U.S. Department of the Interior
Geological Survey

Program MARQHXY:
Marquardt inversion of Hx and Hy frequency soundings from a grounded wire source

by

Walter L. Anderson

Open-File Report 80-901

1980

CONTENTS

DISCLAIMER 3

INTRODUCTION 4

PARAMETERS AND DATA REQUIRED 5

PROGRAM FILES 5

DETAILED PARAMETER AND DATA DEFINITIONS 6

$parms parameters 6

$init parameters 11

DATA MATRIX NOTES 14

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING 16

SPECIAL OBJECT FORMAT PHRASES 17

MULTICS OPERATING INSTRUCTIONS 17

ERROR MESSAGES 18

PRINTED OUTPUT 19

REFERENCES 22

Appendix 1.-- Source listing 23

Source availability 24

Copyright notices 24

Appendix 2.-- Conversion to other systems 105

Appendix 3.-- Test problem input/output listing 106

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 MARQHXY is a general-purpose program for inversion of horizontal magnetic field component (Hx and/or Hy) frequency sounding data obtained on an assumed horizontally stratified earth for the quasi-static case (i.e., neglecting displacement currents). The source is a grounded electric dipole or finite-wire of arbitrary length (positioned along the x-axis and centered at the origin). The Hx and Hy fields are assumed to be measured at the earth's surface. A modified Marquardt (1963) nonlinear least squares algorithm (MARQRT) is used for inversion of frequency sounding data. An adaptive digital filtering algorithm (ZHANKS) developed by Anderson (1979) is used for evaluating all Hankel transforms. See Anderson (1974), or Kauahikaua and Anderson (1977), for the associated forward problem solutions for Hx and Hy about a finite-wire source of arbitrary length. The inverse solution for the vertical magnetic field (Hz) is not provided in the present program; however, a separate inverse program (MARQHZP) for Hz soundings from a grounded wire source has been published by Anderson (1977).

The following program options are currently available:

(1) Simultaneous (or joint) inversion of Hx and Hy frequency soundings for a maximum of 9-layer models.
(2) Mixed frequency (parametric) and/or distance (geometric) sounding inversion. Also, mixed observation types can be used (e.g., amplitude, phase, real or imaginary parts).
(3) Inclusion of additional amplitude shift parameters for both Hx and Hy in the least squares when the correct primary field normalization factors are unknown.
(4) Scaling parameter and observation spaces to constrain the solution space and to reduce round-off effects.
(5) Weighted observations.
(6) Holding certain parameters fixed (constrained).
(7) Object-time format control of reading the observed data matrix.

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

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 MARQHXY are read using Fortran namelist read statements with specific names: $parms and $init. [Note that some parameter relationships occur (e.g., see $parms "k" and $init "mm") due to the general nature of subprogram MARQRT, which was designed for any nonlinear least squares problem.] 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. Preceeding the $parms statement is a required 80-character (or less) title.

The general input order read by program MARQHXY 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 made by repeating steps 1,2,3, and 5 (provided parameters istop=0 and ialt is not 5).

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

PROGRAM FILES

file05	title, input parameters $parms, object format (for reading data matrix on unit ialt=10--default), and $init parameters.
file06	output on-line printer file (see file16 for more detail output).
file10	default input data matrix file read under the object format given in file05. Parameter ialt=10 (default) may be changed to any file number other than 06,13, or 16. Note ialt=05 will mean the data matrix is included immediately after the object-time format on file05.
file13	output scratch disk file used as required during execution of MARQHXY.
file16	output master print-type disk file--contains maximum printable output (if parameter iout=1).

DETAILED 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; the last two parameters represent amplitude shifts for Hx and Hy, respectively (see definitions under parameter b below).
(cref: $init parameter mm and $parms n,b).

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

m= Number of independent variables (m<=4) given in the data matrix (y(i),x(i,j),j=1,m),i=1,n. The value of m must be given as follows:
= 1 when $init parameter -4<=iob<=4 (defines specific observation type in y(i));
= 2 when $init parameter iob=5 (defines mixed observation types in y(i) via x(i,2));
= 4 when $init parameter iob=6 (defines mixed observation types in y(i) via x(i,4) and distance coordinate x0 in x(i,2) and y0 in x(i,3).
(cref: $parms iwt, $init x0, y0, iob, and DATA MATRIX NOTES below for all definitions of x(i,m) used).

