U.S. Department of the Interior
Geological Survey

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

Program MARQDCLAG:
Marquardt inversion of DC-Schlumberger soundings by lagged-convolution

by

Walter L. Anderson

Open-File Report 79-1432

1979

CONTENTS

DISCLAIMER

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

---

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

By Walter L. Anderson

INTRODUCTION

Program MARQDCLAG is a general-purpose program for the least squares inversion of direct-current (DC) Schlumberger sounding data obtained over one-dimensional horizontally stratified earth models. A modified Marquardt (1963) nonlinear least squares algorithm (MARQRT) is used for inversion of Schlumberger soundings. A digital filter developed by Anderson (1975) is employed, along with a fast lagged-convolution adaptive algorithm (RLAGH1), to efficiently and accurately evaluate the necessary Hankel transform integrals for a Schlumberger array configuration (see for example, Zohdy, 1975, p. E4, Eq. 9).

The lagged-convolution method (followed by a cubic spline approximation) runs at least 50% faster than using a direct convolution approach. Using this technique, any AB/2 spacing and range may be used (e.g., up to an arbitrary maximum of 200 points per sounding in the present program).

The following program options are currently available:

(1) Inversion of DC-Schlumberger soundings for a maximum of 10-layer models (i.e., a maximum of 19 unknown model parameters).
(2) Scaling parameter and observation spaces to constrain the solution space and to reduce round-off effects.
(3) Weighted observations.
(4) Holding certain parameters fixed (constrained).
(5) Object-time format control of reading the observed data matrix.

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

The Fortran source listing is given in Appendix 1. A few notes regarding conversion to other systems are given in Appendix 2. Appendix 3 lists the input/output for a sample test problem run on a Honeywell 68/80 system.

PARAMETERS AND DATA REQUIRED

Parameters required by program MARQDCLAG 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 MARQDCLAG 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, and ends with ")" before col. 73.
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 MARQDCLAG.
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-1, where $init parameter mm>0 is the number of layers in the model.
(cref: $init parameter mm and $parms n,b).

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

m= Number of independent variables (m=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 value for 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 MARQDCLAG 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=1.e-2; for 3-figure accuracy, use e=1.e-3, etc. (cref: Marquardt, 1963).

tau= Convergence criterion test parameter (default 1.e-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 1.e-5). Note del is used only when ider=1 for estimated partial derivative calculations.
(cref: $parms ider).

zeta= Singularity criterion for matrix inversion (default 1.e-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 MARQDCLAG, 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 electrical problems, it is recommended that scalep=1 and scaley=1 be generally used for program MARQDCLAG.
(cref: $parms b,k,n)

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

The parameter order must be given as follows:

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

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

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-1.
(cref: $parms k,b).

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

```
$end [end of $init parameters]
```

DATA MATRIX NOTES

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

1. y(i)= i-th observation of the apparent resistivity (in ohm-meters) corresponding to the distance AB/2 given in x(i,1).
2. x(i,1)= i-th distance AB/2 (x(i,1)>0.0 meters), 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).

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING

1. Single sounding using file05 (ialt=5):

```
example 1
$parms n=20,k=5,m=1,iprt=-1,sp=1,sy=1,ialt=5,iwt=1,
 e=.005,b=25,150,35,20,100$
(3f10.0)
26.0        1.          .01
30.         2.          .02
--(etc. for 18 more observations)--
$init mm=3$
```

2. Two separate soundings stored on file10 (uses istop=0 and selective object format control):

```
example 2a (istop=0; col. 1 & 2 from file10)
$parms n=20,k=5,m=1,iprt=-1,sp=1,sy=1,istop=0,e=.01,
 b=15,5,35,20,100$
(2f10.0)
$init mm=3$
example 2b (istop=1; col. 3 & 4 from file10)
$parms istop=1,n=18,b=20,2,30,10,200$
(20x,2f10.0)
$init$
          -- file10 for example 2a & 2b --

10.         1.          21.         1.
9.0         2.          18.         3.
--(etc. for total of 20 lines; only 18 used in example 2b)--
```

SPECIAL OBJECT FORMAT PHRASES

If an existing data matrix file does not have the properly 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 MARQDCLAG by typing: marqdclag

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 MARQDCLAG. 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.e-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 MARQDCLAG.

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.

PRINTED OUTPUT

Results are printed on logical unit 6 (file06) and on unit 16 (file16) if $parms iout=1 (default). Refer to Appendix 3 for a sample output listing of file16.

The following table defines additional names (or terms) used in the printed output files, other than $parms and $init parameters previously defined [also see Marquardt (1963) and IBM Share program 1428 for more details]:

names/terms	definitions
rho(i)	apparent resistivity (in ohm-meters) of layer i, i=1,...,mm.
thick(i)	thickness (in meters) of layer i, i=1,...,mm-1.
iter	Marquardt (1963) major iteration count, where 1<=iter<=niter.
phi	weighted sum-of-squares residual function defined over n observations; i.e., the objective function to be minimized by nonlinear least squares (Marquardt, 1963).
s e	standard error of estimate (or weighted root mean square error) defined as se=sqrt(phi/(n-k+ip)).
length	length of the Marquardt (1963) adjustment vector delta(j),j=1,k at each iteration.
gamma	angle (in degrees) between the gradient and Marquardt (1963) adjustment vector at each iteration.
lambda	Marquardt (1963) lambda factor (=x1 on iter=1) to be added to the diagonal of the Jacobian transpose times Jacobian matrix at each iteration.
-epsilon test	standard convergence test passed whenever abs(delta(j))/(tau+abs(b(j)))<e for all j in (1,k), where delta(j) is the Marquardt (1963) adjustment vector.

-gamma lambda test	alternate convergence test passed whenever lambda>1 and gamma>90 degrees. This criterion is used, rather than the standard epsilon test, when the parameter corrections are dependent on large rounding errors--almost certainly due to the presence of very high correlations among the parameter estimates.
-gamma epsilon test	alternate convergence test passed whenever gamma<gamcr. This criterion is used if parameter increments become small enough to pass the epsilon test as a result of successive halving of the increments. When this occurs, the value of phi is presumed minimized within the limits of the rounding error.
-force off	no convergence occurred after niter iterations. Upon branching to the confidence limit calculations, the program will use the parameter values on the last iteration (i.e., when iter=niter).
obs.y(i)	observed y(i) input dependent variable for i=1,...,n.
cal	calculated dependent variable for i=1,...,n.
res	residual=(obs.y(i)-cal) for i=1,...,n.
%res.err	percent residual error=100*res/cal for i=1,...,n.
x(i,j)	input x(i,j),j=1,m independent variables for i=1,...,n. (see DATA MATRIX NOTES above for specific definitions of x(i,j)).
-unscaled	forced scalep=scaley=0 after the last iteration to produce unscaled statistics on convergence (or if forced off after niter).
partials(i,j)	unscaled partial derivative Jacobian matrix on the last iteration for each parameter (j=1,k), evaluated at observation i=1,...,n.

ptp inverse	inverse of the Jacobian transpose times Jacobian matrix (order k).
correlation matrix	parameter correlation coefficient matrix (order k) derived from the ptp inverse matrix.
std error(j)	parameter standard error defined as error(j)=("-unscaled-"se)*sqrt(ptp(j,j)), for j=1,...,k.
one-parameter	one-parameter lower and upper linear confidence limits, based on Student's t=2.0 (default).
support plane	linear lower and upper support plane confidence limits, based on variance F-ratio statistic ff=4.0 (default).
std.error/parm	parameter relative error defined as std error(j)/parameter value(j), for j=1,k.
resistivity(i)	final resistivity (in ohm-meters) of layer i, i=1,...,mm.
depth(i)	final depth (in meters) to bottom of layer i, i=1,...,mm-1.

REFERENCES

Anderson, W. L., 1975, Improved digital filters for evaluating Fourier and Hankel transform integrals: U.S. Geological Survey Report USGS-GD-75-012, 223 p. available from U.S. Department of Commerce, National Technical Information Service (NTIS), Springfield, Va. 22161 as Report PB-242-800/1WC.

Marquardt, D. W., 1963, An algorithm for least-squares estimation of nonlinear parameters: Journal of the Society for Industrial and Applied Mathematics, v. 11, no. 2, p. 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, p. 250-251.

Zohdy, A. A. R., 1975, Automatic interpretation of Schlumberger sounding curves, using modified Dar Zarrouk functions: U.S. Geological Survey Bulletin 1313-E, 39 p.

Appendix 1.-- Source listing

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

```
C--MARQDCLAG: MARQRT INVERSION OF DC SCHLUMBERGER DATA (5/4/79) 00000010
      SUBROUTINE MARQDCLAG_FCODE(Y,X,B,PRNT,F,IN,IDER)          00000130
      SUBROUTINE MARQDCLAG_PCODE(P,X,B,PRNT,F,IN,IP,IB)         00000670
      SUBROUTINE MARQDCLAG_SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT) 00001170
      SUBROUTINE MARQDCLAG_SUBEND(Y,X,B,K,N,TITLE,IOUT)         00001810
      REAL FUNCTION RHLAG1(FUN,TOL,T0,TM,ALOGT,NEW)             00002080
      SUBROUTINE REED(IUNIT,A,NA,B,NB,*)                        00002440
      SUBROUTINE RITE(IUNIT,A,NA,B,NB)                          00002530
      REAL FUNCTION RKERN(X)                                    00002610
      REAL FUNCTION RFVP(X)                                     00002880
      SUBROUTINE MARQRT(FCODE,PCODE,SUBZ,SUBEND)                00003340
      SUBROUTINE GJR (A,N,EPS,MSING)                            00013100
      SUBROUTINE UNSCAL(BIN,BOUT,SCALEP)                        00013760
      REAL FUNCTION ASINH(X)                                    00013970
      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                           00014050
      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)                 00014280
      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)                      00015480
        REAL FUNCTION RLAGH1(X,FUN,TOL,L,NEW)                   00015700
```

