U.S. Department of the Interior
Geological Survey

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

Program MARQ_TRANS_HCLOOP: Marquardt inversion of transient horizontal coplanar loop soundings

by

Walter L. Anderson

and

James Kauahikaua

OPEN-FILE REPORT 79-773

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 10

DATA MATRIX NOTES 12

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING 12

SPECIAL OBJECT FORMAT PHRASES 12

MULTICS OPERATING INSTRUCTIONS 13

ERROR MESSAGES 13

REFERENCES 14

Appendix 1.-- Source listing 16

Source availability 16

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 authors regarding accuracy or proper functioning of this program on all computer systems.

---

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

By Walter L. Anderson and James Kauahikaua

INTRODUCTION.

Program MARQ_TRANS_HCLOOP is a general-purpose program for inversion of horizontal coplanar loop-loop transient 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 transient sounding data. The associated forward solution for computing the transient response for a horizontal coplanar loop system is taken from Anderson (1979a). Digital filters developed by Anderson (1975, 1979b) are used to evaluate Fourier and Hankel transform integrals, respectively.

The following program options are currently available:

(1) Inversion of horizontal coplanar loop-loop transient soundings (after system deconvolution, stacking, etc.) using a ground loop-loop configuration for a maximum of 10-layer models. See Frischknecht (1967) for details on the horizontal coplanar loop-loop formulation for 2-layer models.
(2) Transient impulse or step response soundings due to a system impulse or step driving current. See Kauahikaua and Anderson (1977) for more details and definitions on transient calculations (for wire-loop systems).
(3) Inclusion of an additional transient amplitude shift parameter in the least squares when the correct normalization factor is unknown.
(4) Scaling parameter and observation spaces to constrain the solution space and to reduce round-off effects.
(5) Weighted observations.
(6) Holding certain parameters fixed (constrained).
(7) 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 MARQ_TRANS_HCLOOP 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 MARQ_TRANS_HCLOOP is:

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

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

PROGRAM FILES.

file05	title, input parameters $parms, object format (for reading the 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 06,13,16, or 20. 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 MARQ_TRANS_HCLOOP.
file16	output master print-type disk file--contains maximum printable output (if parameter iout=1).
file20	output scratch disk file (if parameter ider=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 equal to 2*mm, where $init parameter mm>0 is the number of layers in the model.
(cref: $init parameter mm and $parms n,b).

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

m= Number of independent variables (m=1 required) given in the data matrix (y(i),x(i,j),j=1,m), i=1,n.
(cref: $parms iwt and DATA MATRIX NOTES below).

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 6,13,16, or 20. 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).

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 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=1. 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. When ider=1, file20 is used for scratch disk storage.
(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 $parms parameter sp(=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 in 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 MARQ_TRANS_HCLOOP 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's 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 MARQ_TRANS_HCLOOP, 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 MARQ_TRANS_HCLOOP.
(cref: $parms b,k,n)

b()= Array of initial guesses for all k-parameters. These values must be supplied greater than zero for program MARQ_TRANS_HCLOOP (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

b(2*mm)>0 is the estimated transient amplitude shift parameter used in the model as b(2*mm)*transient. Note that the theoretical loop-loop transient impulse response ($init parameter istep=0) limit approaches 9.0 as time approaches 0.
(cref: $parms k,ip,ib and $init mm,istep).

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):

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

y0= Transmitter-receiver loop-loop separation, where y0>0 meters.

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

istep= 0 (default) for transient impulse response sounding, which corresponds to the time-derivative of Hz when the source loop uses a system step driving current (or Hz when the source loop uses a system impulse driving current).
= 1 for transient step response sounding, which corresponds to Hz when the source loop uses a system step driving current. Note istep=1 is generally used when the transient data is obtained using a SQUID or cryogenic magnetometer. When istep=1, $init parameter ihalf=0 must be used.
(cref: $init ihalf)

ihalf= 0 (default) to use the analytical transient for the halfspace impulse (istep=0) or step response (istep=1) separately from the secondary field numerical integrated transient (note ihalf=0 is usually more accurate than ihalf=1).
= 1 to include the halfspace field with the secondary field to evaluate the numerical integrated transient response. Note ihalf=1 can only be used when istep=0, otherwise, an error condition is given.
(cref: $init istep)

b0 = .01 (default) is the lower induction number limit for which the mutual coupling Z/Z0 frequency response approaches 1.0 for b<b0. b0 must be given as a power of 10**-n.

bm = 100 (default) is the upper induction number limit for which the mutual coupling Z/Z0 frequency response approaches 0.0 for b>bm. bm must be given as a power of 10**n.

Note: b0=.01 and bm=100 are usually adequate for most applications. However, for more accuracy, b0<=.001 and/or bm>=1000 may be used. If parameter nb=0 is used, it is recommended that b0<.01 and bm>100 be used.
(cref: $init nb).

nb = 10 (default) represents the number of induction number points per decade (log cycle) to evaluate the pre-splined frequency response function Z/Z0.

In general, 5<=nb<12 is usually adequate for most applications (nb<5 is not recommended for accuracy reasons). If nb=0 or nb>11 is specified, then a direct mode of evaluating the frequency function is used via a lagged-convolution cosine or sine transform algorithm. When nb=0 or nb>=12, parameters b0<.01 and bm>100 should also be changed. Note nb=0 or nb>=12 is the most accurate method, but more time-consuming than using nb<12. (cref: $init b0,bm).

$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=1. 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 as previously defined. The required data matrix for program MARQ_TRANS_HCLOOP is:

1. y(i)= i-th observation of the transient response (may be shifted or unnormalized) corresponding to time given in x(i,1).
2. x(i,1)= i-th time (x(i,1)>0.0 seconds), where x(i,1)>x(i-1,1), for i=2,3,...,n.
3. x(i,2)= standard deviation of observation i (include only if iwt=1).

EXAMPLE OF INPUT PARAMETERS AND DATA ORDERING.

```
example
$parms n=60,k=4,m=1,iprt=-1,sp=1,sy=2,ialt=5,iwt=1,
 e=.005,b=.1,.2,100,1.5$
(3f10.0)
4.98        .08         .01
5.36        .12         .02
4.85        .16         .04
--(etc for 57 more observations)--
$init mm=2,y0=1000,nb=6,eps=.0001$
```

SPECIAL OBJECT FORMAT PHRASES.

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,t1,2f10.0) 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 the example 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 MARQ_TRANS_HCLOOP by typing:
   marq_trans_hcloop

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 MARQ_TRANS_HCLOOP. Also, file13 and/or file20 (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 MARQ_TRANS_HCLOOP.

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., 1975, Improved digital filters for evaluating Fourier and Hankel transform integrals: U.S. Geol. Sur. Rept. USGS-GD-75-012, 223p. available from U.S. Dept. Comm. NTIS, Springfield, Va. 22161 as Rept. PB-242-800/1WC.

----------, 1979a, Programs TRANS_HCLOOP and TRANS_HZWIRE: Calculation of transient horizontal coplanar loop soundings and transient wire-loop soundings: U.S. Geol. Survey Open-File Rept. 79-590, 46 p.

----------, 1979b, Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics (in press).

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.

Kauahikaua, J., and Anderson, W. L., 1977, Calculation of standard transient and frequency sounding curves for a horizontal wire source of arbitrary length: U.S. Geol. Sur. Rept. USGS-GD-77-007, 63p. available from U.S. Dept. Comm. NTIS, Springfield, Va. 22161 as Rept. PB-274-119.