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

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

rewound and read again). Also, all $parms and $init parameters previously used will be assumed, with the exception of array b(j)--which must always be given.

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

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

= 1 for weighted observations given by the formula wt(i)=1.0/x(i,m+1)**2, where x(i,m+1) is the standard deviation augmented to the data matrix for the given m<=4. Note: wt(i)=1.0 is stored automatically if iwt=0 or when iwt=1 and x(i,m+1)=0.0 (to avoid division by 0).
(cref: $parms n,m, $init iob, and DATA MATRIX NOTES).

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

= 1 to use estimated derivatives, which calls only subroutine fcode. ider=1 option is useful to check the validity of the analytic derivatives, but is not recommended for general use because of accuracy and timing considerations. [However, for this version, ider=1 is required when $init method=0 and 1>0.0.]
(cref: $parms del and $init method,1).

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

= 1 for detailed 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 for achieving a convenient restarting procedure with the last b-vector as a new initial estimate.
(cref: $parms b and Marquardt (1963) for convergence tests used).

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

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

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

e= Convergence criterion test parameter (default 0.5e-4). For example, for 2-figure accuracy, use e=.01; for 3-figure accuracy, use e=.001, etc.
(cref: Marquardt, 1963).

tau= Convergence criterion test parameter (default 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 MARQHXY, the initial parameters b(j)>0 must be given; hence sp=2 should not be used (sp=2 is defined here for possible use in other applications).
(cref: $parms b,k).

sy= scaley (equivalent names) is an observation scaling option.
= 0 (default) to ignore observation scaling (i.e., unscaled observations y(i)).
= 1 to scale observations y(i) using ln(y(i)), provided y(i)>0 for all i=1,2,...,n.
= 2 to scale observations y(i) using arcsinh(y(i)). This option allows for log-type observation scaling whenever y(i) is positive, negative, or zero for any i in (1,n).

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

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

The parameter order must be given as follows:

b(1),b(2),...,b(mm) are the mm layer conductivities (in mhos per meter), and

b(mm+1),b(mm+2),...,b(2*mm-1) are the mm-1 layer thicknesses (in meters); in addition, include

b(2*mm)>0 as the estimated Hx amplitude shift parameter used in the model as b(2*mm)*Hx/Hxp, where Hxp is the primary Hx field; and

b(2*mm+1)>0 as the estimated Hy amplitude shift parameter used in the model as b(2*mm+1)*Hy/Hyp, where Hyp is the primary Hy field.

Note: If only phase data ($init iob=-2 or 2) or multiple distance soundings (iob=6) are used, then the shift parameters should be fixed using $parms ip and ib. Similarly, if only Hx (or Hy) data is given, then the corresponding Hy (or Hx) shift parameter should be fixed (e.g., set to 1.0) via $parms ip and ib.
(cref: $parms k,ip,ib and $init mm,iob).

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

$end [end of $parms namelist]

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

iob= Observation-type defined for y(i): [where we define Zx=b(2*mm)*Hx/Hxp and Zy=b(2*mm+1)*Hy/Hyp];
= 1 (default) defines y(i) as the amplitude of Zx;
= 2 defines y(i) as the phase of Zx, expressed in (-180,+180) degrees. [Note b(2*mm) should be fixed whenever iob=2.]
= 3 defines y(i) as the real-part of Zx;
= 4 defines y(i) as the imaginary-part of Zx;
=-1 defines y(i) as the amplitude of Zy;
=-2 defines y(i) as the phase of Zy, expressed in (-180,+180) degrees. [Note b(2*mm+1) should be fixed whenever iob=-2.]
=-3 defines y(i) as the real-part of Zy;
=-4 defines y(i) as the imaginary-part of Zy;
(Note: for |iob|<=4, m=1 must also be given in $parms.)
= 5 defines mixed observation-type frequency soundings, where the i-th observation type is given by x(i,2)=1.0 for amplitude of Zx, =2.0 for phase of Zx, =3.0 for real of Zx, =4.0 for imaginary of Zx, =-1.0 for amplitude of Zy, =-2.0 for phase of Zy, =-3.0 for real of Zy, or =-4.0 for imaginary of Zy.
(Note: for iob=5, m=2 must also be given in $parms.)
= 6 defines mixed observation-type frequency and/or distance soundings, where the i-th observation type is given by x(i,4) between -4.0 and 4.0 (same definitions as in iob=5 case), and x0=x(i,2) and y0=x(i,3) defines the wire-source and receiver