Source Availability

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

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

```
C--MARQDCLAG: MARQRT INVERSION OF DC SCHLUMBERGER DATA (5/4/79)          00000010
C  BY LAGGED-CONVOLUTION.                                               00000020
C ** HONEYWELL MULTICS VERSION **                                        00000030
C                                                                        00000040
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.            00000050
C                                                                        00000060
      EXTERNAL MARQDCLAG_FCODE,MARQDCLAG_PCODE,                          00000070
     & MARQDCLAG_SUBZ,MARQDCLAG_SUBEND                                   00000080
      CALL MARQRT(MARQDCLAG_FCODE,MARQDCLAG_PCODE,                       00000090
     & MARQDCLAG_SUBZ,MARQDCLAG_SUBEND)                                  00000100
      STOP                                                               00000110
      END                                                                00000120

      SUBROUTINE MARQDCLAG_FCODE(Y,X,B,PRNT,F,IN,IDER)                   00000130
C--FUNCT. EVAL. FOR 'MARQDCLAG' USING FAST LAGGED-CONVOLUTION.           00000140
C                                                                        00000150
C--PARAMETERS--                                                          00000160
C                                                                        00000170
C       Y=      OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)               00000180
C       X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)           00000190
C       B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)               00000200
C       PRNT=   WORK AND PRINT ARRAY (DIM. 5)                            00000210
C       F=      OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. IN   00000220
C       IN=     OBSERVATION NO. TO EVAL. F (1<=IN<=N)                    00000230
C       IDER=   0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED)  00000240
C               1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED)  00000250
C                                                                        00000260
      REAL Y(1),X(200,5),B(1),PRNT(5),RHO(10),H(9),BR(20)                00000270
      EXTERNAL RKERN                                                     00000280
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                                00000290
      COMMON/RPASS/R(200),ALOGR(200),RS(200),R0,RM,NN,IFIRST             00000300
      IF(IN.GT.1.OR.MM.EQ.1) GO TO 20                                    00000310
      DO 10 J=2,MM                                                       00000320
      IF(B(J).EQ.B(J-1))CALL ERRMSG('SOME RHO(J)=RHO(J-1)',4,6,16)       00000330
10    CONTINUE                                                           00000340
20    DO 30 J=1,5                                                        00000350
30    PRNT(J)=X(IN,J)                                                    00000360
      IF(IN.GT.1) GO TO 800                                              00000370
      IF(IDER.EQ.1) GO TO 8001                                           00000380
35    IF(MM.EQ.1) GO TO 45                                               00000390
      DO 40 J=1,M1                                                       00000400
      RHO(J)=B(J)                                                        00000410
40    H(J)=B(J+MM)                                                       00000420
45    RHO(MM)=B(MM)                                                      00000430
C--GET LAGGED-CONVOLUTION RHOA-FUNCTION (ONLY WHEN IN=1 OR IDER=1)       00000440
      NEW=1                                                              00000450
      DO 50 I=1,NN                                                       00000460
      RS(I)=RHO(1)*(RHLAG1(RKERN,EPS,R0,RM,ALOGR(I),NEW)*R(I)*R(I)+1.0)  00000470
50    NEW=0                                                              00000480
      IF(IDER.EQ.0) GO TO 600                                            00000490
C--CALL RITE TO QUICKLY:      WRITE(20) (RS(I),I=1,NN),(B(J),J=1,M21)    00000500
      CALL RITE(20,RS,NN,B,M21)                                          00000510
```

```
      IFIRST=0                                                          00000520
C--GET PRE-SPLINED SOUNDING                                             00000530
600   F=RS(IN)                                                          00000540
      RETURN                                                            00000550
800   IF(IDER.EQ.0) GO TO 600                                           00000560
C--IDER=1 EST.DER.OPTION (B() VARIES PER CALL FOR EACH IN OBS.)         00000570
8001  IF(IFIRST.EQ.1) GO TO 35                                          00000580
      REWIND 20                                                         00000590
C--CALL REED TO QUICKLY:  READ(20,END=35) (RS(I),I=1,NN),(BR(J),J=1,M21)00000600
801   CALL REED(20,RS,NN,BR,M21,$35)                                    00000610
      DO 802 J=1,M21                                                    00000620
      IF(B(J).NE.BR(J)) GO TO 801                                       00000630
802   CONTINUE                                                          00000640
      GO TO 600                                                         00000650
      END                                                               00000660

      SUBROUTINE MARQDCLAG_PCODE(P,X,B,PRNT,F,IN,IP,IB)                 00000670
C--ANALYTIC PARTIALS FOR 'MARQDCLAG' USING FAST LAGGED-CONVOLUTION       00000680
C                                                                       00000690
C  (PCODE ONLY CALLED BY MARQRT IF IDER=0--DEFAULT)                     00000700
C                                                                       00000710
C--PARAMETERS--                                                         00000720
C                                                                       00000730
C       P=      OUTPUT PARTIAL DERIVATIVE ARRAY (DIM. K)              00000740
C               EVALUATED FOR GIVEN X(IN,),B(K) AT OBS. IN            00000750
C       X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)        00000760
C       B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)            00000770
C       PRNT=   WORK AND PRINT ARRAY (DIM. 5)                         00000780
C       F=      LAST FUNCTION VALUE FROM FCODE AT GIVEN IN.           00000790
C               F MAY OR MAY NOT BE NEEDED--BUT AVAILABLE ANYWAY.     00000800
C       IN=     OBSERVATION NO. TO EVAL. P ARRAY (1<=IN<=N)           00000810
C       IP=     NO. PARAMETERS HELD FIXED (IF ANY--IF NONE IP=0).     00000820
C       IB=     ARRAY OF PARAMETER INDICES HELD FIXED IF IP.GT.0      00000830
C               (DIM. 19).                                            00000840
C                                                                       00000850
      INTEGER IB(1)                                                     00000860
      REAL P(1),X(200,5),B(1),PRNT(5),RHO(10),H(9),PM(200,20)           00000870
      EXTERNAL RFVP                                                     00000880
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                               00000890
      COMMON/RPASS/R(200),ALOGR(200),RS(200),R0,RM,NN,IFIRST            00000900
      IF(IN.GT.1) GO TO 50                                              00000910
      DO 30 J=1,M21                                                     00000920
      JJ=J                                                              00000930
      IFIX=0                                                            00000940
      IF(IP.LE.0) GO TO 11                                              00000950
      DO 1 I=1,IP                                                       00000960
      IF(IB(I).EQ.J) IFIX=1                                             00000970
1     CONTINUE                                                          00000980
      IF(IFIX.EQ.1) GO TO 6                                             00000990
C--GET LAGGED-CONVOLUTION PARTIALS OF RHOA-FUNCTION (ONLY WHEN IN=1)    00001000
11    NEW=1                                                             00001010
      DO 5 I=1,NN                                                       00001020
```

```
      PP=RHO(1)*R(I)*R(I)*RHLAG1(RFVP,EPS,RO,RM,ALOGR(I),NEW)          00001030
      IF(J.EQ.1) PP=PP+RS(I)/RHO(1)                                    00001040
      NEW=0                                                            00001050
      PM(I,J)=PP                                                       00001060
      CONTINUE                                                         00001070
      GO TO 30                                                         00001080
      DO 7 I=1,NN                                                      00001090
      PM(I,J)=0.0                                                      00001100
      CONTINUE                                                         00001110
-GET PRE-SPLINED PARTIALS                                              00001120
      DO 60 J=1,M21                                                    00001130
      P(J)=PM(IN,J)                                                    00001140
      RETURN                                                           00001150
      END                                                              00001160

      SUBROUTINE MARQDCLAG_SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)         00001170
-- INITIALIZATION ROUTINE (CALLED ONCE BY MARQRT)                      00001180
                                                                       00001190
  SUBZ IS CALLED BY MARQRT AFTER THE DATA Y(I),X(I,5) ARE READ--       00001200
  SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL $INIT                  00001210
  PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE...            00001220
                                                                       00001230
--PARAMETERS--                                                         00001240
                                                                       00001250
       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.                         00001260
       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY             00001270
               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES              00001280
               PRINTED BY PGM MARQRT...                                00001290
       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)                   00001300
       TITLE=  ALPHA TITLE ARRAY READ IN BY PGM MARQRT.                00001310
       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED                     00001320
               0 IF ONLY UNIT 6 PRINT FILE USED.                       00001330
                                                                       00001340
      CHARACTER*5 TITLE(16)                                            00001350
      REAL Y(1),X(200,5),B(1),PRNT(1),RHO(10),H(9)                     00001360
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                              00001370
      COMMON/RPASS/R(200),ALOGR(200),RS(200),RO,RM,NN,IFIRST           00001380
      NAMELIST/INIT/MM,EPS                                             00001390
      DATA ISUBZ/0/                                                    00001400
      IF(ISUBZ.NE.0) GO TO 10                                          00001410
--PRESET                                                               00001420
      ISUBZ=1                                                          00001430
      MM=1                                                             00001440
      EPS=.1E-5                                                        00001450
0     READ(5,INIT)                                                     00001460
      WRITE(6,20) TITLE,MM,EPS                                         00001470
0     FORMAT('1M A R Q D C L A G -- ',5X,16A5//' MM=',I3/' EPS=',E16.8) 00001480
      IF(IOUT.EQ.1) WRITE(16,20) TITLE,MM,EPS                          00001490
--TEST $INIT PARMS                                                     00001500
      IF(MM.LT.1.OR.MM.GT.10)CALL ERRMSG('MM<1 OR >10    ',3,6,16)     00001510
--TEST X(I, ) DATA BEFORE PROCEEDING                                   00001520
      IF(X(1,1).LE.0.0)CALL ERRMSG('X(1,1)<=0.',2,6,16)                00001530
```

```
      R(1)=X(1,1)                                                       00001540
      ALOGR(1)=ALOG(R(1))                                               00001550
      DO 40 I=2,N                                                       00001560
      IF(X(I,1).LE.X(I-1,1).OR.X(I,1).LE.0.0)                           00001570
     * CALL ERRMSG('SOME X(I,1)<=0.0 OR NOT INCREASING.',7,6,16)         00001580
      R(I)=X(I,1)                                                       00001590
      ALOGR(I)=ALOG(R(I))                                               00001600
)     CONTINUE                                                          00001610
--PRESET SOME GLOBAL CONSTANTS                                          00001620
      IFIRST=1                                                          00001630
      NPRNT=2                                                           00001640
      NN=N                                                              00001650
      R0=.5*X(1,1)                                                      00001660
      RM=X(N,1)                                                         00001670
      M1=MM-1                                                           00001680
      M21=2*MM-1                                                        00001690
      WRITE(6,60) (I,I,I=1,MM)                                          00001700
      IF(IOUT.EQ.1) WRITE(16,60) (I,I,I=1,MM)                           00001710
)     FORMAT(////' PARAMETER ORDER--'//(5X,I3,6X,'  RHO(',I3,')'))      00001720
      IF(MM.EQ.1) GO TO 90                                              00001730
      DO 70 I=1,M1                                                      00001740
      J=MM+I                                                            00001750
      IF(IOUT.EQ.1) WRITE(16,80) J,I                                    00001760
)     WRITE(6,80) J,I                                                   00001770
)     FORMAT(5X,I3,6X,'THICK(',I3,')')                                  00001780
)     RETURN                                                            00001790
      END                                                               00001800

      SUBROUTINE MARQDCLAG_SUBEND(Y,X,B,K,N,TITLE,IOUT)                 00001810
--TERMINATION ROUTINE (CALLED ONCE BY MARQRT)                           00001820
--FOLLOWING CHARACTER STMT. ONLY FOR HONEYWELL MULTICS SYS:             00001830
      CHARACTER*5 TITLE(16)                                             00001840
      REAL Y(1),X(200,5),B(1)                                           00001850
      WRITE(6,10) TITLE                                                 00001860
)     FORMAT(//' ********  E N D  ********  ',6X,16A5//                 00001870
     *' FINAL UNSCALED PARAMETERS--',10X,'RESISTIVITY',11X,'DEPTH'/)     00001880
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                                  00001890
      MM=(K+1)/2                                                        00001900
      DO 30 I=1,MM                                                      00001910
      WRITE(6,20) I,B(I),I,B(I)                                         00001920
   20 FORMAT(5X,I3,4X,E16.8,2X,I3,1X,E16.8)                             00001930
      IF(IOUT.EQ.1) WRITE(16,20) I,B(I),I,B(I)                          00001940
   30 CONTINUE                                                          00001950
      IF(K.EQ.1) GO TO 60                                               00001960
      M2=MM+1                                                           00001970
      D=0.0                                                             00001980
      DO 50 I=M2,K                                                      00001990
      D=D+B(I)                                                          00002000
      L=I-MM                                                            00002010
      WRITE(6,40) I,B(I),L,D                                            00002020
   40 FORMAT(5X,I3,4X,E16.8,24X,I3,1X,E16.8)                            00002030
      IF(IOUT.EQ.1) WRITE(16,40) I,B(I),L,D                             00002040
```

```
   50 CONTINUE                                                          00002050
   60 RETURN                                                            00002060
      END                                                               00002070

      REAL FUNCTION RHLAG1(FUN,TOL,T0,TM,ALOGT,NEW)                     00002080
C--REAL HANKEL-1 TRANSFORM LAG CONVOLUTION & SPLINE INTERPOLATION        00002090
C  VIA LAGGED-CONVOLUTION ROUTINE RLAGH1 DESCRIBED IN                   00002100
C  ANDERSON,1975,NTIS REPT. PB-242-800,P.70-75.                         00002110
C                                                                       00002120
C    FUN = EXTERNAL REAL KERNEL FUNCTION.                               00002130
C    TOL = TOLERANCE REQUESTED FOR RLAGH1                               00002140
C     T0 = TMIN TO USE (E.G., LET T0=.5*TMIN, TMIN=TRUE)                00002150
C     TM = TMAX TO USE (TM>T0)                                          00002160
C  ALOGT = ALOG(T), T= TRANSFORM PARAMETER (T0<=T<=TM) FOR NEW=1 OR 0    00002170
C    NEW = 1 REQUIRED FOR 1ST CALL OR TO RESET SPLINE COEFFICIENTS.      00002180
C    NEW = 0 FOR ALL CALLS AFTER 1ST--USES SPLINE INTERPOLATION ONLY.    00002190
C                                                                       00002200
      REAL ARG(200),Y(200),AR(200),BR(200),CR(200),                     00002210
     & D(2),W1(200),W2(200)                                             00002220
      EXTERNAL FUN                                                      00002230
      DATA D/2*0.0/                                                     00002240
      IF(NEW.EQ.0) GO TO 3                                              00002250
      NT=AINT(5.*ALOG(TM/T0))+5                                         00002260
      IF(NT.GT.200)CALL ERRMSG('IN RHLAG1: NT>200   ',4,6,16)           00002270
      NT1=NT+1                                                          00002280
      X0=ALOG(T0)+.2*NT                                                 00002290
      NU=1                                                              00002300
      DO 1 J=1,NT                                                       00002310
      I=NT1-J                                                           00002320
      X=X0-.2*J                                                         00002330
      ARG(I)=X                                                          00002340
      EX=EXP(X)                                                         00002350
      Y(I)=RLAGH1(X,FUN,TOL,L,NU)/EX                                    00002360
1     NU=0                                                              00002370
      CALL SPLIN1(NT,0.0,ARG,Y,AR,BR,CR,0,D,W1,W2)                      00002380
2     IF(NT.LT.0) CALL ERRMSG('IN RHLAG1: NT<0 AFTER SPLIN1  ',6,6,16)  00002390
3     CALL SPOINT(NT,ARG,Y,AR,BR,CR,ALOGT,X)                            00002400
      RHLAG1=X                                                          00002410
      RETURN                                                            00002420
      END                                                               00002430

      SUBROUTINE REED(IUNIT,A,NA,B,NB,*)                                00002440
C--DYNAMIC ARRAYS A(NA) AND B(NB) VECTOR READ(IUNIT,END=$N) A,B          00002450
C  FOR QUICKER INPUT.                                                   00002460
C                                                                       00002470
      DIMENSION A(NA),B(NB)                                             00002480
      READ(IUNIT,END=1) A,B                                             00002490
      RETURN                                                            00002500
1     RETURN (1)                                                        00002510
      END                                                               00002520
```

```
      SUBROUTINE RITE(IUNIT,A,NA,B,NB)                                  00002530
C--DYNAMIC ARRAYS A(NA) AND B(NB) VECTOR WRITE(IUNIT) A,B                00002540
C  FOR QUICKER OUTPUT.                                                  00002550
C                                                                       00002560
      DIMENSION A(NA),B(NB)                                             00002570
      WRITE(IUNIT) A,B                                                  00002580
      RETURN                                                            00002590
      END                                                               00002600

      REAL FUNCTION RKERN(X)                                            00002610
C--KERNEL FUNCTION USED IN FCODE INTEGRAL                               00002620
C  FOR SCHLUMBERGER APPARENT RESISTIVITY                                00002630
C  IN PROGRAM 'MARQDCLAG'                                               00002640
C                                                                       00002650
      REAL RHO(10),H(9)                                                 00002660
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                               00002670
      X2=-2.0*X                                                         00002680
      V=1.0                                                             00002690
      IF(MM.LE.1) GO TO 30                                              00002700
      I=MM                                                              00002710
   10 I1=I-1                                                            00002720
      T=V/RHO(I1)                                                       00002730
      TR=T*RHO(I)                                                       00002740
      E=X2*H(I1)                                                        00002750
C--HONEYWELL MULTICS TEST  $$$$$$$$$$$$$$$$                             00002760
      IF(E.LT.-88.028) GO TO 40                                         00002770
      T=((1.0-TR)/(1.0+TR))*EXP(E)                                      00002780
      V=(1.0-T)/(1.0+T)                                                 00002790
   20 IF(I.LE.2) GO TO 30                                               00002800
      I=I-1                                                             00002810
      GO TO 10                                                          00002820
   30 RKERN=X*(V-1.0)                                                   00002830
      RETURN                                                            00002840
   40 V=1.0                                                             00002850
      GO TO 20                                                          00002860
      END                                                               00002870

      REAL FUNCTION RFVP(X)                                             00002880
C--RESISTIVITY KERNEL USED IN INTEGRAL OF PARTIAL RHOA W/R B(JJ),        00002890
C  JJ=1,2*MM-1 GIVEN IN COMMON/RESIS/.                                  00002900
C  USED IN PROGRAM 'MARQDCLAG'                                          00002910
C  (RFVP BY RECURRENCE METHOD).                                         00002920
C                                                                       00002930
      REAL RHO(10),H(9),K1                                              00002940
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                               00002950
      X2=-2.0*X                                                         00002960
      JJMM=JJ-MM                                                        00002970
      VM=1.0                                                            00002980
      PV1=0.0                                                           00002990
      IF(MM.EQ.1) GO TO 40                                              00003000
C--INITIALIZE PARTIAL INDEX J1=MM-1 (NUM. INDEX)                        00003010
      J=MM                                                              00003020
```

```
C--LOOP ON J1 INDEX                                                    00003030
   10 J1=J-1                                                           00003040
      E=X2*H(J1)                                                       00003050
      E1=0.0                                                           00003060
C--HONEYWELL MULTICS TEST  $$$$$$$$$$$$$$$$$$                          00003070
      IF(E.GT.-88.028) E1=EXP(E)                                       00003080
      DENK1=1.0/(RHO(J1)+RHO(J)*VM)                                    00003090
      K1=DENK1*(RHO(J1)-RHO(J)*VM)                                     00003100
      DENV1=1.0/(1.0+K1*E1)                                            00003110
      V1=DENV1*(1.0-K1*E1)                                             00003120
      IF(JJ.LE.MM) GO TO 20                                            00003130
C--RECUR FOR PARTIAL W/R H(JJ)                                         00003140
      PEH=0.0                                                          00003150
      IF(JJMM.EQ.J1) PEH=X2*E1                                         00003160
      PKH=-DENK1*RHO(J)*PV1*(1.0+K1)                                   00003170
      PV1=-DENV1*(K1*PEH+E1*PKH)*(1.0+V1)                              00003180
      GO TO 30                                                         00003190
C--RECUR FOR PARTIAL W/R RHO(JJ)                                       00003200
   20 PR1=0.0                                                          00003210
      IF(JJ.EQ.J1) PR1=1.0                                             00003220
      PRM=0.0                                                          00003230
      IF(JJ.EQ.J) PRM=1.0                                              00003240
      PKR=DENK1*(PR1*(1.0-K1)-(1.0+K1)*(RHO(J)*PV1+VM*PRM))            00003250
      PV1=-DENV1*E1*PKR*(1.0+V1)                                       00003260
   30 IF(J.LE.2) GO TO 40                                              00003270
      VM=V1                                                            00003280
      J=J1                                                             00003290
      GO TO 10                                                         00003300
   40 RFVP=X*PV1                                                       00003310
      RETURN                                                           00003320
      END                                                              00003330

      SUBROUTINE MARQRT(FCODE,PCODE,SUBZ,SUBEND)                       00003340
C--(MARQRT)-- GENERAL MARQUARDT NONLINEAR LEAST SQUARES-- 7/11/78.     00003350
C** HONEYWELL MULTICS VERSION **                                        00003360
C  SUBPROGRAM MARQRT IS TO BE LINKED/LOADED WITH USER WRITTEN          00003370
C  SUBROUTINES (FCODE,PCODE,SUBZ, AND SUBEND) FOR                      00003380
C  SPECIFIC NONLINEAR PROBLEM TO BE SOLVED.                            00003390
C                                                                      00003400
C--THE USER MUST DECLARE THE CALLING PARAMETERS FCODE,PCODE,           00003410
C  SUBZ,SUBEND (ANY DESIRED NAMES MAY BE USED) AS EXTERNAL IN          00003420
C  MAIN CALLING PROGRAM; E.G.,                                         00003430
C                                                                      00003440
C       EXTERNAL FCODE,PCODE,SUBZ,SUBEND                               00003450
C       CALL MARQRT(FCODE,PCODE,SUBZ,SUBEND)                           00003460
C       STOP                                                           00003470
C       END                                                            00003480
C                                                                      00003490
C--THIS IS A MODIFIED VERSION OF 'IBM SHARE PROGRAM NO. 1428'.         00003500
C *** MODIFIED BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY,DENVER, COLORADO 00003510
C  FOR NAMELIST INPUT, IMPROVED ESTIMATED DERIVATIVES,                 00003520
C  MODIFIED MARQUARDT LAMBDA DETERMINATION,                            00003530
```

```
C  DATA AND PARAMETER SCALING, WEIGHTED OBSERVATIONS, AND                00003540
C  OTHER CHANGES--ALL DONE IN SINGLE-PRECISION FOR THE                    00003550
C *** HONEYWELL MULTICS SYSTEM ***                                       00003560
C                                                                        00003570
