVD*VMS1*E11)/EVD1                           00012170
  130 PF1=((FM*VMS+VM*PF1+V1*EMS1+E1*VMS1)-F1*                           00012180
     1(VMS1+VM*(FM*EMS1+E1*PF1)+FM*E1*VMS))/DEN                          00012190
  140 IF(J.EQ.1) GO TO 180                                               00012200
      J=J-1                                                              00012210
      VM=V1                                                              00012220
      FM=F1                                                              00012230
      GO TO 70                                                           00012240
C--SPECIAL CASE M=1 (HOMOGENEOUS EARTH)                                  00012250
  150 F1=FM                                                              00012260
      V1=VM                                                              00012270
      J=1                                                                00012280
      EVD=ZERO                                                           00012290
      GO TO 90                                                           00012300
  180 RETURN                                                             00012310
      END                                                                00012320

      SUBROUTINE KELVIN(X,M,B)                                           00012330
C--COMPUTES M(.LE.8) KELVIN FUNCTIONS (ORDERS 0,1) CONSECUTIVELY STORED  00012340
C  IN ARRAY B(M) WHERE:                                                  00012350
C                                                                        00012360
C      X      = DP-ARGUMENT .GT. 0.0D0 (ASYMPTOTIC FORM USED IF X.GE.8.0) 00012370
C      M      = NUMBER OF B'S TO COMPUTE AS DEFINED BELOW (1.GE.M.LE.8)   00012380
C      B(M)   = COMPUTED DP-FUNCTIONS WHERE B IS DEFINED:                 00012390
```

```
C              B(1) = BER(X)  --  ORDER 0                                    00012400
C              B(2) = BEI(X)  --  ORDER 0                                    00012410
C              B(3) = KER(X)  --  ORDER 0                                    00012420
C              B(4) = KEI(X)  --  ORDER 0                                    00012430
C              B(5) = BER1(X)  --  ORDER 1                                   00012440
C              B(6) = BEI1(X)  --  ORDER 1                                   00012450
C              B(7) = KER1(X)  --  ORDER 1                                   00012460
C              B(8) = KEI1(X)  --  ORDER 1                                   00012470
C ** ACCURACY GOOD TO AT LEAST 14 FIGURES FOR ALL X **                       00012480
C NOTE: THIS METHOD OF GENERATING MULTIPLE KELVIN FUNCTIONS WAS CHOSEN       00012490
C       TO REDUCE TOTAL CPU-TIME SINCE MOST APPLICATIONS REQUIRE             00012500
C       MULTIPLE FUNCTION USE AND IS THEREFORE ACCOMPLISHED BY ONE CALL.     00012510
C E.G: TO OBTAIN BER(X),BEI(X),KER(X), AND KEI(X):  CALL KELVIN(X,4,B)       00012520
C  IF X OR M OUT OF RANGE, ROUTINE EXITS WITHOUT ACTION.                     00012530
C                                                                            00012540
      IMPLICIT REAL*8 (A-H,O-Z)                                              00012550
      REAL*8 B(8),CN(8),SN(8)                                                00012560
      DATA CN          /.7071067811865475D0,0.D0,-.7071067811865475D0,       00012570
     * -1.D0,-.7071067811865475D0,0.D0,.7071067811865475D0,1.D0/,            00012580
     * SN    /.7071067811865475D0,1.D0,.7071067811865475D0,0.D0,             00012590
     * -.7071067811865475D0,-1.D0,-.7071067811865475D0,0.D0/                 00012600
      DATA PI4/.7853981633974483D0/,R22/.7071067811865475D0/,                00012610
     * E/0.5D-14/,                                                           00012620
     * PI1/.3183098861837907D0/                                              00012630
      IF(M.LT.1.OR.M.GT.8.OR.X.LE.0.0D0) GO TO 9                             00012640
      IF(X.GE.8.0D0) GO TO 8                                                 00012650
C--SERIES METHODS (X.GT.0.0.AND.X.LT.8.0D0)                                  00012660
      X2=0.5D0*X                                                             00012670
      X4=X2**4                                                               00012680
      T1=-0.25D0*X4                                                          00012690
      S1=T1                                                                  00012700
      T2=0.0D0                                                               00012710
      T3=0.0D0                                                               00012720
      T4=0.0D0                                                               00012730
      T15=0.0D0                                                              00012740
      T26=0.0D0                                                              00012750
      T75=0.0D0                                                              00012760
      T86=0.0D0                                                              00012770
      IF(M.EQ.1) GO TO 100                                                   00012780
      T2=X2**2                                                               00012790
      S2=T2                                                                  00012800
      IF(M.EQ.2) GO TO 100                                                   00012810
      T5=1.5D0                                                               00012820
      S5=T1*T5                                                               00012830
      IF(M.EQ.3) GO TO 100                                                   00012840
      T6=1.0D0                                                               00012850
      S6=T2                                                                  00012860
      IF(M.EQ.4) GO TO 100                                                   00012870
      T3=-0.5D0*X2**3                                                        00012880
      S3=T3                                                                  00012890
      T4=X2                                                                  00012900
      S4=T4                                                                  00012910
```

```
      IF(M.LE.6) GO TO 100                                              00012920
      T7=-0.25D0*X2**3                                                  00012930
      S7=2.0D0*T7*T5                                                    00012940
      T8=X2                                                             00012950
      S8=T8                                                             00012960
  100 TK=2.0D0                                                          00012970
  101 TK2=TK+TK                                                         00012980
      TK21=TK2-1.0D0                                                    00012990
      TK22=TK2-2.0D0                                                    00013000
      RK2=1.0D0/TK2                                                     00013010
      RK21=1.0D0/TK21                                                   00013020
      RK22=1.0D0/TK22                                                   00013030
      R1=-X4*(RK21*RK2)**2                                              00013040
      T1=T1*R1                                                          00013050
      S1=S1+T1                                                          00013060
      IF(M.EQ.1) GO TO 200                                              00013070
      R2=-X4*(RK22*RK21)**2                                             00013080
      T2=T2*R2                                                          00013090
      S2=S2+T2                                                          00013100
      IF(M.EQ.2) GO TO 200                                              00013110
      T5=T5+RK21+RK2                                                    00013120
      T15=T1*T5                                                         00013130
      S5=S5+T15                                                         00013140
      IF(M.EQ.3) GO TO 200                                              00013150
      T6=T6+RK22+RK21                                                   00013160
      T26=T2*T6                                                         00013170
      S6=S6+T26                                                         00013180
      IF(M.EQ.4) GO TO 200                                              00013190
      T3=T3*(-X4*(RK22*RK21**2*RK2))                                    00013200
      S3=S3+T3                                                          00013210
      T4=T4*(-X4*RK22**2*RK21/(TK2-3.0D0))                              00013220
      S4=S4+T4                                                          00013230
      IF(M.LE.6) GO TO 200                                              00013240
      T7=T7*R1                                                          00013250
      T75=TK2*T7*T5                                                     00013260
      S7=S7+T75                                                         00013270
      T8=T8*R2                                                          00013280
      T86=TK21*T8*T6                                                    00013290
      S8=S8+T86                                                         00013300
  200 TK=TK+1.0D0                                                       00013310
      IF(DABS(T1).GT.E.OR.DABS(T2).GT.E.OR.DABS(T15).GT.E.OR.           00013320
     * DABS(T26).GT.E.OR.DABS(T3).GT.E.OR.DABS(T4).GT.E.OR.             00013330
     * DABS(T75).GT.E.OR.DABS(T86).GT.E) GO TO 101                      00013340
      B(1)=1.0D0+S1                                                     00013350
      IF(M.EQ.1) GO TO 9                                                00013360
      B(2)=S2                                                           00013370
      IF(M.EQ.2) GO TO 9                                                00013380
      C=0.11593151565841240-DLOG(X)                                     00013390
      B(3)=C*B(1)+PI4*B(2)+S5                                           00013400
      IF(M.EQ.3) GO TO 9                                                00013410
      B(4)=C*B(2)-PI4*B(1)+S6                                           00013420
      IF(M.EQ.4) GO TO 9                                                00013430
```

```
      B(5)=R22*(S3-S4)                                                  00013440
      IF(M.EQ.5) GO TO 9                                                00013450
      B(6)=R22*(S3+S4)                                                  00013460
      IF(M.EQ.6) GO TO 9                                                00013470
      S7=C*S3-B(1)/X+PI4*S4+S7                                          00013480
      S8=C*S4-B(2)/X-PI4*S3+S8                                          00013490
      B(7)=R22*(S7-S8)                                                  00013500
      IF(M.EQ.7) GO TO 9                                                00013510
      B(8)=R22*(S7+S8)                                                  00013520
    9 RETURN                                                            00013530
C--GENERAL ASYMPTOTIC FORM FOR NU=0,1:                                  00013540
    8 NU=0                                                              00013550
      X2=R22*X                                                          00013560
      X8=8.0D0*X                                                        00013570
      SX=DSQRT(X)                                                       00013580
      EX2=DEXP_(-X2)                                                    00013590
      C1=1.253314137315500D0*EX2/SX                                     00013600
      C2=1.0D0/(2.506628274631001D0*SX*EX2+1.0D-38)                     00013610
      MAXK=30                                                           00013620
      IF(X.LT.15.0D0) MAXK=X+X                                          00013630
    1 XNU=NU                                                            00013640
      XMU=4.0D0*XNU                                                     00013650
      ALP=X2+PI4*(XNU+XNU-0.5D0)                                        00013660
      BETA=ALP+PI4                                                      00013670
      CB=DCOS(BETA)                                                     00013680
      CA=DCOS(ALP)                                                      00013690
      SB=DSIN(BETA)                                                     00013700
      SA=DSIN(ALP)                                                      00013710
      N4=4*NU                                                           00013720
      FM=0.0D0                                                          00013730
      FP=0.0D0                                                          00013740
      GM=0.0D0                                                          00013750
      GP=0.0D0                                                          00013760
      TM=1.0D0                                                          00013770
      TP=1.0D0                                                          00013780
      K=1                                                               00013790
    2 TK=K                                                              00013800
      T=(XMU-(TK+TK-1.0D0)**2)/(TK*X8)                                  00013810
      TPL=DABS(TP)                                                      00013820
      TP=-TP*T                                                          00013830
      IF(DABS(TP).GT.TPL) GO TO 21                                      00013840
      TM=TM*T                                                           00013850
      N=MOD(K,8)                                                        00013860
      IF(N.EQ.0) N=8                                                    00013870
      T1=TP*CN(N)                                                       00013880
      FP=FP+T1                                                          00013890
      T2=TM*CN(N)                                                       00013900
      FM=FM+T2                                                          00013910
      T3=TP*SN(N)                                                       00013920
      GP=GP+T3                                                          00013930
      T4=TM*SN(N)                                                       00013940
      GM=GM+T4                                                          00013950
```

```
      K=K+1                                                          00013960
      IF(K.GT.MAXK) GO TO 3                                          00013970
      GO TO 2                                                        00013980
   21 FP=FP-T1                                                       00013990
      FM=FM-T2                                                       00014000
      GP=GP-T3                                                       00014010
      GM=GM-T4                                                       00014020
    3 FP=FP+1.0D0                                                    00014030
      FM=FM+1.0D0                                                    00014040
      B(N4+4)=C1*(-FM*SB-GM*CB)                                      00014050
      B(N4+3)=C1*(FM*CB-GM*SB)                                       00014060
      B(N4+2)=C2*(FP*SA-GP*CA)+PI1*B(N4+3)                           00014070
      B(N4+1)=C2*(FP*CA+GP*SA)-PI1*B(N4+4)                           00014080
      IF(NU.EQ.1.OR.M.LE.4) GO TO 9                                  00014090
      NU=1                                                           00014100
      GO TO 1                                                        00014110
      END                                                            00014120

      COMPLEX FUNCTION FVP(X)                                        00014130
C--RESISTIVITY KERNEL USED IN INTEGRAL OF PARTIAL RHOA W/R B(JJ).     00014140
C  JJ=1,2*MM-1 GIVEN IN COMMON/RESIST/.                              00014150
C  (FVP BY RECURRENCE METHOD).                                       00014160
C                                                                    00014170
      REAL RHO(10),H(9),K1                                           00014180
      COMMON/RESIST/RHO,H,EPS,R,R2,ALOGR,MM,M1,M21,JJ                00014190
      X2=-2.0*X                                                      00014200
      JJMM=JJ-MM                                                     00014210
      VM=1.0                                                         00014220
      PV1=0.0                                                        00014230
      IF(MM.EQ.1) GO TO 40                                           00014240
C--INITIALIZE PARTIAL INDEX J1=MM-1 (NUM. INDEX)                     00014250
      J=MM                                                           00014260
C--LOOP ON J1 INDEX                                                  00014270
   10 J1=J-1                                                         00014280
      E=X2*H(J1)                                                     00014290
      E1=0.0                                                         00014300
C--HONEYWELL MULTICS TEST  $$$$$$$$$$$$$$$$$                         00014310
      IF(E.GT.-88.028) E1=EXP(E)                                     00014320
      DENK1=1.0/(RHO(J1)+RHO(J)*VM)                                  00014330
      K1=DENK1*(RHO(J1)-RHO(J)*VM)                                   00014340
      DENV1=1.0/(1.0+K1*E1)                                          00014350
      V1=DENV1*(1.0-K1*E1)                                           00014360
      IF(JJ.LE.MM) GO TO 20                                          00014370
C--RECUR FOR PARTIAL W/R H(JJ)                                       00014380
      PEH=0.0                                                        00014390
      IF(JJMM.EQ.J1) PEH=X2*E1                                       00014400
      PKH=-DENK1*RHO(J)*PV1*(1.0+K1)                                 00014410
      PV1=-DENV1*(K1*PEH+E1*PKH)*(1.0+V1)                            00014420
      GO TO 30                                                       00014430
C--RECUR FOR PARTIAL W/R RHO(JJ)                                     00014440
   20 PR1=0.0                                                        00014450
      IF(JJ.EQ.J1) PR1=1.0                                           00014460
```

```
      PRM=0.0                                                          00014470
      IF(JJ.EQ.J) PRM=1.0                                              00014480
      PKR=DENK1*(PR1*(1.0-K1)-(1.0+K1)*(RHO(J)*PV1+VM*PRM))            00014490
      PV1=-DENV1*E1*PKR*(1.0+V1)                                       00014500
   30 IF(J.LE.2) GO TO 40                                              00014510
      VM=V1                                                            00014520
      J=J1                                                             00014530
      GO TO 10                                                         00014540
   40 FVP=CMPLX(X*PV1,0.0)                                             00014550
      RETURN                                                           00014560
      END                                                              00014570