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--MARQ_TRANS_HCLOOP: MARQRT INVERSION OF HZ TRANSIENT DATA (11/2/78) 00000010
      SUBROUTINE REED(IUNIT,A,NA,B,NB,*)                              00000130
      SUBROUTINE RITE(IUNIT,A,NA,B,NB)                                00000220
      SUBROUTINE MARQRT(FCODE,PCODE,SUBZ,SUBEND)                      00000300
      SUBROUTINE GJR (A,N,EPS,MSING)                                  00010060
      SUBROUTINE UNSCAL(BIN,BOUT,SCALEP)                              00010720
      REAL FUNCTION ASINH(X)                                          00010930
      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                 00011010
      SUBROUTINE RECUR1(G,V1,F1)                                      00011240
      SUBROUTINE RECURF(G,DEL,SIG1,V1,F1,PF1,JJ)                      00011550
      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)                       00012170
      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)                            00013370
        REAL FUNCTION RLAGF0(X,FUN,TOL,L,NEW)                         00013590
        REAL FUNCTION RLAGF1(X,FUN,TOL,L,NEW)                         00015980
      COMPLEX FUNCTION FG2(G)                                         00018340
      COMPLEX FUNCTION PFBJG(G)                                       00018480
      COMPLEX FUNCTION PFBJG2(G)                                      00018660
      REAL FUNCTION HCLOOP(B2)                                        00018730
      REAL FUNCTION HSLOOP(B2)                                        00019110
      REAL FUNCTION PHC(B2)                                           00019190
      REAL FUNCTION PHS(B2)                                           00019730
      REAL FUNCTION ERF(X)                                            00019820
      REAL FUNCTION RFLAGS(N,FUN,TOL,T0,TM,T,NEW)                     00019950
      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                     00020360
      SUBROUTINE MARQ_TRANS_HCLOOP_FCODE(Y,X,B,PRNT,F,IN,IDER)        00023780
      SUBROUTINE MARQ_TRANS_HCLOOP_PCODE(P,X,B,PRNT,F,IN,IP,IB)       00024710
      SUBROUTINE MARQ_TRANS_HCLOOP_SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)00025510
      SUBROUTINE MARQ_TRANS_HCLOOP_SUBEND(Y,X,B,K,N,TITLE,IOUT)       00026380
```

Source Availability

The current version of the source code may be obtained by writing directly to the authors. A magnetic tape copy of the source code will be sent to requestors to be copied and returned to the authors. 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--MARQ_TRANS_HCLOOP: MARQRT INVERSION OF HZ TRANSIENT DATA (11/2/78)   00000010
C ** HONEYWELL MULTICS VERSION **                                       00000020
C  FOR GROUND HORIZONTAL COPLANAR LOOPS                                 00000030
C                                                                       00000040
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.           00000050
C                                                                       00000060
      EXTERNAL MARQ_TRANS_HCLOOP_FCODE,MARQ_TRANS_HCLOOP_PCODE,         00000070
     & MARQ_TRANS_HCLOOP_SUBZ,MARQ_TRANS_HCLOOP_SUBEND                  00000080
      CALL MARQRT(MARQ_TRANS_HCLOOP_FCODE,MARQ_TRANS_HCLOOP_PCODE,      00000090
     & MARQ_TRANS_HCLOOP_SUBZ,MARQ_TRANS_HCLOOP_SUBEND)                 00000100
      STOP                                                              00000110
      END                                                               00000120

      SUBROUTINE REED(IUNIT,A,NA,B,NB,*)                                00000130
C--DYNAMIC ARRAYS A(NA) AND B(NB) VECTOR READ(IUNIT,END=$N) A,B          00000140
C  FOR QUICKER INPUT.                                                   00000150
C                                                                       00000160
      DIMENSION A(NA),B(NB)                                             00000170
      READ(IUNIT,END=1) A,B                                             00000180
      RETURN                                                            00000190
1     RETURN (1)                                                        00000200
      END                                                               00000210

      SUBROUTINE RITE(IUNIT,A,NA,B,NB)                                  00000220
C--DYNAMIC ARRAYS A(NA) AND B(NB) VECTOR WRITE(IUNIT) A,B                00000230
C  FOR QUICKER OUTPUT.                                                  00000240
C                                                                       00000250
      DIMENSION A(NA),B(NB)                                             00000260
      WRITE(IUNIT) A,B                                                  00000270
      RETURN                                                            00000280
      END                                                               00000290

      SUBROUTINE MARQRT(FCODE,PCODE,SUBZ,SUBEND)                        00000300
C--(MARQRT)-- GENERAL MARQUARDT NONLINEAR LEAST SQUARES-- 7/11/78.      00000310
C** HONEYWELL MULTICS VERSION **                                        00000320
C  SUBPROGRAM MARQRT IS TO BE LINKED/LOADED WITH USER WRITTEN           00000330
C  SUBROUTINES (FCODE,PCODE,SUBZ, AND SUBEND) FOR                       00000340
C  SPECIFIC NONLINEAR PROBLEM TO BE SOLVED.                             00000350
C                                                                       00000360
C--THE USER MUST DECLARE THE CALLING PARAMETERS FCODE,PCODE,            00000370
C  SUBZ,SUBEND (ANY DESIRED NAMES MAY BE USED) AS EXTERNAL IN           00000380
C  MAIN CALLING PROGRAM; E.G.,                                          00000390
C                                                                       00000400
C       EXTERNAL FCODE,PCODE,SUBZ,SUBEND                                00000410
C       CALL MARQRT(FCODE,PCODE,SUBZ,SUBEND)                            00000420
C       STOP                                                            00000430
C       END                                                             00000440
C                                                                       00000450
C--THIS IS A MODIFIED VERSION OF 'IBM SHARE PROGRAM NO. 1428'.          00000460
C *** MODIFIED BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY,DENVER, COLORADO 00000470
C  FOR NAMELIST INPUT, IMPROVED ESTIMATED DERIVATIVES,                  00000480
C  MODIFIED MARQUARDT LAMBDA DETERMINATION,                             00000490
```

```
C  DATA AND PARAMETER SCALING, WEIGHTED OBSERVATIONS, AND                  00000500
C  OTHER CHANGES--ALL DONE IN SINGLE-PRECISION FOR THE                     00000510
C *** HONEYWELL MULTICS SYSTEM ***                                         00000520
C                                                                          00000530
C--SEE SHARE PROGRAM NO. 1428 AND/OR COMMENTS IN SUBPROGRAMS FCODE,        00000540
C  PCODE,SUBZ AND SUBEND BELOW, FOR DETAILS ON CODING THE                  00000550
C  REQUIRED SUBROUTINES FCODE,PCODE,SUBZ, AND SUBEND.                      00000560
C                                                                          00000570
C--OPERATING NOTE FOR HONEYWELL MULTICS SYSTEM:  $$$$$$$$$$$$$$$$$$$$    00000580
C                                                                          00000590
C  TO OBTAIN ON-LINE (INTERACTIVE) PRINTING ON UNIT 6 AND                  00000600
C  DEFERRED PRINTING ON UNIT 16, USE MULTICS RUN.EC, I.E.,                 00000610
C  'RUN &1' OR 'RUN_EO &1' AND DPRINT '&1.FILE16.LIST' AFTER RUN.          00000620
C                                                                          00000630
C--FOLLOWING CHARACTER STATEMENTS ONLY FOR MULTICS SYSTEM:                 00000640
      CHARACTER*5 TITLE                                                    00000650
      CHARACTER*4 FMT                                                      00000660
      INTEGER SCALEP,SCALEY, SP,SY                                         00000670
      DIMENSION FMT(18),PRNT(5),SPRNT(5),TITLE(16)                         00000680
      DIMENSION BS(20),DB(20),BA(20),G(20),IB(19),SA(20),P(20)             00000690
      DIMENSION A(20,20),B(20), BINV(20)                                   00000700
      DIMENSION X(200,5),Y(200),WT(200)                                    00000710
      DIMENSION XNU(5),SS(4)                                               00000720
      EQUIVALENCE (X(1,5),WT(1)),(IOUT,IFSS1),(IDER,IWS2),                 00000730
     1 (IPRT,IWS3),(NITER,IWS4),(INON,IWS6),(SP,SCALEP),(SY,SCALEY)        00000740
C============================================================              00000750
      NAMELIST/PARMS/N,K,IP,M,IALT,IDER,IPRT,NITER,INON,NPRNT,             00000760
     1 IB,FF,T,E,TAU,XL,GAMCR,DEL,ZETA,B,IOUT,IWT,ISTOP,                   00000770
     2 SCALEP,SCALEY,MODLAM, SP,SY                                         00000780
C============================================================              00000790
      DATA XNU/1.33,1.78,3.16,10.,100./                                    00000800
C                                                                          00000810
C     MAX NO OF PARAMETERS IS K=20  (NOTE: K=N IS ALLOWED)                 00000820
C     MAX NO OF IND VARS IS M=4                                            00000830
C     MAX NO OF OBSERVATIONS IS N=200                                      00000840
C     INTERNAL #IWHER# SWITCH USAGE--                                      00000850
C                  IWHER =-1 MEANS INITIALIZE VIA SUBROUTINE SUBZ.         00000860
C                  IWHER = 0 MEANS START NEW PROBLEM OR END RUN            00000870
C                  IWHER = 1 MEANS GET P(S) AND F                          00000880
C                  IWHER GREATER THAN 1 MEANS GET F ONLY                   00000890
C--FOLLOWING CALL TO SUPPRESS EXP-UNDERFLOW MESSAGES                       00000900
C  FOR THE DEC-10 AND OTHER SYSTEMS:  $$$$$$$$$$$$$$$$$$$$               00000910
C//        CALL ERRSET(0)                                                  00000920
C** FOR THE HONEYWELL MULTICS SYSTEM, USE (INSTEAD) THE FOLLOWING:        00000930
C   IO DETACH ERROR_OUTPUT                                                 00000940
C   IO ATTACH ERROR_OUTPUT DISCARD_                                        00000950
C  (OR-- ON USGS SYS, USE SET_UFL -OFF)                                    00000960
C**                                                                        00000970
C--PRESET GLOBAL PARMS (SOME MAY BE OVERRIDDEN BY $PARMS READ-IN)         00000980
      IP=0                                                                 00000990
      N=0                                                                  00001000
      K=0                                                                  00001010
```

```
      M=0                                                                00001020
      NPRNT=0                                                            00001030
      MODLAM=1                                                           00001040
      ISTOP=1                                                            00001050
      IWT=0                                                              00001060
      IALT=10                                                            00001070
      IOUT=1                                                             00001080
      IDER=0                                                             00001090
      IPRT=0                                                             00001100
      MITER=10                                                           00001110
      INON=1                                                             00001120
      LSCALP=0                                                           00001130
      LSCALY=0                                                           00001140
      FF=4.0E0                                                           00001150
      E=.00005E0                                                         00001160
      TAU=.001E0                                                         00001170
      T=2.0E0                                                            00001180
      DEL=.00001E0                                                       00001190
      ZETA=.1E-30                                                        00001200
      GAMCR=45.0E0                                                       00001210
C                                                                        00001220
   10 GAMMA=0.E0                                                         00001230
      SCALEP=LSCALP                                                      00001240
      SCALEY=LSCALY                                                      00001250
      XLL=0.E0                                                           00001260
      SE=0.0                                                             00001270
      NITER=MITER                                                        00001280
   20 IWHER=0                                                            00001290
      ISS=1                                                              00001300
      INU=4                                                              00001310
      XNUFAC=10.0                                                        00001320
      GO TO 150                                                          00001330
   30 CONTINUE                                                           00001340
      IF (IWHER.GT.0) GO TO 100                                          00001350
      IF (IWHER.EQ.0) GO TO 240                                          00001360
C=================================================================       00001370
C     INITIALIZATION (IWHER=-1, IFSS1=IOUT)                              00001380
      CALL SUBZ (Y,X,BINV,PRNT,NPRNT,N,TITLE,IFSS1)                      00001390
C     ********************************************                       00001400
      IPRNT=NPRNT-1                                                      00001410
      IF(NPRNT.LT.0) IPRNT=IABS(NPRNT)-2                                 00001420
C                                                                        00001430
C--NOTE: IPRNT IS A SPECIAL INDEX USED IN SCALEY=2 CASES                 00001440
C  TO MIX LOG OR ASINH TYPE SCALING WHEN ABS(X(I,IPRNT))=1. OR NOT 1.    00001450
C  RESPECTIVELY, AND ONLY WHEN IPRNT.GT.1                                00001460
      NPRNT=IABS(NPRNT)                                                  00001470
      IF(SCALEY.EQ.0) GO TO 90                                           00001480
      DO 80 I=1,N                                                        00001490
      IF(SCALEY-1) 90,40,60                                              00001500
   40 IF(Y(I).LE.0.)CALL ERRMSG(30HSOME Y(I).LE.0 AND SCALEY=1...,       00001510
     1 6,6,16)                                                           00001520
   50 Y(I)=ALOG(Y(I))                                                    00001530
```

```
      GO TO 80                                                          00001540
   60 IF(IPRNT.LE.1) GO TO 70                                           00001550
      IF(ABS(X(I,IPRNT)).NE.1.0) GO TO 70                               00001560
      IF(Y(I).LE.0.)                                                    00001570
     1CALL ERRMSG(50HSOME Y(I).LE.0 WHEN ABS(X(I,IPRNT))=1 AND SCALEY=2,00001580
     2 10,6,16)                                                         00001590
      GO TO 50                                                          00001600
   70 Y(I)=ASINH(Y(I))                                                  00001610
   80 CONTINUE                                                          00001620
   90 CONTINUE                                                          00001630
      IF (IBOUT.EQ.0) GO TO 150                                         00001640
      GO TO 20                                                          00001650
  100 CONTINUE                                                          00001660
C=======================================================              00001670
C                  COMPUTE F VIA SUBR. FCODE                          00001680
C                  NPRNT IS THE NO OF OTHER WORDS TO BE PRINTED       00001690
C                  THE WORDS TO BE PRINTED ARE IN PRNT(1)...PRNT(5)   00001700
C--CALL FCODE FOR CURRENT BINV AND I-TH OBSERVATION (IFSS2=IDER)       00001710
      CALL FCODE(Y,X,BINV,PRNT,F,I,IFSS2)                               00001720
C     ************************************************               00001730
      FINV=F                                                            00001740
      IF(SCALEY-1) 140,110,120                                          00001750
  110 F=ALOG(F)                                                         00001760
      GO TO 140                                                         00001770
  120 IF(IPRNT.LE.1) GO TO 130                                          00001780
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 110                              00001790
  130 F=ASINH(F)                                                        00001800
  140 CONTINUE                                                          00001810
      IF (IWHER.NE.1) GO TO 150                                         00001820
      IF (IFSS2.NE.0) GO TO 150                                         00001830
C=======================================================              00001840
C                  COMPUTE P(J)=DF/DB VIA SUBR PCODE FOR J=1,K.       00001850
C                      USING X(I,L) AND B(J)                          00001860
C--CALL PCODE FOR CURRENT BINV,FINV AND I-TH OBSERVATION               00001870
      CALL PCODE(P,X,BINV,PRNT,FINV,I,IP,IB)                            00001880
C     ************************************************               00001890
C                THIS IS GENERAL #IWHER# SWITCH                       00001900
  150 CONTINUE                                                          00001910
      IF (IWHER.LT.0) GO TO 320                                         00001920
      IF (IWHER.EQ.0) GO TO 160                                         00001930
C              1  2   3   4   5                                        00001940
      GO TO (490,1560,530,580,590), IWHER                               00001950
C                  READ FIRST CARD OF NEXT CASE                       00001960
  160  ITCT=0                                                           00001970
      IBOUT=0                                                           00001980
C=======================================================              00001990
C  READ $PARMS --$                                                     00002000
C--ALWAYS PRESET XL=.01 (MAY BE OVERRIDDEN BY $$PARMS READ-IN)         00002010
C  AND CLEAR B(I),I=1,20 TO FORCE INITIALIZATION...                    00002020
      XL=.01                                                            00002030
      DO 170 I=1,20                                                     00002040
  170 B(I)=0.E0                                                         00002050
```

```
      READ(5,180) TITLE                                                 00002060
  180 FORMAT(16A5)                                                      00002070
      READ(5,PARMS)                                                     00002080
C--TEST $PARMS                                                          00002090
      IF(N.GT.200.OR.K.GT.20.OR.M.GT.4.OR.IWT.GT.1.OR.IP.GT.19.OR.      00002100
     1 IALT.EQ.6.OR.IALT.EQ.13.OR.IALT.EQ.16.OR.                        00002110
     2 N.LT.1.OR.K.LT.1.OR.M.LT.1.OR.IWT.LT.0.OR.IP.LT.0.OR.            00002120
     3 SCALEY.LT.0.OR.SCALEY.GT.2.OR.SCALEP.LT.0.OR.SCALEP.GT.2.OR.     00002130
     4 N.LT.K) CALL ERRMSG(30HSOME $PARMS OUT OF RANGE..     ,6,6,16)   00002140
      DO 210 I=1,K                                                      00002150
      IF(B(I).EQ.0.E0) CALL ERRMSG(20HSOME B(I) = 0.0      ,4,6,16)     00002160
      BINV(I)=B(I)                                                      00002170
      IF(SCALEP-1) 210,190,200                                          00002180
  190 IF(B(I).LT.0.0)CALL ERRMSG(30HSOME B(I).LT.0. AND SCALEP=1..,     00002190
     1 6,6,16)                                                          00002200
      B(I)=ALOG(B(I))                                                   00002210
      GO TO 210                                                         00002220
  200 B(I)=ASINH(B(I))                                                  00002230
  210 CONTINUE                                                          00002240
      MAXITR=IWS4                                                       00002250
      MITER=NITER                                                       00002260
      ITER=1                                                            00002270
      WRITE (6,2730)                                                    00002280
      IF (IFSS1.NE.1) GO TO 250                                         00002290
      WRITE (16,2730)                                                   00002300
      GO TO 250                                                         00002310
C=======================================================              00002320
C                  END OF LAST PROBLEM                                  00002330
  220 CALL SUBEND(Y,X,BINV,K,N,TITLE,IOUT)                              00002340
C     ************************************************                  00002350
  240 IF(ISTOP.EQ.1.OR.IALT.EQ.5) GO TO 241                             00002360
C--INITIALIZE FOR NEXT PROB (SAME IALT DATA), SINCE ISTOP=0             00002370
      GO TO 10                                                          00002380
C--FOLLOWING CLOSE STMT ONLY FOR HONEYWELL MULTICS:                     00002390
  241 CALL CLOSE_FILE('-ALL')                                           00002400
C     STOP                                                              00002410
      RETURN                                                            00002420
  250 CONTINUE                                                          00002430
      IF (IP.LE.0) GO TO 280                                            00002440
      DO 270 I=1,IP                                                     00002450
      IF (IB(I).GT.0) GO TO 270                                         00002460
      CALL ERRMSG(30HIP.GT.1 BUT SOME IB(I).LE.0...,6,6,16)             00002470
  270 CONTINUE                                                          00002480
  280 CONTINUE                                                          00002490
      IF (K.GT.10) GO TO 290                                            00002500
C--IBKT=1 MEANS USE UPPER A MATRIX FOR SCRATCH STORAGE                  00002510
C       =2 MEANS USE FILE 13 FOR SCRATCH STORAGE                        00002520
      IBKT=1                                                            00002530
      GO TO 300                                                         00002540
  290 IBKT=2                                                            00002550
  300 XKDB=1.E0                                                         00002560
C--READ OBJECT TIME FORMAT FOR DATA ON FILE IALT.                       00002570
```

```
      READ(5,2480) (FMT(I),I=1,18)                                      00002580
      M1=M+IWT                                                          00002590
      DO 310 I=1,N                                                      00002600
      READ(IALT,FMT) Y(I),(X(I,L),L=1,M1)                               00002610
C--SET UP WTS VIA IWT PARM                                              00002620
      WT(I)=1.0E0                                                       00002630
      IF(IWT.EQ.1.AND.X(I,M1).NE.0.0) WT(I)=1.0E0/X(I,M1)**2            00002640
  310 CONTINUE                                                          00002650
      IF(IALT.NE.5) REWIND IALT                                         00002660
      IWHER=-1                                                          00002670
       GO TO 30                                                         00002680
  320 IBKA=1                                                            00002690
C                                                                       00002700
C     ..................................................................00002710
C                   START THE CALCULATION OF THE PTP MATRIX             00002720
      WRITE(6,2520) TITLE                                               00002730
      WRITE (6,2530) N,K,IP,M,GAMCR,DEL,MODLAM,FF,T,E,TAU,XL,ZETA,      00002740
     1 IALT,ISTOP,IWT,IWS2,IWS3,IWS4,IWS6,IFSS1,NPRNT,SCALEP,SCALEY     00002750
      IF(IP.GT.0) WRITE(6,330) (IB(J),J=1,IP)                           00002760
  330 FORMAT(4H IB=,19I3)                                               00002770
      WRITE(6,340) FMT                                                  00002780
  340 FORMAT(5H FMT=,18A4)                                              00002790
      IF(SCALEP.GT.0.AND.IPRT.GE.0) WRITE(6,350) (BINV(J),J=1,K)        00002800
  350 FORMAT(/30H -INITIAL UNSCALED PARAMETERS-/(12X,4E17.8))           00002810
      IF (IFSS1.NE.1) GO TO 360                                         00002820
      WRITE(16,2520) TITLE                                              00002830
      WRITE (16,2530) N,K,IP,M,GAMCR,DEL,MODLAM,FF,T,E,TAU,XL,ZETA,     00002840
     1 IALT,ISTOP,IWT,IWS2,IWS3,IWS4,IWS6,IFSS1,NPRNT,SCALEP,SCALEY     00002850
      IF(IP.GT.0) WRITE(16,330) (IB(J),J=1,IP)                          00002860
      WRITE(16,340) FMT                                                 00002870
      IF(SCALEP.GT.0.AND.IPRT.GE.0) WRITE(16,350) (BINV(J),J=1,K)       00002880
  360 CONTINUE                                                          00002890
  370 CONTINUE                                                          00002900
      DO 380 I=1,K                                                      00002910
      G(I)=0.E0                                                         00002920
      DO 380 J=1,K                                                      00002930
  380 A(I,J)=0.E0                                                       00002940
      IF(IBKA-2) 390,400,400                                            00002950
  390 IFSS3=IWS3                                                        00002960
      IFSS2=IWS2                                                        00002970
      GO TO 410                                                         00002980
  400 IFSS3=1                                                           00002990
      GO TO 420                                                         00003000
  410 IF(IPRT.GE.0) WRITE (6,2540) (B(J),J=1,K)                         00003010
      IF (IFSS1.NE.1) GO TO 420                                         00003020
      IF(IPRT.GE.0) WRITE (16,2540) (B(J),J=1,K)                        00003030
  420 CONTINUE                                                          00003040
  430 FORMAT(/11H -UNSCALED-)                                           00003050
C--THIS IS I=1 TO N SPECIAL NON-DO LOOP                                 00003060
  450 I=1                                                               00003070
      DO 460 J=1,K                                                      00003080
  460 CALL UNSCAL(B(J),BINV(J),SCALEP)                                  00003090
```

```
      IF(IPRT.LT.0) WRITE(6,2540) (BINV(J),J=1,K)                          00003100
      IF(IFSS1.EQ.1.AND.IPRT.LT.0)WRITE(16,2540)(BINV(J),J=1,K)            00003110
      PHI=0.E0                                                             00003120
      IF (IFSS2.EQ.0) GO TO 480                                            00003130
      GO TO 510                                                            00003140
  470 IF (IFSS2.EQ.1) GO TO 520                                            00003150
C     ..................................................                 00003160
C                   THIS IS THE ANALYTICAL P(J) ROUTINE                  00003170
  480 IWHER=1                                                              00003180
C                   GET P(J) AND F                                       00003190
      GO TO 30                                                             00003200
  490 IF (IP.LE.0) GO TO 640                                               00003210
      DO 500 II=1,IP                                                       00003220
      IWS=IB(II)                                                           00003230
  500 P(IWS)=0.E0                                                          00003240
      GO TO 640                                                            00003250
C      ..................................................                00003260
C                   THIS IS THE ESTIMATED P(J) ROUTINE                   00003270
C  (VIA K.M. BROWN S METHOD)                                               00003280
  510 CONTINUE                                                             00003290
      ISW=1                                                                00003300
      IF(XL.LT.0.1E-3) ISW=2                                               00003310
  520 IWHER=3                                                              00003320
      GO TO 30                                                             00003330
  530 FWS=FINV                                                             00003340
      FSAV=F                                                               00003350
      DO 540 II=1,NPRNT                                                    00003360
  540 SPRNT(II)=PRNT(II)                                                   00003370
      J=1                                                                  00003380
  550 IF (IP.LE.0) GO TO 570                                               00003390
      DO 560 II=1,IP                                                       00003400
      IF ((J-IB(II)).EQ.0) GO TO 610                                       00003410
  560 CONTINUE                                                             00003420
  570 HH=DEL*ABS(BINV(J))                                                  00003430
      IF(ISW.EQ.2) HH=1.E3*HH                                              00003440
      IF(HH.LE.5.E-5) HH=5.E-5                                             00003450
      TWS=B(J)                                                             00003460
      TWS1=BINV(J)                                                         00003470
      BINV(J)=TWS1+HH                                                      00003480
      IWHER=4                                                              00003490
      GO TO 30                                                             00003500
  580 B(J)=TWS                                                             00003510
      BINV(J)=TWS1                                                         00003520
      IF(ISW.EQ.1) GO TO 600                                               00003530
C--CENTRAL DIFFERENCES (ISW=2--WHEN XL.LT..1E-3)                           00003540
      FHH=FINV                                                             00003550
      BINV(J)=TWS1-HH                                                      00003560
      IWHER=5                                                              00003570
      GO TO 30                                                             00003580
  590 B(J)=TWS                                                             00003590
      BINV(J)=TWS1                                                         00003600
      P(J)=.5E0*(FHH-FINV)/HH                                              00003610
```

```
      GO TO 620                                                          00003620
C--FORWARD DIFFERENCES (ISW=1--WHEN XL.GE..1E-3)                         00003630
  600 P(J)=(FINV-FWS)/HH                                                 00003640
      GO TO 620                                                          00003650
  610 P(J)=0.E0                                                          00003660
  620 J=J+1                                                              00003670
      IF ((J-K).LE.0) GO TO 550                                          00003680
      FINV=FWS                                                           00003690
      F=FSAV                                                             00003700
      DO 630 II=1,NPRNT                                                  00003710
  630 PRNT(II)=SPRNT(II)                                                 00003720
C                  END OF ESTIMATED P S ROUTINE                          00003730
C      ..................................................                00003740
C                  NOW, USE THE P(J) TO MAKE PARTIALS MATRIX             00003750
C--SET UP FOR SCALING PARTIAL DERIVATIVES AS SELECTED                    00003760
  640 IF(SCALEP-1) 650,710,730                                           00003770
  650 IF(SCALEY-1) 750,660,690                                           00003780
  660 DEN=1.0E0/FINV                                                     00003790
  670 DO 680 JJ=1,K                                                      00003800
  680 P(JJ)=P(JJ)*DEN                                                    00003810
      GO TO 750                                                          00003820
  690 IF(IPRNT.LE.1) GO TO 700                                           00003830
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 660                               00003840
  700 DEN=1.0E0/SQRT(FINV*FINV+1.0E0)                                    00003850
      GO TO 670                                                          00003860
  710 DO 720 JJ=1,K                                                      00003870
  720 P(JJ)=BINV(JJ)*P(JJ)                                               00003880
      GO TO 650                                                          00003890
  730 DO 740 JJ=1,K                                                      00003900
      DEN=BINV(JJ)+SQRT(BINV(JJ)**2+1.0E0)                               00003910
  740 P(JJ)=0.5E0*(DEN+1.0E0/DEN)*P(JJ)                                  00003920
      GO TO 650                                                          00003930
  750 IF(IBKA.EQ.2) WRITE(13) (P(JJ),JJ=1,K)                             00003940
      DO 760 JJ=1,K                                                      00003950
      G(JJ)=G(JJ)+WT(I)*(Y(I)-F)*P(JJ)                                   00003960
      DO 760 II=JJ,K                                                     00003970
      A(II,JJ)=A(II,JJ)+WT(I)*P(II)*P(JJ)                                00003980
  760 A(JJ,II)=A(II,JJ)                                                  00003990
  770 WS=Y(I)-F                                                          00004000
      IF (IFSS3.LE.0) GO TO 810                                          00004010
C--LAST ITERATION RESULTS AND DATA MATRIX FOR PRINTING                   00004020
      IF(I.GT.1) GO TO 771                                               00004030
      IF(IOUT.EQ.0) GO TO 773                                            00004040
      WRITE(16,430)                                                      00004050
      WRITE(16,2550)                                                     00004060
773   IF(IPRT.LT.-1) WRITE(6,772)                                        00004070
772   FORMAT(/11H -UNSCALED-/3X,1HI,4X,3HOBS,11X,3HCAL,11X,3HRES,        00004080
     1 8X,6HX(I,1))                                                      00004090
771   IF(IPRT.LT.-1) WRITE (6,2700) I,Y(I),F,WS,PRNT(1)                  00004100
      IF(NPRNT.GT.0) GO TO 790                                           00004110
      IF (IFSS1.NE.1) GO TO 780                                          00004120
      WRITE (16,2700) I,Y(I),F,WS                                        00004130
```

```
  780 CONTINUE                                                          00004140
      GO TO 810                                                         00004150
790   CONTINUE                                                          00004160
      IF (IFSS1.NE.1) GO TO 800                                         00004170