C--SEE SHARE PROGRAM NO. 1428 AND/OR COMMENTS IN SUBPROGRAMS FCODE,      00003580
C  PCODE,SUBZ AND SUBEND BELOW, FOR DETAILS ON CODING THE                00003590
C  REQUIRED SUBROUTINES FCODE,PCODE,SUBZ, AND SUBEND.                    00003600
C                                                                        00003610
C--OPERATING NOTE FOR HONEYWELL MULTICS SYSTEM:  $$$$$$$$$$$$$$$$$$$$  00003620
C                                                                        00003630
C  TO OBTAIN ON-LINE (INTERACTIVE) PRINTING ON UNIT 6 AND                00003640
C  DEFERRED PRINTING ON UNIT 16, USE MULTICS RUN.EC, I.E.,               00003650
C  'RUN &1' OR 'RUN_EO &1' AND DPRINT '&1.FILE16.LIST' AFTER RUN.        00003660
C                                                                        00003670
C--FOLLOWING CHARACTER STATEMENTS ONLY FOR MULTICS SYSTEM:               00003680
      CHARACTER*5 TITLE                                                  00003690
      CHARACTER*4 FMT                                                    00003700
      INTEGER SCALEP,SCALEY, SP,SY                                       00003710
      DIMENSION FMT(18),PRNT(5),SPRNT(5),TITLE(16)                       00003720
      DIMENSION BS(20),DB(20),BA(20),G(20),IB(19),SA(20),P(20)           00003730
      DIMENSION A(20,20),B(20), BINV(20)                                 00003740
      DIMENSION X(200,5),Y(200),WT(200)                                  00003750
      DIMENSION XNU(5),SS(4)                                             00003760
      EQUIVALENCE (X(1,5),WT(1)),(IOUT,IFSS1),(IDER,IWS2),               00003770
     1 (IPRT,IWS3),(NITER,IWS4),(INON,IWS6),(SP,SCALEP),(SY,SCALEY)      00003780
C=========================================================              00003790
      NAMELIST/PARMS/N,K,IP,M,IALT,IDER,IPRT,NITER,INON,NPRNT,           00003800
     1 IB,FF,T,E,TAU,XL,GAMCR,DEL,ZETA,B,IOUT,IWT,ISTOP,                 00003810
     2 SCALEP,SCALEY,MODLAM, SP,SY                                       00003820
C=========================================================              00003830
      DATA XNU/1.33,1.78,3.16,10.,100./                                  00003840
C                                                                        00003850
C     MAX NO OF PARAMETERS IS K=20  (NOTE: K=N IS ALLOWED)               00003860
C     MAX NO OF IND VARS IS M=4                                          00003870
C     MAX NO OF OBSERVATIONS IS N=200                                    00003880
C     INTERNAL #IWHER# SWITCH USAGE--                                    00003890
C                  IWHER =-1 MEANS INITIALIZE VIA SUBROUTINE SUBZ.       00003900
C                  IWHER = 0 MEANS START NEW PROBLEM OR END RUN          00003910
C                  IWHER = 1 MEANS GET P(S) AND F                        00003920
C                  IWHER GREATER THAN 1 MEANS GET F ONLY                 00003930
C--FOLLOWING CALL TO SUPPRESS EXP-UNDERFLOW MESSAGES                     00003940
C  FOR THE DEC-10 AND OTHER SYSTEMS:  $$$$$$$$$$$$$$$$$$$$              00003950
C//        CALL ERRSET(0)                                                00003960
C** FOR THE HONEYWELL MULTICS SYSTEM, USE (INSTEAD) THE FOLLOWING:       00003970
C   IO DETACH ERROR_OUTPUT                                               00003980
C   IO ATTACH ERROR_OUTPUT DISCARD_                                      00003990
C  (OR-- ON USGS SYS, USE SET_UFL -OFF)                                  00004000
C**                                                                      00004010
C--PRESET GLOBAL PARMS (SOME MAY BE OVERRIDDEN BY $PARMS READ-IN)        00004020
      IP=0                                                               00004030
      N=0                                                                00004040
      K=0                                                                00004050
```

```
      M=0                                                                00004060
      NPRNT=0                                                            00004070
      MODLAM=1                                                           00004080
      ISTOP=1                                                            00004090
      IWT=0                                                              00004100
      IALT=10                                                            00004110
      IOUT=1                                                             00004120
      IDER=0                                                             00004130
      IPRT=0                                                             00004140
      MITER=10                                                           00004150
      INON=1                                                             00004160
      LSCALP=0                                                           00004170
      LSCALY=0                                                           00004180
      FF=4.0E0                                                           00004190
      E=.00005E0                                                         00004200
      TAU=.001E0                                                         00004210
      T=2.0E0                                                            00004220
      DEL=.00001E0                                                       00004230
      ZETA=.1E-30                                                        00004240
      GAMCR=45.0E0                                                       00004250
C                                                                        00004260
   10 GAMMA=0.E0                                                         00004270
      SCALEP=LSCALP                                                      00004280
      SCALEY=LSCALY                                                      00004290
      XLL=0.E0                                                           00004300
      SE=0.0                                                             00004310
      NITER=MITER                                                        00004320
   20 IWHER=0                                                            00004330
      ISS=1                                                              00004340
      INU=4                                                              00004350
      XNUFAC=10.0                                                        00004360
      GO TO 150                                                          00004370
   30 CONTINUE                                                           00004380
      IF (IWHER.GT.0) GO TO 100                                          00004390
      IF (IWHER.EQ.0) GO TO 240                                          00004400
C=====================================================================  00004410
C     INITIALIZATION (IWHER=-1, IFSS1=IOUT)                              00004420
      CALL SUBZ (Y,X,BINV,PRNT,NPRNT,N,TITLE,IFSS1)                      00004430
C     ***************************************************              00004440
      IPRNT=NPRNT-1                                                      00004450
      IF(NPRNT.LT.0) IPRNT=IABS(NPRNT)-2                                 00004460
C                                                                        00004470
C--NOTE: IPRNT IS A SPECIAL INDEX USED IN SCALEY=2 CASES                 00004480
C  TO MIX LOG OR ASINH TYPE SCALING WHEN ABS(X(I,IPRNT))=1. OR NOT 1.    00004490
C  RESPECTIVELY, AND ONLY WHEN IPRNT.GT.1                                00004500
      NPRNT=IABS(NPRNT)                                                  00004510
      IF(SCALEY.EQ.0) GO TO 90                                           00004520
      DO 80 I=1,N                                                        00004530
      IF(SCALEY-1) 90,40,60                                              00004540
   40 IF(Y(I).LE.0.)CALL ERRMSG(30HSOME Y(I).LE.0 AND SCALEY=1...,       00004550
     1 6,6,16)                                                           00004560
   50 Y(I)=ALOG(Y(I))                                                    00004570
```

```
      GO TO 80                                                          00004580
   60 IF(IPRNT.LE.1) GO TO 70                                           00004590
      IF(ABS(X(I,IPRNT)).NE.1.0) GO TO 70                               00004600
      IF(Y(I).LE.0.)                                                    00004610
     1CALL ERRMSG(50HSOME Y(I).LE.0 WHEN ABS(X(I,IPRNT))=1 AND SCALEY=2,00004620
     2 10,6,16)                                                         00004630
      GO TO 50                                                          00004640
   70 Y(I)=ASINH(Y(I))                                                  00004650
   80 CONTINUE                                                          00004660
   90 CONTINUE                                                          00004670
      IF (IBOUT.EQ.0) GO TO 150                                         00004680
      GO TO 20                                                          00004690
  100 CONTINUE                                                          00004700
C=============================================================          00004710
C                  COMPUTE F VIA SUBR. FCODE                          00004720
C                  NPRNT IS THE NO OF OTHER WORDS TO BE PRINTED       00004730
C                  THE WORDS TO BE PRINTED ARE IN PRNT(1)...PRNT(5)   00004740
C--CALL FCODE FOR CURRENT BINV AND I-TH OBSERVATION (IFSS2=IDER)       00004750
      CALL FCODE(Y,X,BINV,PRNT,F,I,IFSS2)                               00004760
C     ************************************************                  00004770
      FINV=F                                                            00004780
      IF(SCALEY-1) 140,110,120                                          00004790
  110 F=ALOG(F)                                                         00004800
      GO TO 140                                                         00004810
  120 IF(IPRNT.LE.1) GO TO 130                                          00004820
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 110                              00004830
  130 F=ASINH(F)                                                        00004840
  140 CONTINUE                                                          00004850
      IF (IWHER.NE.1) GO TO 150                                         00004860
      IF (IFSS2.NE.0) GO TO 150                                         00004870
C=============================================================          00004880
C                  COMPUTE P(J)=DF/DB VIA SUBR PCODE FOR J=1,K.       00004890
C                      USING X(I,L) AND B(J)                          00004900
C--CALL PCODE FOR CURRENT BINV,FINV AND I-TH OBSERVATION               00004910
      CALL PCODE(P,X,BINV,PRNT,FINV,I,IP,IB)                            00004920
C     ************************************************                  00004930
C                THIS IS GENERAL #IWHER# SWITCH                       00004940
  150 CONTINUE                                                          00004950
      IF (IWHER.LT.0) GO TO 320                                         00004960
      IF (IWHER.EQ.0) GO TO 160                                         00004970
C               1   2   3   4   5                                       00004980
      GO TO (490,1560,530,580,590), IWHER                               00004990
C                  READ FIRST CARD OF NEXT CASE                       00005000
  160  ITCT=0                                                           00005010
      IBOUT=0                                                           00005020
C=============================================================          00005030
C  READ $PARMS --$                                                     00005040
C--ALWAYS PRESET XL=.01 (MAY BE OVERRIDDEN BY $$PARMS READ-IN)         00005050
C  AND CLEAR B(I),I=1,20 TO FORCE INITIALIZATION...                    00005060
      XL=.01                                                            00005070
      DO 170 I=1,20                                                     00005080
  170 B(I)=0.E0                                                         00005090
```

```
      READ(5,180) TITLE                                                 00005100
  180 FORMAT(16A5)                                                      00005110
      READ(5,PARMS)                                                     00005120
C--TEST $PARMS                                                          00005130
      IF(N.GT.200.OR.K.GT.20.OR.M.GT.4.OR.IWT.GT.1.OR.IP.GT.19.OR.      00005140
     1 IALT.EQ.6.OR.IALT.EQ.13.OR.IALT.EQ.16.OR.                        00005150
     2 N.LT.1.OR.K.LT.1.OR.M.LT.1.OR.IWT.LT.0.OR.IP.LT.0.OR.            00005160
     3 SCALEY.LT.0.OR.SCALEY.GT.2.OR.SCALEP.LT.0.OR.SCALEP.GT.2.OR.     00005170
     4 N.LT.K) CALL ERRMSG(30HSOME $PARMS OUT OF RANGE..     ,6,6,16)   00005180
      DO 210 I=1,K                                                      00005190
      IF(B(I).EQ.0.E0) CALL ERRMSG(20HSOME B(I) = 0.0      ,4,6,16)     00005200
      BINV(I)=B(I)                                                      00005210
      IF(SCALEP-1) 210,190,200                                          00005220
  190 IF(B(I).LT.0.0)CALL ERRMSG(30HSOME B(I).LT.0. AND SCALEP=1..,     00005230
     1 6,6,16)                                                          00005240
      B(I)=ALOG(B(I))                                                   00005250
      GO TO 210                                                         00005260
  200 B(I)=ASINH(B(I))                                                  00005270
  210 CONTINUE                                                          00005280
      MAXITR=IWS4                                                       00005290
      MITER=NITER                                                       00005300
      ITER=1                                                            00005310
      WRITE (6,2730)                                                    00005320
      IF (IFSS1.NE.1) GO TO 250                                         00005330
      WRITE (16,2730)                                                   00005340
      GO TO 250                                                         00005350
C=======================================================              00005360
C                   END OF LAST PROBLEM                                 00005370
  220 CALL SUBEND(Y,X,BINV,K,N,TITLE,IOUT)                              00005380
C     ************************************************                  00005390
  240 IF(ISTOP.EQ.1.OR.IALT.EQ.5) GO TO 241                             00005400
C--INITIALIZE FOR NEXT PROB (SAME IALT DATA), SINCE ISTOP=0             00005410
      GO TO 10                                                          00005420
C--FOLLOWING CLOSE STMT ONLY FOR HONEYWELL MULTICS:                     00005430
  241 CALL CLOSE_FILE('-ALL')                                           00005440
C     STOP                                                              00005450
      RETURN                                                            00005460
  250 CONTINUE                                                          00005470
      IF (IP.LE.0) GO TO 280                                            00005480
      DO 270 I=1,IP                                                     00005490
      IF (IB(I).GT.0) GO TO 270                                         00005500
      CALL ERRMSG(30HIP.GT.1 BUT SOME IB(I).LE.0...,6,6,16)             00005510
  270 CONTINUE                                                          00005520
  280 CONTINUE                                                          00005530
      IF (K.GT.10) GO TO 290                                            00005540
C--IBKT=1 MEANS USE UPPER A MATRIX FOR SCRATCH STORAGE                  00005550
C       =2 MEANS USE FILE 13 FOR SCRATCH STORAGE                        00005560
      IBKT=1                                                            00005570
      GO TO 300                                                         00005580
  290 IBKT=2                                                            00005590
  300 XKDB=1.E0                                                         00005600
C--READ OBJECT TIME FORMAT FOR DATA ON FILE IALT.                       00005610
```

```
      READ(5,2480) (FMT(I),I=1,18)                                      00005620
      M1=M+IWT                                                          00005630
      DO 310 I=1,N                                                      00005640
      READ(IALT,FMT) Y(I),(X(I,L),L=1,M1)                               00005650
C--SET UP WTS VIA IWT PARM                                              00005660
      WT(I)=1.0E0                                                       00005670
      IF(IWT.EQ.1.AND.X(I,M1).NE.0.0) WT(I)=1.0E0/X(I,M1)**2            00005680
  310 CONTINUE                                                          00005690
      IF(IALT.NE.5) REWIND IALT                                         00005700
      IWHER=-1                                                          00005710
       GO TO 30                                                         00005720
  320 IBKA=1                                                            00005730
C                                                                       00005740
C     .................................................................00005750
C                   START THE CALCULATION OF THE PTP MATRIX             00005760
      WRITE(6,2520) TITLE                                               00005770
      WRITE (6,2530) N,K,IP,M,GAMCR,DEL,MODLAM,FF,T,E,TAU,XL,ZETA,      00005780
     1 IALT,ISTOP,IWT,IWS2,IWS3,IWS4,IWS6,IFSS1,NPRNT,SCALEP,SCALEY     00005790
      IF(IP.GT.0) WRITE(6,330) (IB(J),J=1,IP)                           00005800
  330 FORMAT(4H IB=,19I3)                                               00005810
      WRITE(6,340) FMT                                                  00005820
  340 FORMAT(5H FMT=,18A4)                                              00005830
      IF(SCALEP.GT.0.AND.IPRT.GE.0) WRITE(6,350) (BINV(J),J=1,K)        00005840
  350 FORMAT(/30H -INITIAL UNSCALED PARAMETERS-/(12X,4E17.8))           00005850
      IF (IFSS1.NE.1) GO TO 360                                         00005860
      WRITE(16,2520) TITLE                                              00005870
      WRITE (16,2530) N,K,IP,M,GAMCR,DEL,MODLAM,FF,T,E,TAU,XL,ZETA,     00005880
     1 IALT,ISTOP,IWT,IWS2,IWS3,IWS4,IWS6,IFSS1,NPRNT,SCALEP,SCALEY     00005890
      IF(IP.GT.0) WRITE(16,330) (IB(J),J=1,IP)                          00005900
      WRITE(16,340) FMT                                                 00005910
      IF(SCALEP.GT.0.AND.IPRT.GE.0) WRITE(16,350) (BINV(J),J=1,K)       00005920
  360 CONTINUE                                                          00005930
  370 CONTINUE                                                          00005940
      DO 380 I=1,K                                                      00005950
      G(I)=0.E0                                                         00005960
      DO 380 J=1,K                                                      00005970
  380 A(I,J)=0.E0                                                       00005980
      IF(IBKA-2) 390,400,400                                            00005990
  390 IFSS3=IWS3                                                        00006000
      IFSS2=IWS2                                                        00006010
      GO TO 410                                                         00006020
  400 IFSS3=1                                                           00006030
      GO TO 420                                                         00006040
  410 IF(IPRT.GE.0) WRITE (6,2540) (B(J),J=1,K)                         00006050
      IF (IFSS1.NE.1) GO TO 420                                         00006060
      IF(IPRT.GE.0) WRITE (16,2540) (B(J),J=1,K)                        00006070
  420 CONTINUE                                                          00006080
  430 FORMAT(/11H -UNSCALED-)                                           00006090
C--THIS IS I=1 TO N SPECIAL NON-DO LOOP                                 00006100
  450 I=1                                                               00006110
      DO 460 J=1,K                                                      00006120
  460 CALL UNSCAL(B(J),BINV(J),SCALEP)                                  00006130
```

```
      IF(IPRT.LT.0) WRITE(6,2540) (BINV(J),J=1,K)                          00006140
      IF(IFSS1.EQ.1.AND.IPRT.LT.0)WRITE(16,2540)(BINV(J),J=1,K)            00006150
      PHI=0.E0                                                             00006160
      IF (IFSS2.EQ.0) GO TO 480                                            00006170
      GO TO 510                                                            00006180
  470 IF (IFSS2.EQ.1) GO TO 520                                            00006190
C     ..................................................                 00006200
C                  THIS IS THE ANALYTICAL P(J) ROUTINE                   00006210
  480 IWHER=1                                                              00006220
C                  GET P(J) AND F                                        00006230
      GO TO 30                                                             00006240
  490 IF (IP.LE.0) GO TO 640                                               00006250
      DO 500 II=1,IP                                                       00006260
      IWS=IB(II)                                                           00006270
  500 P(IWS)=0.E0                                                          00006280
      GO TO 640                                                            00006290
C      ..............................................                    00006300
C                  THIS IS THE ESTIMATED P(J) ROUTINE                    00006310
C  (VIA K.M. BROWN S METHOD)                                               00006320
  510 CONTINUE                                                             00006330
      ISW=1                                                                00006340
      IF(XL.LT.0.1E-3) ISW=2                                               00006350
  520 IWHER=3                                                              00006360
      GO TO 30                                                             00006370
  530 FWS=FINV                                                             00006380
      FSAV=F                                                               00006390
      DO 540 II=1,NPRNT                                                    00006400
  540 SPRNT(II)=PRNT(II)                                                   00006410
      J=1                                                                  00006420
  550 IF (IP.LE.0) GO TO 570                                               00006430
      DO 560 II=1,IP                                                       00006440
      IF ((J-IB(II)).EQ.0) GO TO 610                                       00006450
  560 CONTINUE                                                             00006460
  570 HH=DEL*ABS(BINV(J))                                                  00006470
      IF(ISW.EQ.2) HH=1.E3*HH                                              00006480
      IF(HH.LE.5.E-5) HH=5.E-5                                             00006490
      TWS=B(J)                                                             00006500
      TWS1=BINV(J)                                                         00006510
      BINV(J)=TWS1+HH                                                      00006520
      IWHER=4                                                              00006530
      GO TO 30                                                             00006540
  580 B(J)=TWS                                                             00006550
      BINV(J)=TWS1                                                         00006560
      IF(ISW.EQ.1) GO TO 600                                               00006570
C--CENTRAL DIFFERENCES (ISW=2--WHEN XL.LT..1E-3)                           00006580
      FHH=FINV                                                             00006590
      BINV(J)=TWS1-HH                                                      00006600
      IWHER=5                                                              00006610
      GO TO 30                                                             00006620
  590 B(J)=TWS                                                             00006630
      BINV(J)=TWS1                                                         00006640
      P(J)=.5E0*(FHH-FINV)/HH                                              00006650
```

```
 GO TO 620                                                          00006660
RWARD DIFFERENCES (ISW=1--WHEN XL.GE..1E-3)                         00006670
 P(J)=(FINV-FWS)/HH                                                 00006680
 GO TO 620                                                          00006690
 P(J)=0.E0                                                          00006700
 J=J+1                                                              00006710
 IF ((J-K).LE.0) GO TO 550                                          00006720
 FINV=FWS                                                           00006730
 F=FSAV                                                             00006740
 DO 630 II=1,NPRNT                                                  00006750
 PRNT(II)=SPRNT(II)                                                 00006760
             END OF ESTIMATED P S ROUTINE                           00006770
  ..................................................                00006780
             NOW, USE THE P(J) TO MAKE PARTIALS MATRIX              00006790
T UP FOR SCALING PARTIAL DERIVATIVES AS SELECTED                    00006800
 IF(SCALEP-1) 650,710,730                                           00006810
 IF(SCALEY-1) 750,660,690                                           00006820
 DEN=1.0E0/FINV                                                     00006830
 DO 680 JJ=1,K                                                      00006840
 P(JJ)=P(JJ)*DEN                                                    00006850
 GO TO 750                                                          00006860
 IF(IPRNT.LE.1) GO TO 700                                           00006870
 IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 660                               00006880
 DEN=1.0E0/SQRT(FINV*FINV+1.0E0)                                    00006890
 GO TO 670                                                          00006900
 DO 720 JJ=1,K                                                      00006910
 P(JJ)=BINV(JJ)*P(JJ)                                               00006920
 GO TO 650                                                          00006930
 DO 740 JJ=1,K                                                      00006940
 DEN=BINV(JJ)+SQRT(BINV(JJ)**2+1.0E0)                               00006950
 P(JJ)=0.5E0*(DEN+1.0E0/DEN)*P(JJ)                                  00006960
 GO TO 650                                                          00006970
 IF(IBKA.EQ.2) WRITE(13) (P(JJ),JJ=1,K)                             00006980
 DO 760 JJ=1,K                                                      00006990
 G(JJ)=G(JJ)+WT(I)*(Y(I)-F)*P(JJ)                                   00007000
 DO 760 II=JJ,K                                                     00007010
 A(II,JJ)=A(II,JJ)+WT(I)*P(II)*P(JJ)                                00007020
 A(JJ,II)=A(II,JJ)                                                  00007030
 WS=Y(I)-F                                                          00007040
 IF (IFSS3.LE.0) GO TO 810                                          00007050
 T ITERATION RESULTS AND DATA MATRIX FOR PRINTING                   00007060
 IF(I.GT.1) GO TO 771                                               00007070
 IF(IOUT.EQ.0) GO TO 773                                            00007080
 WRITE(16,430)                                                      00007090
 WRITE(16,2550)                                                     00007100
 IF(IPRT.LT.-1) WRITE(6,772)                                        00007110
 FORMAT(/11H -UNSCALED-/3X,1HI,4X,3HOBS,11X,3HCAL,11X,3HRES,        00007120
  8X,6HX(I,1))                                                      00007130
 IF(IPRT.LT.-1) WRITE (6,2700) I,Y(I),F,WS,PRNT(1)                  00007140
 IF(NPRNT.GT.0) GO TO 790                                           00007150
 IF (IFSS1.NE.1) GO TO 780                                          00007160
 WRITE (16,2700) I,Y(I),F,WS                                        00007170
```

```