      COMPLEX FUNCTION KERN(X)                                         00014580
C--KERNEL FUNCTION USED IN FCODE INTEGRAL                              00014590
C  FOR SCHLUMBERGER APPARENT RESISTIVITY                               00014600
C                                                                      00014610
      REAL RHO(10),H(9)                                                00014620
      COMMON/RESIST/RHO,H,EPS,R,R2,ALOGR,MM,M1,M21,JJ                  00014630
      X2=-2.0*X                                                        00014640
      V=1.0                                                            00014650
      IF(MM.LE.1) GO TO 30                                             00014660
      I=MM                                                             00014670
   10 I1=I-1                                                           00014680
      T=V/RHO(I1)                                                      00014690
      TR=T*RHO(I)                                                      00014700
      E=X2*H(I1)                                                       00014710
C--HONEYWELL MULTICS TEST  $$$$$$$$$$$$$$$$$                           00014720
      IF(E.LT.-88.028) GO TO 40                                        00014730
      T=((1.0-TR)/(1.0+TR))*EXP(E)                                     00014740
      V=(1.0-T)/(1.0+T)                                                00014750
   20 IF(I.LE.2) GO TO 30                                              00014760
      I=I-1                                                            00014770
      GO TO 10                                                         00014780
   30 KERN=CMPLX(X*(V-1.0),0.0)                                        00014790
      RETURN                                                           00014800
   40 V=1.0                                                            00014810
      GO TO 20                                                         00014820
      END                                                              00014830

      COMPLEX FUNCTION FG2(G)                                          00014840
C-- F(G)*G*G KERNEL USED BY PROGRAM 'EMLOOPS' FOR THE                  00014850
C   GROUND CASE (A=0). NOTE:  FG2 IS USED IN T0,T1 INTEGRALS           00014860
C   VIA SUBR 'ZHANKS'.                                                 00014870
C                                                                      00014880
      COMPLEX V1,F1,C,ONE,TWO                                          00014890
      DATA ONE,TWO/(1.0,0.0),(2.0,0.0)/                                00014900
      C=CMPLX(G,0.)                                                    00014910
      CALL RECUR1(G,V1,F1)                                             00014920
C//      FG2=(TWO*V1*C*C*C*(F1-ONE))/((C+V1)*(C+V1*F1))                00014930
C  ON MULTICS, REWRITE AS:                                             00014940
      FG2=TWO*V1*(C/(C+V1))*(C/(C+V1*F1))*(F1-ONE)*C                   00014950
      RETURN                                                           00014960
```

```
      END                                                               00014970

      COMPLEX FUNCTION FG(G)                                            00014980
C-- F(G)*G KERNEL USED BY PROGRAM 'EMLOOPS' FOR THE                    00014990
C   GROUND CASE (A=0).  FG IS USED IN T2 INTEGRAL VIA SUBR 'ZHANKS'.    00015000
C                                                                       00015010
      COMPLEX FG2                                                       00015020
      FG=FG2(G)/G                                                       00015030
      RETURN                                                            00015040
      END                                                               00015050

      COMPLEX FUNCTION FG3(G)                                           00015060
C-- F(G)*G**3 KERNEL USED IN PGM 'MARQLOOPS'                           00015070
C                                                                       00015080
      COMPLEX FG2                                                       00015090
      FG3=G*FG2(G)                                                      00015100
      RETURN                                                            00015110
      END                                                               00015120

      COMPLEX FUNCTION RG2(G)                                           00015130
C-- R(G)*G*G*EXP(-G*A) KERNEL USED BY PROGRAM 'EMLOOPS' FOR THE        00015140
C   AIRBORNE CASE (A>0). NOTE:  RG2 IS USED IN T0,T1 INTEGRALS          00015150
C   VIA SUBR 'ZHANKS'.                                                  00015160
C                                                                       00015170
      COMPLEX V1,F1,C,V1F1                                              00015180
      COMMON/AIR/A                                                      00015190
      C=G                                                               00015200
      CALL RECUR1(G,V1,F1)                                              00015210
      V1F1=V1*F1                                                        00015220
      RG2=C*C*(V1F1-C)*CEXP(-C*CMPLX(A,0.))/(V1F1+C)                    00015230
      RETURN                                                            00015240
      END                                                               00015250

      COMPLEX FUNCTION RG(G)                                            00015260
C-- R(G)*G*EXP(-G*A) KERNEL USED BY PROGRAM 'EMLOOPS' FOR THE           00015270
C   AIRBORNE CASE (A>0).  RG IS USED IN T2 INTEGRAL VIA SUBR 'ZHANKS'.  00015280
C                                                                       00015290
      COMPLEX RG2                                                       00015300
      RG=RG2(G)/G                                                       00015310
      RETURN                                                            00015320
      END                                                               00015330

      COMPLEX FUNCTION RG3(G)                                           00015340
C-- R(G)*G**3*EXP(-G*A) KERNEL USED BY PGM 'MARQLOOPS'                 00015350
C                                                                       00015360
      COMPLEX RG2                                                       00015370
      RG3=G*RG2(G)                                                      00015380
      RETURN                                                            00015390
      END                                                               00015400

      SUBROUTINE IKS2(B8,I0K0,I1K1,IKDIF)                               00015410
C--COMPUTE MODIFIED BESSEL FUNCTION (I & K) PRODUCT COMBINATIONS FOR   00015420
```

```
C  PARAMETERS                                                          00015430
C      B8     = DOUBLE PRECISION ARGUMENT (=B/DSQRT(2.D0) HERE)       00015440
C      I0K0   = I0*K0 COMPLEX RESULT                                  00015450
C      I1K1   = I1*K1 COMPLEX RESULT                                  00015460
C      IKDIF  = 4*I1*K1-(B8*DSQRT(I))*(I0*K1-I1*K0) COMPLEX RESULT DONE IN00015470
C               DOUBLE PRECISION BEFORE USING CMPLX.                  00015480
C--SUBROUTINE KELVIN CALLED                                           00015490
C                                                                     00015500
      DOUBLE PRECISION B8,BB(8),BETA,P1,P2,Q1,Q2,R1,R2                00015510
      COMPLEX I0K0,I1K1,IKDIF                                         00015520
      COMPLEX CAMBDA,DENOM,DENOM1,TERM0,TERM1,TERM11,TERM00           00015530
      COMPLEX S00,S11,SI0,SI1,SK0,SK1,ONE                             00015540
      DATA ONE/(1.0,0.0)/                                             00015550
      IF(B8.GT.20D0) GO TO 10                                         00015560
      CALL KELVIN(B8,8,BB)                                            00015570
      P1=-BB(6)*BB(8)+BB(5)*BB(7)                                     00015580
      P2= BB(5)*BB(8)+BB(6)*BB(7)                                     00015590
      I1K1=CMPLX(SNGL(P1),SNGL(P2))                                   00015600
      Q1=-BB(1)*BB(8)-BB(2)*BB(7)                                     00015610
      Q2=BB(1)*BB(7)-BB(2)*BB(8)                                      00015620
      R1=BB(1)*BB(3)-BB(2)*BB(4)                                      00015630
      R2=BB(2)*BB(3)+BB(1)*BB(4)                                      00015640
      I0K0=CMPLX(SNGL(R1),SNGL(R2))                                   00015650
      R1=BB(6)*BB(3)+BB(5)*BB(4)                                      00015660
      R2=BB(6)*BB(4)-BB(5)*BB(3)                                      00015670
      R1=Q1-R1                                                        00015680
      R2=Q2-R2                                                        00015690
      BETA=.7071067811865475D0*B8                                     00015700
      Q1=4.0D0*P1-BETA*(R1-R2)                                        00015710
      Q2=4.0D0*P2-BETA*(R1+R2)                                        00015720
      IKDIF=CMPLX(SNGL(Q1),SNGL(Q2))                                  00015730
      RETURN                                                          00015740
   10 B=SNGL(B8/0.7071067811865475D0)                                 00015750
      TOL=1.E-6                                                       00015760
C--FOR LARGE ARGUMENTS, USE ABRAMOWITZ AND STEGUN                     00015770
C         ASYMPTOTIC FORMULAS FOR LARGE ARGUMENTS                     00015780
C         9.7.1 THROUGH 9.7.5, P. 377-378.                            00015790
      CAMBDA=B*CMPLX(1.0,1.0)/2.                                      00015800
      IKDIF=CMPLX(100.,0.)                                            00015810
      ISIGN=1                                                         00015820
      DENOM=8.*CAMBDA                                                 00015830
      DENOM1=(2.*CAMBDA)**2                                           00015840
      NODD=1                                                          00015850
      TERM0=ONE                                                       00015860
      TERM1=ONE                                                       00015870
      TERM11=ONE                                                      00015880
      TERM00=ONE                                                      00015890
      S00=ONE                                                         00015900
      S11=ONE                                                         00015910
      SI0=ONE                                                         00015920
      SI1=ONE                                                         00015930
      SK0=ONE                                                         00015940
```

```
      SK1=ONE                                                          00015950
    1 NODD2=NODD*NODD                                                  00015960
      OIKDIF=CABS(IKDIF)                                               00015970
      TERM1=TERM1*CMPLX(4.-NODD2,0.)/DENOM                             00015980
      TERM0=TERM0*CMPLX(-FLOAT(NODD2),0.)/DENOM                        00015990
      TERM11=TERM11*CMPLX(NODD*(4.-NODD2)/(NODD+1.),0.)/DENOM1         00016000
      TERM00=TERM00*CMPLX(-FLOAT(NODD*NODD2)/(NODD+1.),0.)/DENOM1      00016010
      ISIGN=-ISIGN                                                     00016020
      S11=S11+ISIGN*TERM11                                             00016030
      S00=S00+ISIGN*TERM00                                             00016040
      SI0=SI0+ISIGN*TERM0                                              00016050
      SI1=SI1+ISIGN*TERM1                                              00016060
      SK0=SK0+TERM0                                                    00016070
      SK1=SK1+TERM1                                                    00016080
      IKDIF=SI0*SK1-SK0*SI1                                            00016090
      NODD=NODD+2                                                      00016100
      IF(ABS(OIKDIF-CABS(IKDIF)).GT.TOL) GO TO 1                       00016110
      DENOM1=ONE/(CAMBDA*CMPLX(2.0,0.0))                               00016120
      I0K0=S00*DENOM1                                                  00016130
      I1K1=S11*DENOM1                                                  00016140
      IKDIF=CMPLX(4.,0.)*I1K1-IKDIF/CMPLX(2.,0.)                       00016150
      RETURN                                                           00016160
      END                                                              00016170

      COMPLEX FUNCTION PFBJG(G)                                        00016180
C-- PARTIAL OF (F W/R B(J),J>=1)*G.                                    00016190
C  J IS GIVEN IN COMMON/PART/J,ISEP ALONG WITH OTHER                   00016200
C  COMMON PARAMETERS.                                                 00016210
C                                                                     00016220
      COMPLEX V1,F1,C,T0,T1,T2,CB,CB2,CB3,CA,ONESG1,TWOSG1,            00016230
     & ZZ0,PF1,TWO,I0K0,I1K1,IKDIF                                     00016240
      COMMON/SHARE/FILL(4),XX,YY,YY2,RHO,RHO2,FILL2,BB,FILL3,          00016250
     & DEL,DEL2,IREST(3)                                               00016260
      COMMON/CTL/T0,T1,T2,CB,CB2,CB3,CA,ONESG1,TWOSG1,I0K0,I1K1,IKDIF, 00016270
     & ZZ0,AMP,FREQ,SIG1,H,EPS,IOB,M1,M21,ILOOPS,IMM                   00016280
      COMMON/PART/J,ISEP                                               00016290
      DATA TWO/(2.0,0.0)/                                              00016300
      CALL RECURF(G,DEL,SIG1,V1,F1,PF1,J)                              00016310
      C=G                                                              00016320
      PFBJG=TWO*C*V1*PF1*C/(C+V1*F1)**2                                00016330
      RETURN                                                           00016340
      END                                                              00016350

      COMPLEX FUNCTION PFBJG2(G)                                       00016360
C-- PARTIAL OF (F W/R B(J),J>=1)*G**2.                                 00016370
C                                                                     00016380
      COMPLEX PFBJG                                                    00016390
      PFBJG2=G*PFBJG(G)                                                00016400
      RETURN                                                           00016410
      END                                                              00016420
```

```
      COMPLEX FUNCTION PRBJG(G)                                          00016430
C-- PARTIAL OF (R W/R B(J),J>=1)*G.                                       00016440
C                                                                        00016450
      COMPLEX PFBJG                                                      00016460
      COMMON/AIR/A                                                       00016470
      PRBJG=PFBJG(G)*CEXP(CMPLX(-G*A,0.0))                               00016480
      RETURN                                                             00016490
      END                                                                00016500

      COMPLEX FUNCTION PRBJG2(G)                                         00016510
C-- PARTIAL OF (R W/R B(J),J>=1)*G**2.                                    00016520
C                                                                        00016530
      COMPLEX PRBJG                                                      00016540
      PRBJG2=G*PRBJG(G)                                                  00016550
      RETURN                                                             00016560
      END                                                                00016570

      SUBROUTINE MODIFY(N)                                               00016580
C--UTILITY TO MODIFY COMMON/SAVE/ AS FOLLOWS:                            00016590
C  N >0 TO REPLACE FSAVE(I)=FSAVE(I)*(GSAVE(I)**N), I=1,NSAVE.           00016600
C  N <0 TO REPLACE FSAVE(I)=FSAVE(I)/(GSAVE(I)**IABS(N)), I=1,NSAVE.     00016610
C--THIS MAY BE USED IN CONJUNCTION WITH SUBPROGRAM 'ZHANKS' TO           00016620
C  MODIFY SAVED KERNELS WHEN USING NEW=0 (SEE ZHANKS).                   00016630
C                                                                        00016640
      COMPLEX FSAVE                                                      00016650
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                            00016660
      IF(N) 5,9,1                                                        00016670
1     IF(N.GT.1) GO TO 3                                                 00016680
      DO 2 I=1,NSAVE                                                     00016690
2     FSAVE(I)=FSAVE(I)*CMPLX(GSAVE(I),0.0)                              00016700
      GO TO 9                                                            00016710
3     DO 4 I=1,NSAVE                                                     00016720
4     FSAVE(I)=FSAVE(I)*CMPLX(GSAVE(I)**N,0.0)                           00016730
      GO TO 9                                                            00016740
5     IF(N.LT.-1) GO TO 3                                                00016750
      DO 6 I=1,NSAVE                                                     00016760
6     FSAVE(I)=FSAVE(I)/CMPLX(GSAVE(I),0.0)                              00016770
9     RETURN                                                             00016780
      END                                                                00016790