separation.
(Note: for iob=6, m=4 must also be given in $parms; also, the Hx and Hy shift parameters b(2*mm) and b(2*mm+1) should be fixed whenever iob=6 option is used.)
(cref: $parms m, b(), $init mm, and DATA MATRIX NOTES).

mm= Number of layers in the model (1<=mm<=9; default mm=1).
Note: make sure $parms k=2*mm+1.
(cref: $parms k,b(), $init iob).

x0= Transmitter-receiver x-separation, where x0>0.0 meters when 0<iob<5 (i.e., Hx data only). Note x0=0.0 is allowed when iob<=-1 for Hy data only. Also, x0 must be given, unless iob=6 is used for distance soundings.
(cref: $init iob and DATA MATRIX NOTES).

y0= Transmitter-receiver y-separation, where y0>0 meters. Note y0 must be given; but when iob=6, y0 is a dummy value here (see DATA MATRIX NOTES, (c)4).
(cref: $init iob and DATA MATRIX NOTES).

l= 0.0 (default) defines a dipole source (recommended for initial studies for any receiver-transmitter separation). For l>0.0 meters, a finite electric wire source is assumed to be positioned along the x-axis from x=-l to x=+l (i.e., the total wire-length is 2*l meters) as described in Anderson (1974). For many cases where the radial separation distance sqrt(x0*x0+y0*y0) is much greater than 2*l, a dipole source may generally be assumed; however, one may always consider the initial dipole solution (when l=0) as a good first approximation to the layering when "near-source" distances are used. Then one may use l>0.0 (and ider=1, which is required if method=0) for the final layered solution after obtaining the l=0 solution from an initial study.
(cref: $parms ider and $init method).

ep= Requested integration accuracy for all finite integrals when l>0.0 is used. (default ep=.1e-2).
(cref: $init parmeters eps,neps,method,ier).

eps= Requested convolution integration tolerance used to compute all Hankel transforms using subprogram ZHANKS (default .1e-5). (Note: eps<=ep is required when l>0.0.)

(cref: $init parameters ep,l).

neps= Approximate number of calls to subprogram fcode before setting ep=eps (default neps=10). This option applies only when l>0.0 finite-wire option is used. Note: neps, ep, and eps may be used to significantly reduce the total computer CPU-time when l>0.0 and xl>=.01, since higher accuracy in the finite integrals are usually not required in the early gradient search stages of the Marquardt algorithm.
(cref: $parms xl and $init parameters ep,eps,l).

method= 0 (default) to use lagged-convolution method (see Anderson, 1975) for all Hankel transforms, and adaptive quintic-spline interpolation integration over the finite-wire length (-l,l) when l>0.0. When l=0 (dipole), method=0 behaves like method=2, and direct convolution is used for all hankel transforms; method=0 is about 10-times faster than method=2 when l>0.0, and usually gives about 3 or more figure accuracy. (Note: when method=0 and l>0.0, only ider=1 can be used in the present version.)
(cref: $parms ider and $init l).

= 2 to use direct convolution method for all Hankel transforms and direct adaptive integration over the finite-wire length when l>0.0. [Note: method=2 is capable of yielding higher accuracy (via parameters ier,ep and eps), but method=2 is not recommended for routine use.]
(cref: $init parameters ep,eps,neps,ier,nfin).

nfin= 1 (default) is used when method=0 to indicate the number of interpolation passes to sample in the lagged-convolution; i.e., the interpolation interval=.2/float(nfin). Normally nfin=1 is adequate for about 3-figure accuracy; however, nfin=2 or 3 will give greater accuracy, but the run time will be 2 or 3 times longer.
(cref: $init parameter method).

ier= Type of adaptive quadrature and convergence test to select for definite integration over a finite wire source (when l>0.0). Parameter ier is ignored for a dipole source (l=0.0).