CONTINUE                                                              00007180
GO TO 810                                                             00007190
CONTINUE                                                              00007200
IF (IFSS1.NE.1) GO TO 800                                             00007210
PERR=0.0                                                              00007220
IF(F.NE.0.0) PERR=100.0*WS/ABS(F)                                     00007230
WRITE (16,2700) I,Y(I),F,WS,PERR,(PRNT(JJ),JJ=1,NPRNT)                00007240
CONTINUE                                                              00007250
WS=Y(I)-F                                                             00007260
PHI=PHI+WT(I)*WS*WS                                                   00007270
I=I+1                                                                 00007280
IF (I.LE.N) GO TO 470                                                 00007290
C IN END OF I=1 TO N NON-DO LOOP                                      00007300
IF(IBKA.NE.2) GO TO 860                                               00007310
WT UNSCALED PARTIALS SAVED ON FILE 13 (WHEN IBKA=2)                   00007320
FORMAT(/20H -UNSCALED PARTIALS-)                                      00007330
IF(IOUT.EQ.1) WRITE(16,820)                                           00007340
REWIND 13                                                             00007350
DO 850 II=1,N                                                         00007360
READ(13) (SA(JJ),JJ=1,K)                                              00007370
FORMAT(2X,I3,5E18.8)                                                  00007380
FORMAT(2X,I3,5E18.8/(5X,5E18.8))                                      00007390
IF(IOUT.EQ.1.AND.K.NE.5) WRITE(16,840) II,(SA(JJ),JJ=1,K)              00007400
IF(IOUT.EQ.1.AND.K.EQ.5) WRITE(16,830) II,(SA(JJ),JJ=1,K)              00007410
CONTINUE                                                              00007420
REWIND 13                                                             00007430
WRITE(6,430)                                                          00007440
IF(IOUT.EQ.1) WRITE(16,430)                                           00007450
CONTINUE                                                              00007460
IF (IP.LE.0) GO TO 890                                                00007470
DO 880 JJ=1,IP                                                        00007480
IWS=IB(JJ)                                                            00007490
DO 870 II=1,K                                                         00007500
A(IWS,II)=0.E0                                                        00007510
A(II,IWS)=0.E0                                                        00007520
A(IWS,IWS)=1.E0                                                       00007530
IF(IBKA-2) 900,1770,1780                                              00007540
C          SAVE SQUARE ROOTS OF DIAGONAL ELEMENTS                     00007550
DO 910 I=1,K                                                          00007560
SA(I)=SQRT(A(I,I))                                                    00007570
DO 950 I=1,K                                                          00007580
DO 930 J=1,K                                                          00007590
WS=SA(I)*SA(J)                                                        00007600
IF (WS.GT.0.E0) GO TO 920                                             00007610
A(I,J)=0.E0                                                           00007620
GO TO 930                                                             00007630
A(I,J)=A(I,J)/WS                                                      00007640
CONTINUE                                                              00007650
IF (SA(I).GT.0.E0) GO TO 940                                          00007660
G(I)=0.E0                                                             00007670
GO TO 950                                                             00007680
G(I)=G(I)/SA(I)                                                       00007690
```

```
) CONTINUE                                                          00007700
  DO 960 I=1,K                                                      00007710
) A(I,I)=1.E0                                                       00007720
  PHIZ=PHI                                                          00007730
             WE NOW HAVE PHI ZERO                                   00007740
  IF(IBKT-1) 970,980,970                                            00007750
) WRITE (13) A                                                      00007760
  REWIND 13                                                         00007770
  GO TO 1000                                                        00007780
) DO 990 II=1,K                                                     00007790
  III=II+10                                                         00007800
  DO 990 JJ=1,K                                                     00007810
) A(III,JJ)=A(II,JJ)                                                00007820
  .....                                                             00007830
) CONTINUE                                                          00007840
  IF (ITCT.GT.0) GO TO 1030                                         00007850
             FIRST ITERATION                                        00007860
  IF (XL.GT.0.E0) GO TO 1010                                        00007870
  XL=0.01E0                                                         00007880
0 ITCT=1                                                            00007890
  DO 1020 J=1,K                                                     00007900
0 BS(J)=B(J)                                                        00007910
             BS(J) CORRESPONDS TO PHIZ                              00007920
0 IBK1=1                                                            00007930
  WS=N-K+IP                                                         00007940
  IF(N.GT.K) SE=SQRT(PHIZ/WS)                                       00007950
  IF (IFSS3.GT.0) GO TO 1040                                        00007960
  WRITE (6,2560) ITER,PHIZ,SE,XLL,GAMMA,XL                          00007970
  IF (IFSS1.NE.1) GO TO 1320                                        00007980
  WRITE (16,2560) ITER,PHIZ,SE,XLL,GAMMA,XL                         00007990
  GO TO 1320                                                        00008000
0   WRITE(6,2490) PHIZ,SE,XL                                        00008010
  IF (IFSS1.NE.1) GO TO 1320                                        00008020
  WRITE (16,2490) PHIZ,SE,XL                                        00008030
  GO TO 1320                                                        00008040
0 PHIL=PHI                                                          00008050
             WE NOW HAVE PHI(LAMBDA)                                00008060
  DO 1060 J=1,K                                                     00008070
  IF(ABS(DB(J)/(ABS(B(J))+TAU)).GE.E) GO TO 1080                    00008080
0 CONTINUE                                                          00008090
  WRITE (6,2680)                                                    00008100
  IF (IFSS1.NE.1) GO TO 1070                                        00008110
  WRITE (16,2680)                                                   00008120
'0 CONTINUE                                                         00008130
  GO TO 1670                                                        00008140
10 IF (IWS4.EQ.0) GO TO 1110                                        00008150
  IF (IWS4.EQ.1) GO TO 1090                                         00008160
  IWS4=IWS4-1                                                       00008170
  ITER=ITER+1                                                       00008180
  GO TO 1110                                                        00008190
90 WRITE (6,2690)                                                   00008200
  IF (IFSS1.NE.1) GO TO 1100                                        00008210
```

```
      WRITE (16,2690)                                                   00008220
 2690 CONTINUE                                                          00008230
      GO TO 1670                                                        00008240
 1190 XKDB=1.E0                                                         00008250
      IF (PHIL.GT.PHIZ) GO TO 1190                                      00008260
      XLS=XL                                                            00008270
      DO 1120 J=1,K                                                     00008280
      BA(J)=B(J)                                                        00008290
 1120 B(J)=BS(J)                                                        00008300
      IF (XL.GT..00000001E0) GO TO 1140                                 00008310
      DO 1130 J=1,K                                                     00008320
      B(J)=BA(J)                                                        00008330
 1130 BS(J)=B(J)                                                        00008340
      GO TO 370                                                         00008350
 1140 XL=XL/XNUFAC                                                      00008360
      IBK1=2                                                            00008370
      GO TO 1320                                                        00008380
 1150 PHL4=PHI                                                          00008390
                WE NOW HAVE PHI(LAMBDA/XNUFAC)                          00008400
      IF (PHL4.GT.PHIZ) GO TO 1170                                      00008410
      DO 1160 J=1,K                                                     00008420
 1160 BS(J)=B(J)                                                        00008430
      GO TO 370                                                         00008440
 1170 XL=XLS                                                            00008450
 1180 CONTINUE                                                          00008460
      DO 1180 J=1,K                                                     00008470
      BS(J)=BA(J)                                                       00008480
      B(J)=BA(J)                                                        00008490
      GO TO 370                                                         00008500
      IBK1=4                                                            00008510
      XLS=XL                                                            00008520
      XL=XL/XNUFAC                                                      00008530
      DO 1200 J=1,K                                                     00008540
      B(J)=BS(J)                                                        00008550
      GO TO 1320                                                        00008560
      IF (PHI.LE.PHIZ) GO TO 1260                                       00008570
      XL=XLS                                                            00008580
      IBK1=3                                                            00008590
      XL=XL*XNUFAC                                                      00008600
      DO 1240 J=1,K                                                     00008610
      B(J)=BS(J)                                                        00008620
      GO TO 1320                                                        00008630
      PHIT4=PHI                                                         00008640
                WE NOW HAVE PHI(XNUFAC*LAMBDA)                          00008650
      IF (PHIT4.GT.PHIZ) GO TO 1280                                     00008660
      DO 1270 J=1,K                                                     00008670
      BS(J)=B(J)                                                        00008680
      GO TO 370                                                         00008690
      IF (GAMMA.GE.GAMCR) GO TO 1220                                    00008700
      XKDB=XKDB/2.E0                                                    00008710
      DO 1290 J=1,K                                                     00008720
      IF(ABS(DB(J)/(ABS(B(J))+TAU)).GE.E) GO TO 1230                    00008730
```

```
 1290 CONTINUE                                                          00008740
      DO 1300 J=1,K                                                     00008750
 1300 B(J)=BS(J)                                                        00008760
      MAXITR=MAXITR-1                                                   00008770
      WRITE (6,2740)                                                    00008780
      IF (IFSS1.NE.1) GO TO 1310                                        00008790
      WRITE (16,2740)                                                   00008800
 1310 CONTINUE                                                          00008810
      GO TO 1670                                                        00008820
C                                                                       00008830
C      ..................................................               00008840
C                  SET UP FOR MATRIX INVERSION                          00008850
 1320 IF(IBKT-1) 1330,1340,1330                                         00008860
 1330 READ (13) A                                                       00008870
      REWIND 13                                                         00008880
      GO TO 1360                                                        00008890
 1340 DO 1350 II=1,K                                                    00008900
      III=II+10                                                         00008910
      DO 1350 JJ=1,K                                                    00008920
 1350 A(II,JJ)=A(III,JJ)                                                00008930
 1360 DO 1370 I=1,K                                                     00008940
 1370 A(I,I)=A(I,I)+XL                                                  00008950
C                  GET INVERSE OF A AND SOLVE FOR DB(J)S                00008960
      IBKM=1                                                            00008970
C      ..................................................               00008980
C                  THIS IS THE MATRIX INVERSION ROUTINE                 00008990
C                  K IS THE SIZE OF THE MATRIX                          00009000
 1380 IF(K.EQ.1) GO TO 1390                                             00009010
      CALL GJR (A,K,ZETA,MSING)                                         00009020
      IF(MSING-1) 1400,1400,1381                                        00009030
1381  CALL ERRMSG(20HSINGULAR MATRIX.....,4,6,16)                       00009040
C--SPECIAL CASE, K=1                                                    00009050
 1390 A(1,1)=1.0/A(1,1)                                                 00009060
 1400 IF(IBKM-1) 1410,1410,1840                                         00009070
C                  END OF MATRIX INVERSION, SOLVE FOR DB(J)             00009080
 1410 DO 1430 I=1,K                                                     00009090
      DB(I)=0.E0                                                        00009100
      DO 1420 J=1,K                                                     00009110
 1420 DB(I)=A(I,J)*G(J)+DB(I)                                           00009120
 1430 DB(I)=XKDB*DB(I)                                                  00009130
      XLL=0.E0                                                          00009140
      DTG=0.E0                                                          00009150
      GTG=0.E0                                                          00009160
      DO 1440 J=1,K                                                     00009170
      DB(J)=DB(J)/SA(J)                                                 00009180
      DTG=DTG+DB(J)*G(J)                                                00009190
      GTG=GTG+G(J)**2                                                   00009200
      B(J)=B(J)+DB(J)                                                   00009210
 1440 XLL=XLL+DB(J)*DB(J)                                               00009220
      KIP=K-IP                                                          00009230
      IF (KIP.EQ.1) GO TO 1480                                          00009240
      CGAM=DTG/SQRT(XLL*GTG)                                            00009250
```

```
      JGAM=1                                                            00009260
      IF (CGAM.GT.0.E0) GO TO 1450                                      00009270
      CGAM=ABS(CGAM)                                                    00009280
      JGAM=2                                                            00009290
 1450 GAMMA=57.2957795E0*(1.5707288E0+CGAM*(-0.2121144E0                00009300
     1+CGAM*(0.074261E0-CGAM*                                           00009310
     2.0187293E0)))*SQRT(1.0E0-CGAM)                                    00009320
      IF(JGAM-1) 1460,1490,1460                                         00009330
 1460 GAMMA=180.E0-GAMMA                                                00009340
      IF (XL.LT.1.0E0) GO TO 1490                                       00009350
      WRITE (6,2670) XL,GAMMA                                           00009360
      IF (IFSS1.NE.1) GO TO 1470                                        00009370
      WRITE (16,2670) XL,GAMMA                                          00009380
 1470 CONTINUE                                                          00009390
      GO TO 1670                                                        00009400
 1480 GAMMA=0.E0                                                        00009410
 1490 XLL=SQRT(XLL)                                                     00009420
      IBK2=1                                                            00009430
      GO TO 1540                                                        00009440
 1500 IF (IFSS3.LE.0) GO TO 1530                                        00009450
      WRITE (6,2500) (DB(J),J=1,K)                                      00009460
      IF (IFSS1.NE.1) GO TO 1510                                        00009470
      WRITE (16,2500) (DB(J),J=1,K)                                     00009480
 1510 CONTINUE                                                          00009490
      WRITE (6,2510) PHI,XL,GAMMA,XLL                                   00009500
      IF (IFSS1.NE.1) GO TO 1520                                        00009510
      WRITE (16,2510) PHI,XL,GAMMA,XLL                                  00009520
 1520 CONTINUE                                                          00009530
C--PRESET XNUFAC--(IF MODLAM=1)                                         00009540
 1530 GO TO (1570,1150,1250,1210),IBK1                                  00009550
C                                                                       00009560
C     .................................................................00009570
C                  CALCULATE PHI                                        00009580
 1540 I=1                                                               00009590
      DO 1550 JJ=1,K                                                    00009600
 1550 CALL UNSCAL(B(JJ),BINV(JJ),SCALEP)                                00009610
      PHI=0.E0                                                          00009620
      IWHER=2                                                           00009630
      GO TO 30                                                          00009640
 1560 PHI=PHI+WT(I)*(Y(I)-F)**2                                         00009650
      I=I+1                                                             00009660
      IF (I.LE.N) GO TO 30                                              00009670
      GO TO (1500,2290,1770,2200,2220,2240),IBK2                        00009680
C=================================================                      00009690
C--DETERMINE AN EFFECTIVE MARQUARDT LAMBDA FACTOR (XNUFAC)              00009700
C  BASED ON HISTORY OF SUM OF SQUARES STORED IN LATEST SS(4)--          00009710
 1570 IF(MODLAM.EQ.0) GO TO 1050                                        00009720
      SS(ISS)=PHI                                                       00009730
      INUO=INU                                                          00009740
      GO TO (1590,1580,1600,1610),ISS                                   00009750
C--MACHINE FAILURE IF ISS.GT.4 OR ISS.LT.1                              00009760
C--   STOP 4                                                            00009770
```

```
 1580 IS1=0                                                              00009780
      IF(SS(2).GT.SS(1)) IS1=1                                           00009790
 1590 ISS=ISS+1                                                          00009800
      GO TO 1660                                                         00009810
 1600 IS2=0                                                              00009820
      IF(SS(3).GT.SS(2)) IS2=1                                           00009830
      IF(IS1.EQ.IS2) GO TO 1590                                          00009840
      INU=INU0-1                                                         00009850
      GO TO 1590                                                         00009860
 1610 IS3=0                                                              00009870
      IF(SS(4).GT.SS(3)) IS3=1                                           00009880
      IF(IS1.EQ.IS2.AND.IS3.EQ.IS2) GO TO 1620                           00009890
      IF(IS1.EQ.0.AND.IS2.EQ.0.AND.IS3.EQ.1) GO TO 1640                  00009900
      IF(IS1.EQ.1.AND.IS2.EQ.0.AND.IS3.EQ.1) GO TO 1640                  00009910
      IF(IS1.EQ.1.AND.IS2.EQ.1.AND.IS3.EQ.0) GO TO 1640                  00009920
      GO TO 1650                                                         00009930
 1620 IF(IS1.EQ.0) GO TO 1630                                            00009940
      IF(INU0.GE.3) GO TO 1650                                           00009950
      INU=3                                                              00009960
      GO TO 1650                                                         00009970
 1630 IF(INU0.GE.5) GO TO 1650                                           00009980
      INU=INU0+1                                                         00009990
      GO TO 1650                                                         00010000
 1640 IF(INU0.LE.1) GO TO 1650                                           00010010
      INU=INU0-1                                                         00010020
 1650 IS1=IS2                                                            00010030
      IS2=IS3                                                            00010040
      SS(3)=SS(4)                                                        00010050
 1660 XNUFAC=XNU(INU)                                                    00010060
      GO TO 1050                                                         00010070
C                                                                        00010080
C                                                                        00010090
C     ..................................................................00010100
C                  THIS IS THE CONFIDENCE LIMIT CALCULATION              00010110
 1670 ITR=MAXITR-IWS4+1                                                  00010120
      WRITE(6,1680) ITR                                                  00010130
 1680 FORMAT(1X,I4,11H ITERATIONS)                                       00010140
      IF(IFSS1.EQ.1) WRITE(16,1680) ITR                                  00010150
      DO 1690 J=1,K                                                      00010160
      CALL UNSCAL(BS(J),BINV(J),SCALEP)                                  00010170
      BS(J)=BINV(J)                                                      00010180
 1690 B(J)=BS(J)                                                         00010190
      WRITE(6,2520) TITLE                                                00010200
      IF (IFSS1.NE.1) GO TO 1700                                         00010210
      WRITE(16,2520) TITLE                                               00010220
 1700 CONTINUE                                                           00010230
      IBKA=2                                                             00010240
C--UNSCALE BOTH PARAMETER AND OBSERVATION SPACES PRIOR                   00010250
C  TO FINAL STATISTICS ON LAST INTERATION--AND WHERE                     00010260
C  IBKA=2, IFSS3=0..                                                     00010270
C               THIS WILL PRINT OBS,CAL,RES,ETC.                         00010280
C               AND SAVE UNSCALED PARTIALS ON FILE FILE13..              00010290
```

```
      IF(IPRT.GE.0) WRITE(6,1710) (BINV(J),J=1,K)                        00010300
 1710 FORMAT(/28H -FINAL UNSCALED PARAMETERS-/(12X,4E17.8))              00010310
      IF(IFSS1.EQ.1.AND.IPRT.GE.0) WRITE(16,1710) (BINV(J),J=1,K)        00010320
      IF(SCALEY.EQ.0) GO TO 1760                                         00010330
      DO 1750 I=1,N                                                      00010340
      IF(SCALEY.NE.1) GO TO 1730                                         00010350
 1720 Y(I)=EXP(Y(I))                                                     00010360
      GO TO 1750                                                         00010370
 1730 IF(IPRNT.LE.1) GO TO 1740                                          00010380
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 1720                              00010390
 1740 Y(I)=SINH(Y(I))                                                    00010400
 1750 CONTINUE                                                           00010410
 1760 LSCALP=SCALEP                                                      00010420
      LSCALY=SCALEY                                                      00010430
      SCALEP=0                                                           00010440
      SCALEY=0                                                           00010450
      GO TO 370                                                          00010460
 1770 CONTINUE                                                           00010470
 1780 WS=N-K+IP                                                          00010480
      IF(N.GT.K) SE=SQRT(PHI/WS)                                         00010490
      PHIZ=PHI                                                           00010500
      WRITE (6,2490) PHIZ,SE,XL                                          00010510
      IF (IFSS1.NE.1) GO TO 1790                                         00010520
      WRITE (16,2490) PHIZ,SE,XL                                         00010530
C                                                                        00010540
C                    WE NOW HAVE MATRIX A                                00010550
 1790 IF(IBKT-1) 1800,1810,1800                                          00010560
 1800 WRITE (13) A                                                       00010570
      REWIND 13                                                          00010580
      GO TO 1830                                                         00010590
 1810 DO 1820 II=1,K                                                     00010600
      III=II+10                                                          00010610
      DO 1820 JJ=1,K                                                     00010620
 1820 A(III,JJ)=A(II,JJ)                                                 00010630
 1830 IBKM=2                                                             00010640
      GO TO 1380                                                         00010650
C                                                                        00010660
C                    WE NOW HAVE C = A INVERSE                           00010670
 1840 DO 1850 J=1,K                                                      00010680
      IF (A(J,J).LT.0.E0) GO TO 1860                                     00010690
 1850 SA(J)=SQRT(A(J,J))                                                 00010700
      GO TO 1870                                                         00010710
 1860 IBOUT=1                                                            00010720
 1870 KST=-4                                                             00010730
      IF (IFSS1.NE.1) GO TO 1880                                         00010740
      WRITE (16,2600)                                                    00010750
 1880 KST=KST+5                                                          00010760
      KEND=KST+4                                                         00010770
      IF (KEND.LT.K) GO TO 1890                                          00010780
      KEND=K                                                             00010790
 1890 DO 1910 I=1,K                                                      00010800