      SUBROUTINE SWAP(ICODE)                                             00016800
C--UTILITY TO SWAP COMMON/SAVE/ AS FOLLOWS:                              00016810
C  ICODE =1 TO SWAP COMMON/SAVE/ TO INTERNAL TEMP STORAGE.               00016820
C        =-1 TO RESWAP INTERNAL TEMP STORAGE TO COMMON/SAVE/.            00016830
C                                                                        00016840
C--THIS MAY BE USED IN CONJUNCTION WITH SUBPROGRAM 'ZHANKS' TO USE       00016850
C  DIFFERENT CLASSES OF INTEGRALS.  ALSO, SEE THE UTILITY                00016860
C  SUBROUTINE 'MODIFY'.                                                  00016870
C                                                                        00016880
      COMPLEX FSAVE,FSWAP                                                00016890
      DIMENSION FSWAP(283),GSWAP(283)                                    00016900
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                            00016910
```

```
      IF(ICODE) 3,1,1                                                   00016920
1     DO 2 I=1,NSAVE                                                    00016930
      FSWAP(I)=FSAVE(I)                                                 00016940
2     GSWAP(I)=GSAVE(I)                                                 00016950
      NSWAP=NSAVE                                                       00016960
      RETURN                                                            00016970
3     DO 4 I=1,NSWAP                                                    00016980
      FSAVE(I)=FSWAP(I)                                                 00016990
4     GSAVE(I)=GSWAP(I)                                                 00017000
      NSAVE=NSWAP                                                       00017010
      RETURN                                                            00017020
      END                                                               00017030

      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                       00017040
C=====================================================================00017050
C  COMPLEX HANKEL TRANSFORMS OF ORDER 0 OR 1 FOR RELATED (SAVED) KERNELS00017060
C  AND FIXED TRANSFORM ARGUMENT B.GT.0.                               00017070
C                                                                     00017080
C--REF: ANDERSON, W.L., 1979 (IN PRESS), GEOPHYSICS, V. , NO. , P. - . 00017090
C                                                                     00017100
C--SUBPROGRAM ZHANKS EVALUATES THE INTEGRAL FROM 0 TO INFINITY OF     00017110
C  FUN(G)*JN(G*B)*DG, DEFINED AS THE COMPLEX HANKEL TRANSFORM OF      00017120
C  ORDER N (=0 OR 1) AND TRANSFORM ARGUMENT B.GT.0.  THE METHOD IS BY 00017130
C  ADAPTIVE DIGITAL FILTERING OF THE COMPLEX KERNEL FUNCTION FUN,     00017140
C  USING DIRECT AND/OR PREVIOUSLY SAVED KERNEL FUNCTION VALUES.       00017150
C                                                                     00017160
C--PARAMETERS (ALL INPUT, EXCEPT NF)                                  00017170
C                                                                     00017180
C     N      = ORDER (=0 OR 1) OF THE HANKEL TRANSFORM TO BE EVALUATED. 00017190
C     B      = REAL TRANSFORM ARGUMENT B.GT.0.0 OF THE HANKEL TRANSFORM. 00017200
C              IF NEW=0, B IS ASSUMED EQUAL TO THE LAST B USED WHEN NEW=100017210
C              (SEE PARAMETER NEW AND SUBPROGRAM USAGE BELOW).        00017220
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)  00017230
C              OF A REAL ARGUMENT G.GT.0. THIS REFERENCE MUST BE SUPPLIED00017240
C              EVEN WHEN NEW=0, SINCE THE ADAPTIVE CONVOLUTION        G0017250
C              MAY NEED SOME DIRECT FUNCTION CALLS (E.G. IF TOL REDUCED).00017260
C              IF PARAMETERS OTHER THAN G ARE REQUIRED IN FUN, USE COMMON00017270
C              IN THE CALLING PROGRAM AND IN SUBPROGRAM FUN.  BOTH    00017280
C              REAL AND IMAGINARY PARTS OF THE COMPLEX FUNCTION FUN(G) 00017290
C              MUST BE CONTINUOUS BOUNDED FUNCTIONS FOR G.GT.0.0. FOR A 00017300
C              REAL FUNCTION F1(G), FUN=CMPLX(F1(G),0.0) MAY BE USED. 00017310
C              TWO INDEPENDENT REAL-FUNCTIONS F1(G),F2(G) MAY BE      00017320
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)). 00017330
C     TOL    = REQUESTED REAL TRUNCATION TOLERANCE ACCEPTED AT THE FILTER00017340
C              TAILS FOR ADAPTIVE FILTERING.  A TRUNCATION CRITERION IS 00017350
C              DEFINED DURING CONVOLUTION IN A FIXED ABSCISSA RANGE AS 00017360
C              THE MAX. ABSOLUTE CONVOLVED PRODUCT TIMES TOL.  TYPICALLY,00017370
C              TOL.LE.0.00001 WOULD GIVE ABOUT .01 PER CENT ACCURACY  00017380
C              FOR WELL-BEHAVED KERNELS AND MODERATE VALUES OF B.  FOR 00017390
C              VERY LARGE OR SMALL B, A VERY SMALL TOL SHOULD BE USED. 00017400
C              IN GENERAL, DECREASING THE TOLERANCE WOULD PRODUCE HIGHER 00017410
C              ACCURACY IN THE CONVOLUTION SINCE MORE FILTER WEIGHTS ARE 00017420
```

```
C               USED (UNLESS EXPONENT UNDERFLOWS OCCUR IN THE KERNEL      00017430
C               EVALUATION -- SEE NOTE (1) BELOW).                        00017440
C               FOR MAXIMUM ACCURACY POSSIBLE, TOL=0.0 MAY BE USED.       00017450
C     NF     = TOTAL NUMBER OF DIRECT FUN CALLS USED DURING CONVOLUTION   00017460
C               FOR ANY VALUE OF NEW (NF IS AN OUTPUT PARAMETER).         00017470
C               NF IS IN THE RANGE 21.LE.NF.LE.283 WHEN NEW=1.  USUALLY,  00017480
C               NF IS MUCH LESS THAN 283 (OR 0) WHEN NEW=0.               00017490
C     NEW   =1 IS REQUIRED FOR THE VERY FIRST CALL TO ZHANKS, OR IF       00017500
C               FORCING DIRECT FUNCTION FUN(G) CALLS, E.G., IF USING      00017510
C               ZHANKS FOR UNRELATED KERNELS.                             00017520
C               NEW=1 INITIALIZES COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE 00017530
C               FOR NSAVE COMPLEX KERNEL VALUES IN FSAVE AND CORRESPONDING00017540
C               REAL ARGUMENTS IN GSAVE FOR THE GIVEN PARAMETER B.        00017550
C     NEW   =0 TO USE RELATED KERNELS (MODIFIED BY USER) CURRENTLY STORED00017560
C               IN COMMON/SAVE/. FUN IS CALLED ONLY IF REQUIRED           00017570
C               DURING THE CONVOLUTION.  ADDITIONAL FUNCTION VALUES WHEN  00017580
C               NEEDED ARE AUTOMATICALLY ADDED TO THE COMMON/SAVE/ BLOCK. 00017590
C                                                                         00017600
C     ******* NOTE THAT IT IS THE USERS RESPONSIBILITY TO MODIFY THE     00017610
C               COMMON FSAVE() VALUES FOR NEW=0 CALLS, EXTERNALLY IN      00017620
C               THE USERS CALLING PROGRAM (SEE SUBPROGRAM USAGE BELOW).   00017630
C                                                                         00017640
C=========================================================================00017650
C--SUBPROGRAM USAGE-- ZHANKS IS CALLED AS FOLLOWS                         00017660
C     ...                                                                 00017670
C     COMPLEX Z1,Z2,ZHANKS,FSAVE                                          00017680
C     COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                             00017690
C     EXTERNAL ZF1,ZF2                                                    00017700
C     ...                                                                 00017710
C     Z1=ZHANKS(N1,B,ZF1,TOL,NF1,1)                                       00017720
C     DO 1 I=1,NSAVE                                                      00017730
C  C--MODIFY FSAVE IN COMMON/SAVE/ TO OBTAIN RELATED ZF2 FROM ZF1.        00017740
C  C--E.G. FSAVE(I)=GSAVE(I)*FSAVE(I) -- FOR RELATION ZF2(G)=G*ZF1(G)     00017750
C   1 CONTINUE                                                            00017760
C     Z2=ZHANKS(N2,B,ZF2,TOL,NF2,0)                                       00017770
C     ...                                                                 00017780
C     END                                                                 00017790
C     COMPLEX FUNCTION ZF1(G)                                             00017800
C     ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF1(G), G.GT.0.      00017810
C     END                                                                 00017820
C     COMPLEX FUNCTION ZF2(G)                                             00017830
C     ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF2(G), G.GT.0.      00017840
C     END                                                                 00017850
C=========================================================================00017860
C--NOTES                                                                  00017870
C       (1).  EXP-UNDERFLOW MAY OCCUR IN EXECUTING THIS SUBPROGRAM.       00017880
C             THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS    00017890
C             EXP-UNDERFLOW TO 0.0.                                       00017900
C       (2).  ANSI FORTRAN (AMERICAN STANDARD X3.9-1966) IS USED, EXCEPT  00017910
C             DATA STATEMENTS MAY NEED TO BE CHANGED FOR SOME COMPILERS.  00017920
C             TO CONVERT ZHANKS TO THE NEW AMERICAN STANDARD FORTRAN      00017930
C             (X3.9-1978), ADD THE FOLLOWING DECLARATION TO THIS ROUTINE  00017940
```

```
C              SAVE Y1,ISAVE                                                    00017950
C        (3).  THE FILTER ABSCISSA CORRESPONDING TO EACH FILTER WEIGHT          00017960
C              IS GENERATED IN DOUBLE-PRECISION (TO REDUCE ROUND-OFF),          00017970
C              BUT IS USED IN SINGLE-PRECISION IN FUNCTION FUN.                 00017980
C        (4).  NO CHECKS ARE MADE ON CALLING PARAMETERS (TO SAVE TIME),         00017990
C              HENCE UNPREDICTABLE RESULTS COULD OCCUR IF ZHANKS                00018000
C              IS CALLED INCORRECTLY (OR IF FUN OR COMMON IS IN ERROR).         00018010
C=====================================================================        00018020
C                                                                              00018030
      COMPLEX FUN,C,CMAX,FSAVE                                                 00018040
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                                  00018050
      DOUBLE PRECISION E,ER,Y1,Y                                               00018060
      DIMENSION T(2),TMAX(2)                                                   00018070
      DIMENSION WT0(283),WA0(76),WB0(76),WC0(76),WD0(55),                      00018080
     * WT1(283),WA1(76),WB1(76),WC1(76),WD1(55)                               00018090
      EQUIVALENCE (WT0(1),WA0(1)),(WT0(77),WB0(1)),(WT0(153),WC0(1)),          00018100
     * (WT0(229),WD0(1)),(WT1(1),WA1(1)),(WT1(77),WB1(1)),                     00018110
     * (WT1(153),WC1(1)),(WT1(229),WD1(1))                                     00018120
      EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                                      00018130
C-----E=DEXP(.2D0), ER=1.0D0/E                                                 00018140
      DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/               00018150
C--J0-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT0 ARRAY)                 00018160
      DATA WA0/                                                                00018170
     * 2.1969101E-11, 4.1201161E-09,-6.1322980E-09, 7.2479291E-09,             00018180
     *-7.9821627E-09, 8.5778983E-09,-9.1157294E-09, 9.6615250E-09,             00018190
     *-1.0207546E-08, 1.0796633E-08,-1.1393033E-08, 1.2049873E-08,             00018200
     *-1.2708789E-08, 1.3446466E-08,-1.4174300E-08, 1.5005577E-08,             00018210
     *-1.5807160E-08, 1.6747136E-08,-1.7625961E-08, 1.8693427E-08,             00018220
     *-1.9650840E-08, 2.0869789E-08,-2.1903555E-08, 2.3305308E-08,             00018230
     *-2.4407377E-08, 2.6033678E-08,-2.7186773E-08, 2.9094334E-08,             00018240
     *-3.0266804E-08, 3.2534013E-08,-3.3672072E-08, 3.6408936E-08,             00018250
     *-3.7425022E-08, 4.0787921E-08,-4.1543242E-08, 4.5756842E-08,             00018260
     *-4.6035233E-08, 5.1425075E-08,-5.0893896E-08, 5.7934897E-08,             00018270
     *-5.6086570E-08, 6.5475248E-08,-6.1539913E-08, 7.4301996E-08,             00018280
     *-6.7117043E-08, 8.4767837E-08,-7.2583120E-08, 9.7366568E-08,             00018290
     *-7.7553611E-08, 1.1279873E-07,-8.1416723E-08, 1.3206914E-07,             00018300
     *-8.3217217E-08, 1.5663185E-07,-8.1482581E-08, 1.8860593E-07,             00018310
     *-7.3963141E-08, 2.3109673E-07,-5.7243707E-08, 2.8867452E-07,             00018320
     *-2.6163525E-08, 3.6808773E-07, 2.7049871E-08, 4.7932617E-07,             00018330
     * 1.1407365E-07, 6.3720626E-07, 2.5241961E-07, 8.6373487E-07,             00018340
     * 4.6831433E-07, 1.1916346E-06, 8.0099716E-07, 1.6696015E-06,             00018350
     * 1.3091334E-06, 2.3701475E-06, 2.0803829E-06, 3.4012978E-06/             00018360
      DATA WB0/                                                                00018370
     * 3.2456774E-06, 4.9240402E-06, 5.0005198E-06, 7.1783540E-06,             00018380
     * 7.6367633E-06, 1.0522038E-05, 1.1590021E-05, 1.5488635E-05,             00018390
     * 1.7510398E-05, 2.2873836E-05, 2.6368006E-05, 3.3864387E-05,             00018400
     * 3.9610390E-05, 5.0230379E-05, 5.9397373E-05, 7.4612122E-05,             00018410
     * 8.8951409E-05, 1.1094809E-04, 1.3308026E-04, 1.6511335E-04,             00018420
     * 1.9895671E-04, 2.4587195E-04, 2.9728181E-04, 3.6629770E-04,             00018430
     * 4.4402013E-04, 5.4589361E-04, 6.6298832E-04, 8.1375348E-04,             00018440
     * 9.8971624E-04, 1.2132772E-03, 1.4772052E-03, 1.8092022E-03,             00018450
     * 2.2045122E-03, 2.6980811E-03, 3.2895354E-03, 4.0238764E-03,             00018460
```