      PERR=0.0                                                          00004180
      IF(F.NE.0.0) PERR=100.0*WS/ABS(F)                                 00004190
      WRITE (16,2700) I,Y(I),F,WS,PERR,(PRNT(JJ),JJ=1,NPRNT)            00004200
  800 CONTINUE                                                          00004210
  810 WS=Y(I)-F                                                         00004220
      PHI=PHI+WT(I)*WS*WS                                               00004230
      I=I+1                                                             00004240
      IF (I.LE.N) GO TO 470                                             00004250
C--THIS IN END OF I=1 TO N NON-DO LOOP                                  00004260
      IF(IBKA.NE.2) GO TO 860                                           00004270
C--PRINT UNSCALED PARTIALS SAVED ON FILE 13 (WHEN IBKA=2)               00004280
  820 FORMAT(/20H -UNSCALED PARTIALS-)                                  00004290
      IF(IOUT.EQ.1) WRITE(16,820)                                       00004300
      REWIND 13                                                         00004310
      DO 850 II=1,N                                                     00004320
      READ(13) (SA(JJ),JJ=1,K)                                          00004330
  830 FORMAT(2X,I3,5E18.8)                                              00004340
  840 FORMAT(2X,I3,5E18.8/(5X,5E18.8))                                  00004350
      IF(IOUT.EQ.1.AND.K.NE.5) WRITE(16,840) II,(SA(JJ),JJ=1,K)         00004360
      IF(IOUT.EQ.1.AND.K.EQ.5) WRITE(16,830) II,(SA(JJ),JJ=1,K)         00004370
  850 CONTINUE                                                          00004380
      REWIND 13                                                         00004390
      WRITE(6,430)                                                      00004400
      IF(IOUT.EQ.1) WRITE(16,430)                                       00004410
  860 CONTINUE                                                          00004420
      IF (IP.LE.0) GO TO 890                                            00004430
      DO 880 JJ=1,IP                                                    00004440
      IWS=IB(JJ)                                                        00004450
      DO 870 II=1,K                                                     00004460
      A(IWS,II)=0.E0                                                    00004470
  870 A(II,IWS)=0.E0                                                    00004480
  880 A(IWS,IWS)=1.E0                                                   00004490
  890 IF(IBKA-2) 900,1770,1780                                          00004500
C                   SAVE SQUARE ROOTS OF DIAGONAL ELEMENTS              00004510
  900 DO 910 I=1,K                                                      00004520
  910 SA(I)=SQRT(A(I,I))                                                00004530
      DO 950 I=1,K                                                      00004540
      DO 930 J=1,K                                                      00004550
      WS=SA(I)*SA(J)                                                    00004560
      IF (WS.GT.0.E0) GO TO 920                                         00004570
      A(I,J)=0.E0                                                       00004580
      GO TO 930                                                         00004590
  920 A(I,J)=A(I,J)/WS                                                  00004600
  930 CONTINUE                                                          00004610
      IF (SA(I).GT.0.E0) GO TO 940                                      00004620
      G(I)=0.E0                                                         00004630
      GO TO 950                                                         00004640
  940 G(I)=G(I)/SA(I)                                                   00004650
```

```
  950 CONTINUE                                                          00004660
      DO 960 I=1,K                                                      00004670
  960 A(I,I)=1.E0                                                       00004680
      PHIZ=PHI                                                          00004690
C                   WE NOW HAVE PHI ZERO                              00004700
      IF(IBKT-1) 970,980,970                                            00004710
  970 WRITE (13) A                                                      00004720
      REWIND 13                                                         00004730
      GO TO 1000                                                        00004740
  980 DO 990 II=1,K                                                     00004750
      III=II+10                                                         00004760
      DO 990 JJ=1,K                                                     00004770
  990 A(III,JJ)=A(II,JJ)                                                00004780
C      ..................................................             00004790
 1000 CONTINUE                                                          00004800
      IF (ITCT.GT.0) GO TO 1030                                         00004810
C                   FIRST ITERATION                                   00004820
      IF (XL.GT.0.E0) GO TO 1010                                        00004830
      XL=0.01E0                                                         00004840
 1010 ITCT=1                                                            00004850
      DO 1020 J=1,K                                                     00004860
 1020 BS(J)=B(J)                                                        00004870
C                   BS(J) CORRESPONDS TO PHIZ                         00004880
 1030 IBK1=1                                                            00004890
      WS=N-K+IP                                                         00004900
      IF(N.GT.K) SE=SQRT(PHIZ/WS)                                       00004910
      IF (IFSS3.GT.0) GO TO 1040                                        00004920
      WRITE (6,2560) ITER,PHIZ,SE,XLL,GAMMA,XL                          00004930
      IF (IFSS1.NE.1) GO TO 1320                                        00004940
      WRITE (16,2560) ITER,PHIZ,SE,XLL,GAMMA,XL                         00004950
      GO TO 1320                                                        00004960
 1040   WRITE(6,2490) PHIZ,SE,XL                                        00004970
      IF (IFSS1.NE.1) GO TO 1320                                        00004980
      WRITE (16,2490) PHIZ,SE,XL                                        00004990
      GO TO 1320                                                        00005000
 1050 PHIL=PHI                                                          00005010
C                   WE NOW HAVE PHI(LAMBDA)                           00005020
      DO 1060 J=1,K                                                     00005030
      IF(ABS(DB(J)/(ABS(B(J))+TAU)).GE.E) GO TO 1080                    00005040
 1060 CONTINUE                                                          00005050
      WRITE (6,2680)                                                    00005060
      IF (IFSS1.NE.1) GO TO 1070                                        00005070
      WRITE (16,2680)                                                   00005080
 1070 CONTINUE                                                          00005090
      GO TO 1670                                                        00005100
 1080 IF (IWS4.EQ.0) GO TO 1110                                         00005110
      IF (IWS4.EQ.1) GO TO 1090                                         00005120
      IWS4=IWS4-1                                                       00005130
      ITER=ITER+1                                                       00005140
      GO TO 1110                                                        00005150
 1090 WRITE (6,2690)                                                    00005160
      IF (IFSS1.NE.1) GO TO 1100                                        00005170
```

```
      WRITE (16,2690)                                                   00005180
 1100 CONTINUE                                                          00005190
      GO TO 1670                                                        00005200
 1110 XKDB=1.E0                                                         00005210
      IF (PHIL.GT.PHIZ) GO TO 1190                                      00005220
      XLS=XL                                                            00005230
      DO 1120 J=1,K                                                     00005240
      BA(J)=B(J)                                                        00005250
 1120 B(J)=BS(J)                                                        00005260
      IF (XL.GT..00000001E0) GO TO 1140                                 00005270
      DO 1130 J=1,K                                                     00005280
      B(J)=BA(J)                                                        00005290
 1130 BS(J)=B(J)                                                        00005300
      GO TO 370                                                         00005310
 1140 XL=XL/XNUFAC                                                      00005320
      IBK1=2                                                            00005330
      GO TO 1320                                                        00005340
 1150 PHL4=PHI                                                          00005350
C                  WE NOW HAVE PHI(LAMBDA/XNUFAC)                      00005360
      IF (PHL4.GT.PHIZ) GO TO 1170                                      00005370
      DO 1160 J=1,K                                                     00005380
 1160 BS(J)=B(J)                                                        00005390
      GO TO 370                                                         00005400
 1170 XL=XLS                                                            00005410
C1170 CONTINUE                                                          00005420
      DO 1180 J=1,K                                                     00005430
      BS(J)=BA(J)                                                       00005440
 1180 B(J)=BA(J)                                                        00005450
      GO TO 370                                                         00005460
 1190 IBK1=4                                                            00005470
      XLS=XL                                                            00005480
      XL=XL/XNUFAC                                                      00005490
      DO 1200 J=1,K                                                     00005500
 1200 B(J)=BS(J)                                                        00005510
      GO TO 1320                                                        00005520
 1210 IF (PHI.LE.PHIZ) GO TO 1260                                       00005530
      XL=XLS                                                            00005540
      IBK1=3                                                            00005550
 1220 XL=XL*XNUFAC                                                      00005560
 1230 DO 1240 J=1,K                                                     00005570
 1240 B(J)=BS(J)                                                        00005580
      GO TO 1320                                                        00005590
 1250 PHIT4=PHI                                                         00005600
C                  WE NOW HAVE PHI(XNUFAC*LAMBDA)                      00005610
      IF (PHIT4.GT.PHIZ) GO TO 1280                                     00005620
 1260 DO 1270 J=1,K                                                     00005630
 1270 BS(J)=B(J)                                                        00005640
      GO TO 370                                                         00005650
 1280 IF (GAMMA.GE.GAMCR) GO TO 1220                                    00005660
      XKDB=XKDB/2.E0                                                    00005670
      DO 1290 J=1,K                                                     00005680
      IF(ABS(DB(J)/(ABS(B(J))+TAU)).GE.E) GO TO 1230                    00005690
```

```
 1290 CONTINUE                                                          00005700
      DO 1300 J=1,K                                                     00005710
 1300 B(J)=BS(J)                                                        00005720
      MAXITR=MAXITR-1                                                   00005730
      WRITE (6,2740)                                                    00005740
      IF (IFSS1.NE.1) GO TO 1310                                        00005750
      WRITE (16,2740)                                                   00005760
 1310 CONTINUE                                                          00005770
      GO TO 1670                                                        00005780
C                                                                       00005790
C      ..................................................               00005800
C                   SET UP FOR MATRIX INVERSION                         00005810
 1320 IF(IBKT-1) 1330,1340,1330                                         00005820
 1330 READ (13) A                                                       00005830
      REWIND 13                                                         00005840
      GO TO 1360                                                        00005850
 1340 DO 1350 II=1,K                                                    00005860
      III=II+10                                                         00005870
      DO 1350 JJ=1,K                                                    00005880
 1350 A(II,JJ)=A(III,JJ)                                                00005890
 1360 DO 1370 I=1,K                                                     00005900
 1370 A(I,I)=A(I,I)+XL                                                  00005910
C                   GET INVERSE OF A AND SOLVE FOR DB(J)S               00005920
      IBKM=1                                                            00005930
C      ..................................................               00005940
C                   THIS IS THE MATRIX INVERSION ROUTINE                00005950
C                   K IS THE SIZE OF THE MATRIX                         00005960
 1380 IF(K.EQ.1) GO TO 1390                                             00005970
      CALL GJR (A,K,ZETA,MSING)                                         00005980
      IF(MSING-1) 1400,1400,1381                                        00005990
1381  CALL ERRMSG(20HSINGULAR MATRIX.....,4,6,16)                       00006000
C--SPECIAL CASE, K=1                                                    00006010
 1390 A(1,1)=1.0/A(1,1)                                                 00006020
 1400 IF(IBKM-1) 1410,1410,1840                                         00006030
C                   END OF MATRIX INVERSION, SOLVE FOR DB(J)            00006040
 1410 DO 1430 I=1,K                                                     00006050
      DB(I)=0.E0                                                        00006060
      DO 1420 J=1,K                                                     00006070
 1420 DB(I)=A(I,J)*G(J)+DB(I)                                           00006080
 1430 DB(I)=XKDB*DB(I)                                                  00006090
      XLL=0.E0                                                          00006100
      DTG=0.E0                                                          00006110
      GTG=0.E0                                                          00006120
      DO 1440 J=1,K                                                     00006130
      DB(J)=DB(J)/SA(J)                                                 00006140
      DTG=DTG+DB(J)*G(J)                                                00006150
      GTG=GTG+G(J)**2                                                   00006160
      B(J)=B(J)+DB(J)                                                   00006170
 1440 XLL=XLL+DB(J)*DB(J)                                               00006180
      KIP=K-IP                                                          00006190
      IF (KIP.EQ.1) GO TO 1480                                          00006200
      CGAM=DTG/SQRT(XLL*GTG)                                            00006210
```

```
      JGAM=1                                                            00006220
      IF (CGAM.GT.0.E0) GO TO 1450                                      00006230
      CGAM=ABS(CGAM)                                                    00006240
      JGAM=2                                                            00006250
 1450 GAMMA=57.2957795E0*(1.5707288E0+CGAM*(-0.2121144E0                00006260
     1+CGAM*(0.074261E0-CGAM*                                           00006270
     2.0187293E0)))*SQRT(1.0E0-CGAM)                                    00006280
      IF(JGAM-1) 1460,1490,1460                                         00006290
 1460 GAMMA=180.E0-GAMMA                                                00006300
      IF (XL.LT.1.0E0) GO TO 1490                                       00006310
      WRITE (6,2670) XL,GAMMA                                           00006320
      IF (IFSS1.NE.1) GO TO 1470                                        00006330
      WRITE (16,2670) XL,GAMMA                                          00006340
 1470 CONTINUE                                                          00006350
      GO TO 1670                                                        00006360
 1480 GAMMA=0.E0                                                        00006370
 1490 XLL=SQRT(XLL)                                                     00006380
      IBK2=1                                                            00006390
      GO TO 1540                                                        00006400
 1500 IF (IFSS3.LE.0) GO TO 1530                                        00006410
      WRITE (6,2500) (DB(J),J=1,K)                                      00006420
      IF (IFSS1.NE.1) GO TO 1510                                        00006430
      WRITE (16,2500) (DB(J),J=1,K)                                     00006440
 1510 CONTINUE                                                          00006450
      WRITE (6,2510) PHI,XL,GAMMA,XLL                                   00006460
      IF (IFSS1.NE.1) GO TO 1520                                        00006470
      WRITE (16,2510) PHI,XL,GAMMA,XLL                                  00006480
 1520 CONTINUE                                                          00006490
C--PRESET XNUFAC--(IF MODLAM=1)                                         00006500
 1530 GO TO (1570,1150,1250,1210),IBK1                                  00006510
C                                                                       00006520
C     .................................................................00006530
C                   CALCULATE PHI                                       00006540
 1540 I=1                                                               00006550
      DO 1550 JJ=1,K                                                    00006560
 1550 CALL UNSCAL(B(JJ),BINV(JJ),SCALEP)                                00006570
      PHI=0.E0                                                          00006580
      IWHER=2                                                           00006590
      GO TO 30                                                          00006600
 1560 PHI=PHI+WT(I)*(Y(I)-F)**2                                         00006610
      I=I+1                                                             00006620
      IF (I.LE.N) GO TO 30                                              00006630
      GO TO (1500,2290,1770,2200,2220,2240),IBK2                        00006640
C=======================================================              00006650
C--DETERMINE AN EFFECTIVE MARQUARDT LAMBDA FACTOR (XNUFAC)              00006660
C  BASED ON HISTORY OF SUM OF SQUARES STORED IN LATEST SS(4)--          00006670
 1570 IF(MODLAM.EQ.0) GO TO 1050                                        00006680
      SS(ISS)=PHI                                                       00006690
      INU0=INU                                                          00006700
      GO TO (1590,1580,1600,1610),ISS                                   00006710
C--MACHINE FAILURE IF ISS.GT.4 OR ISS.LT.1                              00006720
C--   STOP 4                                                            00006730
```

```
 1580 IS1=0                                                              00006740
      IF(SS(2).GT.SS(1)) IS1=1                                           00006750
 1590 ISS=ISS+1                                                          00006760
      GO TO 1660                                                         00006770
 1600 IS2=0                                                              00006780
      IF(SS(3).GT.SS(2)) IS2=1                                           00006790
      IF(IS1.EQ.IS2) GO TO 1590                                          00006800
      INU=INU0-1                                                         00006810
      GO TO 1590                                                         00006820
 1610 IS3=0                                                              00006830
      IF(SS(4).GT.SS(3)) IS3=1                                           00006840
      IF(IS1.EQ.IS2.AND.IS3.EQ.IS2) GO TO 1620                           00006850
      IF(IS1.EQ.0.AND.IS2.EQ.0.AND.IS3.EQ.1) GO TO 1640                  00006860
      IF(IS1.EQ.1.AND.IS2.EQ.0.AND.IS3.EQ.1) GO TO 1640                  00006870
      IF(IS1.EQ.1.AND.IS2.EQ.1.AND.IS3.EQ.0) GO TO 1640                  00006880
      GO TO 1650                                                         00006890
 1620 IF(IS1.EQ.0) GO TO 1630                                            00006900
      IF(INU0.GE.3) GO TO 1650                                           00006910
      INU=3                                                              00006920
      GO TO 1650                                                         00006930
 1630 IF(INU0.GE.5) GO TO 1650                                           00006940
      INU=INU0+1                                                         00006950
      GO TO 1650                                                         00006960
 1640 IF(INU0.LE.1) GO TO 1650                                           00006970
      INU=INU0-1                                                         00006980
 1650 IS1=IS2                                                            00006990
      IS2=IS3                                                            00007000
      SS(3)=SS(4)                                                        00007010
 1660 XNUFAC=XNU(INU)                                                    00007020
      GO TO 1050                                                         00007030
C                                                                        00007040
C                                                                        00007050
C     ..................................................................00007060
C                  THIS IS THE CONFIDENCE LIMIT CALCULATION              00007070
 1670 ITR=MAXITR-IWS4+1                                                  00007080
      WRITE(6,1680) ITR                                                  00007090
 1680 FORMAT(1X,I4,11H ITERATIONS)                                       00007100
      IF(IFSS1.EQ.1) WRITE(16,1680) ITR                                  00007110
      DO 1690 J=1,K                                                      00007120
      CALL UNSCAL(BS(J),BINV(J),SCALEP)                                  00007130
      BS(J)=BINV(J)                                                      00007140
 1690 B(J)=BS(J)                                                         00007150
      WRITE(6,2520) TITLE                                                00007160
      IF (IFSS1.NE.1) GO TO 1700                                         00007170
      WRITE(16,2520) TITLE                                               00007180
 1700 CONTINUE                                                           00007190
      IBKA=2                                                             00007200
C--UNSCALE BOTH PARAMETER AND OBSERVATION SPACES PRIOR                   00007210
C  TO FINAL STATISTICS ON LAST INTERATION--AND WHERE                     00007220
C  IBKA=2, IFSS3=0..                                                     00007230
C               THIS WILL PRINT OBS,CAL,RES,ETC.                         00007240
C               AND SAVE UNSCALED PARTIALS ON FILE FILE13..              00007250
```

```
      IF(IPRT.GE.0) WRITE(6,1710) (BINV(J),J=1,K)                        00007260
 1710 FORMAT(/28H -FINAL UNSCALED PARAMETERS-/(12X,4E17.8))              00007270
      IF(IFSS1.EQ.1.AND.IPRT.GE.0) WRITE(16,1710) (BINV(J),J=1,K)        00007280
      IF(SCALEY.EQ.0) GO TO 1760                                         00007290
      DO 1750 I=1,N                                                      00007300
      IF(SCALEY.NE.1) GO TO 1730                                         00007310
 1720 Y(I)=EXP(Y(I))                                                     00007320
      GO TO 1750                                                         00007330
 1730 IF(IPRNT.LE.1) GO TO 1740                                          00007340
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 1720                              00007350
 1740 Y(I)=SINH(Y(I))                                                    00007360
 1750 CONTINUE                                                           00007370
 1760 LSCALP=SCALEP                                                      00007380
      LSCALY=SCALEY                                                      00007390
      SCALEP=0                                                           00007400
      SCALEY=0                                                           00007410
      GO TO 370                                                          00007420
 1770 CONTINUE                                                           00007430
 1780 WS=N-K+IP                                                          00007440
      IF(N.GT.K) SE=SQRT(PHI/WS)                                         00007450
      PHIZ=PHI                                                           00007460
      WRITE (6,2490) PHIZ,SE,XL                                          00007470
      IF (IFSS1.NE.1) GO TO 1790                                         00007480
      WRITE (16,2490) PHIZ,SE,XL                                         00007490
C                                                                        00007500
C                  WE NOW HAVE MATRIX A                                  00007510
 1790 IF(IBKT-1) 1800,1810,1800                                          00007520
 1800 WRITE (13) A                                                       00007530
      REWIND 13                                                          00007540
      GO TO 1830                                                         00007550
 1810 DO 1820 II=1,K                                                     00007560
      III=II+10                                                          00007570
      DO 1820 JJ=1,K                                                     00007580
 1820 A(III,JJ)=A(II,JJ)                                                 00007590
 1830 IBKM=2                                                             00007600
      GO TO 1380                                                         00007610
C                                                                        00007620
C                  WE NOW HAVE C = A INVERSE                             00007630
 1840 DO 1850 J=1,K                                                      00007640
      IF (A(J,J).LT.0.E0) GO TO 1860                                     00007650
 1850 SA(J)=SQRT(A(J,J))                                                 00007660
      GO TO 1870                                                         00007670
 1860 IBOUT=1                                                            00007680
 1870 KST=-4                                                             00007690
      IF (IFSS1.NE.1) GO TO 1880                                         00007700
      WRITE (16,2600)                                                    00007710
 1880 KST=KST+5                                                          00007720
      KEND=KST+4                                                         00007730
      IF (KEND.LT.K) GO TO 1890                                          00007740
      KEND=K                                                             00007750
 1890 DO 1910 I=1,K                                                      00007760
      IF (IFSS1.NE.1) GO TO 1900                                         00007770
```

```
      WRITE (16,2620) I,(A(I,J),J=KST,KEND)                          00007780
 1900 CONTINUE                                                       00007790
 1910 CONTINUE                                                       00007800
      IF (KEND.LT.K) GO TO 1880                                      00007810
      IF (IBOUT.EQ.0) GO TO 1920                                     00007820
      WRITE (6,2760)                                                 00007830
      IF (IFSS1.NE.1) GO TO 220                                      00007840
      WRITE (16,2760)                                                00007850
      GO TO 220                                                      00007860
 1920 DO 1940 I=1,K                                                  00007870
      DO 1940 J=1,K                                                  00007880
      WS=SA(I)*SA(J)                                                 00007890
      IF (WS.GT.0.E0) GO TO 1930                                     00007900
      A(I,J)=0.E0                                                    00007910
      GO TO 1940                                                     00007920
 1930 A(I,J)=A(I,J)/WS                                               00007930
 1940 CONTINUE                                                       00007940
      DO 1950 J=1,K                                                  00007950
 1950 A(J,J)=1.E0                                                    00007960
      IF (IFSS1.NE.1) GO TO 1960                                     00007970
      WRITE (16,2610)                                                00007980
 1960 CONTINUE                                                       00007990
      KST=-9                                                         00008000
 1970 KST=KST+10                                                     00008010
      KEND=KST+9                                                     00008020
      IF (KEND.LT.K) GO TO 1980                                      00008030
      KEND=K                                                         00008040
 1980 DO 2000 I=1,K                                                  00008050
      IF (IFSS1.NE.1) GO TO 1990                                     00008060
      WRITE (16,2750) I,(A(I,J),J=KST,KEND)                          00008070
 1990 CONTINUE                                                       00008080
 2000 CONTINUE                                                       00008090
      IF (KEND.LT.K) GO TO 1970                                      00008100
C                GET T*SE*SQRT(C(I,I))                              00008110
      DO 2010 J=1,K                                                  00008120
 2010 SA(J)=SE*SA(J)                                                 00008130
      IF(IBKT-1) 2020,2030,2020                                      00008140
 2020 READ (13) A                                                    00008150
      REWIND 13                                                      00008160
      GO TO 2050                                                     00008170
 2030 DO 2040 II=1,K                                                 00008180
      III=II+10                                                      00008190
      DO 2040 JJ=1,K                                                 00008200
 2040 A(II,JJ)=A(III,JJ)                                             00008210
 2050 CONTINUE                                                       00008220
      WRITE (6,2640)                                                 00008230
      IF (IFSS1.NE.1) GO TO 2060                                     00008240
      WRITE (16,2630)                                                00008250
 2060 CONTINUE                                                       00008260
      WS=K-IP                                                        00008270
      DO 2120 J=1,K                                                  00008280
      IF (IP.LE.0) GO TO 2080                                        00008290
```

```
      DO 2070 I=1,IP                                                    00008300
      IF (J.EQ.IB(I)) GO TO 2100                                        00008310
 2070 CONTINUE                                                          00008320
C                                                                       00008330
C--COMPUTE STD.ERR, CONF. LIMITS, AND STD.ERR/PARM.                     00008340
C                                                                       00008350
 2080 HJTD=SQRT(WS*FF)*SA(J)                                            00008360
      STE=SA(J)                                                         00008370
      TWS=STE*T                                                         00008380
      OPL=BINV(J)-TWS                                                   00008390
      OPU=BINV(J)+TWS                                                   00008400
      SPL=BINV(J)-HJTD                                                  00008410
      SPU=BINV(J)+HJTD                                                  00008420
      HJTD=0.0                                                          00008430
      IF(BINV(J).NE.0.0) HJTD=STE/BINV(J)                               00008440
      WRITE (6,2720) J,STE,OPL,OPU,HJTD                                 00008450
      IF (IFSS1.NE.1) GO TO 2090                                        00008460
      WRITE (16,2720) J,STE,OPL,OPU,SPL,SPU,HJTD                        00008470
 2090 CONTINUE                                                          00008480
      GO TO 2120                                                        00008490
 2100 WRITE (6,2570) J                                                  00008500
      IF (IFSS1.NE.1) GO TO 2110                                        00008510
      WRITE (16,2570) J                                                 00008520
 2110 CONTINUE                                                          00008530
 2120 CONTINUE                                                          00008540
C                    NONLINEAR CONFIDENCE LIMIT                         00008550
      IF (IWS6.EQ.1.OR.N.EQ.K) GO TO 220                                00008560
      WS=K-IP                                                           00008570
      WS1=N-K+IP                                                        00008580
      PKN=WS/WS1                                                        00008590
      PC=PHIZ*(1.E0+FF*PKN)                                             00008600
      WRITE (6,2650) PC                                                 00008610
      IF (IFSS1.NE.1) GO TO 2130                                        00008620
      WRITE (16,2650) PC                                                00008630
 2130 CONTINUE                                                          00008640
      WRITE (6,2660)                                                    00008650
      IF (IFSS1.NE.1) GO TO 2140                                        00008660
      WRITE (16,2660)                                                   00008670
 2140 CONTINUE                                                          00008680
      IFSS3=1                                                           00008690
C--       NON- DO LOOP J=1,K                                            00008700
C  (SINCE CONTROL JUMPS OUT AND BACK INSIDE LOOP)                       00008710
      J=1                                                               00008720