      IF (IFSS1.NE.1) GO TO 1900                                         00010810
```

```
      WRITE (16,2620) I,(A(I,J),J=KST,KEND)                          00010820
 1900 CONTINUE                                                       00010830
 1910 CONTINUE                                                       00010840
      IF (KEND.LT.K) GO TO 1880                                      00010850
      IF (IBOUT.EQ.0) GO TO 1920                                     00010860
      WRITE (6,2760)                                                 00010870
      IF (IFSS1.NE.1) GO TO 220                                      00010880
      WRITE (16,2760)                                                00010890
      GO TO 220                                                      00010900
 1920 DO 1940 I=1,K                                                  00010910
      DO 1940 J=1,K                                                  00010920
      WS=SA(I)*SA(J)                                                 00010930
      IF (WS.GT.0.E0) GO TO 1930                                     00010940
      A(I,J)=0.E0                                                    00010950
      GO TO 1940                                                     00010960
 1930 A(I,J)=A(I,J)/WS                                               00010970
 1940 CONTINUE                                                       00010980
      DO 1950 J=1,K                                                  00010990
 1950 A(J,J)=1.E0                                                    00011000
      IF (IFSS1.NE.1) GO TO 1960                                     00011010
      WRITE (16,2610)                                                00011020
 1960 CONTINUE                                                       00011030
      KST=-9                                                         00011040
 1970 KST=KST+10                                                     00011050
      KEND=KST+9                                                     00011060
      IF (KEND.LT.K) GO TO 1980                                      00011070
      KEND=K                                                         00011080
 1980 DO 2000 I=1,K                                                  00011090
      IF (IFSS1.NE.1) GO TO 1990                                     00011100
      WRITE (16,2750) I,(A(I,J),J=KST,KEND)                          00011110
 1990 CONTINUE                                                       00011120
 2000 CONTINUE                                                       00011130
      IF (KEND.LT.K) GO TO 1970                                      00011140
C                  GET T*SE*SQRT(C(I,I))                            00011150
      DO 2010 J=1,K                                                  00011160
 2010 SA(J)=SE*SA(J)                                                 00011170
      IF(IBKT-1) 2020,2030,2020                                      00011180
 2020 READ (13) A                                                    00011190
      REWIND 13                                                      00011200
      GO TO 2050                                                     00011210
 2030 DO 2040 II=1,K                                                 00011220
      III=II+10                                                      00011230
      DO 2040 JJ=1,K                                                 00011240
 2040 A(II,JJ)=A(III,JJ)                                             00011250
 2050 CONTINUE                                                       00011260
      WRITE (6,2640)                                                 00011270
      IF (IFSS1.NE.1) GO TO 2060                                     00011280
      WRITE (16,2630)                                                00011290
 2060 CONTINUE                                                       00011300
      WS=K-IP                                                        00011310
      DO 2120 J=1,K                                                  00011320
      IF (IP.LE.0) GO TO 2080                                        00011330
```

```
      DO 2070 I=1,IP                                                    00011340
      IF (J.EQ.IB(I)) GO TO 2100                                        00011350
 2070 CONTINUE                                                          00011360
C                                                                       00011370
C--COMPUTE STD.ERR, CONF. LIMITS, AND STD.ERR/PARM.                     00011380
C                                                                       00011390
 2080 HJTD=SQRT(WS*FF)*SA(J)                                            00011400
      STE=SA(J)                                                         00011410
      TWS=STE*T                                                         00011420
      OPL=BINV(J)-TWS                                                   00011430
      OPU=BINV(J)+TWS                                                   00011440
      SPL=BINV(J)-HJTD                                                  00011450
      SPU=BINV(J)+HJTD                                                  00011460
      HJTD=0.0                                                          00011470
      IF(BINV(J).NE.0.0) HJTD=STE/BINV(J)                               00011480
      WRITE (6,2720) J,STE,OPL,OPU,HJTD                                 00011490
      IF (IFSS1.NE.1) GO TO 2090                                        00011500
      WRITE (16,2720) J,STE,OPL,OPU,SPL,SPU,HJTD                        00011510
 2090 CONTINUE                                                          00011520
      GO TO 2120                                                        00011530
 2100 WRITE (6,2570) J                                                  00011540
      IF (IFSS1.NE.1) GO TO 2110                                        00011550
      WRITE (16,2570) J                                                 00011560
 2110 CONTINUE                                                          00011570
 2120 CONTINUE                                                          00011580
C                    NONLINEAR CONFIDENCE LIMIT                         00011590
      IF (IWS6.EQ.1.OR.N.EQ.K) GO TO 220                                00011600
      WS=K-IP                                                           00011610
      WS1=N-K+IP                                                        00011620
      PKN=WS/WS1                                                        00011630
      PC=PHIZ*(1.E0+FF*PKN)                                             00011640
      WRITE (6,2650) PC                                                 00011650
      IF (IFSS1.NE.1) GO TO 2130                                        00011660
      WRITE (16,2650) PC                                                00011670
 2130 CONTINUE                                                          00011680
      WRITE (6,2660)                                                    00011690
      IF (IFSS1.NE.1) GO TO 2140                                        00011700
      WRITE (16,2660)                                                   00011710
 2140 CONTINUE                                                          00011720
      IFSS3=1                                                           00011730
C--       NON- DO LOOP J=1,K                                            00011740
C  (SINCE CONTROL JUMPS OUT AND BACK INSIDE LOOP)                       00011750
      J=1                                                               00011760
 2150 IBKP=1                                                            00011770
      DO 2160 JJ=1,K                                                    00011780
 2160 B(JJ)=BS(JJ)                                                      00011790
      IF (IP.LE.0) GO TO 2180                                           00011800
      DO 2170 JJ=1,IP                                                   00011810
      IF (J.EQ.IB(JJ)) GO TO 2380                                       00011820
 2170 CONTINUE                                                          00011830
 2180 DD=-1.E0                                                          00011840
      IBKN=1                                                            00011850
```

```
2190 D=DD                                                              00011860
     B(J)=BS(J)+D*SA(J)                                                00011870
     IBK2=4                                                            00011880
     GO TO 1540                                                        00011890
2200 PHI1=PHI                                                          00011900
     IF (PHI1.GE.PC) GO TO 2230                                        00011910
2210 D=D+DD                                                            00011920
     IF (D/DD.GE.5.E0) GO TO 2420                                      00011930
     B(J)=BS(J)+D*SA(J)                                                00011940
     IBK2=5                                                            00011950
     GO TO 1540                                                        00011960
2220 PHID=PHI                                                          00011970
     IF (PHID.LT.PC) GO TO 2210                                        00011980
     IF (PHID.GE.PC) GO TO 2250                                        00011990
2230 D=D/2.E0                                                          00012000
     IF (D/DD.LE..001E0) GO TO 2420                                    00012010
     B(J)=BS(J)+D*SA(J)                                                00012020
     IBK2=6                                                            00012030
     GO TO 1540                                                        00012040
2240 PHID=PHI                                                          00012050
     IF (PHID.GT.PC) GO TO 2230                                        00012060
2250 XK1=PHIZ/D+PHI1/(1.E0-D)+PHID/(D*(D-1.E0))                        00012070
     XK2=-(PHIZ*(1.E0+D)/D+D/(1.E0-D)*PHI1+PHID/(D*(D-1.E0)))          00012080
     XK3=PHIZ-PC                                                       00012090
     BC=(SQRT(XK2*XK2-4.E0*XK1*XK3)-XK2)/(2.E0*XK1)                    00012100
     IF(IBKN-1) 2260,2260,2270                                         00012110
2260 B(J)=BS(J)-SA(J)*BC                                               00012120
     GO TO 2280                                                        00012130
2270 B(J)=BS(J)+SA(J)*BC                                               00012140
2280 IBK2=2                                                            00012150
     GO TO 1540                                                        00012160
2290 IF(IBKN-1) 2300,2300,2310                                         00012170
2300 IBKN=2                                                            00012180
     DD=1.E0                                                           00012190
     BL=B(J)                                                           00012200
     PL=PHI                                                            00012210
     GO TO 2190                                                        00012220
2310 BU=B(J)                                                           00012230
     PU=PHI                                                            00012240
     GO TO (2320,2340,2360,2400), IBKP                                 00012250
2320 WRITE (6,2620) J,BL,PL,BU,PU                                      00012260
     IF (IFSS1.NE.1) GO TO 2330                                        00012270
     WRITE (16,2620) J,BL,PL,BU,PU                                     00012280
2330 CONTINUE                                                          00012290
     GO TO 2470                                                        00012300
2340 WRITE (6,2590) J,BU,PU                                            00012310
     IF (IFSS1.NE.1) GO TO 2350                                        00012320
     WRITE (16,2590) J,BU,PU                                           00012330
2350 CONTINUE                                                          00012340
     GO TO 2470                                                        00012350
2360 WRITE (6,2620) J,BL,PL                                            00012360
     IF (IFSS1.NE.1) GO TO 2370                                        00012370
```

```
      WRITE (16,2620) J,BL,PL                                          00012380
 2370 CONTINUE                                                         00012390
      GO TO 2470                                                       00012400
 2380 WRITE (6,2570) J                                                 00012410
      IF (IFSS1.NE.1) GO TO 2390                                       00012420
      WRITE (16,2570) J                                                00012430
 2390 CONTINUE                                                         00012440
      GO TO 2470                                                       00012450
 2400 WRITE (6,2580) J                                                 00012460
      IF (IFSS1.NE.1) GO TO 2410                                       00012470
      WRITE (16,2580) J                                                00012480
 2410 CONTINUE                                                         00012490
      GO TO 2470                                                       00012500
 2420 IF(IBKN-1) 2430,2430,2440                                        00012510
                 DELETE LOWER PRINT                                    00012520
 2430 IBKP=2                                                           00012530
      GO TO 2290                                                       00012540
 2440 IF(IBKP-1) 2450,2450,2460                                        00012550
                 DELETE UPPER PRINT                                    00012560
 2450 IBKP=3                                                           00012570
      GO TO 2290                                                       00012580
                 LOWER IS ALREADY DELETED, SO DELETE BOTH              00012590
 2460 IBKP=4                                                           00012600
      GO TO 2290                                                       00012610
 --END OF NON- DO LOOP J=1,K                                           00012620
 2470 J=J+1                                                            00012630
      IF(J.LE.K) GO TO 2150                                            00012640
      GO TO 220                                                        00012650
      .....00012660
 2480 FORMAT(18A4)                                                     00012670
 2490   FORMAT(/13X,4H PHI,14X,4H S E,9X,7H LAMBDA/5X,2E18.8,E13.3)     00012680
 2500 FORMAT (/12H INCREMENTS ,4E17.8/(12X,4E17.8))                    00012690
 2510 FORMAT (13X,4H PHI10X,7H LAMBDA6X,7H GAMMA 6X,7H LENGTH/5X,E18.8,300012700
     1E13.3)                                                           00012710
 2520 FORMAT(16H1M A R Q R T  --,5X,16A5)                              00012720
 2530   FORMAT(/5H N = ,I4,8X,4HK = ,I3,9X,5HIP = ,I3,8X,4HM = ,I2,10X, 00012730
     1 6HGAMCR=,E9.3/5H DEL=,E10.3,2X,9HMODLAM = ,I1,6X,3HFF=,E10.3,3X, 00012740
     2 2HT=,E10.3,4X,2HE=,E10.3/5H TAU=,E10.3,2X,3HXL=,E10.3,3X,       00012750
     3 5HZETA=,E10.3,8H IALT = ,I2,7X,8HISTOP = ,I1/7H IWT = ,I1,9X,   00012760
     4 7HIDER = ,I1,8X,7HIPRT = ,I2,7X,8HNITER = ,I4,4X,7HINON = ,I1/  00012770
     5 8H IOUT = ,I2,7X,                                               00012780
     6 8HNPRNT = ,I1,7X,9HSCALEP = ,I1,6X,9HSCALEY = ,I1/)             00012790
  540 FORMAT (/12H PARAMETERS ,4E17.8/(12X,4E17.8))                    00012800
  550   FORMAT(3X,1HI,4X,8HOBS.Y(I),6X,3HCAL,11X,3HRES,8X,8H%RES.ERR,6X,00012810
     1 6HX(I,1),8X,6HX(I,2),8X,6HX(I,3),8X,6HX(I,4),8X,6HX(I,5))       00012820
  560   FORMAT(/1X,4HITER,8X,4H PHI,14X,4H S E,11X,7H LENGTH,6X,        00012830
     1 7H GAMMA ,6X,7H LAMBDA/1X,I4,2E18.8,3E13.3)                     00012840
  570 FORMAT (2X,I3,20H PARAMETER NOT USED )                           00012850
  580 FORMAT (2X,I3,12H NONE FOUND )                                   00012860
  590 FORMAT (2X,I3,36X,2E18.8)                                        00012870
  600 FORMAT (1H /13H PTP INVERSE )                                    00012880
  610 FORMAT (1H /30H PARAMETER CORRELATION MATRIX )                   00012890
```

```
 2620 FORMAT (2X,I3,5E18.8)                                              00012900
 2630   FORMAT(//4X,13HPARAMETER STD,17X,15HONE - PARAMETER,21X,         00012910
     1 14H SUPPORT PLANE/11X,6H ERROR,12X,6H LOWER,12X,6H UPPER,12X,     00012920
     2 6H LOWER,12X,6H UPPER,10X,14HSTD.ERROR/PARM)                      00012930
 2640   FORMAT(//4X,13HPARAMETER STD,17X,15HONE - PARAMETER/11X,         00012940
     1 6H ERROR,12X,6H LOWER,12X,6H UPPER,10X,14HSTD.ERROR/PARM)         00012950
 2650 FORMAT (//30H  NONLINEAR CONFIDENCE LIMITS //13H PHI CRITICAL,     00012960
     1 E15.8)                                                            00012970
 2660 FORMAT (1H /6H  PARA6X,8H LOWER B8X,10H LOWER PHI10X,8H UPPER B8X, 00012980
     110H UPPER PHI)                                                     00012990
 2670 FORMAT (/19H -GAMMA LAMBDA TEST,5X,2E13.3)                         00013000
 2680 FORMAT (/15H -EPSILON TEST )                                       00013010
 2690 FORMAT (/12H -FORCE OFF )                                          00013020
 2700 FORMAT(1X,I3,2E14.6,E11.3,6E14.6)                                  00013030
 2720 FORMAT (2X,I3,6E18.8)                                              00013040
 2730 FORMAT (1H )                                                       00013050
 2740 FORMAT (/20H -GAMMA EPSILON TEST)                                  00013060
 2750 FORMAT (3X,I5,2X,10F10.4)                                          00013070
 2760 FORMAT (/27H  NEGATIVE DIAGONAL ELEMENT)                           00013080
      END                                                                00013090

      SUBROUTINE GJR (A,N,EPS,MSING)                                     00013100
C     GAUSS-JORDAN-RUTISHAUSER MATRIX INVERSION WITH DOUBLE PIVOTING.    00013110
      DIMENSION A(20,20),B(20),C(20),P(20),Q(20)                         00013120
      INTEGER P,Q                                                        00013130
      MSING=1                                                            00013140
      DO 140 K=1,N                                                       00013150
C     DETERMINATION OF THE PIVOT ELEMENT                                 00013160
      PIVOT=0.E0                                                         00013170
      DO 20 I=K,N                                                        00013180
      DO 20 J=K,N                                                        00013190
      IF(ABS(A(I,J))-ABS(PIVOT)) 20,20,10                                00013200
   10 PIVOT=A(I,J)                                                       00013210
      P(K)=I                                                             00013220
      Q(K)=J                                                             00013230
   20 CONTINUE                                                           00013240
      IF(ABS(PIVOT)-EPS) 220,220,30                                      00013250
C     EXCHANGE OF THE PIVOTAL ROW WITH THE KTH ROW                       00013260
   30 IF (P(K)-K) 40,60,40                                               00013270
   40 DO 50 J=1,N                                                        00013280
      L=P(K)                                                             00013290
      Z=A(L,J)                                                           00013300
      A(L,J)=A(K,J)                                                      00013310
   50 A(K,J)=Z                                                           00013320
C     EXCHANGE OF THE PIVOTAL COLUMN WITH THE KTH COLUMN                 00013330
   60 IF (Q(K)-K) 70,90,70                                               00013340
   70 DO 80 I=1,N                                                        00013350
      L=Q(K)                                                             00013360
      Z=A(I,L)                                                           00013370
      A(I,L)=A(I,K)                                                      00013380
   80 A(I,K)=Z                                                           00013390
   90 CONTINUE                                                           00013400
```

```
C       JORDAN STEP                                                     00013410
        DO 130 J=1,N                                                    00013420
        IF (J-K) 110,100,110                                            00013430
  100 B(J)=1.0E0/PIVOT                                                  00013440
        C(J)=1.0E0                                                      00013450
        GO TO 120                                                       00013460
  110 B(J)=-A(K,J)/PIVOT                                                00013470
        C(J)=A(J,K)                                                     00013480
  120 A(K,J)=0.0E0                                                      00013490
  130 A(J,K)=0.0E0                                                      00013500
        DO 140 I=1,N                                                    00013510
        DO 140 J=1,N                                                    00013520
  140 A(I,J)=A(I,J)+C(I)*B(J)                                           00013530
C       REORDERING THE MATRIX                                           00013540
        DO 200 M=1,N                                                    00013550
        K=N-M+1                                                         00013560
        IF (P(K)-K) 150,170,150                                         00013570
  150 DO 160 I=1,N                                                      00013580
        L=P(K)                                                          00013590
        Z=A(I,L)                                                        00013600
        A(I,L)=A(I,K)                                                   00013610
  160 A(I,K)=Z                                                          00013620
  170 IF (Q(K)-K) 180,200,180                                           00013630
  180 DO 190 J=1,N                                                      00013640
        L=Q(K)                                                          00013650
        Z=A(L,J)                                                        00013660
        A(L,J)=A(K,J)                                                   00013670
  190 A(K,J)=Z                                                          00013680
  200 CONTINUE                                                          00013690
  210 RETURN                                                            00013700
  220 PRINT 230, P(K),Q(K),PIVOT                                        00013710
  230 FORMAT (/16H SINGULAR MATRIX3H I=I3,3H J=I3,7H PIVOT=E16.8/)      00013720
        MSING=2                                                         00013730
        GO TO 210                                                       00013740
        END                                                             00013750

        SUBROUTINE UNSCAL(BIN,BOUT,SCALEP)                              00013760
C// MODIFIED TO TRAP ERRORS >10**38 ON MULTICS                          00013770
C--UNSCALE PARMETER BIN TO BOUT VIA SCALEP                              00013780
        INTEGER SCALEP                                                  00013790
        IF(SCALEP-1) 10,20,30                                           00013800
   10 BOUT=BIN                                                          00013810
        GO TO 40                                                        00013820
   20 IF(BIN.GT.88.028) GO TO 99                                        00013830
        BOUT= EXP_(BIN)                                                 00013840
        GO TO 40                                                        00013850
   30 BOUT= SINH(BIN)                                                   00013860
   40 RETURN                                                            00013870
   99 WRITE(6,699) BIN                                                  00013880
        WRITE(16,699) BIN                                               00013890
699    FORMAT('0"UNSCAL" ARG=',E16.8,' >88.028 FOR EXP_() ON MULTICS'/  00013900
     & ' --CHECK ALL $PARMS AND DATA --IF OK, THEN--'/                   00013910
```

```
     & ' --TRY RESTARTING WITH DIFFERENT SCALING OPTION(S) --OR--'/      00013920
     & ' --RESTART WITH BETTER "GUESSED" STARTING PARAMETERS.')         00013930
      CALL CLOSE_FILE('-ALL')                                           00013940
      STOP                                                              00013950
      END                                                               00013960

      REAL FUNCTION ASINH(X)                                            00013970
C--INVERSE HYPERBOLIC SIN FUNCTION                                      00013980
C                                                                       00013990
      REAL*8 X2                                                         00014000
      X2=X                                                              00014010
      ASINH=DLOG(X2+DSQRT(X2*X2+1.0D0))                                 00014020
      RETURN                                                            00014030
      END                                                               00014040

      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                   00014050
C--ERROR MESSAGE WRITE ROUTINE AND STOP, WHERE--                        00014060
C                                                                       00014070
C       MSG=    ANY MULTIPLE OF 5 CHARACTERS--MAX. OF 120             00014080
C               (USE NH--------- FORM FOR ANSI COMPATIBILITY)          00014090
C       M5=     NO.CHARS IN MSG/5 (REMAINDER MUST BE 0) 1.LE.M5.LE.24 00014100
C       I6=     1ST UNIT  FOR WRITE(I6, ) MSG -- USUALLY I6=6 FOR LPT. 00014110
C               IF I6.LE.0 UNIT I6 IGNORED.                           00014120
C       I9=     2ND UNIT  FOR WRITE(I9, ) MSG --                      00014130
C               IF I9.LE.0, UNIT I9 IGNORED.                          00014140
C--MESSAGE WRITTEN IN FORM--                                           00014150
C       /ERROR--MSG HERE                                              00014160
C                                                                       00014170
      DIMENSION MSG(30)                                                 00014180
      J=5*M5                                                            00014190
      K=J/4+MOD(J,4)                                                    00014200
      IF(I6.GT.0) WRITE(I6,10) (MSG(I),I=1,K)                           00014210
   10 FORMAT(/8H ERROR--,30A4)                                          00014220
      IF(I9.GT.0) WRITE(I9,10) (MSG(I),I=1,K)                           00014230
      CALL CLOSE_FILE('-ALL')                                           00014240
C                                                                       00014250
      STOP                                                              00014260
      END                                                               00014270

      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)                         00014280
C--ONE DIMENSIONAL CUBIC SPLINE COEFFICIENT DETERMINATION.              00014290
C                                                                       00014300
C        BY  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO     00014310
C                                                                       00014320
C  PARMS--- M= NUMBER OF DATA POINTS .GT. 2                             00014330
C           H= EQUAL INTERVAL OPTION WHEN H.GT.0. (USE DUMMY X HERE),   00014340
C              UNEQUAL INTERVALS IF H=0. (X REQUIRED STORAGE)           00014350
C           X= INDEP.VAR WHEN H=0. (DIM .GE. M).                        00014360
C           Y= DEPENDENT VARIABLE  (DIM .GE. M).                        00014370
C           A,B,C=COEFF.ARRAYS (EACH DIM .GE. M)                        00014380
C                RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS OF A,B,&C.   00014390
C                ALSO USED AS WORK ARRAYS DURING EXECUTION.             00014400
```

```
C            IT= TYPE OF BOUNDARY CONDITION SUPPLIED IN D ARRAY. USE      00014410
C               IT=1 IF 1ST DERIVATIVES GIVEN AT END POINTS, OR           00014420
C               IT=0 IF 2ND DERIVATIVES GIVEN AT END POINTS.              00014430