= 1 for absolute error test (not generally recommended to use first).

= 2 (default) for L-one norm error test (recommended first).

= 3 for L-infinity norm error test (try if ier=2 fails, etc.). (Note: ier<=3 uses an adaptive

Newton-Cotes quadrature.)
= 4 for relative error test using adaptive Gaussian quadrature. (Note: ier=4 may be faster than ier=2 because adaptive Gaussian quadrature is generally more efficient than adaptive Newton-Cotes quadrature--but not always.)
= 5 for relative error test using non-adaptive Gaussian quadrature.
= 0 for special case to force ier=2 and to ignore all "ep warning messages", as noted under parameter mev below.
(cref: $init 1,ep,mev).

mev= Maximum allowable complex function evaluations permitted in the adaptive integration procedure when 1>0.0 (default mev=300).

Note: The program will print the message "warning--ep accuracy not achieved in..." if ep accuracy cannot be achieved in approximately mev function evaluations when 1>0.0. In this case, and if ier<=5, the program accepts the integral and continues processing after setting ier=ier+1 (if however, ier>5 is generated, then ier=0 is automatically set to suppress any further ep warning messages). At each warning message, some additional integration information is printed regarding the accuracy (cerr) or actual error obtained. If cerr is reasonably small, one may decide to accept the results and ignore the warning. On the other hand, it may be desirable to rerun the problem with different parameter values used for method, ep, eps, neps, mev, and ier. Also, a data and parameter check should be done before rerunning the program. As a last resort, a special run may be made using ier=0 to bypass any and all ep error messages; however, this may be dangerous. Therefore ier=0 is not recommended for routine work.
(cref: $init 1,method,ep,eps,neps,mev,and ier).

$end [end of $init parameters]

DATA MATRIX NOTES

The data matrix is defined as the sequence of ordered rows: (y(i),x(i,j),j=1,m*), where i=row number 1,2,...,n, and m*=m+1 if iwt=1, otherwise m*=m<=4. The data matrix is read on logical unit ialt (default 10) using an object-time format statement (see any Fortran manual). The number of

items read depends on $parms m,iwt and $init iob as previously defined. The various data matrix options are summarized as follows:

(a) Specific observation type, Hx or Hy frequency sounding (-4<=iob<=4, m=1, and max. 3 items per record):

1. y(i)= i-th observation, where $init -4<=iob<=4 defines the particular type.
2. x(i,1)= i-th frequency (x(i,1)>0.0 Hz.).
3. x(i,2)= standard deviation of observation i (include only if iwt=1).

(b) Mixed observation types, Hx and/or Hy frequency soundings (iob=5, m=2, and max. 4 items per record):

1. y(i)= i-th observation (where actual type is defined by x(i,2)).
2. x(i,1)= i-th frequency (x(i,1)>0.0 Hz.).
3. x(i,2)= observation type in y(i); for Zx=b(2*mm)*Hx/Hxp, use x(i,2)= 1.0 for amplitude, =2.0 for phase (degrees), =3.0 for real part, or =4.0 for imaginary part of Zx; for Zy=b(2*mm+1)*Hy/Hyp, use x(i,2)=-1.0 for amplitude, =-2.0 for phase (degrees), =-3.0 for real part, or =-4.0 for imaginary part of Zy.
4. x(i,3)= standard deviation of observation i (include only if iwt=1).

(c) Mixed observation types, Hx and/or Hy frequency and distance soundings (iob=6, m=4, and max. 6 items per record):

1. y(i)= i-th observation (where actual type is defined by x(i,4)).
2. x(i,1)= i-th frequency (x(i,1)>0.0 Hz.).
3. x(i,2)= i-th transmitter-receiver x0-coordinate, in meters (same rules as with $init x0).
4. x(i,3)= i-th transmitter-receiver y0-coordinate, in meters (same rules as with $init y0).
5. x(i,4)= observation type in y(i), where x(i,4) must be between -4.0 and 4.0 as defined in (b)3 above.
6. x(i,5)= standard deviation of observation i (include only if iwt=1).