```
     * 4.9080203E-03, 6.0010999E-03, 7.3216878E-03, 8.9489225E-03,      00018470
     * 1.0919448E-02, 1.3340696E-02, 1.6276399E-02, 1.9873311E-02,      00018480
     * 2.4233627E-02, 2.9555699E-02, 3.5990069E-02, 4.3791529E-02,      00018490
     * 5.3150319E-02, 6.4341372E-02, 7.7506720E-02, 9.2749987E-02,      00018500
     * 1.0980561E-01, 1.2791555E-01, 1.4525830E-01, 1.5820085E-01,      00018510
     * 1.6058576E-01, 1.4196085E-01, 8.9781222E-02,-1.0238278E-02,      00018520
     *-1.5083434E-01,-2.9059573E-01,-2.9105437E-01,-3.7973244E-02,      00018530
     * 3.8273717E-01, 2.2014118E-01,-4.7342635E-01, 1.9331133E-01,      00018540
     * 5.3839527E-02,-1.1909845E-01, 9.9317051E-02,-6.6152628E-02,      00018550
     * 4.0703241E-02,-2.4358316E-02, 1.4476533E-02,-8.6198067E-03/      00018560
      DATA WC0/                                                        00018570
     * 5.1597053E-03,-3.1074602E-03, 1.8822342E-03,-1.1456545E-03,      00018580
     * 7.0004347E-04,-4.2904226E-04, 2.6354444E-04,-1.6215439E-04,      00018590
     * 9.9891279E-05,-6.1589037E-05, 3.7996921E-05,-2.3452250E-05,      00018600
     * 1.4479572E-05,-8.9417427E-06, 5.5227518E-06,-3.4114252E-06,      00018610
     * 2.1074101E-06,-1.3019229E-06, 8.0433617E-07,-4.9693681E-07,      00018620
     * 3.0702417E-07,-1.8969219E-07, 1.1720069E-07,-7.2412496E-08,      00018630
     * 4.4740283E-08,-2.7643004E-08, 1.7079403E-08,-1.0552634E-08,      00018640
     * 6.5200311E-09,-4.0284597E-09, 2.4890232E-09,-1.5378695E-09,      00018650
     * 9.5019040E-10,-5.8708696E-10, 3.6273937E-10,-2.2412348E-10,      00018660
     * 1.3847792E-10,-8.5560821E-11, 5.2865474E-11,-3.2664392E-11,      00018670
     * 2.0182948E-11,-1.2470979E-11, 7.7057678E-12,-4.7611713E-12,      00018680
     * 2.9415274E-12,-1.8170081E-12, 1.1221034E-12,-6.9271067E-13,      00018690
     * 4.2739744E-13,-2.6344388E-13, 1.6197105E-13,-9.9147443E-14,      00018700
     * 6.0487998E-14,-3.6973097E-14, 2.2817964E-14,-1.4315547E-14,      00018710
     * 9.1574735E-15,-5.9567236E-15, 3.9209969E-15,-2.5911739E-15,      00018720
     * 1.6406939E-15,-8.8248590E-16, 3.0195409E-16, 2.2622634E-17,      00018730
     *-8.0942556E-17,-3.7172363E-17, 1.9299542E-16,-3.3388160E-16,      00018740
     * 4.6174116E-16,-5.8627358E-16, 7.2227767E-16,-8.7972941E-16,      00018750
     * 1.0211793E-15,-1.0940039E-15, 1.0789555E-15,-9.7089714E-16/      00018760
      DATA WD0/                                                        00018770
     * 7.4110927E-16,-4.1700094E-16, 8.5977184E-17, 1.3396469E-16,      00018780
     *-1.7838410E-16, 4.8975421E-17, 1.9398153E-16,-5.0046989E-16,      00018790
     * 8.3280985E-16,-1.1544640E-15, 1.4401527E-15,-1.6637066E-15,      00018800
     * 1.7777129E-15,-1.7322187E-15, 1.5247247E-15,-1.1771155E-15,      00018810
     * 6.9747910E-16,-1.2088956E-16,-4.8382957E-16, 1.0408292E-15,      00018820
     *-1.5220450E-15, 1.9541597E-15,-2.4107448E-15, 2.9241438E-15,      00018830
     *-3.5176475E-15, 4.2276125E-15,-5.0977851E-15, 6.1428456E-15,      00018840
     *-7.3949962E-15, 8.8597601E-15,-1.0515959E-14, 1.2264584E-14,      00018850
     *-1.3949870E-14, 1.5332490E-14,-1.6146782E-14, 1.6084121E-14,      00018860
     *-1.4962523E-14, 1.2794804E-14,-9.9286701E-15, 6.8825809E-15,      00018870
     *-4.0056107E-15, 1.5965079E-15,-7.2732961E-18,-4.0433218E-16,      00018880
     *-6.5679655E-16, 3.3011866E-15,-7.3545910E-15, 1.2394851E-14,      00018890
     *-1.7947697E-14, 2.3774303E-14,-3.0279168E-14, 3.9252831E-14,      00018900
     *-5.5510504E-14, 9.0505371E-14,-1.7064873E-13/                    00018910
C--END OF J0 FILTER WEIGHTS                                            00018920
C                                                                      00018930
C--J1-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT1 ARRAY)         00018940
      DATA WA1/                                                        00018950
     *-4.2129715E-16, 5.3667031E-15,-7.1183962E-15, 8.9478500E-15,      00018960
     *-1.0767891E-14, 1.2362265E-14,-1.3371129E-14, 1.3284178E-14,      00018970
     *-1.1714302E-14, 8.4134738E-15,-3.7726725E-15,-1.4263879E-15,      00018980
```

```
* 6.1279163E-15,-9.1102765E-15, 9.9696405E-15,-9.3649955E-15,      00018990
* 8.6009018E-15,-8.9749846E-15, 1.1153987E-14,-1.4914821E-14,      00019000
* 1.9314024E-14,-2.3172388E-14, 2.5605477E-14,-2.6217555E-14,      00019010
* 2.5057768E-14,-2.2485539E-14, 1.9022752E-14,-1.5198084E-14,      00019020
* 1.1422464E-14,-7.9323958E-15, 4.8421406E-15,-2.1875032E-15,      00019030
*-3.2177842E-17, 1.8637565E-15,-3.3683643E-15, 4.6132219E-15,      00019040
*-5.6209538E-15, 6.4192841E-15,-6.8959928E-15, 6.9895792E-15,      00019050
*-6.5355935E-15, 5.6125163E-15,-4.1453931E-15, 2.6358827E-15,      00019060
*-9.5104370E-16, 1.4600474E-16, 5.6166519E-16, 8.2899246E-17,      00019070
* 5.0032100E-16, 4.3752205E-16, 2.1052293E-15,-9.5451973E-16,      00019080
* 6.4004437E-15,-2.1926177E-15, 1.1651003E-14, 5.8415433E-16,      00019090
* 1.8044664E-14, 1.0755745E-14, 3.0159022E-14, 3.3506138E-14,      00019100
* 5.8709354E-14, 8.1475200E-14, 1.2530006E-13, 1.8519112E-13,      00019110
* 2.7641786E-13, 4.1330823E-13, 6.1506209E-13, 9.1921659E-13,      00019120
* 1.3698462E-12, 2.0447427E-12, 3.0494477E-12, 4.5501001E-12,      00019130
* 6.7870250E-12, 1.0126237E-11, 1.5104976E-11, 2.2536053E-11/      00019140
 DATA WB1/                                                        00019150
* 3.3617368E-11, 5.0153839E-11, 7.4818173E-11, 1.1161804E-10,      00019160
* 1.6651222E-10, 2.4840923E-10, 3.7058109E-10, 5.5284353E-10,      00019170
* 8.2474468E-10, 1.2303750E-09, 1.8355034E-09, 2.7382502E-09,      00019180
* 4.0849867E-09, 6.0940898E-09, 9.0913020E-09, 1.3562651E-08,      00019190
* 2.0233058E-08, 3.0184244E-08, 4.5029477E-08, 6.7176304E-08,      00019200
* 1.0021488E-07, 1.4950371E-07, 2.2303208E-07, 3.3272689E-07,      00019210
* 4.9636623E-07, 7.4049804E-07, 1.1046805E-06, 1.6480103E-06,      00019220
* 2.4585014E-06, 3.6677163E-06, 5.4714550E-06, 8.1626422E-06,      00019230
* 1.2176782E-05, 1.8166179E-05, 2.7099223E-05, 4.0428804E-05,      00019240
* 6.0307294E-05, 8.9971508E-05, 1.3420195E-04, 2.0021123E-04,      00019250
* 2.9860417E-04, 4.4545291E-04, 6.6423156E-04, 9.9073275E-04,      00019260
* 1.4767050E-03, 2.2016806E-03, 3.2788147E-03, 4.8837292E-03,      00019270
* 7.2596811E-03, 1.0788355E-02, 1.5973323E-02, 2.3612041E-02,      00019280
* 3.4655327E-02, 5.0608141E-02, 7.2827752E-02, 1.0337889E-01,      00019290
* 1.4207357E-01, 1.8821315E-01, 2.2996815E-01, 2.5088500E-01,      00019300
* 2.0334626E-01, 6.0665451E-02,-2.0275683E-01,-3.5772336E-01,      00019310
*-1.8280529E-01, 4.7014634E-01, 7.2991233E-03,-3.0614594E-01,      00019320
* 2.4781735E-01,-1.1149185E-01, 2.5985386E-02, 1.0850279E-02,      00019330
*-2.2830217E-02, 2.4644647E-02,-2.2895284E-02, 2.0197032E-02/      00019340
 DATA WC1/                                                        00019350
*-1.7488968E-02, 1.5057670E-02,-1.2953923E-02, 1.1153254E-02,      00019360
*-9.6138436E-03, 8.2952090E-03,-7.1628361E-03, 6.1882910E-03,      00019370
*-5.3482055E-03, 4.6232056E-03,-3.9970542E-03, 3.4560118E-03,      00019380
*-2.9883670E-03, 2.5840861E-03,-2.2345428E-03, 1.9323046E-03,      00019390
*-1.6709583E-03, 1.4449655E-03,-1.2495408E-03, 1.0805480E-03,      00019400
*-9.3441130E-04, 8.0803899E-04,-6.9875784E-04, 6.0425624E-04,      00019410
*-5.2253532E-04, 4.5186652E-04,-3.9075515E-04, 3.3790861E-04,      00019420
*-2.9220916E-04, 2.5269019E-04,-2.1851585E-04, 1.8896332E-04,      00019430
*-1.6340753E-04, 1.4130796E-04,-1.2219719E-04, 1.0567099E-04,      00019440
*-9.1379828E-05, 7.9021432E-05,-6.8334412E-05, 5.9092726E-05,      00019450
*-5.1100905E-05, 4.4189914E-05,-3.8213580E-05, 3.3045496E-05,      00019460
*-2.8576356E-05, 2.4711631E-05,-2.1369580E-05, 1.8479514E-05,      00019470
*-1.5980307E-05, 1.3819097E-05,-1.1950174E-05, 1.0334008E-05,      00019480
*-8.9364160E-06, 7.7278366E-06,-6.6827083E-06, 5.7789251E-06,      00019490
*-4.9973715E-06, 4.3215167E-06,-3.7370660E-06, 3.2316575E-06,      00019500
```

```
     *-2.7946015E-06, 2.4166539E-06,-2.0898207E-06, 1.8071890E-06,      00019510
     *-1.5627811E-06, 1.3514274E-06,-1.1686576E-06, 1.0106059E-06,      00019520
     *-8.7392952E-07, 7.5573750E-07,-6.5353002E-07, 5.6514528E-07,      00019530
     *-4.8871388E-07, 4.2261921E-07,-3.6546333E-07, 3.1603732E-07/      00019540
      DATA WD1/                                                        00019550
     *-2.7329579E-07, 2.3633470E-07,-2.0437231E-07, 1.7673258E-07,      00019560
     *-1.5283091E-07, 1.3216174E-07,-1.1428792E-07, 9.8831386E-08,      00019570
     *-8.5465227E-08, 7.3906734E-08,-6.3911437E-08, 5.5267923E-08,      00019580
     *-4.7793376E-08, 4.1329702E-08,-3.5740189E-08, 3.0906612E-08,      00019590
     *-2.6726739E-08, 2.3112160E-08,-1.9986424E-08, 1.7283419E-08,      00019600
     *-1.4945974E-08, 1.2924650E-08,-1.1176694E-08, 9.6651347E-09,      00019610
     *-8.3580023E-09, 7.2276490E-09,-6.2501673E-09, 5.4048822E-09,      00019620
     *-4.6739154E-09, 4.0418061E-09,-3.4951847E-09, 3.0224895E-09,      00019630
     *-2.6137226E-09, 2.2602382E-09,-1.9545596E-09, 1.6902214E-09,      00019640
     *-1.4616324E-09, 1.2639577E-09,-1.0930164E-09, 9.4519327E-10,      00019650
     *-8.1736202E-10, 7.0681930E-10,-6.1122713E-10, 5.2856342E-10,      00019660
     *-4.5707937E-10, 3.9526267E-10,-3.4180569E-10, 2.9557785E-10,      00019670
     *-2.5560176E-10, 2.2103233E-10,-1.9113891E-10, 1.6528994E-10,      00019680
     *-1.4294012E-10, 1.2361991E-10,-8.2740936E-11/                     00019690
C--END OF J1 FILTER WEIGHTS                                            00019700
C                                                                      00019710
      NONE=0                                                           00019720
      IF(NEW.EQ.0) GO TO 100                                           00019730
      NSAVE=0                                                          00019740
C-----INITIALIZE KERNEL ABSCISSA GENERATION FOR GIVEN B                00019750
      Y1=0.73588526614797944600D0/DBLE(B)                              00019760
  100 ZHANKS=(0.0,0.0)                                                 00019770
      CMAX=(0.0,0.0)                                                   00019780
      NF=0                                                             00019790
      Y=Y1                                                             00019800
C-----BEGIN RIGHT-SIDE CONVOLUTION AT WEIGHT 131 (EITHER NEW=1 OR 0)   00019810
      ASSIGN 110 TO M                                                  00019820
      I=131                                                            00019830
      Y=Y*E                                                            00019840
      GO TO 200                                                        00019850
  110 TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                                 00019860
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                                 00019870
      I=I+1                                                            00019880
      Y=Y*E                                                            00019890
      IF(I.LE.149) GO TO 200                                           00019900
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) NONE=1                     00019910
C-----ESTABLISH TRUNCATION CRITERION (CMAX=CMPLX(TMAX(1),TMAX(2))      00019920
      CMAX=TOL*CMAX                                                    00019930
      ASSIGN 120 TO M                                                  00019940
      GO TO 200                                                        00019950
C-----CHECK FOR FILTER TRUNCATION AT RIGHT END                         00019960
  120 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130      00019970
      I=I+1                                                            00019980
      Y=Y*E                                                            00019990
      IF(I.LE.283) GO TO 200                                           00020000
  130 Y=Y1                                                             00020010
C-----CONTINUE WITH LEFT-SIDE CONVOLUTION AT WEIGHT 130                00020020
```