 2150 IBKP=1                                                            00008730
      DO 2160 JJ=1,K                                                    00008740
 2160 B(JJ)=BS(JJ)                                                      00008750
      IF (IP.LE.0) GO TO 2180                                           00008760
      DO 2170 JJ=1,IP                                                   00008770
      IF (J.EQ.IB(JJ)) GO TO 2380                                       00008780
 2170 CONTINUE                                                          00008790
 2180 DD=-1.E0                                                          00008800
      IBKN=1                                                            00008810
```

```
2190 D=DD                                                              00008820
     B(J)=BS(J)+D*SA(J)                                                00008830
     IBK2=4                                                            00008840
     GO TO 1540                                                        00008850
2200 PHI1=PHI                                                          00008860
     IF (PHI1.GE.PC) GO TO 2230                                        00008870
2210 D=D+DD                                                            00008880
     IF (D/DD.GE.5.E0) GO TO 2420                                      00008890
     B(J)=BS(J)+D*SA(J)                                                00008900
     IBK2=5                                                            00008910
     GO TO 1540                                                        00008920
2220 PHID=PHI                                                          00008930
     IF (PHID.LT.PC) GO TO 2210                                        00008940
     IF (PHID.GE.PC) GO TO 2250                                        00008950
2230 D=D/2.E0                                                          00008960
     IF (D/DD.LE..001E0) GO TO 2420                                    00008970
     B(J)=BS(J)+D*SA(J)                                                00008980
     IBK2=6                                                            00008990
     GO TO 1540                                                        00009000
2240 PHID=PHI                                                          00009010
     IF (PHID.GT.PC) GO TO 2230                                        00009020
2250 XK1=PHIZ/D+PHI1/(1.E0-D)+PHID/(D*(D-1.E0))                        00009030
     XK2=-(PHIZ*(1.E0+D)/D+D/(1.E0-D)*PHI1+PHID/(D*(D-1.E0)))          00009040
     XK3=PHIZ-PC                                                       00009050
     BC=(SQRT(XK2*XK2-4.E0*XK1*XK3)-XK2)/(2.E0*XK1)                    00009060
     IF(IBKN-1) 2260,2260,2270                                         00009070
2260 B(J)=BS(J)-SA(J)*BC                                               00009080
     GO TO 2280                                                        00009090
2270 B(J)=BS(J)+SA(J)*BC                                               00009100
2280 IBK2=2                                                            00009110
     GO TO 1540                                                        00009120
2290 IF(IBKN-1) 2300,2300,2310                                         00009130
2300 IBKN=2                                                            00009140
     DD=1.E0                                                           00009150
     BL=B(J)                                                           00009160
     PL=PHI                                                            00009170
     GO TO 2190                                                        00009180
2310 BU=B(J)                                                           00009190
     PU=PHI                                                            00009200
     GO TO (2320,2340,2360,2400), IBKP                                 00009210
2320 WRITE (6,2620) J,BL,PL,BU,PU                                      00009220
     IF (IFSS1.NE.1) GO TO 2330                                        00009230
     WRITE (16,2620) J,BL,PL,BU,PU                                     00009240
2330 CONTINUE                                                          00009250
     GO TO 2470                                                        00009260
2340 WRITE (6,2590) J,BU,PU                                            00009270
     IF (IFSS1.NE.1) GO TO 2350                                        00009280
     WRITE (16,2590) J,BU,PU                                           00009290
2350 CONTINUE                                                          00009300
     GO TO 2470                                                        00009310
2360 WRITE (6,2620) J,BL,PL                                            00009320
     IF (IFSS1.NE.1) GO TO 2370                                        00009330
```

```
      WRITE (16,2620) J,BL,PL                                          00009340
 2370 CONTINUE                                                         00009350
      GO TO 2470                                                       00009360
 2380 WRITE (6,2570) J                                                 00009370
      IF (IFSS1.NE.1) GO TO 2390                                       00009380
      WRITE (16,2570) J                                                00009390
 2390 CONTINUE                                                         00009400
      GO TO 2470                                                       00009410
 2400 WRITE (6,2580) J                                                 00009420
      IF (IFSS1.NE.1) GO TO 2410                                       00009430
      WRITE (16,2580) J                                                00009440
 2410 CONTINUE                                                         00009450
      GO TO 2470                                                       00009460
 2420 IF(IBKN-1) 2430,2430,2440                                        00009470
C                  DELETE LOWER PRINT                                 00009480
 2430 IBKP=2                                                           00009490
      GO TO 2290                                                       00009500
 2440 IF(IBKP-1) 2450,2450,2460                                        00009510
C                  DELETE UPPER PRINT                                 00009520
 2450 IBKP=3                                                           00009530
      GO TO 2290                                                       00009540
C                  LOWER IS ALREADY DELETED, SO DELETE BOTH           00009550
 2460 IBKP=4                                                           00009560
      GO TO 2290                                                       00009570
C--END OF NON- DO LOOP J=1,K                                          00009580
 2470 J=J+1                                                            00009590
      IF(J.LE.K) GO TO 2150                                            00009600
      GO TO 220                                                        00009610
C     ..................................................................00009620
 2480 FORMAT(18A4)                                                     00009630
 2490   FORMAT(/13X,4H PHI,14X,4H S E,9X,7H LAMBDA/5X,2E18.8,E13.3)     00009640
 2500 FORMAT (/12H INCREMENTS ,4E17.8/(12X,4E17.8))                    00009650
 2510 FORMAT (13X,4H PHI10X,7H LAMBDA6X,7H GAMMA 6X,7H LENGTH/5X,E18.8,300009660
     1E13.3)                                                           00009670
 2520 FORMAT(16H1M A R Q R T  --,5X,16A5)                              00009680
 2530   FORMAT(/5H N = ,I4,8X,4HK = ,I3,9X,5HIP = ,I3,8X,4HM = ,I2,10X, 00009690
     1 6HGAMCR=,E9.3/5H DEL=,E10.3,2X,9HMODLAM = ,I1,6X,3HFF=,E10.3,3X, 00009700
     2 2HT=,E10.3,4X,2HE=,E10.3/5H TAU=,E10.3,2X,3HXL=,E10.3,3X,       00009710
     3 5HZETA=,E10.3,8H IALT = ,I2,7X,8HISTOP = ,I1/7H IWT = ,I1,9X,   00009720
     4 7HIDER = ,I1,8X,7HIPRT = ,I2,7X,8HNITER = ,I4,4X,7HINON = ,I1/  00009730
     5 8H IOUT = ,I2,7X,                                               00009740
     6 8HNPRNT = ,I1,7X,9HSCALEP = ,I1,6X,9HSCALEY = ,I1/)             00009750
 2540 FORMAT (/12H PARAMETERS ,4E17.8/(12X,4E17.8))                    00009760
 2550   FORMAT(3X,1HI,4X,8HOBS.Y(I),6X,3HCAL,11X,3HRES,8X,8H%RES.ERR,6X,00009770
     1 6HX(I,1),8X,6HX(I,2),8X,6HX(I,3),8X,6HX(I,4),8X,6HX(I,5))       00009780
 2560   FORMAT(/1X,4HITER,8X,4H PHI,14X,4H S E,11X,7H LENGTH,6X,       00009790
     1 7H GAMMA ,6X,7H LAMBDA/1X,I4,2E18.8,3E13.3)                     00009800
 2570 FORMAT (2X,I3,20H PARAMETER NOT USED )                           00009810
 2580 FORMAT (2X,I3,12H NONE FOUND )                                   00009820
 2590 FORMAT (2X,I3,36X,2E18.8)                                        00009830
 2600 FORMAT (1H /13H PTP INVERSE )                                    00009840
 2610 FORMAT (1H /30H PARAMETER CORRELATION MATRIX )                   00009850
```

```
 2620 FORMAT (2X,I3,5E18.8)                                              00009860
 2630   FORMAT(//4X,13HPARAMETER STD,17X,15HONE - PARAMETER,21X,          00009870
     1 14H SUPPORT PLANE/11X,6H ERROR,12X,6H LOWER,12X,6H UPPER,12X,     00009880
     2 6H LOWER,12X,6H UPPER,10X,14HSTD.ERROR/PARM)                      00009890
 2640   FORMAT(//4X,13HPARAMETER STD,17X,15HONE - PARAMETER/11X,          00009900
     1 6H ERROR,12X,6H LOWER,12X,6H UPPER,10X,14HSTD.ERROR/PARM)         00009910
 2650 FORMAT (//30H  NONLINEAR CONFIDENCE LIMITS //13H PHI CRITICAL,     00009920
     1 E15.8)                                                            00009930
 2660 FORMAT (1H /6H  PARA6X,8H LOWER B8X,10H LOWER PHI10X,8H UPPER B8X,00009940
     110H UPPER PHI)                                                     00009950
 2670 FORMAT (/19H -GAMMA LAMBDA TEST,5X,2E13.3)                         00009960
 2680 FORMAT (/15H -EPSILON TEST )                                       00009970
 2690 FORMAT (/12H -FORCE OFF )                                          00009980
 2700 FORMAT(1X,I3,2E14.6,E11.3,6E14.6)                                  00009990
 2720 FORMAT (2X,I3,6E18.8)                                              00010000
 2730 FORMAT (1H )                                                       00010010
 2740 FORMAT (/20H -GAMMA EPSILON TEST)                                  00010020
 2750 FORMAT (3X,I5,2X,10F10.4)                                          00010030
 2760 FORMAT (/27H  NEGATIVE DIAGONAL ELEMENT)                           00010040
      END                                                                00010050

      SUBROUTINE GJR (A,N,EPS,MSING)                                     00010060
C     GAUSS-JORDAN-RUTISHAUSER MATRIX INVERSION WITH DOUBLE PIVOTING.    00010070
      DIMENSION A(20,20),B(20),C(20),P(20),Q(20)                         00010080
      INTEGER P,Q                                                        00010090
      MSING=1                                                            00010100
      DO 140 K=1,N                                                       00010110
C     DETERMINATION OF THE PIVOT ELEMENT                                 00010120
      PIVOT=0.E0                                                         00010130
      DO 20 I=K,N                                                        00010140
      DO 20 J=K,N                                                        00010150
      IF(ABS(A(I,J))-ABS(PIVOT)) 20,20,10                                00010160
   10 PIVOT=A(I,J)                                                       00010170
      P(K)=I                                                             00010180
      Q(K)=J                                                             00010190
   20 CONTINUE                                                           00010200
      IF(ABS(PIVOT)-EPS) 220,220,30                                      00010210
C     EXCHANGE OF THE PIVOTAL ROW WITH THE KTH ROW                       00010220
   30 IF (P(K)-K) 40,60,40                                               00010230
   40 DO 50 J=1,N                                                        00010240
      L=P(K)                                                             00010250
      Z=A(L,J)                                                           00010260
      A(L,J)=A(K,J)                                                      00010270
   50 A(K,J)=Z                                                           00010280
C     EXCHANGE OF THE PIVOTAL COLUMN WITH THE KTH COLUMN                 00010290
   60 IF (Q(K)-K) 70,90,70                                               00010300
   70 DO 80 I=1,N                                                        00010310
      L=Q(K)                                                             00010320
      Z=A(I,L)                                                           00010330
      A(I,L)=A(I,K)                                                      00010340
   80 A(I,K)=Z                                                           00010350
   90 CONTINUE                                                           00010360
```

```
C       JORDAN STEP                                                     00010370
        DO 130 J=1,N                                                    00010380
        IF (J-K) 110,100,110                                            00010390
  100 B(J)=1.0E0/PIVOT                                                  00010400
        C(J)=1.0E0                                                      00010410
        GO TO 120                                                       00010420
  110 B(J)=-A(K,J)/PIVOT                                                00010430
        C(J)=A(J,K)                                                     00010440
  120 A(K,J)=0.0E0                                                      00010450
  130 A(J,K)=0.0E0                                                      00010460
        DO 140 I=1,N                                                    00010470
        DO 140 J=1,N                                                    00010480
  140 A(I,J)=A(I,J)+C(I)*B(J)                                           00010490
C       REORDERING THE MATRIX                                           00010500
        DO 200 M=1,N                                                    00010510
        K=N-M+1                                                         00010520
        IF (P(K)-K) 150,170,150                                         00010530
  150 DO 160 I=1,N                                                      00010540
        L=P(K)                                                          00010550
        Z=A(I,L)                                                        00010560
        A(I,L)=A(I,K)                                                   00010570
  160 A(I,K)=Z                                                          00010580
  170 IF (Q(K)-K) 180,200,180                                           00010590
  180 DO 190 J=1,N                                                      00010600
        L=Q(K)                                                          00010610
        Z=A(L,J)                                                        00010620
        A(L,J)=A(K,J)                                                   00010630
  190 A(K,J)=Z                                                          00010640
  200 CONTINUE                                                          00010650
  210 RETURN                                                            00010660
  220 PRINT 230, P(K),Q(K),PIVOT                                        00010670
  230 FORMAT (/16H SINGULAR MATRIX3H I=I3,3H J=I3,7H PIVOT=E16.8/)      00010680
        MSING=2                                                         00010690
        GO TO 210                                                       00010700
        END                                                             00010710

      SUBROUTINE UNSCAL(BIN,BOUT,SCALEP)                                00010720
C// MODIFIED TO TRAP ERRORS >10**38 ON MULTICS                          00010730
C--UNSCALE PARMETER BIN TO BOUT VIA SCALEP                              00010740
      INTEGER SCALEP                                                    00010750
      IF(SCALEP-1) 10,20,30                                             00010760
   10 BOUT=BIN                                                          00010770
      GO TO 40                                                          00010780
   20 IF(BIN.GT.88.028) GO TO 99                                        00010790
      BOUT= EXP_(BIN)                                                   00010800
      GO TO 40                                                          00010810
   30 BOUT= SINH(BIN)                                                   00010820
   40 RETURN                                                            00010830
   99 WRITE(6,699) BIN                                                  00010840
      WRITE(16,699) BIN                                                 00010850
699   FORMAT('0"UNSCAL" ARG=',E16.8,' >88.028 FOR EXP_() ON MULTICS'/   00010860
     & ' --CHECK ALL $PARMS AND DATA --IF OK, THEN--'/                  00010870
```

```
     & ' --TRY RESTARTING WITH DIFFERENT SCALING OPTION(S) --OR--'/      00010880
     & ' --RESTART WITH BETTER "GUESSED" STARTING PARAMETERS.')         00010890
      CALL CLOSE_FILE('-ALL')                                           00010900
      STOP                                                              00010910
      END                                                               00010920

      REAL FUNCTION ASINH(X)                                            00010930
C--INVERSE HYPERBOLIC SIN FUNCTION                                      00010940
C                                                                       00010950
      REAL*8 X2                                                         00010960
      X2=X                                                              00010970
      ASINH=DLOG(X2+DSQRT(X2*X2+1.0D0))                                 00010980
      RETURN                                                            00010990
      END                                                               00011000

      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                   00011010
C--ERROR MESSAGE WRITE ROUTINE AND STOP, WHERE--                        00011020
C                                                                       00011030
C       MSG=     ANY MULTIPLE OF 5 CHARACTERS--MAX. OF 120             00011040
C                (USE NH-------- FORM FOR ANSI COMPATIBILITY)          00011050
C       M5=      NO.CHARS IN MSG/5 (REMAINDER MUST BE 0) 1.LE.M5.LE.24 00011060
C       I6=      1ST UNIT  FOR WRITE(I6, ) MSG -- USUALLY I6=6 FOR LPT. 00011070
C                IF I6.LE.0 UNIT I6 IGNORED.                           00011080
C       I9=      2ND UNIT  FOR WRITE(I9, ) MSG --                      00011090
C                IF I9.LE.0, UNIT I9 IGNORED.                          00011100
C--MESSAGE WRITTEN IN FORM--                                            00011110
C       /ERROR--MSG HERE                                                00011120
C                                                                       00011130
      DIMENSION MSG(30)                                                 00011140
      J=5*M5                                                            00011150
      K=J/4+MOD(J,4)                                                    00011160
      IF(I6.GT.0) WRITE(I6,10) (MSG(I),I=1,K)                           00011170
   10 FORMAT(/8H ERROR--,30A4)                                          00011180
      IF(I9.GT.0) WRITE(I9,10) (MSG(I),I=1,K)                           00011190
      CALL CLOSE_FILE('-ALL')                                           00011200
C                                                                       00011210
      STOP                                                              00011220
      END                                                               00011230

      SUBROUTINE RECUR1(G,V1,F1)                                        00011240
C--BACKWARD RECURRENCE FOR COMPLEX V1,F1 GIVEN REAL*4 ARGUMENT G AND:   00011250
COMMON/MODEL/ PARAMETERS:                                               00011260
C     K(10) = NORMALIZED CONDUCTIVITY ARRAY (M VALUES,WHERE K(1)=1.0).  00011270
C     D(9)  = LAYER THICKNESS ARRAY (M-1 VALUES) D=2*THICKNESS/DEL.     00011280
C     M     = NUMBER LAYERS (M.GE.1.AND.M.LE.10)                        00011290
C             SPECIAL CASE WHEN M=1 (HOMOGENEOUS--D IGNORED)            00011300
C                                                                       00011310
C--NOTE: G,K,D ARE REAL*4                                               00011320
C                                                                       00011330
C                                                                       00011340
      COMMON/MODEL/K,D,M                                                00011350
      REAL*4 K(10),D(9)                                                 00011360
```

```
      COMPLEX C,VM,V1,F1,EVD,ONE                                        00011370
      DATA ONE/(1.0,0.0)/                                               00011380
      F1=ONE                                                            00011390
      G2=G*G                                                            00011400
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                      00011410
      IF(M.EQ.1) GO TO 2                                                00011420
      J=M-1                                                             00011430
    1 V1=CSQRT(CMPLX(G2,2.0*K(J)))                                      00011440
      EVD=CEXP(-V1*D(J))                                                00011450
      C=(ONE-EVD)/(ONE+EVD)                                             00011460
      F1=(VM*F1+V1*C)/(V1+VM*F1*C)                                      00011470
      IF(J.EQ.1) GO TO 3                                                00011480
      J=J-1                                                             00011490
      VM=V1                                                             00011500
      GO TO 1                                                           00011510
    2 V1=VM                                                             00011520
    3 RETURN                                                            00011530
      END                                                               00011540

      SUBROUTINE RECURF(G,DEL,SIG1,V1,F1,PF1,JJ)                        00011550
C--GET PF1=PARTIAL OF F1 W/R PARM. JJ, EVALUATED AT                     00011560
C  THE GIVEN G,DEL, AND SIG1 (OTHER MODEL PARMS IN COMMON/MODEL/)       00011570
C  ALSO GIVEN ARE V1,F1 AS IN RECUR1.                                   00011580
C                                                                       00011590
      IMPLICIT COMPLEX (A-H,O-Z)                                        00011600
      REAL   K,D,G,G2,DEL,SIG1                                          00011610
      COMMON/MODEL/K,D,M                                                00011620
      DIMENSION K(10),D(9)                                              00011630
      DATA ONE,ZERO,CI/(1.0,0.0),(0.0,0.0),(0.0,1.0)/                   00011640
      TWODEL=CMPLX(2.0/DEL,0.0)                                         00011650
      JJM=JJ-M                                                          00011660
      FM=ONE                                                            00011670
      PF1=ZERO                                                          00011680
   30 G2=G*G                                                            00011690
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                      00011700
   50 IF(M.EQ.1) GO TO 150                                              00011710
C--INITIALIZE PARTIAL INDEX J=M-1 (NUM. INDEX)                          00011720
      J=M-1                                                             00011730
C--LOOP ON J INDEX                                                      00011740
   70 V1=CSQRT(CMPLX(G2,2.0*K(J)))                                      00011750
      EVD=CEXP(-V1*D(J))                                                00011760
   90 EVD1=ONE+EVD                                                      00011770
      E1=(ONE-EVD)/EVD1                                                 00011780
      E11=ONE+E1                                                        00011790
      T=VM*FM                                                           00011800
      DEN=V1+T*E1                                                       00011810
      F1=(T+V1*E1)/DEN                                                  00011820
      IF(JJ.LE.M) GO TO 100                                             00011830
C--RECUR FOR PF1 W/R DIST                                               00011840
      EMD1=ZERO                                                         00011850
      IF(JJM.EQ.J) EMD1=(TWODEL*V1*EVD*E11)/EVD1                        00011860
      PF1=((VM*PF1+V1*EMD1)-F1*VM*(FM*EMD1+E1*PF1))/DEN                 00011870
```

```
      GO TO 140                                                          00011880
C--RECUR FOR PF1 W/R SIGMA                                               00011890
  100 VMS=ZERO                                                           00011900
      VMS1=ZERO                                                          00011910
      EMS1=ZERO                                                          00011920
      IF(JJ.EQ.1) GO TO 110                                              00011930
      IF(J+1.EQ.JJ) VMS=CI/(SIG1*VM)                                     00011940
      IF(J.EQ.JJ) VMS1=CI/(SIG1*V1)                                      00011950
      GO TO 120                                                          00011960
  110 IF(M.GT.1) VMS=-CI*K(J+1)/(SIG1*VM)                                00011970
      IF(J.GT.1) VMS1=-CI*K(J)/(SIG1*V1)                                 00011980
  120 IF(JJ.NE.J) GO TO 130                                              00011990
      IF(J.EQ.1) EMS1=(EVD*V1*D(1)*E11)/(2.0*SIG1*EVD1)                  00012000
      IF(J.GT.1) EMS1=(D(J)*EVD*VMS1*E11)/EVD1                           00012010
  130 PF1=((FM*VMS+VM*PF1+V1*EMS1+E1*VMS1)-F1*                           00012020
     1(VMS1+VM*(FM*EMS1+E1*PF1)+FM*E1*VMS))/DEN                          00012030
  140 IF(J.EQ.1) GO TO 180                                               00012040
      J=J-1                                                              00012050
      VM=V1                                                              00012060
      FM=F1                                                              00012070
      GO TO 70                                                           00012080
C--SPECIAL CASE M=1 (HOMOGENEOUS EARTH)                                  00012090
  150 F1=FM                                                              00012100
      V1=VM                                                              00012110
      J=1                                                                00012120
      EVD=ZERO                                                           00012130
      GO TO 90                                                           00012140
  180 RETURN                                                             00012150
      END                                                                00012160

      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)                          00012170
C--ONE DIMENSIONAL CUBIC SPLINE COEFFICIENT DETERMINATION.               00012180
C                                                                        00012190
C        BY  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO      00012200
C                                                                        00012210
C  PARMS--- M= NUMBER OF DATA POINTS .GT. 2                              00012220
C           H= EQUAL INTERVAL OPTION WHEN H.GT.0. (USE DUMMY X HERE),    00012230
C              UNEQUAL INTERVALS IF H=0. (X REQUIRED STORAGE)            00012240
C           X= INDEP.VAR WHEN H=0. (DIM .GE. M).                         00012250
C           Y= DEPENDENT VARIABLE  (DIM .GE. M).                         00012260
C           A,B,C=COEFF.ARRAYS (EACH DIM .GE. M)                         00012270
C                 RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS OF A,B,&C.   00012280
C                 ALSO USED AS WORK ARRAYS DURING EXECUTION.             00012290
C           IT= TYPE OF BOUNDARY CONDITION SUPPLIED IN D ARRAY. USE      00012300
C              IT=1 IF 1ST DERIVATIVES GIVEN AT END POINTS, OR           00012310
C              IT=0 IF 2ND DERIVATIVES GIVEN AT END POINTS.              00012320
C           D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M RESPECTIVELY.     00012330
C           P,S= WORK ARRAYS (EACH DIM=M).                               00012340
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE.               00012350
C  THE RESULTING CUBIC SPLINE IS OF THE FORM:                            00012360
C     Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3             00012370
C       FOR I=1,2,...,M-1                                                00012380
```

```
C                                                                       00012390
C                                                                       00012400
      REAL*4  X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL               00012410
      IF(IT.LT.0.OR.IT.GT.1.OR.H.LT.0..OR.M.LT.3) GO TO 999             00012420
      N=M-1                                                             00012430
      IF(IT.EQ.0) GO TO 20                                              00012440
C--1ST DERIVATIVE BOUNDARIES GIVEN                                      00012450
      NE=N-1                                                            00012460
      IF(H) 999,11,1                                                    00012470
C--EQUAL SPACING H .GT. 0. AND IT=1                                     00012480
    1 HH=3.0/H                                                          00012490
      DO 2 I=1,NE                                                       00012500
      B(I)=4.0                                                          00012510
      C(I)=1.0                                                          00012520
      A(I)=1.0                                                          00012530
    2 P(I)=HH*(Y(I+2)-Y(I))                                             00012540
      P(1)=P(1)-D(1)                                                    00012550
      P(NE)=P(NE)-D(2)                                                  00012560
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE                       00012570
    3 C(1)=C(1)/B(1)                                                    00012580
      P(1)=P(1)/B(1)                                                    00012590
      DO 4 I=2,NE                                                       00012600
      MUL=1.0/(B(I)-A(I)*C(I-1))                                        00012610
      C(I)=MUL*C(I)                                                     00012620
    4 P(I)=MUL*(P(I)-A(I)*P(I-1))                                       00012630
C--OBTAIN SPLINE COEFFICIENTS                                           00012640
      A(NE+IT)=P(NE)                                                    00012650
      I=NE-1                                                            00012660
    5 A(I+IT)=P(I)-C(I)*A(I+IT+1)                                       00012670
      I=I-1                                                             00012680
       IF(I.GE.1) GO TO 5                                               00012690
      IF(IT.EQ.0) GO TO 6                                               00012700
      A(1)=D(1)                                                         00012710
      A(M)=D(2)                                                         00012720
    6 IF(H.EQ.0.) GO TO 14                                              00012730
      HH=1.0/H                                                          00012740
      DO 7 I=1,N                                                        00012750
      MUL=HH*(Y(I+1)-Y(I))                                              00012760
      B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))                               00012770
    7 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))                                 00012780
      RETURN                                                            00012790
C--UNEQUAL SPACING H=0.. AND IT=1                                       00012800
   11 DO 12 I=1,N                                                       00012810
   12 S(I+1)=X(I+1)-X(I)                                                00012820
      DO 13 I=1,NE                                                      00012830
      B(I)=2.0*(S(I+1)+S(I+2))                                          00012840
      C(I)=S(I+1)                                                       00012850
      A(I)=S(I+2)                                                       00012860
   13 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/     00012870
     $ (S(I+1)*S(I+2))                                                  00012880
      P(1)=P(1)-S(3)*D(1)                                               00012890
      P(NE)=P(NE)-S(N)*D(2)                                             00012900
```

```
      GO TO 3                                                          00012910
   14 DO 15 I=1,N                                                      00012920
      HH=1.0/S(I+1)                                                    00012930
      MUL=(Y(I+1)-Y(I))*HH**2                                          00012940
      B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH                                00012950
   15 C(I)=-2.0*MUL*HH+(A(I+1)+A(I))*HH**2                             00012960
      RETURN                                                           00012970
C--2ND DERIVATIVE BOUNDARIES GIVEN                                     00012980
   20 NE=N+1                                                           00012990
      IF(H) 999,31,21                                                  00013000
C--EQUAL SPACING H .GT. 0 AND IT=0                                     00013010
   21 HH=3.0/H                                                         00013020
      DO 22 I=2,N                                                      00013030
      B(I)=4.0                                                         00013040
      C(I)=1.0                                                         00013050
      A(I)=1.0                                                         00013060
   22 P(I)=HH*(Y(I+1)-Y(I-1))                                          00013070
      B(1)=2.0                                                         00013080
      B(NE)=2.0                                                        00013090
      C(1)=1.0                                                         00013100
      C(NE)=1.0                                                        00013110
      A(NE)=1.0                                                        00013120
      P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)                                   00013130
      P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)                                  00013140
      GO TO 3                                                          00013150
C--UNEQUAL SPACING H=0 AND IT=0                                        00013160
   31 DO 32 I=1,N                                                      00013170
   32 S(I+1)=X(I+1)-X(I)                                               00013180
      N1=N-1                                                           00013190
      DO 33 I=1,N1                                                     00013200
      B(I+1)=2.0*(S(I+1)+S(I+2))                                       00013210
      C(I+1)=S(I+1)                                                    00013220
      A(I+1)=S(I+2)                                                    00013230
   33 P(I+1)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/  00013240