The data matrix should be grouped or ordered with equal consecutive frequencies (and/or distances, if used) with respect to each observation type (for example, see the grouping used in appendix 3). This ordering is not mandatory, but it will significantly reduce the total

calculation time when l=0.0 and ider=0 (default case).

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING

1. Specific observation type; phase of Hy (iob=-2), finite-wire source (l>0.0), fixed shift parameter (required since iob=-2):

```
example 1.
$parms n=60,k=7,m=1,iprt=-1,sp=1,sy=2,ialt=5,
 e=.005,niter=4,xl=.1,
 ip=2,ib=6,7,
 b=.1,.2,.3,10,20,2*1$
(2f10.0)
-2.8        1.
-4.15       1.2
-8.1        1.6
-10.2       2.
--(etc. for 56 more observations)--
$init mm=3,iob=-2,y0=100,x0=100,l=200,
 ep=.01,eps=.001,neps=30,ier=4$
```

2. Mixed observation types (real and imaginary parts), Hx and Hy soundings, dipole-source (l=0.0), unknown shift parameters:

```
example 2.
$parms n=100,k=7,m=2,iprt=-2,sp=1,sy=2,ialt=5,
 b=.1,.2,.3,10,20,.5,1.5$
(3f10.0)
1.01        1.          3.
-2.3        1.          4.
0.987       1.          -3.
-5.23       1.          -4.
0.79        1.6         3.
-2.34       1.6         4.
0.867       1.6         -3.
-10.23      1.6         -4.
--(etc. for rest of soundings)--
$init mm=3,iob=5,x0=100,y0=200$
```

3. Distance Hx and Hy amplitude soundings, dipole-source (l=0), weighted observations (iwt=1), and fixed Hy shift parameter:

```
example 3.
$parms n=50,k=7,m=4,iprt=-1,sp=1,sy=2,ialt=5,iwt=1,
 b=.1,.2,.3,10,20,2,1, ip=1,ib=7$
(6f10.0)
1.98        1.          100.        200.        1.          .02
0.998       1.          100.        200.        -1.         .03
```

```
--(etc. for rest of freq. sounding at this spacing)--
1.56      1.2       300.      400.      1.        .02
0.97      1.2       300.      400.      -1.       .025
1.32      4.        300.      400.      1.        .04
0.832     4.        300.      400.      -1.       .045
--(etc. for rest of freq. sounding at this spacing)--
$init mm=3,iob=6,y0=1$
```

SPECIAL OBJECT FORMAT PHRASES

One may use special Fortran object formats to skip observations without changing the data matrix. For example, if we wish to use only the Hy amplitude data in example 3 above, we could set n=25 and use the format (/6f10.0). Similarly, if we wanted only Hx amplitudes to be used in example 3, then the format (6f10.0/) would accomplish the desired result.

Also, 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,t1,3f10.0) will select y(i) using col.41-50 and x(i,1) beginning at col.1.

MULTICS OPERATING INSTRUCTIONS

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

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

3. Attach "file10" to an input data matrix ascii file if ialt=10 (default) is used. If ialt=5 is selected, then ignore this step, but include the data matrix following the object-time format on "file05"--see examples 1-3 above. In practice, it is usually best to use distinct files file05 and file10 for parameters and data respectively. To attach file10, type:
   io attach file10 vfile_ data_file_name

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

5. Execute program MARQHXY by typing: marqhxy

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 found on disk on Multics), and "file06" is always the on-line terminal print file. File16 should either be deleted or printed on a line printer after running program MARQHXY. Also, file13 (if used) should be deleted after running the program. To submit the job as a batch job (called an absentee job 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 1.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 MARQHXY.

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
sigma(i)	conductivity (in mhos/meter) 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 (=xl 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., 1974, Electromagnetic fields about a finite electric wire source: U.S. Geological Survey Report USGS-GD-74-041, 205 p. avail. from U.S. Department Commerce National Technical Information Service (NTIS), Springfield, Va., 22161 as Report PB-238-199/4WC.

-------, 1975, Improved digital filters for evaluating Fourier and Hankel transform integrals: U.S. Geological Survey Report USGS-GD-75-012, 223 p. avail. from U.S. Department Commerce NTIS, Springfield, Va., 22161 as Report PB-242-800/1WC.