C            D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M RESPECTIVELY.     00014440
C            P,S= WORK ARRAYS (EACH DIM=M).                               00014450
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE.                00014460
C  THE RESULTING CUBIC SPLINE IS OF THE FORM:                             00014470
C     Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3              00014480
C       FOR I=1,2,...,M-1                                                 00014490
C                                                                         00014500
C                                                                         00014510
      REAL*4  X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL                 00014520
      IF(IT.LT.0.OR.IT.GT.1.OR.H.LT.0..OR.M.LT.3) GO TO 999               00014530
      N=M-1                                                               00014540
      IF(IT.EQ.0) GO TO 20                                                00014550
C--1ST DERIVATIVE BOUNDARIES GIVEN                                        00014560
      NE=N-1                                                              00014570
      IF(H) 999,11,1                                                      00014580
C--EQUAL SPACING H .GT. 0. AND IT=1                                       00014590
    1 HH=3.0/H                                                            00014600
      DO 2 I=1,NE                                                         00014610
      B(I)=4.0                                                            00014620
      C(I)=1.0                                                            00014630
      A(I)=1.0                                                            00014640
    2 P(I)=HH*(Y(I+2)-Y(I))                                               00014650
      P(1)=P(1)-D(1)                                                      00014660
      P(NE)=P(NE)-D(2)                                                    00014670
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE                         00014680
    3 C(1)=C(1)/B(1)                                                      00014690
      P(1)=P(1)/B(1)                                                      00014700
      DO 4 I=2,NE                                                         00014710
      MUL=1.0/(B(I)-A(I)*C(I-1))                                          00014720
      C(I)=MUL*C(I)                                                       00014730
    4 P(I)=MUL*(P(I)-A(I)*P(I-1))                                         00014740
C--OBTAIN SPLINE COEFFICIENTS                                             00014750
      A(NE+IT)=P(NE)                                                      00014760
      I=NE-1                                                              00014770
    5 A(I+IT)=P(I)-C(I)*A(I+IT+1)                                         00014780
      I=I-1                                                               00014790
       IF(I.GE.1) GO TO 5                                                 00014800
      IF(IT.EQ.0) GO TO 6                                                 00014810
      A(1)=D(1)                                                           00014820
      A(M)=D(2)                                                           00014830
    6 IF(H.EQ.0.) GO TO 14                                                00014840
      HH=1.0/H                                                            00014850
      DO 7 I=1,N                                                          00014860
      MUL=HH*(Y(I+1)-Y(I))                                                00014870
      B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))                                 00014880
    7 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))                                   00014890
      RETURN                                                              00014900
C--UNEQUAL SPACING H=0.. AND IT=1                                         00014910
   11 DO 12 I=1,N                                                         00014920
```

```
   12 S(I+1)=X(I+1)-X(I)                                                00014930
      DO 13 I=1,NE                                                      00014940
      B(I)=2.0*(S(I+1)+S(I+2))                                          00014950
      C(I)=S(I+1)                                                       00014960
      A(I)=S(I+2)                                                       00014970
   13 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/     00014980
     $ (S(I+1)*S(I+2))                                                  00014990
      P(1)=P(1)-S(3)*D(1)                                               00015000
      P(NE)=P(NE)-S(N)*D(2)                                             00015010
      GO TO 3                                                           00015020
   14 DO 15 I=1,N                                                       00015030
      HH=1.0/S(I+1)                                                     00015040
      MUL=(Y(I+1)-Y(I))*HH**2                                           00015050
      B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH                                 00015060
   15 C(I)=-2.0*MUL*HH+(A(I+1)+A(I))*HH**2                              00015070
      RETURN                                                            00015080
C--2ND DERIVATIVE BOUNDARIES GIVEN                                      00015090
   20 NE=N+1                                                            00015100
      IF(H) 999,31,21                                                   00015110
C--EQUAL SPACING H .GT. 0 AND IT=0                                      00015120
   21 HH=3.0/H                                                          00015130
      DO 22 I=2,N                                                       00015140
      B(I)=4.0                                                          00015150
      C(I)=1.0                                                          00015160
      A(I)=1.0                                                          00015170
   22 P(I)=HH*(Y(I+1)-Y(I-1))                                           00015180
      B(1)=2.0                                                          00015190
      B(NE)=2.0                                                         00015200
      C(1)=1.0                                                          00015210
      C(NE)=1.0                                                         00015220
      A(NE)=1.0                                                         00015230
      P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)                                    00015240
      P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)                                   00015250
      GO TO 3                                                           00015260
C--UNEQUAL SPACING H=0 AND IT=0                                         00015270
   31 DO 32 I=1,N                                                       00015280
   32 S(I+1)=X(I+1)-X(I)                                                00015290
      N1=N-1                                                            00015300
      DO 33 I=1,N1                                                      00015310
      B(I+1)=2.0*(S(I+1)+S(I+2))                                        00015320
      C(I+1)=S(I+1)                                                     00015330
      A(I+1)=S(I+2)                                                     00015340
   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)))/   00015350
     *      (S(I+1)*S(I+2))                                            00015360
      B(1)=2.0                                                          00015370
      B(NE)=2.0                                                         00015380
      C(1)=1.0                                                          00015390
      C(NE)=1.0                                                         00015400
      A(NE)=1.0                                                         00015410
      P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)                           00015420
      P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)                          00015430
      GO TO 3                                                           00015440
```

```
  999 M=-IABS(M)                                                        00015450
      RETURN                                                            00015460
      END                                                               00015470

      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)                              00015480
C--GIVEN CUBIC SPLINE COEFF'S A,B,C,AND M OBS.DATA ARRAYS X,Y           00015490
C  SPOINT EVALUATES THE PIECEWISE CUBIC SPLINE ORDINATE YY AT THE       00015500
C  ABSCISSA XX, WHERE XX IS IN THE CLOSED INTERVAL (X(1),X(M)).         00015510
C  NOTE: IF COMPUTING OVER EQUAL INTERVALS, USE THE SUBR 'CUBIC'        00015520
C  WHICH REQUIRES ONLY ONE CALL.                                        00015530
C                                                                       00015540
      DIMENSION X(1),Y(1),A(1),B(1),C(1)                                00015550
      IF(XX.LT.X(1).OR.XX.GT.X(M)) GO TO 9                              00015560
      M1=M-1                                                            00015570
      DO 1 I=1,M1                                                       00015580
      J=I                                                               00015590
      IF(XX.LE.X(I+1)) GO TO 2                                          00015600
    1 CONTINUE                                                          00015610
    9 WRITE(6,60) XX,X(1),X(M)                                          00015620
   60 FORMAT('0ERROR IN SPOINT CALL--XX=',E16.8,' NOT IN CLOSED INTERVAL00015630
     * (',E16.8,',',E16.8,')')                                          00015640
      RETURN                                                            00015650
    2 Z=XX-X(J)                                                         00015660
      YY=Y(J)+((C(J)*Z+B(J))*Z+A(J))*Z                                  00015670
      RETURN                                                            00015680
      END                                                               00015690

        REAL FUNCTION RLAGH1(X,FUN,TOL,L,NEW)                           00015700
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE               00015710
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*J1(G*B)*DG' DEFINED AS THE    00015720
C  REAL HANKEL TRANSFORM OF ORDER 1 AND ARGUMENT X(=ALOG(B))            00015730
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND               00015740
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....       00015750
C                                                                       00015760
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.                    00015770
C                                                                       00015780
C--PARAMETERS:                                                          00015790
C                                                                       00015800
C     * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE HANKEL TRANSFORM 00015810
C                'RLAGH1' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,  00015820
C                SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT, 00015830
C                THEN SUBPROGRAM 'RHANK1' IS ADVISED FOR GENERAL USE.   00015840
C                (ALSO SEE PARM 'NEW' & NOTES (2)-(3) BELOW).           00015850
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).     00015860
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN   00015870
C                 CALLING PROGRAM AND IN SUBPROGRAM FUN.                00015880
C                 THE REAL FUNCTION FUN SHOULD BE A MONOTONE            00015890
C                 DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00015900
C       TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,     00015910
C                 IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00015920
C                 THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,      00015930
C                 TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON       00015940
```

```
C                 THE FUNCTION FUN AND PARAMETER X...IN GENERAL,          00015950
C               A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'    00015960
C               BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY  00015970
C               RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN   00015980
C               APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,     00015990
C               ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...    00016000
C       L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE          00016010
C                 CONVOLUTION (L DEPENDS ON TOL AND FUN).                 00016020
C               MIN.L=15 AND MAX.L=236--WHICH COULD                       00016030
C                 OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING    00016040
C                 VERY FAST...                                            00016050
C     * NEW=    1 IS NECESSARY 1ST TIME OR BRAND NEW X.                   00016060
C               0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20          00016070
C                 IS ASSUMED INTERNALLY BY THIS ROUTINE.                  00016080
C                 NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL                00016090
C                 STILL ASSUME X=(LAST X)-0.20 ANYWAY...                  00016100
C                 IT IS THE USERS RESPONSIBILITY TO NORMALIZE              00016110
C                 BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).  00016120
C               THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT         00016130
C                 TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A              00016140
C                 SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING   00016150
C                 ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...           00016160
C                 THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED              00016170
C                 KERNELS WILL BE USED IN THE LAGGED CONVOLUTION          00016180
C                 WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS      00016190
C                 WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)              00016200
C                                                                         00016210
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGH1; THE HANKEL      00016220
C  TRANSFORM IS THEN RLAGH1/B WHICH IS TO BE COMPUTED AFTER EXIT FROM     00016230
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...            00016240
C                                                                         00016250
C--USAGE-- 'RLAGH1' IS CALLED AS FOLLOWS:                                 00016260
C     ...                                                                 00016270
C     EXTERNAL RF                                                         00016280
C     ...                                                                 00016290
C     R=RLAGH1(ALOG(B),RF,TOL,L,NEW)/B                                    00016300
C     ...                                                                 00016310
C     END                                                                 00016320
C     REAL FUNCTION RF(G)                                                 00016330
C     ...USER SUPPLIED CODE...                                            00016340
C     END                                                                 00016350
C                                                                         00016360
C--NOTES:                                                                 00016370
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM       00016380
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS         00016390
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....                                00016400
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION     00016410
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN             00016420
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,     00016430
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'       00016440
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING    00016450
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).       00016460
```

```
C      FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                    00016470
C                 ...                                                      00016480
C                 NB=AINT(5.*ALOG(BMAX/BMIN))+1                            00016490
C                 NB1=NB+1                                                 00016500
C                 X0=ALOG(BMAX)+.2                                         00016510
C                 NEW=1                                                    00016520
C                 DO 1 J=1,NB                                              00016530
C                 I=NB1-J                                                  00016540
C                 X=X0-.2*J                                                00016550
C                 ARG(I)=EXP(X)                                            00016560
C                 ANS(I)=RLAGH1(X,RF,TOL,L,NEW)/ARG(I)                     00016570
C      1          NEW=0                                                    00016580
C                 ...                                                      00016590
C      (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR      00016600
C      ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,    00016610
C      TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)            00016620
C      SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.                 00016630
C      (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY            00016640
C      ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW                00016650
C      BMAX,BMIN AND BY SETTING NEW=1....                                  00016660
C      (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE  00016670
C                                                                          00016680
C                                                                          00016690
        DIMENSION KEY(236),SAVE(236)                                       00016700
        DIMENSION WT(236),W1(76),W2(76),W3(76),W4(8)                       00016710
        EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)),          00016720
     1 (WT(229),W4(1))                                                     00016730
C--J1-EXTENDED FILTER WEIGHT ARRAYS:                                       00016740
       DATA W1/                                                            00016750
     1-8.8863805E-10, 1.1293811E-09,-1.2050872E-09, 1.2696232E-09,         00016760
     2-1.3223909E-09, 1.3642393E-09,-1.3969439E-09, 1.4225941E-09,         00016770
     3-1.4427475E-09, 1.4580582E-09,-1.4682563E-09, 1.4732179E-09,         00016780
     4-1.4735606E-09, 1.4719870E-09,-1.4727091E-09, 1.4828225E-09,         00016790
     5-1.5102619E-09, 1.5667752E-09,-1.6634522E-09, 1.8172900E-09,         00016800
     6-2.0412753E-09, 2.3595230E-09,-2.7861077E-09, 3.3592871E-09,         00016810
     7-4.0940172E-09, 5.0571015E-09,-6.2604109E-09, 7.8269461E-09,         00016820
     8-9.7514701E-09, 1.2267639E-08,-1.5312389E-08, 1.9339924E-08,         00016830
     9-2.4126297E-08, 3.0576829E-08,-3.8060204E-08, 4.8423732E-08,         00016840
     1-6.0051116E-08, 7.6787475E-08,-9.4700993E-08, 1.2192844E-07,         00016850
     2-1.4918997E-07, 1.9392737E-07,-2.3464786E-07, 3.0911127E-07,         00016860
     3-3.6815394E-07, 4.9413800E-07,-5.7554168E-07, 7.9301529E-07,         00016870
     4-8.9502818E-07, 1.2794292E-06,-1.3811469E-06, 2.0789668E-06,         00016880
     5-2.1069398E-06, 3.4103188E-06,-3.1584463E-06, 5.6639045E-06,         00016890
     6-4.6059955E-06, 9.5561672E-06,-6.4142855E-06, 1.6440205E-05,         00016900
     7-8.2010619E-06, 2.8945217E-05,-8.6348466E-06, 5.2317398E-05,         00016910
     8-3.9915035E-06, 9.7273612E-05, 1.5220520E-05, 1.8614373E-04,         00016920
     9 7.2023760E-05, 3.6620099E-04, 2.2062958E-04, 7.3874539E-04,         00016930
     1 5.8623480E-04, 1.5226779E-03, 1.4538718E-03, 3.1930365E-03/         00016940
       DATA W2/                                                            00016950
     1 3.4640868E-03, 6.7790882E-03, 8.0328420E-03, 1.4484339E-02,         00016960
     2 1.8201316E-02, 3.0866143E-02, 4.0106549E-02, 6.4527872E-02,         00016970
     3 8.4285526E-02, 1.2773175E-01, 1.6020907E-01, 2.1948043E-01,         00016980
```

```
     4 2.3636305E-01, 2.4895051E-01, 1.2586300E-01,-5.1060445E-02,      00016990
     5-3.4376222E-01,-2.9042175E-01, 1.1564736E-01, 4.9253231E-01,      00017000
     6-4.6748595E-01, 1.5280945E-01, 3.3348541E-02,-8.2485252E-02,      00017010
     7 7.9740630E-02,-6.6934498E-02, 5.5150465E-02,-4.5868721E-02,      00017020
     8 3.8651958E-02,-3.2935834E-02, 2.8303994E-02,-2.4475127E-02,      00017030
     9 2.1259541E-02,-1.8526278E-02, 1.6182037E-02,-1.4158101E-02,      00017040
     1 1.2402225E-02,-1.0873526E-02, 9.5392016E-03,-8.3723743E-03,      00017050
     2 7.3506490E-03,-6.4551136E-03, 5.6696335E-03,-4.9803353E-03,      00017060
     3 4.3752213E-03,-3.8438703E-03, 3.3772023E-03,-2.9672872E-03,      00017070
     4 2.6071877E-03,-2.2908274E-03, 2.0128794E-03,-1.7686706E-03,      00017080
     5 1.5540998E-03,-1.3655666E-03, 1.1999089E-03,-1.0543497E-03,      00017090
     6 9.2644973E-04,-8.1406593E-04, 7.1531559E-04,-6.2854459E-04,      00017100
     7 5.5229955E-04,-4.8530352E-04, 4.2643446E-04,-3.7470650E-04,      00017110
     8 3.2925334E-04,-2.8931382E-04, 2.5421910E-04,-2.2338147E-04,      00017120
     9 1.9628455E-04,-1.7247455E-04, 1.5155278E-04,-1.3316889E-04,      00017130
     1 1.1701502E-04,-1.0282066E-04, 9.0348135E-05,-7.9388568E-05/      00017140
      DATA W3/                                                         00017150
     1 6.9758436E-05,-6.1296474E-05, 5.3860978E-05,-4.7327436E-05,      00017160
     2 4.1586435E-05,-3.6541840E-05, 3.2109174E-05,-2.8214208E-05,      00017170
     3 2.4791718E-05,-2.1784390E-05, 1.9141864E-05,-1.6819888E-05,      00017180
     4 1.4779578E-05,-1.2986765E-05, 1.1411426E-05,-1.0027182E-05,      00017190
     5 8.8108499E-06,-7.7420630E-06, 6.8029235E-06,-5.9777053E-06,      00017200
     6 5.2525892E-06,-4.6154325E-06, 4.0555653E-06,-3.5636118E-06,      00017210
     7 3.1313335E-06,-2.7514911E-06, 2.4177236E-06,-2.1244417E-06,      00017220
     8 1.8667342E-06,-1.6402859E-06, 1.4413051E-06,-1.2664597E-06,      00017230
     9 1.1128220E-06,-9.7781908E-07, 8.5919028E-07,-7.5494920E-07,      00017240
     1 6.6335060E-07,-5.8286113E-07, 5.1213358E-07,-4.4998431E-07,      00017250
     2 3.9537334E-07,-3.4738689E-07, 3.0522189E-07,-2.6817250E-07,      00017260
     3 2.3561831E-07,-2.0701397E-07, 1.8188012E-07,-1.5979545E-07,      00017270
     4 1.4038968E-07,-1.2333746E-07, 1.0835294E-07,-9.5185048E-08,      00017280
     5 8.3613184E-08,-7.3443411E-08, 6.4505118E-08,-5.6648167E-08,      00017290
     6 4.9740428E-08,-4.3665572E-08, 3.8321109E-08,-3.3616717E-08,      00017300
     7 2.9472836E-08,-2.5819439E-08, 2.2594957E-08,-1.9745353E-08,      00017310
     8 1.7223359E-08,-1.4987869E-08, 1.3003472E-08,-1.1240058E-08,      00017320
     9 9.6723739E-09,-8.2794392E-09, 7.0438407E-09,-5.9509676E-09,      00017330
     1 4.9882405E-09,-4.1443813E-09, 3.4088114E-09,-2.7712762E-09/      00017340
      DATA W4/                                                         00017350
     1 2.2217311E-09,-1.7504755E-09, 1.3485207E-09,-1.0080937E-09,      00017360
     2 7.2300885E-10,-4.8860666E-10, 3.0121413E-10,-9.1649798E-11/      00017370
C--$$ENDATA                                                           00017380
C                                                                     00017390
      IF(NEW) 10,30,10                                                00017400
10    LAG=-1                                                          00017410
      X0=-X-17.0                                                      00017420
      DO 20 IR=1,236                                                  00017430
20    KEY(IR)=0                                                       00017440
30    LAG=LAG+1                                                       00017450
      RLAGH1=0.0                                                      00017460
      CMAX=0.0                                                        00017470
      L=0                                                             00017480
      ASSIGN 110 TO M                                                 00017490
      I=86                                                            00017500
```

```
      GO TO 200                                                          00017510
110   CMAX=AMAX1(ABS(C),CMAX)                                            00017520
      I=I+1                                                              00017530
      IF(I.LE.98) GO TO 200                                              00017540
      IF(CMAX.EQ.0.0) GO TO 150                                          00017550
      CMAX=TOL*CMAX                                                      00017560
      ASSIGN 120 TO M                                                    00017570
      I=85                                                               00017580
      GO TO 200                                                          00017590
120   IF(ABS(C).LE.CMAX) GO TO 130                                       00017600
      I=I-1                                                              00017610
      IF(I.GT.0) GO TO 200                                               00017620
130   ASSIGN 140 TO M                                                    00017630
      I=99                                                               00017640
      GO TO 200                                                          00017650
140   IF(ABS(C).LE.CMAX) GO TO 190                                       00017660
      I=I+1                                                              00017670
      IF(I.LE.236) GO TO 200                                             00017680
      GO TO 190                                                          00017690
150   ASSIGN 160 TO M                                                    00017700
      I=1                                                                00017710
      GO TO 200                                                          00017720
160   IF(C.EQ.0.0) GO TO 170                                             00017730
      I=I+1                                                              00017740
      IF(I.LE.85) GO TO 200                                              00017750
170   ASSIGN 180 TO M                                                    00017760
      I=236                                                              00017770
      GO TO 200                                                          00017780
180   IF(C.EQ.0.0) GO TO 190                                             00017790
      I=I-1                                                              00017800
      IF(I.GE.99) GO TO 200                                              00017810
190   RETURN                                                             00017820
C---STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)              00017830
200   LOOK=I+LAG                                                         00017840
      IQ=LOOK/237                                                        00017850
      IR=MOD(LOOK,237)                                                   00017860
      IF(IR.EQ.0) IR=1                                                   00017870
      IROLL=IQ*236                                                       00017880
      IF(KEY(IR).LE.IROLL) GO TO 220                                     00017890
210   C=SAVE(IR)*WT(I)                                                   00017900
      RLAGH1=RLAGH1+C                                                    00017910
      L=L+1                                                              00017920
      GO TO M,(110,120,140,160,180)                                      00017930
220   KEY(IR)=IROLL+IR                                                   00017940
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                              00017950
      GO TO 210                                                          00017960
      END                                                                00017970
```