     *      (S(I+1)*S(I+2))                                           00013250
      B(1)=2.0                                                         00013260
      B(NE)=2.0                                                        00013270
      C(1)=1.0                                                         00013280
      C(NE)=1.0                                                        00013290
      A(NE)=1.0                                                        00013300
      P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)                          00013310
      P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)                         00013320
      GO TO 3                                                          00013330
  999 M=-IABS(M)                                                       00013340
      RETURN                                                           00013350
      END                                                              00013360

      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)                             00013370
C--GIVEN CUBIC SPLINE COEFF'S A,B,C,AND M OBS.DATA ARRAYS X,Y          00013380
C  SPOINT EVALUATES THE PIECEWISE CUBIC SPLINE ORDINATE YY AT THE      00013390
C  ABSCISSA XX, WHERE XX IS IN THE CLOSED INTERVAL (X(1),X(M)).        00013400
C  NOTE: IF COMPUTING OVER EQUAL INTERVALS, USE THE SUBR 'CUBIC'       00013410
```

```
C  WHICH REQUIRES ONLY ONE CALL.                                          00013420
C                                                                        00013430
      DIMENSION X(1),Y(1),A(1),B(1),C(1)                                 00013440
      IF(XX.LT.X(1).OR.XX.GT.X(M)) GO TO 9                               00013450
      M1=M-1                                                             00013460
      DO 1 I=1,M1                                                        00013470
      J=I                                                                00013480
      IF(XX.LE.X(I+1)) GO TO 2                                           00013490
    1 CONTINUE                                                           00013500
    9 WRITE(6,60) XX,X(1),X(M)                                           00013510
   60 FORMAT('0ERROR IN SPOINT CALL--XX=',E16.8,' NOT IN CLOSED INTERVAL00013520
     * (',E16.8,',',E16.8,')')                                          00013530
      RETURN                                                             00013540
    2 Z=XX-X(J)                                                          00013550
      YY=Y(J)+((C(J)*Z+B(J))*Z+A(J))*Z                                   00013560
      RETURN                                                             00013570
      END                                                                00013580

        REAL FUNCTION RLAGF0(X,FUN,TOL,L,NEW)                            00013590
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE               00013600
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*COS(G*B)*DG' DEFINED AS THE    00013610
C  REAL FOURIER COSINE TRANSFORM WITH ARGUMENT X(=ALOG(B))               00013620
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND                00013630
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....        00013640
C                                                                        00013650
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.                     00013660
C                                                                        00013670
C--PARAMETERS:                                                           00013680
C                                                                        00013690
C     * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM00013700
C                'RLAGF0' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,   00013710
C                SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT,  00013720
C                THEN SUBPROGRAM 'RFOUR0' IS ADVISED FOR GENERAL USE.    00013730
C                (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).            00013740
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).      00013750
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN    00013760
C                 CALLING PROGRAM AND IN SUBPROGRAM FUN.                 00013770
C                 THE REAL FUNCTION FUN SHOULD BE A MONOTONE             00013780
C                 DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00013790
C       TOL=        REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,    00013800
C                   IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00013810
C                   THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,     00013820
C                   TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON      00013830
C                   THE FUNCTION FUN AND PARAMETER X...IN GENERAL,       00013840
C                 A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY' 00013850
C                 BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00013860
C                 RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00013870
C                 APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,  00013880
C                 ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...  00013890
C       L=          RESULTING NO. FILTER WTS. USED IN THE VARIABLE        00013900
C                   CONVOLUTION (L DEPENDS ON TOL AND FUN).              00013910
C                 MIN.L=24 AND MAX.L=281--WHICH COULD                    00013920
```

```
C                  OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING  00013930
C                  VERY FAST...                                          00013940
C     * NEW=     1 IS NECESSARY 1ST TIME OR BRAND NEW X.                 00013950
C                0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20        00013960
C                  IS ASSUMED INTERNALLY BY THIS ROUTINE.                00013970
C                  NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL              00013980
C                  STILL ASSUME X=(LAST X)-0.20 ANYWAY...                00013990
C                  IT IS THE USERS RESPONSIBILITY TO NORMALIZE           00014000
C                  BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).00014010
C                THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT      00014020
C                  TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A            00014030
C                  SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00014040
C                  ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...         00014050
C                  THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED            00014060
C                  KERNELS WILL BE USED IN THE LAGGED CONVOLUTION        00014070
C                  WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS    00014080
C                  WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)            00014090
C                                                                        00014100
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF0; THE FOURIER    00014110
C  TRANSFORM IS THEN RLAGF0/B WHICH IS TO BE COMPUTED AFTER EXIT FROM    00014120
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...           00014130
C                                                                        00014140
C--USAGE-- 'RLAGF0' IS CALLED AS FOLLOWS:                                00014150
C     ...                                                                00014160
C     EXTERNAL RF                                                        00014170
C     ...                                                                00014180
C     R=RLAGF0(ALOG(B),RF,TOL,L,NEW)/B                                   00014190
C     ...                                                                00014200
C     END                                                                00014210
C     REAL FUNCTION RF(G)                                                00014220
C     ...USER SUPPLIED CODE...                                           00014230
C     END                                                                00014240
C                                                                        00014250
C--NOTES:                                                                00014260
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM      00014270
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS        00014280
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....                               00014290
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION    00014300
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN            00014310
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,    00014320
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'      00014330
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING   00014340
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).     00014350
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                   00014360
C                  ...                                                   00014370
C                  NB=AINT(5.*ALOG(BMAX/BMIN))+1                         00014380
C                  NB1=NB+1                                              00014390
C                  X0=ALOG(BMAX)+.2                                      00014400
C                  NEW=1                                                 00014410
C                  DO 1 J=1,NB                                           00014420
C                  I=NB1-J                                               00014430
C                  X=X0-.2*J                                             00014440
```

```
C                ARG(I)=EXP(X)                                                  00014450
C                ANS(I)=RLAGF0(X,RF,TOL,L,NEW)/ARG(I)                           00014460
C      1         NEW=0                                                          00014470
C                ...                                                            00014480
C      (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR           00014490
C      ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,         00014500
C      TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)                 00014510
C      SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.                      00014520
C      (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY                 00014530
C      ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW                     00014540
C      BMAX,BMIN AND BY SETTING NEW=1....                                       00014550
C      (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE00014560
C                                                                               00014570
C                                                                               00014580
        DIMENSION KEY(281),SAVE(281)                                            00014590
        DIMENSION YT(281),Y1(76),Y2(76),Y3(76),Y4(53)                           00014600
        EQUIVALENCE (YT(1),Y1(1)),(YT(77),Y2(1)),(YT(153),Y3(1)),               00014610
     1 (YT(229),Y4(1))                                                          00014620
C--COS-EXTENDED FILTER WEIGHT ARRAYS:                                           00014630
       DATA Y1/                                                                 00014640
     1 5.1178101E-14, 2.9433849E-14, 2.5492522E-14, 1.9034819E-14,              00014650
     2 6.4179780E-14, 1.3085746E-15, 1.1989957E-13,-1.2216234E-14,              00014660
     3 1.7534103E-13, 7.9373498E-15, 2.1235658E-13, 7.9981520E-14,              00014670
     4 2.3815757E-13, 1.9714260E-13, 2.8920132E-13, 3.4161340E-13,              00014680
     5 4.0349917E-13, 5.2203885E-13, 5.9837223E-13, 7.8015306E-13,              00014690
     6 8.8911655E-13, 1.1709731E-12, 1.3165595E-12, 1.7578463E-12,              00014700
     7 1.9538564E-12, 2.6289768E-12, 2.9167697E-12, 3.9044344E-12,              00014710
     8 4.3927341E-12, 5.7526904E-12, 6.6569552E-12, 8.4555678E-12,              00014720
     9 1.0063229E-11, 1.2487964E-11, 1.5134682E-11, 1.8501488E-11,              00014730
     1 2.2720051E-11, 2.7452598E-11, 3.4025443E-11, 4.0875985E-11,              00014740
     2 5.0751668E-11, 6.1094382E-11, 7.5492982E-11, 9.1445759E-11,              00014750
     3 1.1227336E-10, 1.3676464E-10, 1.6720269E-10, 2.0423244E-10,              00014760
     4 2.4932743E-10, 3.0470661E-10, 3.7198526E-10, 4.5449934E-10,              00014770
     5 5.5502537E-10, 6.7793669E-10, 8.2810001E-10, 1.0112626E-09,              00014780
     6 1.2354800E-09, 1.5085255E-09, 1.8432253E-09, 2.2503397E-09,              00014790
     7 2.7499027E-09, 3.3569525E-09, 4.1025670E-09, 5.0077487E-09,              00014800
     8 6.1205950E-09, 7.4703399E-09, 9.1312760E-09, 1.1143911E-08,              00014810
     9 1.3622929E-08, 1.6623917E-08, 2.0324094E-08, 2.4798610E-08,              00014820
     1 3.0321709E-08, 3.6992986E-08, 4.5237482E-08, 5.5183434E-08/              00014830
       DATA Y2/                                                                 00014840
     1 6.7491070E-08, 8.2317946E-08, 1.0069271E-07, 1.2279375E-07,              00014850
     2 1.5022907E-07, 1.8316969E-07, 2.2413747E-07, 2.7322865E-07,              00014860
     3 3.3441046E-07, 4.0756197E-07, 4.9894278E-07, 6.0793233E-07,              00014870
     4 7.4443665E-07, 9.0679753E-07, 1.1107379E-06, 1.3525651E-06,              00014880
     5 1.6573073E-06, 2.0174273E-06, 2.4728798E-06, 3.0090445E-06,              00014890
     6 3.6838816E-06, 4.4879625E-06, 5.5059521E-06, 6.6935820E-06,              00014900
     7 8.2160716E-06, 9.9828691E-06, 1.2260527E-05, 1.4888061E-05,              00014910
     8 1.8296530E-05, 2.2202672E-05, 2.7305154E-05, 3.3109672E-05,              00014920
     9 4.0751046E-05, 4.9372484E-05, 6.0820947E-05, 7.3619571E-05,              00014930
     1 9.0780005E-05, 1.0976837E-04, 1.3550409E-04, 1.6365676E-04,              00014940
     2 2.0227521E-04, 2.4398338E-04, 3.0197018E-04, 3.6370760E-04,              00014950
     3 4.5083748E-04, 5.4213338E-04, 6.7315347E-04, 8.0800951E-04,              00014960
```

```
     4 1.0051938E-03, 1.2041401E-03, 1.5011708E-03, 1.7942344E-03,      00014970
     5 2.2421056E-03, 2.6730676E-03, 3.3490681E-03, 3.9815050E-03,      00014980
     6 5.0028666E-03, 5.9285668E-03, 7.4730905E-03, 8.8233510E-03,      00014990
     7 1.1160132E-02, 1.3119627E-02, 1.6653199E-02, 1.9472767E-02,      00015000
     8 2.4800811E-02, 2.8793704E-02, 3.6762063E-02, 4.2228780E-02,      00015010
     9 5.3905163E-02, 6.0804660E-02, 7.7081738E-02, 8.3874501E-02,      00015020
     1 1.0377190E-01, 1.0377718E-01, 1.1892208E-01, 9.0437429E-02/      00015030
      DATA Y3/                                                         00015040
     1 7.1685138E-02,-3.9473064E-02,-1.5078720E-01,-4.0489859E-01,      00015050
     2-5.6018995E-01,-6.8050388E-01,-1.5094224E-01, 6.6304064E-01,      00015060
     3 1.3766748E+00,-8.0373222E-01,-1.0869629E+00, 1.2812892E+00,      00015070
     4-5.0341082E-01,-4.4274455E-02, 2.0913102E-01,-1.9999661E-01,      00015080
     5 1.5207664E-01,-1.0920260E-01, 7.8169956E-02,-5.6651561E-02,      00015090
     6 4.1611799E-02,-3.0880012E-02, 2.3072559E-02,-1.7311631E-02,      00015100
     7 1.3021442E-02,-9.8085025E-03, 7.3943529E-03,-5.5769518E-03,      00015110
     8 4.2073164E-03,-3.1745026E-03, 2.3954154E-03,-1.8076122E-03,      00015120
     9 1.3640816E-03,-1.0293934E-03, 7.7682952E-04,-5.8623518E-04,      00015130
     1 4.4240399E-04,-3.3386183E-04, 2.5195025E-04,-1.9013541E-04,      00015140
     2 1.4348659E-04,-1.0828284E-04, 8.1716174E-05,-6.1667509E-05,      00015150
     3 4.6537684E-05,-3.5119887E-05, 2.6503388E-05,-2.0000904E-05,      00015160
     4 1.5093768E-05,-1.1390572E-05, 8.5959318E-06,-6.4869407E-06,      00015170
     5 4.8953713E-06,-3.6942830E-06, 2.7878625E-06,-2.1038241E-06,      00015180
     6 1.5875917E-06,-1.1980090E-06, 9.0398030E-07,-6.8208296E-07,      00015190
     7 5.1458650E-07,-3.8817581E-07, 2.9272267E-07,-2.2067921E-07,      00015200
     8 1.6623514E-07,-1.2514102E-07, 9.4034535E-08,-7.0556837E-08,      00015210
     9 5.2741581E-08,-3.9298610E-08, 2.9107255E-08,-2.1413893E-08,      00015220
     1 1.5742032E-08,-1.1498608E-08, 8.7561571E-09,-7.2959446E-09/      00015230
      DATA Y4/                                                         00015240
     1 6.8816619E-09,-8.9679825E-09, 1.4258275E-08,-1.9564299E-08,      00015250
     2 2.0235313E-08,-1.4725545E-08, 5.4632820E-09, 3.5995580E-09,      00015260
     3-9.5287133E-09, 1.1460041E-08,-1.0250532E-08, 7.4641748E-09,      00015270
     4-4.4703465E-09, 2.0499053E-09,-4.4806353E-10,-4.0374336E-10,      00015280
     5 7.0321001E-10,-6.7067960E-10, 4.9130404E-10,-2.8840747E-10,      00015290
     6 1.2373144E-10,-1.5260443E-11,-4.2027559E-11, 6.1885474E-11,      00015300
     7-5.9273937E-11, 4.6588766E-11,-3.2054182E-11, 1.9831637E-11,      00015310
     8-1.1210098E-11, 5.9567021E-12,-3.2427812E-12, 2.1353868E-12,      00015320
     9-1.8476851E-12, 1.8438474E-12,-1.8362842E-12, 1.7241847E-12,      00015330
     1-1.5161479E-12, 1.2627657E-12,-1.0129176E-12, 7.9578625E-13,      00015340
     2-6.2131435E-13, 4.8745900E-13,-3.8703630E-13, 3.1172547E-13,      00015350
     3-2.5397802E-13, 2.0824130E-13,-1.7123163E-13, 1.4113344E-13,      00015360
     4-1.1687986E-13, 9.7664016E-14,-8.2977176E-14, 7.2515267E-14,      00015370
     5-5.6047478E-14/                                                  00015380
C--$$ENDATA                                                            00015390
      IF(NEW) 10,30,10                                                 00015400
10    LAG=-1                                                           00015410
      X0=-X-30.30251236                                                00015420
      DO 20 IR=1,281                                                   00015430
20    KEY(IR)=0                                                        00015440
30    LAG=LAG+1                                                        00015450
      RLAGF0=0.0                                                       00015460
      CMAX=0.0                                                         00015470
      L=0                                                              00015480
```

```
      ASSIGN 110 TO M                                                   00015490
      I=149                                                             00015500
      GO TO 200                                                         00015510
110   CMAX=AMAX1(ABS(C),CMAX)                                           00015520
      I=I+1                                                             00015530
      IF(I.LE.170) GO TO 200                                            00015540
      IF(CMAX.EQ.0.0) GO TO 150                                         00015550
      CMAX=TOL*CMAX                                                     00015560
      ASSIGN 120 TO M                                                   00015570
      I=148                                                             00015580
      GO TO 200                                                         00015590
120   IF(ABS(C).LE.CMAX) GO TO 130                                      00015600
      I=I-1                                                             00015610
      IF(I.GT.0) GO TO 200                                              00015620
130   ASSIGN 140 TO M                                                   00015630
      I=171                                                             00015640
      GO TO 200                                                         00015650
140   IF(ABS(C).LE.CMAX) GO TO 190                                      00015660
      I=I+1                                                             00015670
      IF(I.LE.281) GO TO 200                                            00015680
      GO TO 190                                                         00015690
150   ASSIGN 160 TO M                                                   00015700
      I=1                                                               00015710
      GO TO 200                                                         00015720
160   IF(C.EQ.0.0) GO TO 170                                            00015730
      I=I+1                                                             00015740
      IF(I.LE.148) GO TO 200                                            00015750
170   ASSIGN 180 TO M                                                   00015760
      I=281                                                             00015770
      GO TO 200                                                         00015780
180   IF(C.EQ.0.0) GO TO 190                                            00015790
      I=I-1                                                             00015800
      IF(I.GE.171) GO TO 200                                            00015810
190   RETURN                                                            00015820
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)              00015830
200   LOOK=I+LAG                                                        00015840
      IQ=LOOK/282                                                       00015850
      IR=MOD(LOOK,282)                                                  00015860
      IF(IR.EQ.0) IR=1                                                  00015870
      IROLL=IQ*281                                                      00015880
      IF(KEY(IR).LE.IROLL) GO TO 220                                    00015890
210   C=SAVE(IR)*YT(I)                                                  00015900
      RLAGF0=RLAGF0+C                                                   00015910
      L=L+1                                                             00015920
      GO TO M,(110,120,140,160,180)                                     00015930
220   KEY(IR)=IROLL+IR                                                  00015940
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                             00015950
      GO TO 210                                                         00015960
      END                                                               00015970

        REAL FUNCTION RLAGF1(X,FUN,TOL,L,NEW)                           00015980
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE               00015990
```

```
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*SIN(G*B)*DG' DEFINED AS THE   00016000
C  REAL FOURIER SINE TRANSFORM WITH ARGUMENT X(=ALOG(B))                00016010
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND               00016020
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....       00016030
C                                                                       00016040
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.                    00016050
C                                                                       00016060
C--PARAMETERS:                                                          00016070
C                                                                       00016080
C     * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM00016090
C                'RLAGF1' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,  00016100
C                SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT, 00016110
C                THEN SUBPROGRAM 'RFOUR1' IS ADVISED FOR GENERAL USE.   00016120
C                (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).           00016130
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).     00016140
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN   00016150
C                   CALLING PROGRAM AND IN SUBPROGRAM FUN.              00016160
C                   THE REAL FUNCTION FUN SHOULD BE A MONOTONE          00016170
C                   DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00016180
C       TOL=        REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,   00016190
C                   IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00016200
C                   THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,    00016210
C                   TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON     00016220
C                   THE FUNCTION FUN AND PARAMETER X...IN GENERAL,      00016230
C                 A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY' 00016240
C                 BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00016250
C                 RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00016260
C                 APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,  00016270
C                 ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE... 00016280
C       L=          RESULTING NO. FILTER WTS. USED IN THE VARIABLE      00016290
C                   CONVOLUTION (L DEPENDS ON TOL AND FUN).             00016300
C                 MIN.L=20 AND MAX.L=266--WHICH COULD                   00016310
C                   OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING 00016320
C                   VERY FAST...                                        00016330
C     * NEW=      1 IS NECESSARY 1ST TIME OR BRAND NEW X.               00016340
C                 0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20      00016350
C                   IS ASSUMED INTERNALLY BY THIS ROUTINE.              00016360
C                   NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL            00016370
C                   STILL ASSUME X=(LAST X)-0.20 ANYWAY...              00016380
C                   IT IS THE USERS RESPONSIBILITY TO NORMALIZE          00016390
C                   BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).00016400
C                 THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT     00016410
C                   TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A          00016420
C                   SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00016430
C                   ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...       00016440
C                   THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED          00016450
C                   KERNELS WILL BE USED IN THE LAGGED CONVOLUTION      00016460
C                   WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS  00016470
C                   WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)          00016480
C                                                                       00016490
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF1; THE FOURIER   00016500
C  TRANSFORM IS THEN RLAGF1/B WHICH IS TO BE COMPUTED AFTER EXIT FROM   00016510
```

```
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...                00016520
C                                                                            00016530
C--USAGE-- 'RLAGF1' IS CALLED AS FOLLOWS:                                    00016540
C      ...                                                                   00016550
C      EXTERNAL RF                                                           00016560
C      ...                                                                   00016570
C      R=RLAGF1(ALOG(B),RF,TOL,L,NEW)/B                                      00016580
C      ...                                                                   00016590
C      END                                                                   00016600
C      REAL FUNCTION RF(G)                                                   00016610
C      ...USER SUPPLIED CODE...                                              00016620
C      END                                                                   00016630
C                                                                            00016640
C--NOTES:                                                                    00016650
C        (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM         00016660
C      BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS           00016670
C      ANY & ALL EXP-UNDERFLOW'S TO 0.0....                                  00016680
C      (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION       00016690
C      METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN               00016700
C      THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,       00016710
C      PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'         00016720
C      BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING      00016730
C      ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).         00016740
C      FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                      00016750
C                 ...                                                        00016760
C                 NB=AINT(5.*ALOG(BMAX/BMIN))+1                              00016770
C                 NB1=NB+1                                                   00016780
C                 X0=ALOG(BMAX)+.2                                           00016790
C                 NEW=1                                                      00016800
C                 DO 1 J=1,NB                                                00016810
C                 I=NB1-J                                                    00016820
C                 X=X0-.2*J                                                  00016830
C                 ARG(I)=EXP(X)                                              00016840
C                 ANS(I)=RLAGF1(X,RF,TOL,L,NEW)/ARG(I)                       00016850
C      1          NEW=0                                                      00016860
C                 ...                                                        00016870
C      (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR        00016880
C      ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,      00016890
C      TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)              00016900
C      SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.                   00016910
C      (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY              00016920
C      ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW                  00016930
C      BMAX,BMIN AND BY SETTING NEW=1....                                    00016940
C      (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE    00016950
C                                                                            00016960
C                                                                            00016970
        DIMENSION KEY(266),SAVE(266)                                         00016980
        DIMENSION WT(266),W1(76),W2(76),W3(76),W4(38)                        00016990
        EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)),            00017000
     1 (WT(229),W4(1))                                                       00017010
C--SIN-EXTENDED FILTER WEIGHT ARRAYS:                                        00017020
      DATA W1/                                                               00017030
```

```
1-1.1113940E-09,-1.3237246E-12, 1.5091739E-12,-1.6240954E-12,      00017040
2 1.7236636E-12,-1.8227727E-12, 1.9255992E-12,-2.0335514E-12,      00017050
3 2.1473541E-12,-2.2675549E-12, 2.3946842E-12,-2.5292661E-12,      00017060
4 2.6718110E-12,-2.8227693E-12, 2.9825171E-12,-3.1514006E-12,      00017070
5 3.3297565E-12,-3.5179095E-12, 3.7163306E-12,-3.9256378E-12,      00017080
6 4.1464798E-12,-4.3794552E-12, 4.6252131E-12,-4.8845227E-12,      00017090
7 5.1582809E-12,-5.4474462E-12, 5.7530277E-12,-6.0760464E-12,      00017100
8 6.4175083E-12,-6.7783691E-12, 7.1595239E-12,-7.5618782E-12,      00017110
9 7.9864477E-12,-8.4344110E-12, 8.9072422E-12,-9.4067705E-12,      00017120
1 9.9349439E-12,-1.0493731E-11, 1.1084900E-11,-1.1709937E-11,      00017130
2 1.2370354E-11,-1.3067414E-11, 1.3802200E-11,-1.4575980E-11,      00017140
3 1.5390685E-11,-1.6249313E-11, 1.7155934E-11,-1.8115250E-11,      00017150
4 1.9131898E-11,-2.0209795E-11, 2.1352159E-11,-2.2561735E-11,      00017160
5 2.3840976E-11,-2.5192263E-11, 2.6618319E-11,-2.8122547E-11,      00017170
6 2.9709129E-11,-3.1382870E-11, 3.3149030E-11,-3.5013168E-11,      00017180
7 3.6981050E-11,-3.9058553E-11, 4.1251694E-11,-4.3566777E-11,      00017190
8 4.6010537E-11,-4.8590396E-11, 5.1314761E-11,-5.4193353E-11,      00017200
9 5.7236720E-11,-6.0455911E-11, 6.3861222E-11,-6.7461492E-11,      00017210
1 7.1265224E-11,-7.5279775E-11, 7.9512249E-11,-8.3971327E-11/      00017220
 DATA W2/                                                         00017230
1 8.8668961E-11,-9.3621900E-11, 9.8851764E-11,-1.0438319E-10,      00017240
2 1.1024087E-10,-1.1644680E-10, 1.2301979E-10,-1.2997646E-10,      00017250
3 1.3733244E-10,-1.4510363E-10, 1.5330772E-10,-1.6196550E-10,      00017260
4 1.7110130E-10,-1.8074257E-10, 1.9091922E-10,-2.0166306E-10,      00017270
5 2.1300756E-10,-2.2498755E-10, 2.3763936E-10,-2.5100098E-10,      00017280
6 2.6511250E-10,-2.8001616E-10, 2.9575691E-10,-3.1238237E-10,      00017290
7 3.2994314E-10,-3.4849209E-10, 3.6808529E-10,-3.8878042E-10,      00017300
8 4.1063982E-10,-4.3372666E-10, 4.5811059E-10,-4.8386049E-10,      00017310
9 5.1105728E-10,-5.3977672E-10, 5.7011632E-10,-6.0215516E-10,      00017320
1 6.3601273E-10,-6.7175964E-10, 7.0955028E-10,-7.4942601E-10,      00017330
2 7.9161025E-10,-8.3606980E-10, 8.8317110E-10,-9.3270330E-10,      00017340
3 9.8533749E-10,-1.0404508E-09, 1.0993731E-09,-1.1605442E-09,      00017350
4 1.2267391E-09,-1.2942905E-09, 1.3691677E-09,-1.4429912E-09,      00017360
5 1.5288164E-09,-1.6077524E-09, 1.7085998E-09,-1.7890471E-09,      00017370
6 1.9129068E-09,-1.9857116E-09, 2.1491608E-09,-2.1926779E-09,      00017380
7 2.4312660E-09,-2.3959044E-09, 2.7872500E-09,-2.5610596E-09,      00017390
8 3.2762318E-09,-2.6082940E-09, 4.0261453E-09,-2.3560563E-09,      00017400
9 5.3176554E-09,-1.3960161E-09, 7.7708747E-09, 1.1853546E-09,      00017410
1 1.2760851E-08, 7.4264707E-09, 2.3342187E-08, 2.1869851E-08/      00017420
 DATA W3/                                                         00017430
1 4.6306744E-08, 5.4631686E-08, 9.6763087E-08, 1.2823337E-07,      00017440
2 2.0832812E-07, 2.9280540E-07, 4.5580888E-07, 6.5992437E-07,      00017450
3 1.0056815E-06, 1.4779183E-06, 2.2284335E-06, 3.2994604E-06,      00017460
4 4.9485823E-06, 7.3545473E-06, 1.1001083E-05, 1.6380539E-05,      00017470
5 2.4469550E-05, 3.6469246E-05, 5.4441527E-05, 8.1176726E-05,      00017480
6 1.2113828E-04, 1.8066494E-04, 2.6954609E-04, 4.0202288E-04,      00017490
7 5.9969995E-04, 8.9437312E-04, 1.3338166E-03, 1.9886697E-03,      00017500
8 2.9643943E-03, 4.4168923E-03, 6.5773518E-03, 9.7855105E-03,      00017510
9 1.4539361E-02, 2.1558670E-02, 3.1871864E-02, 4.6903518E-02,      00017520
1 6.8559512E-02, 9.9170152E-02, 1.4120770E-01, 1.9610835E-01,      00017530
2 2.6192603E-01, 3.2743321E-01, 3.6407406E-01, 3.1257559E-01,      00017540
3 9.0460168E-02,-3.6051039E-01,-8.6324760E-01,-8.1178720E-01,      00017550
```

```
     4 5.2205241E-01, 1.5449873E+00,-1.1817933E+00,-2.6759896E-01,      00017560
     5 8.0869203E-01,-6.2757149E-01, 3.4062630E-01,-1.5885304E-01,      00017570
     6 7.0472984E-02,-3.1624462E-02, 1.4894068E-02,-7.4821176E-03,      00017580
     7 4.0035936E-03,-2.2543784E-03, 1.3160358E-03,-7.8636604E-04,      00017590
     8 4.7658745E-04,-2.9125817E-04, 1.7885105E-04,-1.1012416E-04,      00017600
     9 6.7910334E-05,-4.1914054E-05, 2.5881544E-05,-1.5985851E-05,      00017610
     1 9.8751880E-06,-6.1008526E-06, 3.7692543E-06,-2.3287953E-06/      00017620
      DATA W4/                                                          00017630
     1 1.4388425E-06,-8.8899353E-07, 5.4926991E-07,-3.3937048E-07,      00017640
     2 2.0968284E-07,-1.2955437E-07, 8.0046336E-08,-4.9457371E-08,      00017650
     3 3.0557711E-08,-1.8880390E-08, 1.1665454E-08,-7.2076428E-09,      00017660
     4 4.4533423E-09,-2.7515696E-09, 1.7001092E-09,-1.0504494E-09,      00017670
     5 6.4904567E-10,-4.0102999E-10, 2.4778763E-10,-1.5310321E-10,      00017680
     6 9.4600354E-11,-5.8453314E-11, 3.6119400E-11,-2.2320056E-11,      00017690
     7 1.3793460E-11,-8.5242656E-12, 5.2675102E-12,-3.2543076E-12,      00017700
     8 2.0097689E-12,-1.2405412E-12, 7.6530538E-13,-4.7191929E-13,      00017710