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

-------, 1979, Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics, vol. 44, no. 7, p. 1287-1305.

Herriot, J.G., and Reinsch, C.H., 1976, Algorithm 507, Procedures for quintic natural spline interpolation: ACM Transactions on Mathematical Software, v. 2, no. 3, p. 281-289.

Kauahikaua, J., and Anderson, W.L., 1977, Calculation of standard transient and frequency sounding curves for a horizontal wire source of arbitrary length: U.S. Geological Survey Report USGS-GD-77-007, 63 p. avail. from U.S. Department Commerce NTIS, Springfield, Va. 22161 as Report PB-274-119.

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.

Patterson, T.N.L., 1973, Alogrithm for automatic numerical integration over a finite interval [D1]: Association for Computing Machinery Communication, v. 16, no. 11, p. 694-699.

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.

Appendix 1.-- Source listing

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

```
C--MARQHXY--MARQUARDT INVERSION OF HX,HY-FINITE WIRE DATA-- 7/31/78.    00000010
      SUBROUTINE MARQRT(FCODE,PCODE,SUBZ,SUBEND)                        00000230
      SUBROUTINE GJR (A,N,EPS,MSING)                                    00009990
      SUBROUTINE UNSCAL(BIN,BOUT,SCALEP)                                00010650
      REAL FUNCTION ASINH(X)                                            00010860
      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                   00010940
      SUBROUTINE POLAR2(Z,AMP,PHZ180)                                   00011170
      SUBROUTINE RECUR1(G,V1,F1)                                        00011460
      SUBROUTINE RECURF(G,DEL,SIG1,V1,F1,PF1,JJ)                        00011770
      SUBROUTINE RECURS(G,V1,F1)                                        00012390
      COMPLEX FUNCTION F3(G)                                            00012700
      COMPLEX FUNCTION F4(G)                                            00012780
      COMPLEX FUNCTION GF4(G)                                           00012840
      COMPLEX FUNCTION FINITE(FUNC,BFIN)                                00012900
       COMPLEX FUNCTION FINHX(B)                                        00013880
      COMPLEX FUNCTION FINHY(B)                                         00014130
      SUBROUTINE SETRH0(X)                                              00014460
      COMPLEX FUNCTION ZHY(B,NEW,R)                                     00014720
        COMPLEX FUNCTION ZLAGH0(X,FUN,TOL,L,NEW)                        00014930
      COMPLEX FUNCTION CANC4(A1,B1,EP,M,N,FUN,MF,ESUM)                  00017170
      SUBROUTINE CQUAD(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)            00018740
      COMPLEX FUNCTION CQSUB(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)  00022260
      COMPLEX FUNCTION CQSUBA(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)00023790
      COMPLEX FUNCTION FUNINT(X)                                        00025200
      SUBROUTINE QUINT(NY,Y,B,C,D,E,F)                                  00025360
      SUBROUTINE QPOINT(NY,Y,B,C,D,E,F,X1,DELX,XX,YY)                   00026130
      SUBROUTINE KELVIN(X,M,B)                                          00026320
      SUBROUTINE IK1(B8,I1K1)                                           00028120
      SUBROUTINE IKS(B8,I1K1,IKDIF)                                     00028420
      SUBROUTINE BESSIK(B,ZBES)                                         00029060
      SUBROUTINE IKSALL(B8,I0K0,I1K1,I0K1,I1K0,IKDIF)                   00029230
       SUBROUTINE FINF3(B1,B2,F31,F32)                                  00030060
      COMPLEX FUNCTION HX03(X)                                          00030180
      COMPLEX FUNCTION HY03(X)                                          00030600
      COMPLEX FUNCTION F3PJ(G)                                          00031050
      COMPLEX FUNCTION F4PJ(G)                                          00031210
      COMPLEX FUNCTION PHXPJ(X)                                         00031280
      COMPLEX FUNCTION PHYPJ(X)                                         00031850
      SUBROUTINE PRMHXY                                                 00032470
      SUBROUTINE MODIFY(N)                                              00032730
      SUBROUTINE SWAP(ICODE)                                            00032950
      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                       00033190
      SUBROUTINE FCODE(Y,X,B,PRNT,F,I,IDER)                             00036610
      SUBROUTINE PCODE(P,X,B,PRNT,F,I,IP,IB)                            00038000
      SUBROUTINE SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)                    00039030
      SUBROUTINE SUBEND(Y,X,B,K,N,TITLE,IOUT)                           00040350
```

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 is usually recorded in the following mode (unless otherwise requested):

> Industry compatible: 9-track, unlabeled, EBCDIC mode, odd-parity, 800 bpi density, 80-character records (blocked, 50-card images per block), and contained on one file.