```
      ASSIGN 140 TO M                                                   00020030
      I=130                                                             00020040
      GO TO 200                                                         00020050
C-----CHECK FOR FILTER TRUNCATION AT LEFT END                           00020060
  140 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2).AND.             00020070
     * NONE.EQ.0) GO TO 190                                            00020080
      I=I-1                                                             00020090
      Y=Y*ER                                                            00020100
      IF(I.GT.0) GO TO 200                                              00020110
C-----RETURN WITH ISAVE=1 PRESET FOR POSSIBLE NEW=0 USE.                00020120
  190 ISAVE=1                                                           00020130
C-----NORMALIZE BY B TO ACCOUNT FOR INTEGRATION RANGE CHANGE            00020140
      ZHANKS=ZHANKS/B                                                   00020150
      RETURN                                                            00020160
C-----SAVE/RETRIEVE PSEUDO-SUBROUTINE (CALL FUN ONLY WHEN NECESSARY)    00020170
  200 G=SNGL(Y)                                                         00020180
      IF(NEW) 300,210,300                                               00020190
  210 IF(ISAVE.GT.NSAVE) GO TO 300                                      00020200
      ISAVE0=ISAVE                                                      00020210
  220 IF(G.EQ.GSAVE(ISAVE)) GO TO 240                                   00020220
      ISAVE=ISAVE+1                                                     00020230
      IF(ISAVE.LE.NSAVE) GO TO 220                                      00020240
      ISAVE=ISAVE0                                                      00020250
C-----G NOT IN COMMON/SAVE/----- EVALUATE FUN.                          00020260
      GO TO 300                                                         00020270
C-----G FOUND IN COMMON/SAVE/----- USE FSAVE AS GIVEN.                  00020280
  240 C=FSAVE(ISAVE)                                                    00020290
      ISAVE=ISAVE+1                                                     00020300
C-----SWITCH ON ORDER N                                                 00020310
  250 IF(N) 270,260,270                                                 00020320
  260 C=C*WT0(I)                                                        00020330
      GO TO 280                                                         00020340
  270 C=C*WT1(I)                                                        00020350
  280 ZHANKS=ZHANKS+C                                                   00020360
      GO TO M,(110,120,140)                                             00020370
C-----DIRECT FUN EVALUATION (AND ADD TO END OF COMMON/SAVE/)            00020380
  300 NSAVE=NSAVE+1                                                     00020390
      C=FUN(G)                                                          00020400
      NF=NF+1                                                           00020410
      FSAVE(NSAVE)=C                                                    00020420
      GSAVE(NSAVE)=G                                                    00020430
      GO TO 250                                                         00020440
      END                                                               00020450

      SUBROUTINE FCODE(Y,X,B,PRNT,F,IN,IDER)                            00020460
C--FUNCTION EVALUATION FOR 'MARQLOOPS' (ALL OPTIONS--SEE DOCUMENTATION).00020470
C                                                                       00020480
C--PARAMETERS--                                                         00020490
C                                                                       00020500
C       Y=      OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)              00020510
C       X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)          00020520
C       B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)              00020530
```

```
C       PRNT=   WORK AND PRINT ARRAY (DIM. 5)                                   00020540
C       F=      OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. IN          00020550
C       IN=     OBSERVATION NO. TO EVAL. F (1<=IN<=N)                           00020560
C       IDER=   0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED)         00020570
C               1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED)         00020580
C                                                                               00020590
      REAL*8 B8                                                                 00020600
      REAL Y(1),X(200,5),B(1),PRNT(5),F,K(10),D(9)                              00020610
      COMPLEX ZHANKS,CA,ONESG1,TWOSG1,                                          00020620
     & T0,T1,T2,TERM1,TERM2,ZZ0,CB,CB2,CB3,I1K1,I0K0,IKDIF,                     00020630
     & ZERO,ONE,ONEI,ONE1,TWO,TWOI,TWO2,THREE,THREEI,THREE3,FOUR,               00020640
     & EIGHT,EIGHTI,NINE,NINE9,TENI,TWELVE,TWELV2,SIXTEN                        00020650
      EXTERNAL FG2,FG,RG2,RG,KERN                                               00020660
      COMMON/SHARE/FILL(4),XX,YY,YY2,RHO,RHO2,FILL2,BB,FILL3,DEL,DEL2,          00020670
     & IREST(3)                                                                 00020680
      COMMON/MODEL/K,D,M                                                        00020690
      COMMON/AIR/A                                                              00020700
      COMMON/CTL/T0,T1,T2,CB,CB2,CB3,CA,ONESG1,TWOSG1,I0K0,I1K1,IKDIF,          00020710
     & ZZ0,AMP,FREQ,SIG1,H,EPS,IOB,M1,M21,ILOOPS,IMM                            00020720
      COMMON/PART/JJ,ISEP                                                       00020730
      COMMON/RESIST/RRHO(10),HH(9),EEPS,RR,RR2,XJUNK,MMMM(4)                    00020740
      DATA ZERO/(0.,0.)/,ONE/(1.,0.)/,ONEI/(0.,1.)/,ONE1/(1.,1.)/,              00020750
     & TWO/(2.,0.)/,TWOI/(0.,2.)/,TWO2/(-2.,2.)/,THREE/(3.,0.)/,               00020760
     & FOUR/(4.,0.)/,                                                           00020770
     & THREEI/(0.,3.)/,THREE3/(3.,3.)/,EIGHT/(8.,0.)/,EIGHTI/(0.,8.)/,         00020780
     & NINE/(9.,0.)/,NINE9/(9.,9.)/,TENI/(0.,10.)/,TWELVE/(12.,0.)/,           00020790
     & TWELV2/(12.,12.)/,SIXTEN/(16.,0.)/                                       00020800
      IF(IN.GT.1.OR.M.EQ.1) GO TO 20                                            00020810
      DO 10 J=2,M                                                               00020820
      IF(B(J).EQ.B(J-1))CALL ERRMSG(20HSOME SIG(J)=SIG(J-1),4,6,16)             00020830
10    CONTINUE                                                                  00020840
20    DO 30 J=1,5                                                               00020850
30    PRNT(J)=X(IN,J)                                                           00020860
      ISEP=0                                                                    00020870
      IF(IOB.NE.6) GO TO 40                                                     00020880
      IF(YY.NE.PRNT(4).AND.PRNT(4).GT.0.0) ISEP=1                               00020890
      IF(H.NE.ABS(PRNT(4)).AND.PRNT(4).LE.0.0) ISEP=-1                          00020900
40    FREQ=PRNT(1)                                                              00020910
      IF(IOB.GE.5.AND.PRNT(3).EQ.5.0) GO TO 300                                 00020920
      LOOPS=IFIX(PRNT(2))                                                       00020930
      ILOOPS=IABS(LOOPS)                                                        00020940
      NEW=1                                                                     00020950
      IF(IN.EQ.1.OR.IDER.NE.0.OR.FREQ.NE.FREQL.OR.ISEP.NE.0) GO TO 50           00020960
      NEW=0                                                                     00020970
      IF(LOOPS.NE.LOOPSL) GO TO 100                                             00020980
      IF(IOB.EQ.5) GO TO 190                                                    00020990
      IF(IOB.EQ.6.AND.ISEP.EQ.0) GO TO 190                                      00021000
50    SIG1=B(1)                                                                 00021010
      ONESG1=CMPLX(SIG1,0.0)                                                    00021020
      TWOSG1=CMPLX(2.0*SIG1,0.0)                                                00021030
      DEL2=1.0/(39.47841762E-7*SIG1*FREQ)                                       00021040
      DEL=SQRT(DEL2)                                                            00021050
```

```
      IF(IOB.NE.6.OR.ISEP.EQ.0) GO TO 60                                00021060
      IF(ISEP.LT.0) H=ABS(PRNT(4))                                      00021070
      IF(ISEP.GT.0) YY=PRNT(4)                                          00021080
60    YY2=YY*YY                                                         00021090
      RHO=YY                                                            00021100
      RHO2=RHO*RHO                                                      00021110
      IF(M.EQ.1) GO TO 90                                               00021120
      DO 80 J=1,M1                                                      00021130
      K(J)=B(J)/SIG1                                                    00021140
80    D(J)=2.0*B(J+M)/DEL                                               00021150
90    K(M)=B(M)/SIG1                                                    00021160
      BB=RHO/DEL                                                        00021170
      CB=CMPLX(BB,0.)                                                   00021180
      CB2=CB*CB                                                         00021190
      CB3=CB*CB2                                                        00021200
      A=H/DEL                                                           00021210
      CA=CMPLX(A,0.0)                                                   00021220
      T0=ZERO                                                           00021230
      T1=ZERO                                                           00021240
      T2=ZERO                                                           00021250
C--SWITCH FOR GROUND (A=0) OR AIRBORNE (A>0) CASE.                      00021260
100   IF(A.GT.0.0) GO TO 2000                                           00021270
C--GROUND CASE (A=0), GET T0,T1,T2 INTEGRALS AS REQUIRED.               00021280
      IF(M.EQ.1) GO TO 806                                              00021290
      GO TO (801,802,8029,804,803),ILOOPS                               00021300
801   T0=ZHANKS(0,BB,FG2,EPS,NW0,NEW)                                   00021310
      GO TO 806                                                         00021320
802   T1=ZHANKS(1,BB,FG2,EPS,NW1,NEW)                                   00021330
      GO TO 806                                                         00021340
8029  IF(NEW.EQ.1) GO TO 803                                            00021350
8030  CALL MODIFY(-1)                                                   00021360
803   T2=ZHANKS(1,BB,FG,EPS,NW2,NEW)                                    00021370
      CALL MODIFY(1)                                                    00021380
      GO TO 806                                                         00021390
804   T0=ZHANKS(0,BB,FG2,EPS,NW0,NEW)                                   00021400
      NEW=0                                                             00021410
      GO TO 8030                                                        00021420
C--GET Z/Z0 FOR GROUND CASE (A=0)                                       00021430
806   GO TO (1001,1002,1003,1004,1005),ILOOPS                           00021440
1001  TERM1=-ONEI*(NINE-(NINE+NINE9*CB+EIGHTI*CB2+                      00021450
     & TWO2*CB3)*CEXP(-CB*ONE1))/CB2                                    00021460
      TERM2=CB3*T0                                                      00021470
      GO TO 1006                                                        00021480
1002  B8=.7071067811865475D0*DBLE(BB)                                   00021490
      CALL IKS2(B8,I0K0,I1K1,IKDIF)                                     00021500
      TERM1=EIGHT*(IKDIF-FOUR*I1K1)                                     00021510
      TERM1= (TWOI*CB2*(I1K1-I0K0)+TERM1+SIXTEN*I1K1)                   00021520
      TERM2=CB3*T1                                                      00021530
      GO TO 1006                                                        00021540
1003  TERM1=(TWO*CB2+THREEI-ONEI*(THREE+THREE3*CB+TWOI*CB2)*            00021550
     & CEXP(-CB*ONE1))/CB2                                              00021560
      TERM2=CB2*T2                                                      00021570
```

```
      GO TO 1006                                                        00021580
1004  TERM1=-ONEI*(TWELVE+TWELV2*CB+TENI*CB2+TWO2*CB3)*CEXP(-CB*ONE1)/  00021590
     & (TWO*CB2)+TWO*(ONE+THREEI/CB2)                                   00021600
      TERM2=CB2*(T2-CB*T0)/TWO                                          00021610
      GO TO 1006                                                        00021620
1005  TERM1=-ONEI*(THREE-(THREE+THREE3*CB+TWOI*CB2)*                    00021630
     & CEXP(-CB*ONE1))/CB2                                              00021640
      TERM2=-CB2*T2                                                     00021650
C--COMPUTE MUTUAL COUPLING (AIRBORNE OR GROUND CASE)                    00021660
1006  ZZ0=TERM1+TERM2                                                   00021670
      IF(IMM.LT.0) ZZ0=B(M21+1)*ZZ0                                     00021680
      GO TO 3000                                                        00021690
C--AIRBORNE CASE (A>0), GET T0,T1,T2 INTEGRALS AS SELECTED              00021700
2000  GO TO (2001,2002,20029,2004,2003),ILOOPS                          00021710
2001  T0=ZHANKS(0,BB,RG2,EPS,NW0,NEW)                                   00021720
      GO TO 2006                                                        00021730
2002  T1=ZHANKS(1,BB,RG2,EPS,NW1,NEW)                                   00021740
      GO TO 2006                                                        00021750
20029 IF(NEW.EQ.1) GO TO 2003                                           00021760
20030 CALL MODIFY(-1)                                                   00021770
2003  T2=ZHANKS(1,BB,RG,EPS,NW2,NEW)                                    00021780
      CALL MODIFY(1)                                                    00021790
      GO TO 2006                                                        00021800
2004  T0=ZHANKS(0,BB,RG2,EPS,NW0,NEW)                                   00021810
      NEW=0                                                             00021820
      GO TO 20030                                                       00021830
C--GET Z/Z0 FOR AIRBORNE CASE (A>0)                                     00021840
2006  GO TO (2011,2012,2013,2014,2015),ILOOPS                           00021850
2011  TERM1=ONE                                                         00021860
      TERM2=CB3*T0                                                      00021870
      GO TO 1006                                                        00021880
2012  TERM1=ZERO                                                        00021890
      TERM2=CB3*T1                                                      00021900
      GO TO 1006                                                        00021910
2013  TERM1=ONE                                                         00021920
      TERM2=CB2*T2                                                      00021930
      GO TO 1006                                                        00021940
2014  TERM1=ONE                                                         00021950
      TERM2=CB2*(T2-CB*T0)/TWO                                          00021960
      GO TO 1006                                                        00021970
2015  TERM1=ONE                                                         00021980
      TERM2=-CB2*T2                                                     00021990
      GO TO 1006                                                        00022000
C////////////////////                                                   00022010
3000  GO TO (140,150,160,170,190,190),IOB                               00022020
140   F=CABS(ZZ0)                                                       00022030
      AMP=F                                                             00022040
      GO TO 180                                                         00022050
150   CALL POLAR2(ZZ0,AMP,F)                                            00022060
      GO TO 180                                                         00022070
160   F=REAL(ZZ0)                                                       00022080
      GO TO 180                                                         00022090
```