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	(delete unless supported on system)
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. MARQDCLAG_FCODE, MARQDCLAG_PCODE, etc.) should be renamed to 6 or less characters for most other systems.

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

```
test100
$parms n=19,k=5,m=1,sp=1,sy=1,iprt=-1,e=.01,
 b=100,2,100,20,90$
(2e16.8)
$init mm=3,eps=.1e-5$
```

file10

```
  0.99216102e+02  0.10000000e+02
  0.97641480e+02  0.14677993e+02
  0.93269240e+02  0.21544347e+02
  0.82618054e+02  0.31622776e+02
  0.62050463e+02  0.46415887e+02
  0.34484983e+02  0.68129204e+02
  0.12309198e+02  0.99999994e+02
  0.33087195e+01  0.14677992e+03
  0.22332367e+01  0.21544345e+03
  0.30596437e+01  0.31622774e+03
  0.44167510e+01  0.46415884e+03
  0.63473979e+01  0.68129199e+03
  0.90466573e+01  0.99999988e+03
  0.12748527e+02  0.14677991e+04
  0.17695841e+02  0.21544344e+04
  0.24085289e+02  0.31622772e+04
  0.31978206e+02  0.46415881e+04
  0.41190794e+02  0.68129195e+04
  0.51210185e+02  0.99999982e+04
```

```
m a r q d c l a g --      test100

mm=  3
eps=  0.10000000e-05


parameter order--

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