     9 2.9084993E-13,-1.7923661E-13, 1.1018948E-13,-6.7885902E-14,      00017720
     1 4.2025050E-14,-2.1314731E-14/                                    00017730
C--$$ENDATA                                                             00017740
C                                                                       00017750
      IF(NEW) 10,30,10                                                  00017760
10    LAG=-1                                                            00017770
      X0=-X-38.30455704                                                 00017780
      DO 20 IR=1,266                                                    00017790
20    KEY(IR)=0                                                         00017800
30    LAG=LAG+1                                                         00017810
      RLAGF1=0.0                                                        00017820
      CMAX=0.0                                                          00017830
      L=0                                                               00017840
      ASSIGN 110 TO M                                                   00017850
      I=191                                                             00017860
      GO TO 200                                                         00017870
110   CMAX=AMAX1(ABS(C),CMAX)                                           00017880
      I=I+1                                                             00017890
      IF(I.LE.208) GO TO 200                                            00017900
      IF(CMAX.EQ.0.0) GO TO 150                                         00017910
      CMAX=TOL*CMAX                                                     00017920
      ASSIGN 120 TO M                                                   00017930
      I=190                                                             00017940
      GO TO 200                                                         00017950
120   IF(ABS(C).LE.CMAX) GO TO 130                                      00017960
      I=I-1                                                             00017970
      IF(I.GT.0) GO TO 200                                              00017980
130   ASSIGN 140 TO M                                                   00017990
      I=209                                                             00018000
      GO TO 200                                                         00018010
140   IF(ABS(C).LE.CMAX) GO TO 190                                      00018020
      I=I+1                                                             00018030
      IF(I.LE.266) GO TO 200                                            00018040
      GO TO 190                                                         00018050
150   ASSIGN 160 TO M                                                   00018060
      I=1                                                               00018070
```

```
      GO TO 200                                                          00018080
160   IF(C.EQ.0.0) GO TO 170                                             00018090
      I=I+1                                                              00018100
      IF(I.LE.190) GO TO 200                                             00018110
170   ASSIGN 180 TO M                                                    00018120
      I=266                                                              00018130
      GO TO 200                                                          00018140
180   IF(C.EQ.0.0) GO TO 190                                             00018150
      I=I-1                                                              00018160
      IF(I.GE.209) GO TO 200                                             00018170
190   RETURN                                                             00018180
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)               00018190
200   LOOK=I+LAG                                                         00018200
      IQ=LOOK/267                                                        00018210
      IR=MOD(LOOK,267)                                                   00018220
      IF(IR.EQ.0) IR=1                                                   00018230
      IROLL=IQ*266                                                       00018240
      IF(KEY(IR).LE.IROLL) GO TO 220                                     00018250
210   C=SAVE(IR)*WT(I)                                                   00018260
      RLAGF1=RLAGF1+C                                                    00018270
      L=L+1                                                              00018280
      GO TO M,(110,120,140,160,180)                                      00018290
220   KEY(IR)=IROLL+IR                                                   00018300
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                              00018310
      GO TO 210                                                          00018320
      END                                                                00018330

      COMPLEX FUNCTION FG2(G)                                            00018340
C-- F(G)*G*G KERNEL USED BY PROGRAM 'EMLOOPS' FOR THE                    00018350
C   GROUND CASE (A=0). NOTE:  FG2 IS USED IN T0,T1 INTEGRALS             00018360
C   VIA SUBR 'ZHANKS'.                                                   00018370
C                                                                        00018380
      COMPLEX V1,F1,C,ONE,TWO                                            00018390
      DATA ONE,TWO/(1.0,0.0),(2.0,0.0)/                                  00018400
      C=CMPLX(G,0.)                                                      00018410
      CALL RECUR1(G,V1,F1)                                               00018420
C//      FG2=(TWO*V1*C*C*C*(F1-ONE))/((C+V1)*(C+V1*F1))                  00018430
C  ON MULTICS, REWRITE AS:                                               00018440
      FG2=TWO*V1*(C/(C+V1))*(C/(C+V1*F1))*(F1-ONE)*C                     00018450
      RETURN                                                             00018460
      END                                                                00018470

      COMPLEX FUNCTION PFBJG(G)                                          00018480
C-- PARTIAL OF (F W/R B(J),J>=1)*G.                                      00018490
C  J IS GIVEN IN COMMON/PART/J,ISEP ALONG WITH OTHER                     00018500
C  COMMON PARAMETERS.                                                    00018510
C                                                                        00018520
      COMPLEX V1,F1,C,T0,T1,T2,CB,CB2,CB3,CA,ONESG1,TWOSG1,              00018530
     & ZZ0,PF1,TWO,I0K0,I1K1,IKDIF                                       00018540
      COMMON/SHARE/FILL(4),XX,YY,YY2,RHO,RHO2,FILL2,BB,FILL3,            00018550
     & DEL,DEL2,IREST(3)                                                 00018560
      COMMON/CTL/T0,T1,T2,CB,CB2,CB3,CA,ONESG1,TWOSG1,I0K0,I1K1,IKDIF,   00018570
```

```
     & ZZ0,AMP,FREQ,SIG1,H,EPS,IOB,M1,M21,ILOOPS,IMM                    00018580
      COMMON/PART/J,ISEP                                               00018590
      DATA TWO/(2.0,0.0)/                                              00018600
      CALL RECURF(G,DEL,SIG1,V1,F1,PF1,J)                              00018610
      C=G                                                              00018620
      PFBJG=TWO*C*V1*PF1*C/(C+V1*F1)**2                                00018630
      RETURN                                                           00018640
      END                                                              00018650

      COMPLEX FUNCTION PFBJG2(G)                                       00018660
C-- PARTIAL OF (F W/R B(J),J>=1)*G**2.                                 00018670
C                                                                      00018680
      COMPLEX PFBJG                                                    00018690
      PFBJG2=G*PFBJG(G)                                                00018700
      RETURN                                                           00018710
      END                                                              00018720

      REAL FUNCTION HCLOOP(B2)                                         00018730
C--COSINE-TRANSFORM KERNEL FOR HORIZONTAL COPLANAR LOOP-LOOP           00018740
C  TRANSIENT FAST FORWARD SOLUTION (A=0 GROUND CASE)                   00018750
C                                                                      00018760
      COMPLEX ZHANKS,ONEI,ONE1,TWO2,NINE,NINE9,EIGHTI,CB,CB2,CB3,Z     00018770
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN     00018780
      COMMON/LOOP/D(10),RR,EPS,B0,BM,M1,IHALF                          00018790
      COMMON/MODEL/RK(10),DD(9),MM                                     00018800
      EXTERNAL FG2                                                     00018810
      DATA ONE1/(1.,1.)/,ONEI/(0.,1.)/,TWO2/(-2.,2.)/,NINE/(9.,0.)/,   00018820
     & NINE9/(9.,9.)/,EIGHTI/(0.,8.)/                                  00018830
      B=SQRT(B2)                                                       00018840
      IF(B.LT.B0) GO TO 2                                              00018850
      IF(B.GT.BM) GO TO 3                                              00018860
      IF(ISPLN.EQ.1) GO TO 4                                           00018870
      CB=B                                                             00018880
      CB2=B2                                                           00018890
      CB3=CB*CB2                                                       00018900
      Z=(0.0,0.0)                                                      00018910
      IF(IHALF.EQ.1) Z=-ONEI*(NINE-(NINE+NINE9*CB+EIGHTI*CB2+TWO2*CB3)* 00018920
     & CEXP(-CB*ONE1))/CB2                                             00018930
      IF(MM.EQ.1) GO TO 1                                              00018940
      DEL=RR/B                                                         00018950
      DO 10 I=1,M1                                                     00018960
10    DD(I)=2.0*D(I)/DEL                                               00018970
      Z=Z+CB3*ZHANKS(0,B,FG2,EPS,LL,1)                                 00018980
1     HCLOOP=REAL(Z)                                                   00018990
      RETURN                                                           00019000
2     IF(IHALF.EQ.0) GO TO 3                                           00019010
      HCLOOP=1.0                                                       00019020
      RETURN                                                           00019030
3     HCLOOP=0.0                                                       00019040
      RETURN                                                           00019050
C--ISPLN=1 (0<NB<12 OPTION)                                            00019060
C  INTERPOLATE PRE-SPLINED FREQ FUNCTION                               00019070
```

```
4      CALL SPOINT(NS,XS,YS,AS,BS,CS,B,HCLOOP)                          00019080
       RETURN                                                           00019090
       END                                                              00019100

       REAL FUNCTION HSLOOP(B2)                                         00019110
C--SINE-TRANSFORM KERNEL FOR HORIZONTAL COPLANAR LOOP-LOOP              00019120
C  TRANSIENT STEP RESPONSE FORWARD SOLUTION                             00019130
C--CALLS FUNCTION HCLOOP (IMPULSE RESPONSE KERNEL)                      00019140
C                                                                       00019150
       HSLOOP=HCLOOP(B2)/B2                                             00019160
       RETURN                                                           00019170
       END                                                              00019180

       REAL FUNCTION PHC(B2)                                            00019190
C--COSINE-TRANSFORM KERNEL FOR PARTIALS OF HORIZONTAL COPLANAR LOOP-LOOP00019200
C  TRANSIENT FOR THE GROUND CASE (A=0)                                  00019210
C  WITH RESPECT TO PARAMETER INDEX IN COMMON/PART/                      00019220
C  (USED IN PROGRAM 'MARQ_TRANS_HCLOOP')                                00019230
C                                                                       00019240
       REAL K(10),DD(9),D(10)                                           00019250
       COMPLEX ZHANKS,CB,CB2,CB3,ZTEMP,T0,T1,THREE,TWOSG1,C13(13),      00019260
      & ONEI,ONE1,ONEM1,TWO,TWO2,NINE,NINE9,EIGHTI,Z                    00019270
       COMMON/PASS/SIG1,TMIN,TMAX,T00,TM,DB,                            00019280
      * ISTEP,M21,M2,JSPLN,NN,IFIRST                                    00019290
       COMMON/LOOP/D,RR,EPS,B0,BM,M1,IHALF                              00019300
       COMMON/SPLN/XP(200),YP(200),AP(200),BP(200),CP(200),NP,ISPLN     00019310
       COMMON/MODEL/K,DD,M                                              00019320
       COMMON/SHARE/FILL(12),DEL,FILL1,IREST(3)                         00019330
       COMMON/CTL/C13,FILL2(2),SSIG1,FILL3(2),JREST(5)                  00019340
       COMMON/PART/INDEX,JUNK                                           00019350
       EXTERNAL FG2,PFBJG2                                              00019360
       DATA THREE/(3.0,0.0)/,ONEI/(0.0,1.0)/,ONE1/(1.0,1.0)/,           00019370
      & ONEM1/(1.,-1.)/,TWO/(2.,0.)/,TWO2/(-2.,2.)/,                   00019380
      & NINE/(9.,0.)/,NINE9/(9.,9.)/,EIGHTI/(0.,8.)/                   00019390
       B=SQRT(B2)                                                       00019400
       IF(B.LT.B0.OR.B.GT.BM) GO TO 100                                 00019410
       IF(ISPLN.EQ.1) GO TO 60                                          00019420
       CB=B                                                             00019430
       CB2=B2                                                           00019440
       CB3=CB*CB2                                                       00019450
       SSIG1=SIG1                                                       00019460
       DEL=RR/B                                                         00019470
       TWOSG1=CMPLX(2.*SIG1,0.)                                         00019480
       IF(M.EQ.1) GO TO 10                                              00019490
       DO 1 I=1,M1                                                      00019500
1      DD(I)=2.*D(I)/DEL                                                00019510
10     IF(INDEX.GT.1) GO TO 40                                          00019520
       ZTEMP=(0.0,0.0)                                                  00019530
       IF(M1.EQ.0) GO TO 32                                             00019540
       T0=ZHANKS(0,B,FG2,EPS,LW,1)                                      00019550
       T1=ZHANKS(1,B,FG2,EPS,LW,0)                                      00019560
       ZTEMP=CB3*(ZHANKS(0,B,PFBJG2,EPS,LW,1)-                          00019570
```

```
     & (CB*T1-THREE*T0)/TWOSG1)                                          00019580
32    Z=CEXP(-CB*ONE1)                                                   00019590
      Z=ZTEMP-ONEI*(Z*(ONEI-ONEM1*CB-TWO*CB2)-(NINE-                     00019600
     & (NINE+NINE9*CB+EIGHTI*CB2+TWO2*CB3)*Z)/CB2)/CMPLX(SIG1,0.0)       00019610
      GO TO 50                                                           00019620
40    Z=CB3*ZHANKS(0,B,PFBJG2,EPS,LW,1)                                  00019630
50    PHC=REAL(Z)                                                        00019640
      RETURN                                                             00019650
C--ISPLN=1 (0<NB<12 OPTION)                                              00019660
C  INTERPOLATE PARTIAL DER. PRE-SPLINED FREQ FUNCTION                    00019670
60    CALL SPOINT(NP,XP,YP,AP,BP,CP,B,PHC)                               00019680
      RETURN                                                             00019690
100   PHC=0.0                                                            00019700
      RETURN                                                             00019710
      END                                                                00019720

      REAL FUNCTION PHS(B2)                                              00019730
C--SINE-TRANSFROM KERNEL FOR PARTIALS OF HORIZONTAL COPLANAR LOOP-LOOP   00019740
C  TRANSIENT FOR THE GROUND CASE (A=0)                                   00019750
C--CALLS PHC KERNEL FUNCTION                                             00019760
C  (USED IN PROGRAM 'MARQ_TRANS_HCLOOP')                                 00019770
C                                                                        00019780
      PHS=PHC(B2)/B2                                                     00019790
      RETURN                                                             00019800
      END                                                                00019810

      REAL FUNCTION ERF(X)                                               00019820
C                                                                        00019830
C        ERF COMPUTES THE ERROR FUNCTION TO SIX                          00019840
C        PLACE ACCURACY.                                                 00019850
C                                                                        00019860
      DIMENSION A(5)                                                     00019870
      DATA A/0.254829592,-0.284496736,1.421413741,-1.453152027,          00019880
     1 1.061405429/                                                      00019890
      T=1./(1.+0.3275911*X)                                              00019900
      ERF=T*(A(1)+T*(A(2)+T*(A(3)+T*(A(4)+T*A(5)))))                     00019910
      ERF=1.-ERF*EXP_(-X*X)                                              00019920
      RETURN                                                             00019930
      END                                                                00019940

      REAL FUNCTION RFLAGS(N,FUN,TOL,T0,TM,T,NEW)                        00019950
C--FOURIER TRANSFORM LAG CONVOLUTION & SPLINE INTERPOLATION               00019960
C  GIVES FOURIER COSINE OR SINE TRANSFORMS VIA RLAGF0,RLAGF1             00019970
C  REF: ANDERSON,1975,NTIS REPT. PB-242-800,P.76-87.                     00019980
C                                                                        00019990
C      N = 0 FOR COSINE TRANSFORM (VIA RLAGF0)                           00020000
C      N = 1 FOR SINE TRANSFORM (VIA RLAGF1)                             00020010
C    FUN = EXTERNAL REAL KERNEL FUNCTION.                                00020020
C    TOL = TOLERANCE REQUESTED FOR RLAGF0 OR RLAGF1                      00020030
C     T0 = TMIN TO USE (E.G., LET T0=.5*TMIN, TMIN=TRUE)                 00020040
C     TM = TMAX TO USE (TM>T0)                                           00020050
C      T = TRANSFORM PARAMETER (T0<=T<=TM) FOR THIS CALL (NEW=1 OR 0)    00020060
```

```
C     NEW = 1 REQUIRED FOR 1ST CALL OR TO RESET SPLINE COEFFICIENTS.     00020070
C     NEW = 0 FOR ALL CALLS AFTER 1ST--USES SPLINE INTERPOLATION ONLY.   00020080
C                                                                        00020090
      REAL ARG(200),Y(200),AR(200),BR(200),CR(200),                      00020100
     & D(2),W1(200),W2(200)                                              00020110
      EXTERNAL FUN                                                       00020120
      DATA D/2*0.0/                                                      00020130
      IF(NEW.EQ.0) GO TO 3                                               00020140
      NT=AINT(5.*ALOG(TM/T0))+5                                          00020150
      IF(NT.GT.200)CALL ERRMSG('IN RFLAGS: NT>200   ',4,6,16)            00020160
      NT1=NT+1                                                           00020170
      X0=ALOG(T0)+.2*NT                                                  00020180
      NU=1                                                               00020190
      DO 1 J=1,NT                                                        00020200
      I=NT1-J                                                            00020210
      X=X0-.2*J                                                          00020220
      EX=EXP(X)                                                          00020230
      ARG(I)=EX                                                          00020240
      IF(N.EQ.0) Y(I)=RLAGF0(X,FUN,TOL,L,NU)/EX                          00020250
      IF(N.NE.0) Y(I)=RLAGF1(X,FUN,TOL,L,NU)/EX                          00020260
1     NU=0                                                               00020270
      CALL SPLIN1(NT,0.0,ARG,Y,AR,BR,CR,0,D,W1,W2)                       00020280
2     IF(NT.LT.0) CALL ERRMSG('IN RFLAGS: NT<0 AFTER SPLIN1  ',6,6,16)   00020290
3     IF(T.LT.T0) CALL ERRMSG('IN RFLAGS: T<T0',3,6,16)                  00020300
      IF(T.GT.TM) CALL ERRMSG('IN RFLAGS: T>TM',3,6,16)                  00020310
      CALL SPOINT(NT,ARG,Y,AR,BR,CR,T,X)                                 00020320
      RFLAGS=X                                                           00020330
      RETURN                                                             00020340
      END                                                                00020350

      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                        00020360
C=======================================================================00020370
C  COMPLEX HANKEL TRANSFORMS OF ORDER 0 OR 1 FOR RELATED (SAVED) KERNELS00020380
C  AND FIXED TRANSFORM ARGUMENT B.GT.0.                                  00020390
C                                                                        00020400
C--REF: ANDERSON, W.L., 1979 (IN PRESS), GEOPHYSICS.                     00020410
C                                                                        00020420
C--SUBPROGRAM ZHANKS EVALUATES THE INTEGRAL FROM 0 TO INFINITY OF        00020430
C  FUN(G)*JN(G*B)*DG, DEFINED AS THE COMPLEX HANKEL TRANSFORM OF         00020440
C  ORDER N (=0 OR 1) AND TRANSFORM ARGUMENT B.GT.0.  THE METHOD IS BY    00020450
C  ADAPTIVE DIGITAL FILTERING OF THE COMPLEX KERNEL FUNCTION FUN,        00020460
C  USING DIRECT AND/OR PREVIOUSLY SAVED KERNEL FUNCTION VALUES.          00020470
C                                                                        00020480
C--PARAMETERS (ALL INPUT, EXCEPT NF)                                     00020490
C                                                                        00020500
C     N      = ORDER (=0 OR 1) OF THE HANKEL TRANSFORM TO BE EVALUATED.  00020510
C     B      = REAL TRANSFORM ARGUMENT B.GT.0.0 OF THE HANKEL TRANSFORM. 00020520
C              IF NEW=0, B IS ASSUMED EQUAL TO THE LAST B USED WHEN NEW=100020530
C              (SEE PARAMETER NEW AND SUBPROGRAM USAGE BELOW).           00020540
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)    00020550
C              OF A REAL ARGUMENT G.GT.0. THIS REFERENCE MUST BE SUPPLIED00020560
C              EVEN WHEN NEW=0, SINCE THE ADAPTIVE CONVOLUTION           00020570
```

```
C              MAY NEED SOME DIRECT FUNCTION CALLS (E.G. IF TOL REDUCED).00020580
C              IF PARAMETERS OTHER THAN G ARE REQUIRED IN FUN, USE COMMON00020590
C              IN THE CALLING PROGRAM AND IN SUBPROGRAM FUN.  BOTH       00020600
C              REAL AND IMAGINARY PARTS OF THE COMPLEX FUNCTION FUN(G)   00020610
C              MUST BE CONTINUOUS BOUNDED FUNCTIONS FOR G.GT.0.0. FOR A  00020620
C              REAL FUNCTION F1(G), FUN=CMPLX(F1(G),0.0) MAY BE USED.    00020630
C              TWO INDEPENDENT REAL-FUNCTIONS F1(G),F2(G) MAY BE         00020640
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)). 00020650
C     TOL    = REQUESTED REAL TRUNCATION TOLERANCE ACCEPTED AT THE FILTER00020660
C              TAILS FOR ADAPTIVE FILTERING.  A TRUNCATION CRITERION IS  00020670
C              DEFINED DURING CONVOLUTION IN A FIXED ABSCISSA RANGE AS   00020680
C              THE MAX. ABSOLUTE CONVOLVED PRODUCT TIMES TOL.  TYPICALLY,00020690
C              TOL.LE.0.00001 WOULD GIVE ABOUT .01 PER CENT ACCURACY     00020700
C              FOR WELL-BEHAVED KERNELS AND MODERATE VALUES OF B.  FOR   00020710
C              VERY LARGE OR SMALL B, A VERY SMALL TOL SHOULD BE USED.   00020720
C              IN GENERAL, DECREASING THE TOLERANCE WOULD PRODUCE HIGHER 00020730
C              ACCURACY IN THE CONVOLUTION SINCE MORE FILTER WEIGHTS ARE 00020740
C              USED (UNLESS EXPONENT UNDERFLOWS OCCUR IN THE KERNEL      00020750
C              EVALUATION -- SEE NOTE (1) BELOW).                        00020760
C              FOR MAXIMUM ACCURACY POSSIBLE, TOL=0.0 MAY BE USED.       00020770
C     NF     = TOTAL NUMBER OF DIRECT FUN CALLS USED DURING CONVOLUTION  00020780
C              FOR ANY VALUE OF NEW (NF IS AN OUTPUT PARAMETER).         00020790
C              NF IS IN THE RANGE 21.LE.NF.LE.283 WHEN NEW=1.  USUALLY,  00020800
C              NF IS MUCH LESS THAN 283 (OR 0) WHEN NEW=0.               00020810
C     NEW   =1 IS REQUIRED FOR THE VERY FIRST CALL TO ZHANKS, OR IF      00020820
C              FORCING DIRECT FUNCTION FUN(G) CALLS, E.G., IF USING      00020830
C              ZHANKS FOR UNRELATED KERNELS.                             00020840
C              NEW=1 INITIALIZES COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE 00020850
C              FOR NSAVE COMPLEX KERNEL VALUES IN FSAVE AND CORRESPONDING00020860
C              REAL ARGUMENTS IN GSAVE FOR THE GIVEN PARAMETER B.        00020870
C     NEW   =0 TO USE RELATED KERNELS (MODIFIED BY USER) CURRENTLY STORED00020880
C              IN COMMON/SAVE/. FUN IS CALLED ONLY IF REQUIRED           00020890
C              DURING THE CONVOLUTION.  ADDITIONAL FUNCTION VALUES WHEN  00020900
C              NEEDED ARE AUTOMATICALLY ADDED TO THE COMMON/SAVE/ BLOCK. 00020910
C                                                                        00020920
C     ******* NOTE THAT IT IS THE USERS RESPONSIBILITY TO MODIFY THE     00020930
C              COMMON FSAVE() VALUES FOR NEW=0 CALLS, EXTERNALLY IN      00020940
C              THE USERS CALLING PROGRAM (SEE SUBPROGRAM USAGE BELOW).   00020950
C                                                                        00020960
C========================================================================00020970
C--SUBPROGRAM USAGE-- ZHANKS IS CALLED AS FOLLOWS                        00020980
C     ...                                                                00020990
C     COMPLEX Z1,Z2,ZHANKS,FSAVE                                         00021000
C     COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                            00021010
C     EXTERNAL ZF1,ZF2                                                   00021020
C     ...                                                                00021030
C     Z1=ZHANKS(N1,B,ZF1,TOL,NF1,1)                                      00021040
C     DO 1 I=1,NSAVE                                                     00021050
C  C--MODIFY FSAVE IN COMMON/SAVE/ TO OBTAIN RELATED ZF2 FROM ZF1.       00021060
C  C--E.G. FSAVE(I)=GSAVE(I)*FSAVE(I) -- FOR RELATION ZF2(G)=G*ZF1(G)    00021070
C    1 CONTINUE                                                         00021080
C     Z2=ZHANKS(N2,B,ZF2,TOL,NF2,0)                                      00021090
```

```
C      ...                                                              00021100
C      END                                                              00021110
C      COMPLEX FUNCTION ZF1(G)                                          00021120
C      ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF1(G), G.GT.0.   00021130
C      END                                                              00021140
C      COMPLEX FUNCTION ZF2(G)                                          00021150
C      ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF2(G), G.GT.0.   00021160
C      END                                                              00021170
C=====================================================================00021180
C--NOTES                                                               00021190
C        (1).  EXP-UNDERFLOW MAY OCCUR IN EXECUTING THIS SUBPROGRAM.   00021200
C              THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS 00021210
C              EXP-UNDERFLOW TO 0.0.                                    00021220
C        (2).  ANSI FORTRAN (AMERICAN STANDARD X3.9-1966) IS USED, EXCEPT00021230
C              DATA STATEMENTS MAY NEED TO BE CHANGED FOR SOME COMPILERS.00021240
C              TO CONVERT ZHANKS TO THE NEW AMERICAN STANDARD FORTRAN   00021250
C              (X3.9-1978), ADD THE FOLLOWING DECLARATION TO THIS ROUTINE00021260
C              SAVE Y1,ISAVE                                            00021270
C        (3).  THE FILTER ABSCISSA CORRESPONDING TO EACH FILTER WEIGHT  00021280
C              IS GENERATED IN DOUBLE-PRECISION (TO REDUCE ROUND-OFF),  00021290
C              BUT IS USED IN SINGLE-PRECISION IN FUNCTION FUN.         00021300
C        (4).  NO CHECKS ARE MADE ON CALLING PARAMETERS (TO SAVE TIME), 00021310
C              HENCE UNPREDICTABLE RESULTS COULD OCCUR IF ZHANKS        00021320
C              IS CALLED INCORRECTLY (OR IF FUN OR COMMON IS IN ERROR). 00021330
C=====================================================================00021340
C                                                                      00021350
      COMPLEX FUN,C,CMAX,FSAVE                                         00021360
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                          00021370
      DOUBLE PRECISION E,ER,Y1,Y                                       00021380
      DIMENSION T(2),TMAX(2)                                           00021390
      DIMENSION WT0(283),WA0(76),WB0(76),WC0(76),WD0(55),              00021400
     * WT1(283),WA1(76),WB1(76),WC1(76),WD1(55)                       00021410
      EQUIVALENCE (WT0(1),WA0(1)),(WT0(77),WB0(1)),(WT0(153),WC0(1)),  00021420
     * (WT0(229),WD0(1)),(WT1(1),WA1(1)),(WT1(77),WB1(1)),             00021430
     * (WT1(153),WC1(1)),(WT1(229),WD1(1))                             00021440
      EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                              00021450
C-----E=DEXP(.2D0), ER=1.0D0/E                                         00021460
      DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/       00021470
C--J0-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT0 ARRAY)         00021480
      DATA WA0/                                                        00021490
     * 2.1969101E-11, 4.1201161E-09,-6.1322980E-09, 7.2479291E-09,     00021500
     *-7.9821627E-09, 8.5778983E-09,-9.1157294E-09, 9.6615250E-09,     00021510
     *-1.0207546E-08, 1.0796633E-08,-1.1393033E-08, 1.2049873E-08,     00021520
     *-1.2708789E-08, 1.3446466E-08,-1.4174300E-08, 1.5005577E-08,     00021530
     *-1.5807160E-08, 1.6747136E-08,-1.7625961E-08, 1.8693427E-08,     00021540
     *-1.9650840E-08, 2.0869789E-08,-2.1903555E-08, 2.3305308E-08,     00021550
     *-2.4407377E-08, 2.6033678E-08,-2.7186773E-08, 2.9094334E-08,     00021560
     *-3.0266804E-08, 3.2534013E-08,-3.3672072E-08, 3.6408936E-08,     00021570
     *-3.7425022E-08, 4.0787921E-08,-4.1543242E-08, 4.5756842E-08,     00021580
     *-4.6035233E-08, 5.1425075E-08,-5.0893896E-08, 5.7934897E-08,     00021590
     *-5.6086570E-08, 6.5475248E-08,-6.1539913E-08, 7.4301996E-08,     00021600
     *-6.7117043E-08, 8.4767837E-08,-7.2583120E-08, 9.7366568E-08,     00021610
```

```
*-7.7553611E-08, 1.1279873E-07,-8.1416723E-08, 1.3206914E-07,      00021620
*-8.3217217E-08, 1.5663185E-07,-8.1482581E-08, 1.8860593E-07,      00021630
*-7.3963141E-08, 2.3109673E-07,-5.7243707E-08, 2.8867452E-07,      00021640
*-2.6163525E-08, 3.6808773E-07, 2.7049871E-08, 4.7932617E-07,      00021650
* 1.1407365E-07, 6.3720626E-07, 2.5241961E-07, 8.6373487E-07,      00021660
* 4.6831433E-07, 1.1916346E-06, 8.0099716E-07, 1.6696015E-06,      00021670
* 1.3091334E-06, 2.3701475E-06, 2.0803829E-06, 3.4012978E-06/      00021680
 DATA WB0/                                                        00021690
* 3.2456774E-06, 4.9240402E-06, 5.0005198E-06, 7.1783540E-06,      00021700
* 7.6367633E-06, 1.0522038E-05, 1.1590021E-05, 1.5488635E-05,      00021710
* 1.7510398E-05, 2.2873836E-05, 2.6368006E-05, 3.3864387E-05,      00021720
* 3.9610390E-05, 5.0230379E-05, 5.9397373E-05, 7.4612122E-05,      00021730
* 8.8951409E-05, 1.1094809E-04, 1.3308026E-04, 1.6511335E-04,      00021740
* 1.9895671E-04, 2.4587195E-04, 2.9728181E-04, 3.6629770E-04,      00021750
* 4.4402013E-04, 5.4589361E-04, 6.6298832E-04, 8.1375348E-04,      00021760
* 9.8971624E-04, 1.2132772E-03, 1.4772052E-03, 1.8092022E-03,      00021770
* 2.2045122E-03, 2.6980811E-03, 3.2895354E-03, 4.0238764E-03,      00021780
* 4.9080203E-03, 6.0010999E-03, 7.3216878E-03, 8.9489225E-03,      00021790
* 1.0919448E-02, 1.3340696E-02, 1.6276399E-02, 1.9873311E-02,      00021800
* 2.4233627E-02, 2.9555699E-02, 3.5990069E-02, 4.3791529E-02,      00021810
* 5.3150319E-02, 6.4341372E-02, 7.7506720E-02, 9.2749987E-02,      00021820
* 1.0980561E-01, 1.2791555E-01, 1.4525830E-01, 1.5820085E-01,      00021830
* 1.6058576E-01, 1.4196085E-01, 8.9781222E-02,-1.0238278E-02,      00021840
*-1.5083434E-01,-2.9059573E-01,-2.9105437E-01,-3.7973244E-02,      00021850
* 3.8273717E-01, 2.2014118E-01,-4.7342635E-01, 1.9331133E-01,      00021860
* 5.3839527E-02,-1.1909845E-01, 9.9317051E-02,-6.6152628E-02,      00021870
* 4.0703241E-02,-2.4358316E-02, 1.4476533E-02,-8.6198067E-03/      00021880
 DATA WC0/                                                        00021890
* 5.1597053E-03,-3.1074602E-03, 1.8822342E-03,-1.1456545E-03,      00021900
* 7.0004347E-04,-4.2904226E-04, 2.6354444E-04,-1.6215439E-04,      00021910
* 9.9891279E-05,-6.1589037E-05, 3.7996921E-05,-2.3452250E-05,      00021920
* 1.4479572E-05,-8.9417427E-06, 5.5227518E-06,-3.4114252E-06,      00021930
* 2.1074101E-06,-1.3019229E-06, 8.0433617E-07,-4.9693681E-07,      00021940
* 3.0702417E-07,-1.8969219E-07, 1.1720069E-07,-7.2412496E-08,      00021950
* 4.4740283E-08,-2.7643004E-08, 1.7079403E-08,-1.0552634E-08,      00021960
* 6.5200311E-09,-4.0284597E-09, 2.4890232E-09,-1.5378695E-09,      00021970
* 9.5019040E-10,-5.8708696E-10, 3.6273937E-10,-2.2412348E-10,      00021980
* 1.3847792E-10,-8.5560821E-11, 5.2865474E-11,-3.2664392E-11,      00021990
* 2.0182948E-11,-1.2470979E-11, 7.7057678E-12,-4.7611713E-12,      00022000
* 2.9415274E-12,-1.8170081E-12, 1.1221034E-12,-6.9271067E-13,      00022010
* 4.2739744E-13,-2.6344388E-13, 1.6197105E-13,-9.9147443E-14,      00022020
* 6.0487998E-14,-3.6973097E-14, 2.2817964E-14,-1.4315547E-14,      00022030
* 9.1574735E-15,-5.9567236E-15, 3.9209969E-15,-2.5911739E-15,      00022040
* 1.6406939E-15,-8.8248590E-16, 3.0195409E-16, 2.2622634E-17,      00022050
*-8.0942556E-17,-3.7172363E-17, 1.9299542E-16,-3.3388160E-16,      00022060
* 4.6174116E-16,-5.8627358E-16, 7.2227767E-16,-8.7972941E-16,      00022070
* 1.0211793E-15,-1.0940039E-15, 1.0789555E-15,-9.7089714E-16/      00022080
 DATA WD0/                                                        00022090
* 7.4110927E-16,-4.1700094E-16, 8.5977184E-17, 1.3396469E-16,      00022100
*-1.7838410E-16, 4.8975421E-17, 1.9398153E-16,-5.0046989E-16,      00022110
* 8.3280985E-16,-1.1544640E-15, 1.4401527E-15,-1.6637066E-15,      00022120
* 1.7777129E-15,-1.7322187E-15, 1.5247247E-15,-1.1771155E-15,      00022130
```

```
     * 6.9747910E-16,-1.2088956E-16,-4.8382957E-16, 1.0408292E-15,      00022140
     *-1.5220450E-15, 1.9541597E-15,-2.4107448E-15, 2.9241438E-15,      00022150
     *-3.5176475E-15, 4.2276125E-15,-5.0977851E-15, 6.1428456E-15,      00022160
     *-7.3949962E-15, 8.8597601E-15,-1.0515959E-14, 1.2264584E-14,      00022170
     *-1.3949870E-14, 1.5332490E-14,-1.6146782E-14, 1.6084121E-14,      00022180
     *-1.4962523E-14, 1.2794804E-14,-9.9286701E-15, 6.8825809E-15,      00022190
     *-4.0056107E-15, 1.5965079E-15,-7.2732961E-18,-4.0433218E-16,      00022200
     *-6.5679655E-16, 3.3011866E-15,-7.3545910E-15, 1.2394851E-14,      00022210
     *-1.7947697E-14, 2.3774303E-14,-3.0279168E-14, 3.9252831E-14,      00022220
     *-5.5510504E-14, 9.0505371E-14,-1.7064873E-13/                     00022230
C--END OF J0 FILTER WEIGHTS                                             00022240
C                                                                       00022250
C--J1-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT1 ARRAY)          00022260