```
170    F=AIMAG(ZZ0)                                                    00022100
180    LOOPSL=LOOPS                                                    00022110
       FREQL=FREQ                                                      00022120
       RETURN                                                          00022130
190    IOBS=PRNT(3)                                                    00022140
200    GO TO (140,150,160,170),IOBS                                    00022150
C--COMPUTE APPARENT RESISTIVITY (IOB>=5, PRNT(3)=5.0)                  00022160
300    RR=PRNT(1)                                                      00022170
       CALL SWAP(1)                                                    00022180
       RR2=RR*RR                                                       00022190
       RRHO(M)=1.0/B(M)                                                00022200
       IF(M.EQ.1) GO TO 320                                            00022210
       DO 310 J=1,M1                                                   00022220
       RRHO(J)=1.0/B(J)                                                00022230
310    HH(J)=B(J+M)                                                    00022240
320    F=RRHO(1)*(RR2*REAL(ZHANKS(1,RR,KERN,EEPS,LL,1))+1.0)           00022250
       CALL SWAP(-1)                                                   00022260
       RETURN                                                          00022270
       END                                                             00022280

       SUBROUTINE PCODE(P,X,B,PRNT,F,IN,IP,IB)                         00022290
C--ANALYTIC PARTIALS W/R PARAMETERS IN B(K) AND IN COMMON               00022300
C  FOR PROGRAM 'MARQLOOPS'.                                            00022310
C                                                                      00022320
C  (PCODE ONLY CALLED BY MARQRT IF IDER=0--DEFAULT)                    00022330
C                                                                      00022340
C--PARAMETERS--                                                        00022350
C                                                                      00022360
C       P=      OUTPUT PARTIAL DERIVATIVE ARRAY (DIM. K)               00022370
C               EVALUATED FOR GIVEN X(IN,),B(K) AT OBS. IN             00022380
C       X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)         00022390
C       B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)             00022400
C       PRNT=   WORK AND PRINT ARRAY (DIM. 5)                          00022410
C       F=      LAST FUNCTION VALUE FROM FCODE AT GIVEN IN.            00022420
C               F MAY OR MAY NOT BE NEEDED--BUT AVAILABLE ANYWAY.      00022430
C       IN=     OBSERVATION NO. TO EVAL. P ARRAY (1<=IN<=N)            00022440
C       IP=     NO. PARAMETERS HELD FIXED (IF ANY--IF NONE IP=0).      00022450
C       IB=     ARRAY OF PARAMETER INDICES HELD FIXED IF IP.GT.0       00022460
C               (DIM. 19).                                             00022470
C                                                                      00022480
       LOGICAL SHIFT                                                   00022490
       INTEGER IB(1)                                                   00022500
       REAL P(1),X(200,5),B(1),PRNT(5)                                 00022510
       COMPLEX Z(19),ZHANKS,ZZ0,T0,T1,T2,CB,CB2,CB3,CA,ONESG1,         00022520
      & TWOSG1,ZERO,TWO,THREE,ZTEMP,ZSIGN,ONE,ONEI,NINE,NINE9,EIGHTI,  00022530
      & SIXI,ONE1,I0K0,I1K1,IKDIF,SIX,THREEM,THREEI,THREE3,TWOI,SIX6,  00022540
      & TWOM2,FOUR,ONEM1,TWO2,EIGHT,TWELVE,TWELV2,TENI,HALF2,ZZ01      00022550
       EXTERNAL RG3,PRBJG2,RG2,PRBJG,FG2,PFBJG2,FG3,PFBJG,FVP          00022560
       COMMON/SHARE/FILL(4),XX,YY,YY2,RHO,RHO2,FILL2,BB,               00022570
      1 FILL3,DEL,DEL2,IREST(3)                                        00022580
       COMMON/RESIST/RRHO(10),HH(9),EEPS,RR,RR2,XJUNK,MMMM(4)          00022590
       COMMON/AIR/A                                                    00022600
```

```
      COMMON/CTL/T0,T1,T2,CB,CB2,CB3,CA,ONESG1,TWOSG1,I0K0,I1K1,IKDIF,  00022610
     & ZZ0,AMP,FREQ,SIG1,H,EPS,IOB,M1,M21,ILOOPS,IMM                    00022620
      COMMON/PART/JJ,ISEP                                               00022630
      DATA ZERO/(0.0,0.0)/,TWO/(2.0,0.0)/,THREE/(3.0,0.0)/,ONE/(1.,0.)/,00022640
     & ONEI/(0.0,1.0)/,NINE/(9.0,0.0)/,NINE9/(9.0,9.0)/,SIXI/(0.0,6.0)/,00022650
     & EIGHTI/(0.,8.)/,TENI/(0.0,10.0)/,ONE1/(1.0,1.0)/,SIX/(6.0,0.0)/, 00022660
     & THREEM/(-3.0,3.0)/,THREEI/(0.0,3.0)/,THREE3/(3.0,3.0)/,          00022670
     & TWOI/(0.0,2.0)/,SIX6/(6.0,6.0)/,TWOM2/(2.0,-2.0)/,FOUR/(4.0,0.0)/00022680
     &,ONEM1/(1.0,-1.0)/,TWO2/(-2.,2.)/,EIGHT/(8.0,0.0)/,               00022690
     & TWELVE/(12.0,0.0)/,TWELV2/(12.,12.)/,HALF2/(.5,.5)/              00022700
C--GET PARTIALS W/R SIGMA(JJ), JJ IN (1,M), OR                          00022710
C  W/R DIST(JJ-M),JJ IN (M+1,2*M-1), M=NO.LAYERS (M21=2*M-1 IN COMMON). 00022720
C  (ANY PARM MAY BE HELD FIXED VIA IP,IB()).                            00022730
C                                                                       00022740
C--SWAP (SAVE) FSAVE() = FG2 OR RG2 FROM LAST FCODE CALL.               00022750
      CALL SWAP(1)                                                      00022760
      M2=M21                                                            00022770
      IF(IMM.LT.0) M2=M21+1                                             00022780
      SHIFT=.FALSE.                                                     00022790
      DO 2 J=1,M2                                                       00022800
      IF(IMM.LT.0.AND.J.EQ.M2) SHIFT=.TRUE.                             00022810
      JJ=J                                                              00022820
      IF(IP.LE.0) GO TO 30                                              00022830
      P(J)=0.0                                                          00022840
      DO 20 I=1,IP                                                      00022850
      IF(IB(I).EQ.J) GO TO 2                                            00022860
20    CONTINUE                                                          00022870
30    IF(IOB.GE.5.AND.PRNT(3).EQ.5.0) GO TO 300                         00022880
      IF(IN.EQ.1.OR.FREQ.NE.FREQLL.OR.ILOOPS.NE.LOOPSL) GO TO 40        00022890
      IF(IOB.EQ.5) GO TO 2130                                           00022900
      IF(IOB.EQ.6.AND.ISEP.EQ.0) GO TO 2130                             00022910
C--SWITCH FOR GROUND (A=0) OR AIRBORNE (A>0) CASE.                      00022920
40    IF(SHIFT) GO TO 2140                                              00022930
      IF(A.GT.0.0) GO TO 2000                                           00022940
C--GROUND CASE (A=0)                                                    00022950
      ZTEMP=ZERO                                                        00022960
      GO TO (101,102,103,104,105),ILOOPS                                00022970
101   IF(J.GT.1) GO TO 1011                                             00022980
      IF(M1.EQ.0) GO TO 1012                                            00022990
      T1=ZHANKS(1,BB,FG2,EPS,NW1,0)                                     00023000
      ZTEMP=CB3*(ZHANKS(0,BB,PFBJG2,EPS,NW0,1)-                         00023010
     & (CB*T1-THREE*T0)/TWOSG1)                                         00023020
1012  Z(J)=CEXP(-CB*ONE1)                                               00023030
      Z(J)=ZTEMP-ONEI*(Z(J)*(ONEI-ONEM1*CB-TWO*CB2)-(NINE-              00023040
     & (NINE+NINE9*CB+EIGHTI*CB2+TWO2*CB3)*                             00023050
     & Z(J))/CB2)/ONESG1                                                00023060
      GO TO 2080                                                        00023070
1011  Z(J)=CB3*ZHANKS(0,BB,PFBJG2,EPS,NW0,1)                            00023080
      GO TO 2080                                                        00023090
102   IF(J.GT.1) GO TO 1021                                             00023100
      IF(M1.EQ.0) GO TO 1022                                            00023110
      CALL MODIFY(1)                                                    00023120
```

```
      ZTEMP=T1/ONESG1+CB*ZHANKS(0,BB,FG3,EPS,NW0,0)/TWOSG1                00023130
      ZTEMP=CB3*(ZHANKS(1,BB,PFBJG2,EPS,NW1,1)+ZTEMP)                     00023140
1022  Z(J)=ZTEMP+TWO*(-I1K1*(EIGHT+ONEI*CB2)+                             00023150
     & (FOUR*I1K1-IKDIF)*(ONEI*CB2+FOUR)+                                 00023160
     & ONEI*CB2*(I0K0-I1K1))/ONESG1                                       00023170
      GO TO 2080                                                          00023180
1021  Z(J)=CB3*ZHANKS(1,BB,PFBJG2,EPS,NW1,1)                              00023190
      GO TO 2080                                                          00023200
103   ZSIGN=ONE                                                           00023210
1030  IF(J.GT.1) GO TO 1031                                               00023220
      IF(M1.EQ.0) GO TO 1032                                              00023230
      ZTEMP=(T2+CB*ZHANKS(0,BB,FG2,EPS,NW0,0))/TWOSG1                     00023240
      ZTEMP=ZSIGN*CB2*(ZHANKS(1,BB,PFBJG,EPS,NW1,1)+ZTEMP)                00023250
1032  Z(J)=CEXP(-CB*ONE1)                                                 00023260
      IF(ILOOPS.EQ.3) Z(J)=(TWO-(ONE+ONE1*CB)*Z(J)-                       00023270
     & (TWO*CB2+THREEI-ONEI*(THREE+THREE3*CB+TWOI*CB2)*Z(J))/CB2)/        00023280
     & ONESG1 +ZTEMP                                                      00023290
      IF(ILOOPS.EQ.5) Z(J)=ONEI*(-CB2*Z(J)*(ONEI-ONEM1*CB)                00023300
     & +THREE-(THREE+THREE3*CB+TWOI*CB2)*                                 00023310
     & Z(J))/(CB2*ONESG1) +ZTEMP                                          00023320
      GO TO 2080                                                          00023330
1031  Z(J)=ZSIGN*CB2*ZHANKS(1,BB,PFBJG,EPS,NW1,1)                         00023340
      GO TO 2080                                                          00023350
104   IF(J.GT.1) GO TO 1041                                               00023360
      IF(M1.EQ.0) GO TO 1042                                              00023370
      T1=ZHANKS(1,BB,FG2,EPS,NW1,0)                                       00023380
      ZTEMP=ZHANKS(1,BB,PFBJG,EPS,NW1,1)                                  00023390
      CALL MODIFY(1)                                                      00023400
      ZTEMP=CB2*(TWOSG1*ZTEMP-TWOSG1*CB*                                  00023410
     & ZHANKS(0,BB,PFBJG2,EPS,NW0,0)+CB2*T1+T2-TWO*CB*T0)/(FOUR*ONESG1)   00023420
1042  Z(J)=-ONEI*CEXP(-CB*ONE1)*(-(TWELVE+TWELV2*CB+TENI*CB2+TWO2*CB3)*   00023430
     & (HALF2*CB+ONE)+CB*(SIX6+TENI*CB+THREEM*CB2))/(TWOSG1*CB2)-         00023440
     & SIXI/(ONESG1*CB2)                                                  00023450
      GO TO 2080                                                          00023460
1041  ZTEMP=ZHANKS(1,BB,PFBJG,EPS,NW1,1)                                  00023470
      CALL MODIFY(1)                                                      00023480
      Z(J)=CB2*(ZTEMP-CB*ZHANKS(0,BB,PFBJG2,EPS,NW0,0))/TWO               00023490
      GO TO 2080                                                          00023500
105   ZSIGN=-ONE                                                          00023510
      GO TO 1030                                                          00023520
C--AIRBORNE CASE (A>0)                                                    00023530
2000  GO TO (2001,2002,2003,2004,2005),ILOOPS                             00023540
2001  IF(J.GT.1) GO TO 20011                                              00023550
      CALL MODIFY(1)                                                      00023560
      ZTEMP=CA*ZHANKS(0,BB,RG3,EPS,NW0,0)+                                00023570
     & CB*ZHANKS(1,BB,RG3,EPS,NW1,0)-THREE*T0                             00023580
      Z(J)=-CB3*(ZTEMP-TWOSG1*ZHANKS(0,BB,PRBJG2,EPS,NW0,1))/TWOSG1       00023590
      GO TO 2080                                                          00023600
20011 Z(J)=CB3*ZHANKS(0,BB,PRBJG2,EPS,NW0,1)                              00023610
      GO TO 2080                                                          00023620
2002  IF(J.GT.1) GO TO 20021                                              00023630
      CALL MODIFY(1)                                                      00023640
```