```
m a r q r t  --     test100

n =   19        k =   5         ip =   0        n =  1          gamcr=0.450e+02
del= 0.100e-04  modlam = 1      ff= 0.400e+01   t= 0.200e+01    e= 0.100e-01
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 = 1

fmt=(2e16.8)

parameters     0.10000000e+03   0.20000000e+01   0.10000000e+03   0.20000000e+02
               0.90000001e+02

iter          phi                s e              length        gamma         lambda
   1    0.72233043e+01    0.71829681e+00    0.000e+00    0.000e+00    0.100e-01

parameters     0.93147235e+02   0.35045735e+01   0.73368705e+02   0.28684894e+02
               0.32705987e+03

iter          phi                s e              length        gamma         lambda
   2    0.74758029e+00    0.23108135e+00    0.307e+01    0.744e+02    0.100e-01

parameters     0.10079156e+03   0.18226764e+01   0.82807919e+02   0.29685390e+02
               0.17471028e+03

iter          phi                s e              length        gamma         lambda
   3    0.71294581e-01    0.71361545e-01    0.918e+00    0.648e+02    0.100e-02

parameters     0.10025946e+03   0.12658436e+01   0.89015429e+02   0.29826803e+02
               0.12617731e+03

iter          phi                s e              length        gamma         lambda
   4    0.29696215e-02    0.14564197e-01    0.494e+00    0.805e+02    0.100e-03

parameters     0.10003814e+03   0.10634684e+01   0.90078474e+02   0.29966763e+02
               0.10641991e+03

iter          phi                s e              length        gamma         lambda
   5    0.11215077e-03    0.28303302e-02    0.244e+00    0.847e+02    0.100e-05

parameters     0.10000338e+03   0.10104126e+01   0.90182087e+02   0.29994992e+02
               0.10116146e+03

iter          phi                s e              length        gamma         lambda
   6    0.15531955e-05    0.33308037e-03    0.720e-01    0.866e+02    0.100e-07

parameters     0.10000155e+03   0.10064672e+01   0.90187436e+02   0.29996680e+02
               0.10076925e+03

iter          phi                s e              length        gamma         lambda
   7    0.10162922e-05    0.26942958e-03    0.551e-02    0.870e+02    0.100e-07

-epsilon test
   7 iterations
```

```
m a r q d c l a g  --      test100

parameters      0.10000155e+03    0.10064672e+01    0.90187436e+02    0.29996680e+02
                0.10076925e+03

-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   0.992161e+02   0.992175e+02  -0.138e-02  -0.139373e-02   0.100000e+02   0.000000e+00
   2   0.976415e+02   0.976425e+02  -0.101e-02  -0.103628e-02   0.146780e+02   0.000000e+00
   3   0.932692e+02   0.932694e+02  -0.149e-03  -0.159509e-03   0.215443e+02   0.000000e+00
   4   0.826181e+02   0.826170e+02   0.108e-02   0.131132e-02   0.316228e+02   0.000000e+00
   5   0.620505e+02   0.620486e+02   0.189e-02   0.305398e-02   0.464159e+02   0.000000e+00
   6   0.344850e+02   0.344833e+02   0.169e-02   0.489099e-02   0.681292e+02   0.000000e+00
   7   0.123092e+02   0.123107e+02  -0.148e-02  -0.120500e-01   0.100000e+03   0.000000e+00
   8   0.330872e+01   0.330845e+01   0.271e-03   0.819633e-02   0.146780e+03   0.000000e+00
   9   0.223324e+01   0.223330e+01  -0.658e-04  -0.294780e-02   0.215443e+03   0.000000e+00
  10   0.305964e+01   0.306119e+01  -0.154e-02  -0.503688e-01   0.316228e+03   0.000000e+00
  11   0.441675e+01   0.441845e+01  -0.170e-02  -0.383870e-01   0.464159e+03   0.000000e+00
  12   0.634740e+01   0.634788e+01  -0.482e-03  -0.759157e-02   0.681292e+03   0.000000e+00
  13   0.904666e+01   0.904499e+01   0.166e-02   0.183948e-01   0.100000e+04   0.000000e+00
  14   0.127485e+02   0.127440e+02   0.449e-02   0.352651e-01   0.146780e+04   0.000000e+00
  15   0.176958e+02   0.176893e+02   0.652e-02   0.368613e-01   0.215443e+04   0.000000e+00
  16   0.240853e+02   0.240774e+02   0.788e-02   0.327316e-01   0.316228e+04   0.000000e+00
  17   0.319782e+02   0.319726e+02   0.561e-02   0.175425e-01   0.464159e+04   0.000000e+00
  18   0.411908e+02   0.411936e+02  -0.281e-02  -0.682608e-02   0.681292e+04   0.000000e+00
  19   0.512102e+02   0.512298e+02  -0.196e-01  -0.383147e-01   0.100000e+05   0.000000e+00

-unscaled partials-
   1    0.99201328e+00    0.14514259e-01    0.80209570e-07    0.76149634e-01   -0.62947668e-05
   2    0.97597121e+00    0.43554298e-01    0.25310942e-06    0.22159164e+00   -0.19819414e-04
   3    0.93143482e+00    0.12359397e+00    0.79675114e-06    0.58998469e+00   -0.62089696e-04
   4    0.82298020e+00    0.31540736e+00    0.24950508e-05    0.13253906e+01   -0.19244692e-03
   5    0.61370108e+00    0.67246267e+00    0.77285716e-05    0.22184523e+01   -0.58323224e-03
   6    0.33359555e+00    0.11139075e+01    0.23413947e-04    0.23714793e+01   -0.16879863e-02
   7    0.10867666e+00    0.14274816e+01    0.67983501e-04    0.13703727e+01   -0.44685468e-02
   8    0.16372229e-01    0.16439952e+01    0.18373455e-03    0.34773475e+00   -0.10147347e-01
   9    0.85482280e-03    0.20935950e+01    0.45107928e-03    0.28731693e-01   -0.18731277e-01
  10    0.14185010e-03    0.29366455e+01    0.10128122e-02    0.42434211e-03   -0.28973182e-01
  11    0.18920638e-03    0.41784766e+01    0.21511975e-02   -0.35156263e-03   -0.41645737e-01
  12    0.24552271e-03    0.58859953e+01    0.44269344e-02   -0.53138840e-03   -0.58711728e-01
  13    0.31248072e-03    0.81597120e+01    0.88845959e-02   -0.76162669e-03   -0.81423244e-01
  14    0.39203002e-03    0.11066569e+02    0.17372031e-01   -0.10560938e-02   -0.11045757e+00
  15    0.49133023e-03    0.14573977e+02    0.32953740e-01   -0.14120164e-02   -0.14549053e+00
  16    0.59122530e-03    0.18463225e+02    0.60271330e-01   -0.18080459e-02   -0.18433787e+00
  17    0.68913474e-03    0.22246770e+02    0.10548095e+00   -0.21937903e-02   -0.22213074e+00
  18    0.76576773e-03    0.25157980e+02    0.17515063e+00   -0.24929854e-02   -0.25121262e+00
  19    0.79945411e-03    0.26307420e+02    0.27356955e+00   -0.26158218e-02   -0.26270031e+00

-unscaled-

              phi               s e             lambda
        0.57056858e-03    0.63839564e-02    0.100e-07

ptp inverse
   1    0.40273971e+00    0.27149084e+00    0.60330760e-01   -0.22023259e+00    0.27239444e+02
   2    0.27149084e+00    0.18283031e+01    0.55475607e+00   -0.77937145e+00    0.18353926e+03
   3    0.60332763e-01    0.55476956e+00    0.38373134e+02   -0.20593226e+00    0.78772529e+02
   4   -0.22023260e+00   -0.77937149e+00   -0.20592653e+00    0.43038671e+00   -0.78217005e+02
   5    0.27239446e+02    0.18353927e+03    0.78771180e+02   -0.78217008e+02    0.18442858e+05
```

parameter correlation matrix

1	1.0000	0.3164	0.0153	-0.5290	0.3161
2	0.3164	1.0000	0.0662	-0.8786	0.9995
3	0.0153	0.0662	1.0000	-0.0507	0.0936
4	-0.5290	-0.8786	-0.0507	1.0000	-0.8779
5	0.3161	0.9995	0.0936	-0.8779	1.0000

	parameter std error	one - parameter lower	one - parameter upper	support plane lower	support plane upper	std.error/para
1	0.40513721e-02	0.99993446e+02	0.10000965e+03	0.99983431e+02	0.10001967e+03	0.40513093e-04
2	0.86320512e-02	0.98920310e+00	0.10237313e+01	0.96786350e+00	0.10450709e+01	0.85765845e-02
3	0.39546089e-01	0.90108343e+02	0.90266528e+02	0.90010580e+02	0.90364291e+02	0.43848779e-03
4	0.41881221e-02	0.29988304e+02	0.30005056e+02	0.29977950e+02	0.30015410e+02	0.13961952e-03
5	0.86696991e+00	0.99035311e+02	0.10250319e+03	0.96892043e+02	0.10464646e+03	0.86035164e-02

******** e n d ******** test100

final uncaled parameters--

			resistivity		depth
1	0.10000155e+03	1	0.10000155e+03		
2	0.10064672e+01	2	0.10064672e+01		
3	0.90187436e+02	3	0.90187436e+02		
4	0.29996680e+02			1	0.29996680e+02
5	0.10076925e+03			2	0.13076593e+03