      DATA WA1/                                                         00022270
     *-4.2129715E-16, 5.3667031E-15,-7.1183962E-15, 8.9478500E-15,      00022280
     *-1.0767891E-14, 1.2362265E-14,-1.3371129E-14, 1.3284178E-14,      00022290
     *-1.1714302E-14, 8.4134738E-15,-3.7726725E-15,-1.4263879E-15,      00022300
     * 6.1279163E-15,-9.1102765E-15, 9.9696405E-15,-9.3649955E-15,      00022310
     * 8.6009018E-15,-8.9749846E-15, 1.1153987E-14,-1.4914821E-14,      00022320
     * 1.9314024E-14,-2.3172388E-14, 2.5605477E-14,-2.6217555E-14,      00022330
     * 2.5057768E-14,-2.2485539E-14, 1.9022752E-14,-1.5198084E-14,      00022340
     * 1.1422464E-14,-7.9323958E-15, 4.8421406E-15,-2.1875032E-15,      00022350
     *-3.2177842E-17, 1.8637565E-15,-3.3683643E-15, 4.6132219E-15,      00022360
     *-5.6209538E-15, 6.4192841E-15,-6.8959928E-15, 6.9895792E-15,      00022370
     *-6.5355935E-15, 5.6125163E-15,-4.1453931E-15, 2.6358827E-15,      00022380
     *-9.5104370E-16, 1.4600474E-16, 5.6166519E-16, 8.2899246E-17,      00022390
     * 5.0032100E-16, 4.3752205E-16, 2.1052293E-15,-9.5451973E-16,      00022400
     * 6.4004437E-15,-2.1926177E-15, 1.1651003E-14, 5.8415433E-16,      00022410
     * 1.8044664E-14, 1.0755745E-14, 3.0159022E-14, 3.3506138E-14,      00022420
     * 5.8709354E-14, 8.1475200E-14, 1.2530006E-13, 1.8519112E-13,      00022430
     * 2.7641786E-13, 4.1330823E-13, 6.1506209E-13, 9.1921659E-13,      00022440
     * 1.3698462E-12, 2.0447427E-12, 3.0494477E-12, 4.5501001E-12,      00022450
     * 6.7870250E-12, 1.0126237E-11, 1.5104976E-11, 2.2536053E-11/      00022460
      DATA WB1/                                                         00022470
     * 3.3617368E-11, 5.0153839E-11, 7.4818173E-11, 1.1161804E-10,      00022480
     * 1.6651222E-10, 2.4840923E-10, 3.7058109E-10, 5.5284353E-10,      00022490
     * 8.2474468E-10, 1.2303750E-09, 1.8355034E-09, 2.7382502E-09,      00022500
     * 4.0849867E-09, 6.0940898E-09, 9.0913020E-09, 1.3562651E-08,      00022510
     * 2.0233058E-08, 3.0184244E-08, 4.5029477E-08, 6.7176304E-08,      00022520
     * 1.0021488E-07, 1.4950371E-07, 2.2303208E-07, 3.3272689E-07,      00022530
     * 4.9636623E-07, 7.4049804E-07, 1.1046805E-06, 1.6480103E-06,      00022540
     * 2.4585014E-06, 3.6677163E-06, 5.4714550E-06, 8.1626422E-06,      00022550
     * 1.2176782E-05, 1.8166179E-05, 2.7099223E-05, 4.0428804E-05,      00022560
     * 6.0307294E-05, 8.9971508E-05, 1.3420195E-04, 2.0021123E-04,      00022570
     * 2.9860417E-04, 4.4545291E-04, 6.6423156E-04, 9.9073275E-04,      00022580
     * 1.4767050E-03, 2.2016806E-03, 3.2788147E-03, 4.8837292E-03,      00022590
     * 7.2596811E-03, 1.0788355E-02, 1.5973323E-02, 2.3612041E-02,      00022600
     * 3.4655327E-02, 5.0608141E-02, 7.2827752E-02, 1.0337889E-01,      00022610
     * 1.4207357E-01, 1.8821315E-01, 2.2996815E-01, 2.5088500E-01,      00022620
     * 2.0334626E-01, 6.0665451E-02,-2.0275683E-01,-3.5772336E-01,      00022630
     *-1.8280529E-01, 4.7014634E-01, 7.2991233E-03,-3.0614594E-01,      00022640
     * 2.4781735E-01,-1.1149185E-01, 2.5985386E-02, 1.0850279E-02,      00022650
```

```
     *-2.2830217E-02, 2.4644647E-02,-2.2895284E-02, 2.0197032E-02/      00022660
      DATA WC1/                                                         00022670
     *-1.7488968E-02, 1.5057670E-02,-1.2953923E-02, 1.1153254E-02,      00022680
     *-9.6138436E-03, 8.2952090E-03,-7.1628361E-03, 6.1882910E-03,      00022690
     *-5.3482055E-03, 4.6232056E-03,-3.9970542E-03, 3.4560118E-03,      00022700
     *-2.9883670E-03, 2.5840861E-03,-2.2345428E-03, 1.9323046E-03,      00022710
     *-1.6709583E-03, 1.4449655E-03,-1.2495408E-03, 1.0805480E-03,      00022720
     *-9.3441130E-04, 8.0803899E-04,-6.9875784E-04, 6.0425624E-04,      00022730
     *-5.2253532E-04, 4.5186652E-04,-3.9075515E-04, 3.3790861E-04,      00022740
     *-2.9220916E-04, 2.5269019E-04,-2.1851585E-04, 1.8896332E-04,      00022750
     *-1.6340753E-04, 1.4130796E-04,-1.2219719E-04, 1.0567099E-04,      00022760
     *-9.1379828E-05, 7.9021432E-05,-6.8334412E-05, 5.9092726E-05,      00022770
     *-5.1100905E-05, 4.4189914E-05,-3.8213580E-05, 3.3045496E-05,      00022780
     *-2.8576356E-05, 2.4711631E-05,-2.1369580E-05, 1.8479514E-05,      00022790
     *-1.5980307E-05, 1.3819097E-05,-1.1950174E-05, 1.0334008E-05,      00022800
     *-8.9364160E-06, 7.7278366E-06,-6.6827083E-06, 5.7789251E-06,      00022810
     *-4.9973715E-06, 4.3215167E-06,-3.7370660E-06, 3.2316575E-06,      00022820
     *-2.7946015E-06, 2.4166539E-06,-2.0898207E-06, 1.8071890E-06,      00022830
     *-1.5627811E-06, 1.3514274E-06,-1.1686576E-06, 1.0106059E-06,      00022840
     *-8.7392952E-07, 7.5573750E-07,-6.5353002E-07, 5.6514528E-07,      00022850
     *-4.8871388E-07, 4.2261921E-07,-3.6546333E-07, 3.1603732E-07/      00022860
      DATA WD1/                                                         00022870
     *-2.7329579E-07, 2.3633470E-07,-2.0437231E-07, 1.7673258E-07,      00022880
     *-1.5283091E-07, 1.3216174E-07,-1.1428792E-07, 9.8831386E-08,      00022890
     *-8.5465227E-08, 7.3906734E-08,-6.3911437E-08, 5.5267923E-08,      00022900
     *-4.7793376E-08, 4.1329702E-08,-3.5740189E-08, 3.0906612E-08,      00022910
     *-2.6726739E-08, 2.3112160E-08,-1.9986424E-08, 1.7283419E-08,      00022920
     *-1.4945974E-08, 1.2924650E-08,-1.1176694E-08, 9.6651347E-09,      00022930
     *-8.3580023E-09, 7.2276490E-09,-6.2501673E-09, 5.4048822E-09,      00022940
     *-4.6739154E-09, 4.0418061E-09,-3.4951847E-09, 3.0224895E-09,      00022950
     *-2.6137226E-09, 2.2602382E-09,-1.9545596E-09, 1.6902214E-09,      00022960
     *-1.4616324E-09, 1.2639577E-09,-1.0930164E-09, 9.4519327E-10,      00022970
     *-8.1736202E-10, 7.0681930E-10,-6.1122713E-10, 5.2856342E-10,      00022980
     *-4.5707937E-10, 3.9526267E-10,-3.4180569E-10, 2.9557785E-10,      00022990
     *-2.5560176E-10, 2.2103233E-10,-1.9113891E-10, 1.6528994E-10,      00023000
     *-1.4294012E-10, 1.2361991E-10,-8.2740936E-11/                     00023010
C--END OF J1 FILTER WEIGHTS                                             00023020
C                                                                       00023030
      NONE=0                                                            00023040
      IF(NEW.EQ.0) GO TO 100                                            00023050
      NSAVE=0                                                           00023060
C-----INITIALIZE KERNEL ABSCISSA GENERATION FOR GIVEN B                 00023070
      Y1=0.73588526614797944600D0/DBLE(B)                               00023080
  100 ZHANKS=(0.0,0.0)                                                  00023090
      CMAX=(0.0,0.0)                                                    00023100
      NF=0                                                              00023110
      Y=Y1                                                              00023120
C-----BEGIN RIGHT-SIDE CONVOLUTION AT WEIGHT 131 (EITHER NEW=1 OR 0)    00023130
      ASSIGN 110 TO M                                                   00023140
      I=131                                                             00023150
      Y=Y*E                                                             00023160
      GO TO 200                                                         00023170
```

```
  110 TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                                  00023180
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                                  00023190
      I=I+1                                                             00023200
      Y=Y*E                                                             00023210
      IF(I.LE.149) GO TO 200                                            00023220
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) NONE=1                      00023230
C-----ESTABLISH TRUNCATION CRITERION (CMAX=CMPLX(TMAX(1),TMAX(2))       00023240
      CMAX=TOL*CMAX                                                     00023250
      ASSIGN 120 TO M                                                   00023260
      GO TO 200                                                         00023270
C-----CHECK FOR FILTER TRUNCATION AT RIGHT END                          00023280
  120 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130       00023290
      I=I+1                                                             00023300
      Y=Y*E                                                             00023310
      IF(I.LE.283) GO TO 200                                            00023320
  130 Y=Y1                                                              00023330
C-----CONTINUE WITH LEFT-SIDE CONVOLUTION AT WEIGHT 130                 00023340
      ASSIGN 140 TO M                                                   00023350
      I=130                                                             00023360
      GO TO 200                                                         00023370
C-----CHECK FOR FILTER TRUNCATION AT LEFT END                           00023380
  140 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2).AND.             00023390
     * NONE.EQ.0) GO TO 190                                            00023400
      I=I-1                                                             00023410
      Y=Y*ER                                                            00023420
      IF(I.GT.0) GO TO 200                                              00023430
C-----RETURN WITH ISAVE=1 PRESET FOR POSSIBLE NEW=0 USE.                00023440
  190 ISAVE=1                                                           00023450
C-----NORMALIZE BY B TO ACCOUNT FOR INTEGRATION RANGE CHANGE            00023460
      ZHANKS=ZHANKS/B                                                   00023470
      RETURN                                                            00023480
C-----SAVE/RETRIEVE PSEUDO-SUBROUTINE (CALL FUN ONLY WHEN NECESSARY)    00023490
  200 G=SNGL(Y)                                                         00023500
      IF(NEW) 300,210,300                                               00023510
  210 IF(ISAVE.GT.NSAVE) GO TO 300                                      00023520
      ISAVE0=ISAVE                                                      00023530
  220 IF(G.EQ.GSAVE(ISAVE)) GO TO 240                                   00023540
      ISAVE=ISAVE+1                                                     00023550
      IF(ISAVE.LE.NSAVE) GO TO 220                                      00023560
      ISAVE=ISAVE0                                                      00023570
C-----G NOT IN COMMON/SAVE/----- EVALUATE FUN.                          00023580
      GO TO 300                                                         00023590
C-----G FOUND IN COMMON/SAVE/----- USE FSAVE AS GIVEN.                  00023600
  240 C=FSAVE(ISAVE)                                                    00023610
      ISAVE=ISAVE+1                                                     00023620
C-----SWITCH ON ORDER N                                                 00023630
  250 IF(N) 270,260,270                                                 00023640
  260 C=C*WT0(I)                                                        00023650
      GO TO 280                                                         00023660
  270 C=C*WT1(I)                                                        00023670
  280 ZHANKS=ZHANKS+C                                                   00023680
      GO TO M,(110,120,140)                                             00023690
```

```
C-----DIRECT FUN EVALUATION (AND ADD TO END OF COMMON/SAVE/)                00023700
  300 NSAVE=NSAVE+1                                                        00023710
      C=FUN(G)                                                             00023720
      NF=NF+1                                                              00023730
      FSAVE(NSAVE)=C                                                       00023740
      GSAVE(NSAVE)=G                                                       00023750
      GO TO 250                                                            00023760
      END                                                                  00023770

      SUBROUTINE MARQ_TRANS_HCLOOP_FCODE(Y,X,B,PRNT,F,IN,IDER)             00023780
C--FUNCT. EVAL. FOR 'MARQ_TRANS_HCLOOP'                                    00023790
C                                                                          00023800
C--PARAMETERS--                                                            00023810
C       Y=       OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)                00023820
C       X=       OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)            00023830
C       B=       CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)                00023840
C       PRNT=    WORK AND PRINT ARRAY (DIM. 5)                             00023850
C       F=       OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. IN    00023860
C       IN=      OBSERVATION NO. TO EVAL. F (1<=IN<=N)                     00023870
C       IDER=    0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED)   00023880
C                1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED)   00023890
C                                                                          00023900
      REAL Y(1),X(200,5),B(1),PRNT(5),K(10),DD(9),D(10),DER(2),V(200),     00023910
     * BV(20)                                                              00023920
      EXTERNAL HCLOOP,HSLOOP                                               00023930
      COMMON/PASS/SIG1,TMIN,TMAX,T0,TM,DB,                                 00023940
     * ISTEP,M21,M2,JSPLN,NN,IFIRST                                        00023950
      COMMON/TCOM/T(200),THC(200)                                          00023960
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN         00023970
      COMMON/LOOP/D,RR,EPS,B0,BM,M1,IHALF                                  00023980
      COMMON/MODEL/K,DD,M                                                  00023990
C--STATEMENT FUNCTION HZ0(U) IS HALFSPACE IMPULSE RESPONSE                 00024000
      HZ0(U)=9.*ERF(U)-1.128379167*U*EXP_(-U*U)*(9.+2.*U*U*(3.+2.*U*U))    00024010
C--STATEMENT FUNCTION HZ1(U) IS HALFSPACE STEP RESPONSE                    00024020
      HZ1(U)=1.+ERF(U)*(9./(2.*U*U)-1.)-EXP_(-U*U)*.5641895835*            00024030
     * (9./U+4.*U)                                                         00024040
      DATA DER/2*0.0/                                                      00024050
      IF(IN.GT.1.OR.M.EQ.1) GO TO 20                                       00024060
      DO 10 J=2,M                                                          00024070
      IF(B(J).EQ.B(J-1)) CALL ERRMSG('SOME SIG(J)=SIG(J-1)',4,6,16)        00024080
10    CONTINUE                                                             00024090
20    DO 30 J=1,5                                                          00024100
30    PRNT(J)=X(IN,J)                                                      00024110
      IF(IN.GT.1) GO TO 800                                                00024120
      IF(IDER.EQ.1) GO TO 8001                                             00024130
35    SIG1=B(1)                                                            00024140
      IF(M.EQ.1) GO TO 45                                                  00024150
      DO 40 J=1,M1                                                         00024160
      D(J)=B(M+J)                                                          00024170
40    K(J)=B(J)/SIG1                                                       00024180
45    K(M)=B(M)/SIG1                                                       00024190
      TCON=6.28318531E-7*SIG1*RR*RR                                        00024200
```

```
      IF(JSPLN.EQ.0) GO TO 49                                          00024210
C--GET PRE-SPLINED FREQ FUNCTION (0<NB<12 OPTION)                      00024220
      NS=0                                                             00024230
      TEM=B0/DB                                                        00024240
      ISPLN=0                                                          00024250
46    TEM=TEM*DB                                                       00024260
      IF(TEM.GT.BM) GO TO 47                                           00024270
      NS=NS+1                                                          00024280
      IF(NS.GT.200) CALL ERRMSG('NS>200 IN FCODE',3,6,16)              00024290
      XS(NS)=TEM                                                       00024300
      YS(NS)=HCLOOP(TEM*TEM)                                           00024310
      GO TO 46                                                         00024320
47    CALL SPLIN1(NS,0.0,XS,YS,AS,BS,CS,0,DER,T,V)                     00024330
      ISPLN=1                                                          00024340
49    T0=.5*TMIN/TCON                                                  00024350
      TM=TMAX/TCON                                                     00024360
      NEW=1                                                            00024370
      DO 70 I=1,NN                                                     00024380
      T(I)=X(I,1)/TCON                                                 00024390
      TRANS=0.0                                                        00024400
      IF(M.EQ.1.AND.IHALF.EQ.0) GO TO 64                               00024410
C--GET TRANSIENT VIA RFLAGS (LAGGED-CONVOLUTION IN TIME)               00024420
      IF(ISTEP.EQ.0)                                                   00024430
     * TRANS=.63661977*RFLAGS(0,HCLOOP,EPS,T0,TM,T(I),NEW)             00024440
      IF(ISTEP.EQ.1)                                                   00024450
     * TRANS=.63661977*RFLAGS(1,HSLOOP,EPS,T0,TM,T(I),NEW)             00024460
      NEW=0                                                            00024470
      IF(IHALF.EQ.1) GO TO 65                                          00024480
64    U=1.0/SQRT(2.*T(I))                                              00024490
      IF(ISTEP.EQ.0) TRANS=TRANS+HZ0(U)                                00024500
      IF(ISTEP.EQ.1) TRANS=TRANS+HZ1(U)                                00024510
65    V(I)=TRANS                                                       00024520
70    CONTINUE                                                         00024530
      IF(IDER.EQ.0) GO TO 600                                          00024540
      CALL RITE(20,V,NN,B,M2)                                          00024550
      IFIRST=0                                                         00024560
C--GET PRE-SPLINED TRANSIENT (EITHER ISTEP=0 OR 1)                     00024570
600   F=B(M2)*V(IN)                                                    00024580
      THC(IN)=F                                                        00024590
      RETURN                                                           00024600
800   IF(IDER.EQ.0) GO TO 600                                          00024610
C--IDER=1 EST.DER.OPTION (B() VARIES PER CALL FOR EACH IN OBS.)        00024620
8001  IF(IFIRST.EQ.1) GO TO 35                                         00024630
      REWIND 20                                                        00024640
801   CALL REED(20,V,NN,BV,M2,$35)                                     00024650
      DO 802 J=1,M2                                                    00024660
      IF(B(J).NE.BV(J)) GO TO 801                                      00024670
802   CONTINUE                                                         00024680
      GO TO 600                                                        00024690
      END                                                              00024700
```

```
      SUBROUTINE MARQ_TRANS_HCLOOP_PCODE(P,X,B,PRNT,F,IN,IP,IB)          00024710
C--ANALYTIC PARTIALS W/R PARAMETERS IN B(K) AND IN COMMON                00024720
C  FOR PROGRAM 'MARQ_TRANS_HCLOOP'                                       00024730
C                                                                        00024740
C  (PCODE ONLY CALLED BY MARQRT IF IDER=0--DEFAULT)                      00024750
C                                                                        00024760
C--PARAMETERS--                                                          00024770
C                                                                        00024780
C       P=      OUTPUT PARTIAL DERIVATIVE ARRAY (DIM. K)                 00024790
C               EVALUATED FOR GIVEN X(IN,),B(K) AT OBS. IN               00024800
C       X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)           00024810
C       B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)               00024820
C       PRNT=   WORK AND PRINT ARRAY (DIM. 5)                            00024830
C       F=      LAST FUNCTION VALUE FROM FCODE AT GIVEN IN.              00024840
C               F MAY OR MAY NOT BE NEEDED--BUT AVAILABLE ANYWAY.        00024850
C       IN=     OBSERVATION NO. TO EVAL. P ARRAY (1<=IN<=N)              00024860
C       IP=     NO. PARAMETERS HELD FIXED (IF ANY--IF NONE IP=0).        00024870
C       IB=     ARRAY OF PARAMETER INDICES HELD FIXED IF IP.GT.0         00024880
C               (DIM. 19).                                               00024890
C                                                                        00024900
      INTEGER IB(1)                                                      00024910
      REAL P(1),X(200,5),B(1),PRNT(5),                                   00024920
     * DER(2),W1(200),W2(200),                                          00024930
     * K(10),DD(9),D(10),PM(200,20)                                      00024940
      EXTERNAL PHC,PHS                                                   00024950
      COMMON/PASS/SIG1,TMIN,TMAX,T0,TM,DB,                               00024960
     * ISTEP,M21,M2,JSPLN,NN,IFIRST                                      00024970
      COMMON/TCOM/T(200),THC(200)                                        00024980
      COMMON/LOOP/D,RR,EPS,B0,BM,M1,IHALF                                00024990
      COMMON/SPLN/XP(200),YP(200),AP(200),BP(200),CP(200),NP,ISPLN       00025000
      COMMON/MODEL/K,DD,M                                                00025010
      COMMON/PART/INDEX,JUNK                                             00025020
      DATA DER/2*0.0/                                                    00025030
      IF(IN.GT.1) GO TO 50                                               00025040
      DO 30 J=1,M21                                                      00025050
      INDEX=J                                                            00025060
      IFIX=0                                                             00025070
      IF(IP.LE.0) GO TO 11                                               00025080
      DO 1 I=1,IP                                                        00025090
      IF(IB(I).EQ.J) IFIX=1                                              00025100
1     CONTINUE                                                           00025110
      IF(IFIX.EQ.1) GO TO 6                                              00025120
11    IF(JSPLN.EQ.0) GO TO 4                                             00025130
C--GET PRE-SPLINED PARTIAL DER. FREQ. FUNCTION (0<NB<12 OPTION)          00025140
      NP=0                                                               00025150
      TEM=B0/DB                                                          00025160
      ISPLN=0                                                            00025170
2     TEM=TEM*DB                                                         00025180
      IF(TEM.GT.BM) GO TO 3                                              00025190
      NP=NP+1                                                            00025200
      XP(NP)=TEM                                                         00025210
C--YP=PHC APPLIES TO EITHER ISTEP=0 OR 1 (SEE SUBR PHS USE)              00025220
```

```
      YP(NP)=PHC(TEM*TEM)                                                00025230
      GO TO 2                                                            00025240
3     CALL SPLIN1(NP,0.0,XP,YP,AP,BP,CP,0,DER,W1,W2)                     00025250
      ISPLN=1                                                            00025260
4     NEW=1                                                              00025270
      DO 5 I=1,NN                                                        00025280
C--GET PARTIAL OF TRANSIENT VIA RFLAGS (LAGGED-CONVOLUTION IN TIME)      00025290
      IF(ISTEP.EQ.0)                                                     00025300
     * TP=.63661977*RFLAGS(0,PHC,EPS,TO,TM,T(I),NEW)                     00025310
      IF(ISTEP.EQ.1)                                                     00025320
     * TP=.63661977*RFLAGS(1,PHS,EPS,TO,TM,T(I),NEW)                     00025330
      NEW=0                                                              00025340
      PM(I,J)=B(M2)*TP                                                   00025350
5     CONTINUE                                                           00025360
      GO TO 30                                                           00025370
6     DO 7 I=1,NN                                                        00025380
7     PM(I,J)=0.0                                                        00025390
30    CONTINUE                                                           00025400
50    DO 60 J=1,M21                                                      00025410
60    P(J)=PM(IN,J)                                                      00025420
      P(M2)=0.0                                                          00025430
      IF(IP.EQ.0) GO TO 65                                               00025440
      DO 61 I=1,IP                                                       00025450
      IF(IB(I).EQ.M2) GO TO 70                                           00025460
61    CONTINUE                                                           00025470
65    P(M2)=THC(IN)/B(M2)                                                00025480
70    RETURN                                                             00025490
      END                                                                00025500

      SUBROUTINE MARQ_TRANS_HCLOOP_SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)   00025510
C--'MARQ_TRANS_HCLOOP' INITIALIZATION (CALLED ONCE BY 'MARQRT')           00025520
C                                                                        00025530
C  SUBZ IS CALLED BY MARQRT AFTER THE DATA Y(I),X(I,5) ARE READ--        00025540
C  SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL $INIT                    00025550
C  PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE.                00025560
C                                                                        00025570
C--PARAMETERS--                                                          00025580
C       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.                          00025590
C       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY              00025600
C               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES               00025610
C               PRINTED BY PGM MARQRT...                                 00025620
C       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)                    00025630
C       TITLE=  ALPHA TITLE ARRAY READ IN BY PGM MARQRT.                 00025640
C       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED                      00025650
C               0 IF ONLY UNIT 6 PRINT FILE USED.                        00025660
C                                                                        00025670
      CHARACTER*5 TITLE(16)                                              00025680
      REAL D(10),DD(9),Y(1),X(200,5),B(1),PRNT(1),K(10)                  00025690
      COMMON/LOOP/D,RR,EPS,RO,BM,M1,IHALF                                00025700
      COMMON/PASS/SIG1,TMIN,TMAX,TO,TM,DB,                               00025710
     & ISTEP,M21,M2,JSPLN,NN,IFIRST                                      00025720
      COMMON/SPLN/FILL(1000),NS,ISPLN                                    00025730
```

```
      COMMON/MODEL/K,DD,MM                                                  00025740
      NAMELIST/INIT/MM,Y0,EPS,B0,BM,IHALF,NB,ISTEP                          00025750
      DATA ISUBZ/0/                                                         00025760
      IF(ISUBZ.NE.0) GO TO 10                                               00025770
C--PRESET                                                                   00025780
      ISUBZ=1                                                               00025790
      MM=1                                                                  00025800
      IHALF=0                                                               00025810
      B0=.01                                                                00025820
      BM=100.                                                               00025830
      NB=10                                                                 00025840
      Y0=0.0                                                                00025850
      EPS=.1E-5                                                             00025860
      ISTEP=0                                                               00025870
10    READ(5,INIT)                                                          00025880
      WRITE(6,20) TITLE                                                     00025890
20    FORMAT('1M A R Q _ T R A N S _ H C L O O P  --  ',5X,16A5/)            00025900
      IF(IOUT.EQ.1) WRITE(16,20) TITLE                                      00025910
      WRITE(6,30) MM,Y0,EPS,B0,BM,IHALF,ISTEP,NB                            00025920
      IF(IOUT.EQ.1) WRITE(16,30) MM,Y0,EPS,B0,BM,IHALF,ISTEP,NB             00025930
30    FORMAT(' MM=',I3,12X,'Y0=',E13.6,4X,'EPS=',E13.6/                     00025940
     & ' B0=',E13.6,2X,'BM=',E13.6,4X,'IHALF=',I3/                          00025950
     & ' ISTEP=',I3,9X,'NB=',I4)                                            00025960
C--TEST $INIT PARMS                                                         00025970
      IF(MM.LT.1.OR.MM.GT.10.OR.Y0.LE.0.0.OR.IHALF.LT.0.OR.IHALF.GT.1       00025980
     & .OR.ISTEP.LT.0.OR.ISTEP.GT.1.OR.BM.LE.B0.OR.B0.LE.0.0.OR.            00025990
     & (IHALF.EQ.1.AND.ISTEP.EQ.1))                                         00026000
     & CALL ERRMSG('SOME $INIT PARMS OUT OF RANGE.',6,6,16)                 00026010
C--TEST X(I, ) DATA BEFORE PROCEEDING                                       00026020
      DO 40 I=2,N                                                           00026030
      IF(X(I,1).LE.X(I-1,1).OR.X(I,1).LE.0.0)                               00026040
     & CALL ERRMSG('SOME X(I,1)<=0.0 OR NOT INCREASING.',7,6,16)            00026050
40    CONTINUE                                                              00026060
C--PRESET SOME GLOBAL CONSTANTS                                             00026070
      IFIRST=1                                                              00026080
      NN=N                                                                  00026090
      RR=Y0                                                                 00026100
      TMIN=X(1,1)                                                           00026110
      TMAX=X(N,1)                                                           00026120
      ISPLN=0                                                               00026130
      JSPLN=0                                                               00026140
      IF(NB.GT.0.AND.NB.LT.12.AND.MM.GT.1) JSPLN=1                          00026150
      IF(JSPLN.EQ.1) DB=EXP(2.30258509/FLOAT(NB))                           00026160
      WRITE(6,50)                                                           00026170
      IF(IOUT.EQ.1) WRITE(16,50)                                            00026180
50    FORMAT(////' PARAMETER ORDER--')                                      00026190
      M1=MM-1                                                               00026200
      M21=2*MM-1                                                            00026210
      M2=M21+1                                                              00026220
      WRITE(6,110) (I,I,I=1,MM)                                             00026230
      IF(IOUT.EQ.1) WRITE(16,110) (I,I,I=1,MM)                              00026240
110   FORMAT(5X,I3,6X,6HSIGMA(,I3,1H))                                      00026250
```

```
      IF(MM.EQ.1) GO TO 132                                              00026260
      DO 120 I=1,M1                                                      00026270
      J=MM+I                                                             00026280
      IF(IOUT.EQ.1) WRITE(16,130) J,I                                    00026290
120   WRITE(6,130) J,I                                                   00026300
130   FORMAT(5X,I3,6X,6HTHICK(,I3,1H))                                   00026310
132   WRITE(6,131) M2,M2                                                 00026320
131   FORMAT(5X,I3,10X,'B(',I3,') SHIFT PARAMETER IN B(2*MM)*TRANSIENT')00026330
      IF(IOUT.EQ.1) WRITE(16,131) M2,M2                                  00026340
      NPRNT=2                                                            00026350
      RETURN                                                             00026360
      END                                                                00026370

      SUBROUTINE MARQ_TRANS_HCLOOP_SUBEND(Y,X,B,K,N,TITLE,IOUT)          00026380
C-- 'MARQ_TRANS_HCLOOP' TERMINATION ROUTINE (CALLED ONCE BY MARQRT)      00026390
C  (PARAMETERS SAME AS IN SUBROUTINE FCODE,PCODE, OR SUBZ)               00026400
C  B= FINAL SOLUTION VECTOR OBTAINED BY PGM MARQRT.                     00026410
C                                                                       00026420
C--FOLLOWING CHARACTER STMT. ONLY FOR HONEYWELL MULTICS SYS:           00026430
      CHARACTER*5 TITLE(16)                                              00026440
      REAL Y(1),X(200,5),B(1)                                            00026450
      WRITE(6,10) TITLE                                                  00026460
   10 FORMAT('1M A R Q _ T R A N S _ H C L O O P  --',5X,16A5//          00026470
     1 28H FINAL UNSCALED PARAMETERS--,10X,11HRESISTIVITY,11X,5HDEPTH/)  00026480
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                                   00026490
      MM=(K+1)/2                                                         00026500
      DO 30 I=1,MM                                                       00026510
      R=1.0/B(I)                                                         00026520
      WRITE(6,20) I,B(I),I,R                                             00026530
   20 FORMAT(5X,I3,4X,E16.8,2X,I3,1X,E16.8)                              00026540
      IF(IOUT.EQ.1) WRITE(16,20) I,B(I),I,R                              00026550
   30 CONTINUE                                                           00026560
      IF(K.LE.2) GO TO 52                                                00026570
      M2=MM+1                                                            00026580
      K1=K-1                                                             00026590
      D=0.0                                                              00026600
      DO 50 I=M2,K1                                                      00026610
      D=D+B(I)                                                           00026620
      L=I-MM                                                             00026630
      WRITE(6,40) I,B(I),L,D                                             00026640
   40 FORMAT(5X,I3,4X,E16.8,24X,I3,1X,E16.8)                             00026650
      IF(IOUT.EQ.1) WRITE(16,40) I,B(I),L,D                              00026660
   50 CONTINUE                                                           00026670
   52 WRITE(6,51) K,B(K)                                                 00026680
   51 FORMAT(5X,I3,4X,E16.8)                                             00026690
      IF(IOUT.EQ.1) WRITE(16,51) K,B(K)                                  00026700
   60 RETURN                                                             00026710
      END                                                                00026720
```