```
      ZTEMP=-CA*ZHANKS(1,BB,RG3,EPS,NW1,0)+                              00023650
     & CB*ZHANKS(0,BB,RG3,EPS,NW0,0)+TWO*T1                              00023660
      Z(J)=CB3*(ZTEMP+TWOSG1*ZHANKS(1,BB,PRBJG2,EPS,NW1,1))/TWOSG1       00023670
      GO TO 2080                                                         00023680
20021 Z(J)=CB3*ZHANKS(1,BB,PRBJG2,EPS,NW1,1)                             00023690
      GO TO 2080                                                         00023700
2003  ZSIGN=ONE                                                          00023710
20030 IF(J.GT.1) GO TO 20031                                             00023720
      ZTEMP=-CA*ZHANKS(1,BB,RG2,EPS,NW1,0)+CB*                           00023730
     & ZHANKS(0,BB,RG2,EPS,NW0,0)+T2                                     00023740
      Z(J)=ZSIGN*CB2*(ZTEMP+TWOSG1*ZHANKS(1,BB,PRBJG,EPS,NW1,1))/TWOSG1  00023750
      GO TO 2080                                                         00023760
20031 Z(J)=ZSIGN*CB2*ZHANKS(1,BB,PRBJG,EPS,NW1,1)                        00023770
      GO TO 2080                                                         00023780
2004  IF(J.GT.1) GO TO 20041                                             00023790
      T1=ZHANKS(1,BB,RG2,EPS,NW1,0)                                      00023800
      CALL MODIFY(1)                                                     00023810
      Z(J)=-CA*T1+CA*CB*ZHANKS(0,BB,RG3,EPS,NW0,0)+                      00023820
     & CB2*ZHANKS(1,BB,RG3,EPS,NW1,0)+T2-TWO*CB*T0                       00023830
      ZTEMP=ZHANKS(1,BB,PRBJG,EPS,NW1,1)                                 00023840
      CALL MODIFY(1)                                                     00023850
      ZTEMP=TWOSG1*(ZTEMP+CB*ZHANKS(0,BB,PRBJG2,EPS,NW0,0))              00023860
      Z(J)=CB2*(Z(J)+ZTEMP)/(TWO*TWOSG1)                                 00023870
      GO TO 2080                                                         00023880
20041 ZTEMP=ZHANKS(1,BB,PRBJG,EPS,NW1,1)                                 00023890
      CALL MODIFY(1)                                                     00023900
      Z(J)=CB2*(ZTEMP-CB*ZHANKS(0,BB,PRBJG2,EPS,NW0,0))/TWO              00023910
      GO TO 2080                                                         00023920
2005  ZSIGN=-ONE                                                         00023930
      GO TO 20030                                                        00023940
C--SWITCH ON IOB                                                         00023950
2080  IF(IMM.LT.0.AND.J.LT.M2) Z(J)=Z(J)*B(M2)                           00023960
      GO TO (2090,2100,2110,2120,2130,2130),IOB                          00023970
2090  PP=(REAL(ZZ0)*REAL(Z(J))+AIMAG(ZZ0)*AIMAG(Z(J)))/AMP               00023980
      GO TO 2160                                                         00023990
2100  PP=0.0                                                             00024000
      IF(SHIFT) GO TO 2160                                               00024010
      PP=57.29577951*(REAL(ZZ0)*AIMAG(Z(J))-                             00024020
     & AIMAG(ZZ0)*REAL(Z(J)))/(AMP*AMP)                                  00024030
      GO TO 2160                                                         00024040
2110  PP=REAL(Z(J))                                                      00024050
      GO TO 2160                                                         00024060
2120  PP=AIMAG(Z(J))                                                     00024070
      GO TO 2160                                                         00024080
2130  IOBS=PRNT(3)                                                       00024090
      GO TO (2090,2100,2110,2120),IOBS                                   00024100
C--ANALYTIC PARTIAL OF APP.RES. (IOB>=5, PRNT(3)=5.0)                    00024110
300   PP=0.0                                                             00024120
      IF(SHIFT) GO TO 2160                                               00024130
      MMM1(4)=JJ                                                         00024140
      PP=RRHO(1)*RR2*REAL(ZHANKS(1,RR,FVP,EEPS,LL,1))                    00024150
      IF(J.EQ.1) PP=PP+F/RRHO(1)                                         00024160
```

```
      IF(J.LE.MMMM(1)) PP=-PP*RRHO(J)**2                                00024170
      GO TO 2160                                                        00024180
2140  Z(J)=ZZ0/B(J)                                                     00024190
      ZZ01=ZZ0                                                          00024200
      ZZ0=ZZ0/B(J)                                                      00024210
      GO TO 2080                                                        00024220
2160  P(J)=PP                                                           00024230
      IF(SHIFT) ZZ0=ZZ01                                                00024240
2     CONTINUE                                                          00024250
C--RESTORE ORIG. FSAVE                                                  00024260
      CALL SWAP(-1)                                                     00024270
      FREQLL=FREQ                                                       00024280
      LOOPSL=ILOOPS                                                     00024290
      RETURN                                                            00024300
      END                                                               00024310

      SUBROUTINE SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)                    00024320
C--'MARQLOOPS' INITIALIZATION ROUTINE (CALLED ONCE BY MARQRT)           00024330
C  SUBZ IS CALLED BY MARQRT AFTER THE DATA Y(I),X(I,5) ARE READ--       00024340
C  SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL $INIT                  00024350
C  PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE...            00024360
C                                                                       00024370
C--PARAMETERS--                                                         00024380
C                                                                       00024390
C       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.                         00024400
C       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY             00024410
C               =-3 IF M=2 OR IOB.LT.5                                  00024420
C               =4 IF M=3 OR IOB=5                                      00024430
C               =-5 IF M=4 OR IOB=6                                     00024440
C               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES              00024450
C               PRINTED BY PGM MARQRT...                                00024460
C       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)                   00024470
C       TITLE=  ALPHA TITLE ARRAY READ IN BY PGM MARQRT.                00024480
C       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED                     00024490
C               0 IF ONLY UNIT 6 PRINT FILE USED.                       00024500
C                                                                       00024510
C--FOLLOWING CHARACTER STMT. ONLY FOR HONEYWELL MULTICS SYS:            00024520
      CHARACTER*5 TITLE(16)                                             00024530
      COMPLEX T0,T1,T2,CB,CB2,CB3,CA,ONESG1,TWOSG1,I0K0,I1K1,IKDIF,ZZ0  00024540
      REAL    Y(1),X(200,5),B(1),PRNT(1),EPS                            00024550
      REAL    K(10),D(9)                                                00024560
      COMMON/MODEL/K,D,MM                                               00024570
      COMMON/SHARE/FILL(4),X0,Y0,YY2,RHO,RHO2,FILL2,BB,FILL3,DEL,DEL2,  00024580
     & IREST(3)                                                         00024590
      COMMON/CTL/T0,T1,T2,CB,CB2,CB3,CA,ONESG1,TWOSG1,I0K0,I1K1,IKDIF,  00024600
     & ZZ0,AMP,FREQ,SIG1,H,EPS,IOB,M1,M21,ILOOPS,IMM                    00024610
      COMMON/RESIST/RRHO(10),HH(9),EEPS,RR,RR2,XJUNK,MMMM(4)            00024620
      NAMELIST/INIT/MM,Y0,H,IOB,EPS                                     00024630
      DATA ISUBZ/0/                                                     00024640
      IF(ISUBZ.NE.0) GO TO 10                                           00024650
C--PRESET                                                               00024660
      ISUBZ=1                                                           00024670
```

```
      MM=1                                                              00024680
      IOB=1                                                             00024690
      YO=0.0                                                            00024700
      EPS=.1E-5                                                         00024710
      H=0.0                                                             00024720
   10 READ(5,INIT)                                                      00024730
      WRITE(6,20) TITLE                                                 00024740
   20 FORMAT(21H1M A R Q L O O P S --,5X,16A5/)                          00024750
      IF(IOUT.EQ.1) WRITE(16,20) TITLE                                  00024760
      WRITE(6,30) IOB,MM,YO,H,EPS                                       00024770
      IF(IOUT.EQ.1)                                                     00024780
     1 WRITE(16,30) IOB,MM,YO,H,EPS                                     00024790
   30 FORMAT(7H IOB = ,I1,9X,5HMM = ,I3,8X,3HYO=,E12.5,11H    H=Z+H"=,   00024800
     1 E12.5/5H EPS=,E11.5)                                             00024810
C--TEST $INIT PARMS                                                     00024820
      IMM=MM                                                            00024830
      MM=IABS(MM)                                                       00024840
      IF(MM.LT.1.OR.MM.GT.10.OR.(YO.EQ.0.0.AND.IOB.LT.6).OR.            00024850
     4IOB.LT.1.OR.IOB.GT.6.OR.H.LT.0.0)                                 00024860
     5CALL ERRMSG(30HSOME $INIT PARMS OUT OF RANGE ,6,6,16)             00024870
C--TEST X(I, ) DATA FOR GIVEN IOB BEFORE PROCEEDING--                   00024880
   40 DO 70 I=1,N                                                       00024890
      IF(X(I,1).LE.0.0) CALL ERRMSG(                                    00024900
     121HSOME FREQ=X(I,1).LE.0,5,6,16)                                  00024910
      IF(IFIX(X(I,2)).LT.1.OR.IFIX(X(I,2)).GT.5) CALL ERRMSG(           00024920
     &   30HSOME LOOPS=X(I,2) OUT OF RANGE,6,6,16)                      00024930
      IF(IOB-5) 70,50,50                                                00024940
   50 IF(IFIX(X(I,3)).LT.1.OR.IFIX(X(I,3)).GT.5) CALL ERRMSG(           00024950
     140HSOME IOBS=X(I,3) OUT OF RANGE WHEN IOB>4,8,6,16)               00024960
   70 CONTINUE                                                          00024970
      IF(IMM.LT.0.AND.B(2*MM).EQ.0.0)                                   00024980
     & CALL ERRMSG(25HMM<0 & B(2*IABS(MM))=0.0 ,5,6,16)                 00024990
C--PRESET SOME GLOBAL CONSTANTS                                         00025000
      WRITE(6,90)                                                       00025010
      IF(IOUT.EQ.1) WRITE(16,90)                                        00025020
90    FORMAT(////18H PARAMETER ORDER--/)                                00025030
  100 M1=MM-1                                                           00025040
      M21=2*MM-1                                                        00025050
      WRITE(6,110) (I,I,I=1,MM)                                         00025060
      IF(IOUT.EQ.1) WRITE(16,110) (I,I,I=1,MM)                          00025070
  110 FORMAT(5X,I3,6X,6HSIGMA(,I3,1H))                                  00025080
      IF(MM.EQ.1) GO TO 132                                             00025090
      DO 120 I=1,M1                                                     00025100
      J=MM+I                                                            00025110
      IF(IOUT.EQ.1) WRITE(16,130) J,I                                   00025120
  120 WRITE(6,130) J,I                                                  00025130
  130 FORMAT(5X,I3,6X,6HTHICK(,I3,1H))                                  00025140
132   IF(IMM.GT.0) GO TO 140                                            00025150
      J=M21+1                                                           00025160
      WRITE(6,131) J,J                                                  00025170
131   FORMAT(5X,I3,10X,                                                 00025180
     & 2HB(,I3,35H) SHIFT PARAMETER IN B(2*|MM|)*Z/ZO)                  00025190
```

```
      IF(IOUT.EQ.1) WRITE(16,131) J,J                                   00025200
C--X(I,1)=FREQ, X(I,2)=LOOPS TYPE(IF IOB=5), X(I,3)=IOB TYPE(IF IOB>4), 00025210
C  X(I,M+1)=STD.DEV. (IF IWT=1)                                         00025220
C  NOTE-- M=2 REQUIRED IN PGM MARQRT WHEN IOB.LE.4, AND                 00025230
C  M=3 IS NECESSARY WHEN IOB=5...                                       00025240
C  ALSO, M=4 IS NECESSARY WHEN IOB=6...                                 00025250
  140 NPRNT=-3                                                          00025260
      IF(IOB.EQ.5) NPRNT=4                                              00025270
      IF(IOB.EQ.6) NPRNT=-5                                             00025280
      IF(IOB.LT.5) GO TO 150                                            00025290
      MMMM(1)=MM                                                        00025300
      MMMM(2)=M1                                                        00025310
      MMMM(3)=M21                                                       00025320
      EEPS=.001*EPS                                                     00025330
  150 RETURN                                                            00025340
      END                                                               00025350

      SUBROUTINE SUBEND(Y,X,B,K,N,TITLE,IOUT)                           00025360
C-- 'MARQLOOPS' TERMINATION ROUTINE (CALLED ONCE BY MARQRT)             00025370
C  (PARAMETERS SAME AS IN SUBROUTINE FCODE,PCODE, OR SUBZ)              00025380
C  B= FINAL SOLUTION VECTOR OBTAINED BY PGM MARQRT.                     00025390
C                                                                       00025400
C--FOLLOWING CHARACTER STMT. ONLY FOR HONEYWELL MULTICS SYS:            00025410
      CHARACTER*5 TITLE(16)                                             00025420
      REAL Y(1),X(200,5),B(1)                                           00025430
      WRITE(6,10) TITLE                                                 00025440
   10 FORMAT(21H1M A R Q L O O P S --,5X,16A5//                         00025450
     1 28H FINAL UNSCALED PARAMETERS--,10X,11HRESISTIVITY,11X,5HDEPTH/) 00025460
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                                  00025470
      MM=(K+1)/2                                                        00025480
      DO 30 I=1,MM                                                      00025490
      R=1.0/B(I)                                                        00025500
      WRITE(6,20) I,B(I),I,R                                            00025510
   20 FORMAT(5X,I3,4X,E16.8,2X,I3,1X,E16.8)                             00025520
      IF(IOUT.EQ.1) WRITE(16,20) I,B(I),I,R                             00025530
   30 CONTINUE                                                          00025540
      IF(K.LE.2) GO TO 52                                               00025550
      M2=MM+1                                                           00025560
      K1=K                                                              00025570
      IF(MOD(K,2).EQ.0) K1=K-1                                          00025580
      D=0.0                                                             00025590
      DO 50 I=M2,K1                                                     00025600
      D=D+B(I)                                                          00025610
      L=I-MM                                                            00025620
      WRITE(6,40) I,B(I),L,D                                            00025630
   40 FORMAT(5X,I3,4X,E16.8,24X,I3,1X,E16.8)                            00025640
      IF(IOUT.EQ.1) WRITE(16,40) I,B(I),L,D                             00025650
   50 CONTINUE                                                          00025660
52    IF(K1.EQ.K) GO TO 60                                              00025670
      WRITE(6,51) K,B(K)                                                00025680
51    FORMAT(5X,I3,4X,E16.8)                                            00025690
      IF(IOUT.EQ.1) WRITE(16,51) K,B(K)                                 00025700
```

```
60 RETURN                                                    00025710
   END                                                       00025720
```

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. Subprogram ERRMSG 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 MARQRT, arrays TITLE and FMT).

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

```
test2_13_x2
$parms n=36,m=3,k=4,
 sp=1,sy=2,iprt=-1,e=.001,
 b=.03,3,300,3$
(2e16.8,2f10.0)
$init mm=-2,y0=200,h=0,eps=.1e-5,iob=5$
```

file10

```
 0.20320306e+01  0.10000000e+01        1.        3.
 0.44297530e-01  0.10000000e+01        1.        4.
 0.20168934e+01  0.10000000e+01        3.        3.
 0.27006604e-01  0.10000000e+01        3.        4.
 0.20766954e+01  0.31622777e+01        1.        3.
 0.66449506e-01  0.31622777e+01        1.        4.
 0.20423448e+01  0.31622777e+01        3.        3.
 0.45585428e-01  0.31622777e+01        3.        4.
 0.21370684e+01  0.10000000e+02        1.        3.
 0.72628196e-01  0.10000000e+02        1.        4.
 0.20809538e+01  0.10000000e+02        3.        3.
 0.63852344e-01  0.10000000e+02        3.        4.
 0.21911114e+01  0.31622777e+02        1.        3.
 0.63235732e-01  0.31622777e+02        1.        4.
 0.21218432e+01  0.31622777e+02        3.        3.
 0.91763178e-01  0.31622777e+02        3.        4.
 0.22395336e+01  0.99999999e+02        1.        3.
 0.55342732e-01  0.99999999e+02        1.        4.
 0.21657660e+01  0.99999999e+02        3.        3.
 0.18011268e+00  0.99999999e+02        3.        4.
 0.23523172e+01  0.31622776e+03        1.        3.
 0.19828064e-01  0.31622776e+03        1.        4.
 0.22930036e+01  0.31622776e+03        3.        3.
 0.44294072e+00  0.31622776e+03        3.        4.
 0.25638946e+01  0.99999998e+03        1.        3.
-0.48680944e+00  0.99999998e+03        1.        4.
 0.28433346e+01  0.99999998e+03        3.        3.
 0.85697380e+00  0.99999998e+03        3.        4.
 0.14623892e+01  0.31622776e+04        1.        3.
-0.16139775e+01  0.31622776e+04        1.        4.
 0.37544432e+01  0.31622776e+04        3.        3.
 0.71218392e+00  0.31622776e+04        3.        4.
-0.77266146e-01  0.99999998e+04        1.        3.
-0.71643852e+00  0.99999998e+04        1.        4.
 0.40172924e+01  0.99999998e+04        3.        3.
 0.19771922e+00  0.99999998e+04        3.        4.
```