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).

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

6. Some systems may require statement order changes. For example, the statement functions HZ0(U) and HZ1(U) used in subprogram MARQ_TRANS_HCLOOP_FCODE appear before the DATA statement on Multics. Most systems require the statement functions to be the first executable statements of a subprogram (i.e., placed after the DATA statement).

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

```
test2x
$parms n=21,k=4,m=1,iprt=-1,sp=1,sy=2,e=.005,
 b=.18,1.8,480,1$
(2e16.8)
$init mm=2,y0=1000$
```

file10

```
  0.90011072e+01   0.12566370e-02
  0.89979557e+01   0.19916355e-02
  0.90034355e+01   0.31565295e-02
  0.89840084e+01   0.50027620e-02
  0.86500809e+01   0.79288434e-02
  0.65377956e+01   0.12566370e-01
  0.28415456e+01   0.19916354e-01
  0.47536118e+00   0.31565294e-01
 -0.70070705e-01   0.50027619e-01
 -0.68711989e-01   0.79288432e-01
 -0.49168396e-01   0.12566370e+00
 -0.39093065e-01   0.19916354e+00
 -0.28908257e-01   0.31565293e+00
 -0.18966294e-01   0.50027619e+00
 -0.10976795e-01   0.79288432e+00
 -0.56447714e-02   0.12566369e+01
 -0.26225012e-02   0.19916354e+01
 -0.11203003e-02   0.31565293e+01
 -0.44699982e-03   0.50027618e+01
 -0.16876354e-03   0.79288431e+01
 -0.60367182e-04   0.12566369e+02
```

```
m a r q _ t r a n s _ h c l o o p  --        test2x

mm=  2            y0= 0.100000e+04    eps= 0.100000e-05
b0= 0.100000e-01  bm= 0.100000e+03    ihalf=  0
istep=  0         nb=  10



parameter order--

      1       sigma(  1)
      2       sigma(  2)
      3       thick(  1)
      4           b(  4) shift parameter in b(2*mm)*transient
```

```
m a r q r t  --     test2x

n =   21        k =   4         ip =   0        m =  1          gamcr=0.450e+02
del= 0.100e-04  modlam = 1      ff= 0.400e+01   t= 0.200e+01    e= 0.500e-02
tau= 0.100e-02  xl= 0.100e-01   zeta= 0.100e-30 ialt = 10       istop = 1
iwt = 0         ider = 0        iprt = -1       niter =   10    inon = 1
iout =  1       nprnt = 2       scalep = 1      scaley = 2

fmt=(2e16.8)

parameters     0.18000000e+00    0.18000000e+01    0.48000000e+03    0.10000000e+01


iter          phi                s e               length        gamma         lambda
   1     0.15631875e+00     0.95891713e-01     0.000e+00     0.000e+00     0.100e-01

parameters     0.19949690e+00    0.13355602e+01    0.48679296e+03    0.11017284e+01


iter          phi                s e               length        gamma         lambda
   2     0.68669731e-01     0.63556243e-01     0.330e+00     0.323e+02     0.100e-02

parameters     0.19992716e+00    0.18529584e+01    0.50149283e+03    0.10021142e+01


iter          phi                s e               length        gamma         lambda
   3     0.14996319e-03     0.29700781e-02     0.342e+00     0.570e+02     0.100e-03

parameters     0.19998967e+00    0.19984927e+01    0.50010699e+03    0.10003098e+01


iter          phi                s e               length        gamma         lambda
   4     0.68806012e-06     0.20118182e-03     0.757e-01     0.458e+02     0.100e-04

parameters     0.19999975e+00    0.19999480e+01    0.50000395e+03    0.10000500e+01


iter          phi                s e               length        gamma         lambda
   5     0.17290846e-07     0.31892140e-04     0.801e-03     0.495e+02     0.100e-06

parameters     0.19999995e+00    0.20000097e+01    0.50000049e+03    0.10000018e+01


iter          phi                s e               length        gamma         lambda
   6     0.19047550e-10     0.10585103e-05     0.577e-04     0.227e+02     0.100e-08

-epsilon test
   6 iterations
```

```
m a r q r t  --     test2x

parameters     0.19999995e+00    0.20000097e+01    0.50000049e+03    0.10000018e+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   0.900111e+01   0.900112e+01 -0.178e-04 -0.197333e-03   0.125664e-02   0.000000e+00
   2   0.899796e+01   0.899797e+01 -0.104e-04 -0.115262e-03   0.199164e-02   0.000000e+00
   3   0.900344e+01   0.900343e+01  0.715e-06  0.794425e-05   0.315653e-02   0.000000e+00
   4   0.898401e+01   0.898402e+01 -0.132e-04 -0.147286e-03   0.500276e-02   0.000000e+00
   5   0.865008e+01   0.865010e+01 -0.205e-04 -0.237038e-03   0.792884e-02   0.000000e+00
   6   0.653780e+01   0.653780e+01 -0.316e-05 -0.483197e-04   0.125664e-01   0.000000e+00
   7   0.284155e+01   0.284155e+01 -0.358e-05 -0.125857e-03   0.199164e-01   0.000000e+00
   8   0.475361e+00   0.475363e+00 -0.146e-05 -0.307200e-03   0.315653e-01   0.000000e+00
   9  -0.700707e-01  -0.700721e-01  0.136e-05  0.194313e-02   0.500276e-01   0.000000e+00
  10  -0.687120e-01  -0.687126e-01  0.635e-06  0.924375e-03   0.792884e-01   0.000000e+00
  11  -0.491684e-01  -0.491680e-01 -0.376e-06 -0.765242e-03   0.125664e+00   0.000000e+00
  12  -0.390931e-01  -0.390924e-01 -0.715e-06 -0.182847e-02   0.199164e+00   0.000000e+00
  13  -0.289083e-01  -0.289084e-01  0.104e-06  0.359213e-03   0.315653e+00   0.000000e+00
  14  -0.189663e-01  -0.189667e-01  0.453e-06  0.238763e-02   0.500276e+00   0.000000e+00
  15  -0.109768e-01  -0.109769e-01  0.795e-07  0.724356e-03   0.792884e+00   0.000000e+00
  16  -0.564477e-02  -0.564461e-02 -0.159e-06 -0.281210e-02   0.125664e+01   0.000000e+00
  17  -0.262250e-02  -0.262235e-02 -0.147e-06 -0.560689e-02   0.199164e+01   0.000000e+00
  18  -0.112030e-02  -0.112045e-02  0.154e-06  0.137707e-01   0.315653e+01   0.000000e+00
  19  -0.447000e-03  -0.447273e-03  0.273e-06  0.610523e-01   0.500276e+01   0.000000e+00
  20  -0.168764e-03  -0.168654e-03 -0.109e-06 -0.648586e-01   0.792884e+01   0.000000e+00
  21  -0.603672e-04  -0.602915e-04 -0.757e-07 -0.125551e+00   0.125664e+02   0.000000e+00

-unscaled partials-
   1   -0.43911720e+02    0.14035163e-03   -0.22917123e-04    0.90011088e+01
   2   -0.44264772e+02   -0.28002650e-03    0.40154912e-04    0.89979500e+01
   3   -0.43683447e+02    0.45555476e-03   -0.70119429e-04    0.90034187e+01
   4   -0.43892993e+02   -0.77580160e-03    0.10630879e-03    0.89840056e+01
   5   -0.32084282e+02    0.42384879e-03   -0.34429490e-04    0.86500857e+01
   6    0.45337027e+01   -0.13813308e-01    0.17084068e-02    0.65377871e+01
   7    0.23153695e+02   -0.49847416e-01    0.27835362e-02    0.28415441e+01
   8    0.11444843e+02   -0.39707705e-01    0.26979173e-03    0.47536179e+00
   9    0.19960354e+01    0.72334061e-02   -0.12409876e-02   -0.70071944e-01
  10    0.31349028e+00    0.27755916e-01   -0.73354375e-03   -0.68712504e-01
  11    0.30199091e-01    0.23792337e-01   -0.22251252e-03   -0.49167933e-01
  12   -0.11668202e+00    0.14439505e-01   -0.33670024e-04   -0.39092282e-01
  13   -0.11687817e+00    0.62199329e-02    0.23994578e-04   -0.28908310e-01
  14   -0.71411775e-01    0.10776179e-02    0.29848109e-04   -0.18966713e-01
  15   -0.33944706e-01   -0.10040271e-02    0.19823012e-04   -0.10976855e-01
  16   -0.13564581e-01   -0.12526995e-02    0.10002448e-04   -0.56446028e-02
  17   -0.47517256e-02   -0.87000432e-03    0.42295691e-05   -0.26223496e-02
  18   -0.15043275e-02   -0.47236263e-03    0.15752777e-05   -0.11204526e-02
  19   -0.44116746e-03   -0.22086626e-03    0.53367987e-06   -0.44727210e-03
  20   -0.12230380e-03   -0.93303665e-04    0.16834587e-06   -0.16865386e-03
  21   -0.32422866e-04   -0.36662995e-04    0.50152947e-07   -0.60291379e-04

-unscaled-

          phi              s e             lambda
       0.10472713e-08    0.78488341e-05    0.100e-08

ptp inverse
   1    0.13534518e-02    0.12088236e+00   -0.14621157e+02    0.59956966e-02
   2    0.12088238e+00    0.40710004e+03    0.51803760e+04    0.47897704e+00
```

3	-0.14621157e+02	0.51803760e+04	0.34715959e+06	-0.69304172e+02
4	0.59956967e-02	0.47897701e+00	-0.69304173e+02	0.28949363e-01

parameter correlation matrix

1	1.0000	0.1629	-0.6745	0.9579
2	0.1629	1.0000	0.4358	0.1395
3	-0.6745	0.4358	1.0000	-0.6913
4	0.9579	0.1395	-0.6913	1.0000

	parameter std error	one - parameter lower	one - parameter upper	support plane lower	support plane upper	std.error/parm
1	0.28875302e-06	0.19999937e+00	0.20000053e+00	0.19999879e+00	0.20000110e+00	0.14437655e-05
2	0.15836373e-03	0.19996929e+01	0.20003264e+01	0.19993762e+01	0.20006431e+01	0.79181481e-04
3	0.46245527e-02	0.49999124e+03	0.50000974e+03	0.49998199e+03	0.50001899e+03	0.92490961e-05
4	0.13354408e-05	0.99999910e+00	0.10000044e+01	0.99999643e+00	0.10000071e+01	0.13354385e-05

```
m a r q _ t r a n s _ h c l o o p  --     test2x

final unscaled parameters--          resistivity          depth

      1      0.19999995e+00     1   0.50000013e+01
      2      0.20000097e+01     2   0.49999759e+00
      3      0.50000049e+03                          1   0.50000049e+03
      4      0.10000018e+01
```