```
m a r q l o o p s --     test2_13_x2

iob = 5          mm =  -2         y0= 0.20000e+03     h=z+h'= 0.00000e+00
eps=0.10000e-05


parameter order--

      1       sigma(  1)
      2       sigma(  2)
      3       thick(  1)
      4           b(  4) shift parameter in b(2*|mm|)*z/z0
```

```
m a r q r t --     test2_13_x2

n =   36        k =   4         ip =   0        m =  3          gamcr=0.450e+02
del= 0.100e-04  modlam = 1      ff= 0.400e+01   t= 0.200e+01    e= 0.100e-02
tau= 0.100e-02  xl= 0.100e-01   zeta= 0.100e-30 ialt = 10       istop = 1
ivt = 0         ider = 0        iprt = -1       niter =   10    inon = 1
iout =  1       nprnt = 4       scalep = 1      sceley = 2

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

parameters     0.30000000e-01   0.30000000e+01   0.30000001e+03   0.30000000e+01


iter        phi                 s e              length       gamma        lambda
   1    0.33978403e+01    0.32585658e+00    0.000e+00    0.000e+00    0.100e-01

parameters     0.22252883e-01   0.19507815e+00   0.18799332e+03   0.20191569e+01


iter        phi                 s e              length       gamma        lambda
   2    0.58069640e-01    0.42599017e-01    0.282e+01    0.685e+02    0.100e-02

parameters     0.20068671e-01   0.19664699e+01   0.23848450e+03   0.20070843e+01


iter        phi                 s e              length       gamma        lambde
   3    0.65995735e-02    0.14360943e-01    0.233e+01    0.588e+02    0.100e-03

parameters     0.20013396e-01   0.17968894e+01   0.19435543e+03   0.20011275e+01


iter        phi                 s e              length       gamma        lambda
   4    0.19964231e-03    0.24977634e-02    0.224e+00    0.464e+02    0.100e-04

parameters     0.20000287e-01   0.19849703e+01   0.19993730e+03   0.20000511e+01


iter        phi                 s e              length       gamma        lambda
   5    0.15904880e-06    0.70500177e-04    0.103e+00    0.813e+02    0.100e-06

parameters     0.20000001e-01   0.19997391e+01   0.19999975e+03   0.20000013e+01


iter        phi                 s e              length       gamma        lambda
   6    0.13717088e-09    0.20704082e-05    0.742e-02    0.107e+02    0.100e-08

-epsilon test
   6 iterations
```

m a r q r t -- test2_13_x2

parameters 0.20000001e-01 0.19997391e+01 0.19999975e+03 0.20000013e+01

-unscaled-

i	obs.y(i)	cal	res	%res.err	x(i,1)	x(i,2)	x(i,3)	x(i,4)	x(i,5)
1	0.203203e+01	0.203203e+01	-0.274e-05	-0.134930e-03	0.100000e+01	0.100000e+01	0.300000e+01	0.000000e+00	
2	0.442975e-01	0.442950e-01	0.250e-05	0.564428e-02	0.100000e+01	0.100000e+01	0.400000e+01	0.000000e+00	
3	0.201689e+01	0.201689e+01	0.566e-06	0.280751e-04	0.100000e+01	0.300000e+01	0.300000e+01	0.000000e+00	
4	0.270066e-01	0.270142e-01	-0.762e-05	-0.282077e-01	0.100000e+01	0.300000e+01	0.400000e+01	0.000000e+00	
5	0.207670e+01	0.207669e+01	0.411e-05	0.198042e-03	0.316228e+01	0.100000e+01	0.300000e+01	0.000000e+00	
6	0.664495e-01	0.664482e-01	0.130e-05	0.196221e-02	0.316228e+01	0.100000e+01	0.400000e+01	0.000000e+00	
7	0.204234e+01	0.204234e+01	0.176e-05	0.860941e-04	0.316228e+01	0.300000e+01	0.300000e+01	0.000000e+00	
8	0.455854e-01	0.455907e-01	-0.530e-05	-0.116194e-01	0.316228e+01	0.300000e+01	0.400000e+01	0.000000e+00	
9	0.213707e+01	0.213706e+01	0.370e-05	0.172924e-03	0.100000e+02	0.100000e+01	0.300000e+01	0.000000e+00	
10	0.726282e-01	0.726296e-01	-0.141e-05	-0.194395e-02	0.100000e+02	0.100000e+01	0.400000e+01	0.000000e+00	
11	0.208095e+01	0.208095e+01	0.298e-05	0.143215e-03	0.100000e+02	0.300000e+01	0.300000e+01	0.000000e+00	
12	0.638523e-01	0.638533e-01	-0.956e-06	-0.149646e-02	0.100000e+02	0.300000e+01	0.400000e+01	0.000000e+00	
13	0.219111e+01	0.219111e+01	0.122e-05	0.557660e-04	0.316228e+02	0.100000e+01	0.300000e+01	0.000000e+00	
14	0.632357e-01	0.632385e-01	-0.281e-05	-0.444465e-02	0.316228e+02	0.100000e+01	0.400000e+01	0.000000e+00	
15	0.212184e+01	0.212184e+01	0.191e-05	0.898912e-04	0.316228e+02	0.300000e+01	0.300000e+01	0.000000e+00	
16	0.917632e-01	0.917651e-01	-0.196e-05	-0.214042e-02	0.316228e+02	0.300000e+01	0.400000e+01	0.000000e+00	
17	0.223953e+01	0.223953e+01	-0.387e-06	-0.172996e-04	0.100000e+03	0.100000e+01	0.300000e+01	0.000000e+00	
18	0.553427e-01	0.553454e-01	-0.264e-05	-0.476217e-02	0.100000e+03	0.100000e+01	0.400000e+01	0.000000e+00	
19	0.216577e+01	0.216577e+01	-0.298e-07	-0.137606e-05	0.100000e+03	0.300000e+01	0.300000e+01	0.000000e+00	
20	0.180113e+00	0.180115e+00	-0.192e-05	-0.106413e-02	0.100000e+03	0.300000e+01	0.400000e+01	0.000000e+00	
21	0.235232e+01	0.235232e+01	-0.197e-05	-0.836176e-04	0.316228e+03	0.100000e+01	0.300000e+01	0.000000e+00	
22	0.198281e-01	0.198292e-01	-0.113e-05	-0.571356e-02	0.316228e+03	0.100000e+01	0.400000e+01	0.000000e+00	
23	0.229300e+01	0.229301e+01	-0.191e-05	-0.831812e-04	0.316228e+03	0.300000e+01	0.300000e+01	0.000000e+00	
24	0.442941e+00	0.442942e+00	-0.137e-05	-0.309500e-03	0.316228e+03	0.300000e+01	0.400000e+01	0.000000e+00	
25	0.256389e+01	0.256390e+01	-0.235e-05	-0.918283e-04	0.100000e+04	0.100000e+01	0.300000e+01	0.000000e+00	
26	-0.486809e+00	-0.486810e+00	0.663e-06	0.136214e-03	0.100000e+04	0.100000e+01	0.400000e+01	0.000000e+00	
27	0.284333e+01	0.284334e+01	-0.215e-05	-0.754665e-04	0.100000e+04	0.300000e+01	0.300000e+01	0.000000e+00	
28	0.856974e+00	0.856974e+00	-0.462e-06	-0.539031e-04	0.100000e+04	0.300000e+01	0.400000e+01	0.000000e+00	
29	0.146239e+01	0.146239e+01	-0.417e-06	-0.285309e-04	0.316228e+04	0.100000e+01	0.300000e+01	0.000000e+00	
30	-0.161398e+01	-0.161398e+01	0.715e-06	0.443163e-04	0.316228e+04	0.100000e+01	0.400000e+01	0.000000e+00	
31	0.375444e+01	0.375445e+01	-0.232e-05	-0.619154e-04	0.316228e+04	0.300000e+01	0.300000e+01	0.000000e+00	
32	0.712184e+00	0.712184e+00	-0.432e-06	-0.606772e-04	0.316228e+04	0.300000e+01	0.400000e+01	0.000000e+00	
33	-0.772661e-01	-0.772657e-01	-0.468e-06	-0.605086e-03	0.100000e+05	0.100000e+01	0.300000e+01	0.000000e+00	
34	-0.716439e+00	-0.716440e+00	0.104e-05	0.145592e-03	0.100000e+05	0.100000e+01	0.400000e+01	0.000000e+00	
35	0.401729e+01	0.401730e+01	-0.280e-05	-0.697339e-04	0.100000e+05	0.300000e+01	0.300000e+01	0.000000e+00	
36	0.197719e+00	0.197719e+00	-0.708e-07	-0.357985e-04	0.100000e+05	0.300000e+01	0.400000e+01	0.000000e+00	

-unscaled partials-

1	0.17262209e+00	0.13879137e-01	-0.16549475e-03	0.10160160e+01
2	0.60665254e+00	0.10006387e-01	-0.32389581e-03	0.22147500e-01
3	0.10490855e-01	0.75711063e-02	-0.91090482e-04	0.10084457e+01
4	0.10753983e+00	0.63502372e-02	-0.20836427e-03	0.13507103e-01
5	0.72469123e+00	0.23571716e-01	-0.51807294e-03	0.10383449e+01
6	0.13040773e+01	0.56634533e-02	-0.55442062e-03	0.33224079e-01
7	0.40176390e-01	0.14039399e-01	-0.30637547e-03	0.10211708e+01
8	0.31215058e+00	0.50891716e-02	-0.39616734e-03	0.22795347e-01
9	0.20274968e+01	0.24269018e-01	-0.10954997e-02	0.10685316e+01
10	0.19610915e+01	-0.47270534e-02	-0.59115513e-03	0.36314779e-01
11	0.12861678e+00	0.16844848e-01	-0.72749815e-03	0.10404747e+01
12	0.88718758e+00	-0.70120920e-03	-0.51899821e-03	0.31926628e-01
13	0.37032560e+01	0.15270643e-01	-0.15897797e-02	0.10955543e+01
14	0.17909158e+01	-0.10315294e-01	-0.26104449e-03	0.31619250e-01
15	0.41409716e+00	0.12984165e-01	-0.12222745e-02	0.10609199e+01

16	0.25462214e+01	-0.59076946e-02	-0.38391284e-03	0.45882540e-01
17	0.53083103e+01	0.57766199e-02	-0.17053233e-02	0.11197662e+01
18	0.48543589e+00	-0.96747704e-02	0.43902761e-03	0.27672665e-01
19	0.18793786e+01	0.63560304e-02	-0.15235095e-02	0.10823823e+01
20	0.72996366e+01	-0.73135994e-02	0.13637500e-03	0.90057240e-01
21	0.10295393e+02	-0.17405788e-02	-0.72366689e-03	0.11761588e+01
22	-0.61231194e+01	-0.54206447e-02	0.14125811e-02	0.99145918e-02
23	0.11141708e+02	-0.16404022e-03	-0.10015492e-02	0.11465020e+01
24	0.17496983e+02	-0.52561966e-02	0.10804311e-02	0.22147090e+00
25	-0.15922312e+01	-0.13174144e-02	0.80579026e-03	0.12819476e+01
26	-0.46830600e+02	0.97893859e-03	0.11729535e-03	-0.24340489e+00
27	0.38243128e+02	-0.16146865e-02	0.58018177e-03	0.14216674e+01
28	0.12034123e+02	0.29349094e-04	0.55659075e-03	0.42843684e+00
29	-0.93423745e+02	0.75868838e-04	-0.13913099e-03	0.73119432e+00
30	-0.17336390e+02	-0.14711669e-03	0.44808414e-04	-0.80698859e+00
31	0.30047072e+02	0.13970887e-03	-0.71954482e-04	0.18772215e+01
32	-0.23061813e+02	0.24896946e-04	-0.10289584e-03	0.35609194e+00
33	-0.96192133e+01	0.17277282e-05	-0.39792033e-05	-0.38632815e-01
34	0.56936467e+02	-0.17008476e-05	-0.73628934e-07	-0.35821955e+00
35	-0.14965304e+01	0.56257342e-06	0.57040427e-06	0.20086462e+01
36	-0.12977272e+02	0.10812326e-05	-0.18312836e-05	0.98859582e-01

-unscaled-

phi	s e	lambda
0.21193404e-09	0.25735071e-05	0.100e-08

ptp inverse

1	0.55358493e-04	0.11264848e-01	-0.52153011e-01	-0.25057514e-03
2	0.11264848e-01	0.59046687e+03	0.33232466e+04	-0.19048575e+01
3	-0.52153010e-01	0.33232466e+04	0.83191006e+05	0.12955757e+02
4	-0.25057514e-03	-0.19048576e+01	0.12955756e+02	0.52648876e-01

parameter correlation matrix

1	1.0000	0.0623	-0.0243	-0.1468
2	0.0623	1.0000	0.4742	-0.3416
3	-0.0243	0.4742	1.0000	0.1958
4	-0.1468	-0.3416	0.1958	1.0000

parameter	std	one - parameter		support plane		
	error	lower	upper	lower	upper	std.error/parm
1	0.19147739e-07	0.19999963e-01	0.20000039e-01	0.19999925e-01	0.20000078e-01	0.95738687e-06
2	0.62534997e-04	0.19996141e+01	0.19998642e+01	0.19994890e+01	0.19999893e+01	0.31271577e-04
3	0.74227281e-03	0.19999826e+03	0.20000123e+03	0.19999678e+03	0.20000272e+03	0.37113687e-05
4	0.59050002e-06	0.20000001e+01	0.20000025e+01	0.19999989e+01	0.20000037e+01	0.29524981e-06

```
m a r q l o o p s --     test2_13_x2

final unscaled parameters--          resistivity          depth

      1      0.20000001e-01     1   0.49999997e+02
      2      0.19997391e+01     2   0.50006522e+00
      3      0.19999975e+03                          1   0.19999975e+03
      4      0.20000013e+01
```