Copyright Notices

(1). Subprogram QUINT was converted to FORTRAN from the original ALGOL program published by Herriot and Reinsch (1976): Copyright 1976, Association for Computing Machinery, Inc.; permission to republish, all or in part, was granted by ACM.

(2). Subprograms CQUAD, CQSUB, and CQSUBA are modified versions of subprograms QUAD, QSUB, and QSUBA, respectively, which were published by Patterson (1973): Copyright 1973, Association for Computing Machinery, Inc.; permission to republish, all or in part, was granted by ACM.

---

* present address is:

U.S. Geological Survey
Mail Stop 964
Box 25046, Federal Center
Denver, Colorado 80225

```
C--MARQHXY--MARQUARDT INVERSION OF HX,HY-FINITE WIRE DATA-- 7/31/78.     00000010
C** HONEYWELL MULTICS VERSION **                                         00000020
C** THIS VERSION IS FOR IMPROVED SPEED WHEN L>0.0 AND METHOD=0 (THE      00000030
C   DEFAULT OPTION).  METHOD=0,L>0.0 USES LAG-CONVOLUTION (SUBR ZLAGH1, 00000040
C   ZLAGH0), QUINTIC SPLINES (SUBR QUINT, QPOINT), AND                   00000050
C   AUTOMATIC INTEGRATION (VIA SUBR FINITE AND CANC4).  HOWEVER,         00000060
C   ONE SHOULD ALWAYS USE L=0.0 (DIPOLE) FOR INITIAL RUNS UNTIL A        00000070
C   LAYER SOLUTION IS OBTAINED, THEN USE L>0.0 FOR A FINAL SOLUTION--    00000080
C   IF NECESSARY.                                                        00000090
C                                                                        00000100
C** METHOD=0, L>0.0 IS ABOUT 10-TIMES FASTER THAN METHOD=2 AND SHOULD    00000110
C   GIVE ABOUT 3 TO 5 FIGURE ACCURACY IN THE FINITE INTEGRALS, DEPENDING00000120
C   ON PARAMETERS: EP, EPS, AND NEPS (ALSO IER=2 OR 4 SHOULD BE USED).   00000130
C                                                                        00000140
C--BY W.L.ANDERSON, U.G.GEOLOGICAL SURVEY, DENVER, COLORADO.             00000150
C                                                                        00000160
C  SUBROUTINES FCODE,PCODE,SUBZ,AND SUBEND TO LINK WITH PGM MARQRT.      00000170
C                                                                        00000180
      EXTERNAL FCODE,PCODE,SUBZ,SUBEND                                   00000190
      CALL MARQRT(FCODE,PCODE,SUBZ,SUBEND)                               00000200
      STOP                                                               00000210
      END                                                                00000220

      SUBROUTINE MARQRT(FCODE,PCODE,SUBZ,SUBEND)                         00000230
C--(MARQRT)-- GENERAL MARQUARDT NONLINEAR LEAST SQUARES-- 7/11/78.       00000240
C** HONEYWELL MULTICS VERSION **                                         00000250
C  SUBPROGRAM MARQRT IS TO BE LINKED/LOADED WITH USER WRITTEN            00000260
C  SUBROUTINES (FCODE,PCODE,SUBZ, AND SUBEND) FOR                       00000270
C  SPECIFIC NONLINEAR PROBLEM TO BE SOLVED.                             00000280
C                                                                        00000290
C--THE USER MUST DECLARE THE CALLING PARAMETERS FCODE,PCODE,            00000300
C  SUBZ,SUBEND (ANY DESIRED NAMES MAY BE USED) AS EXTERNAL IN           00000310
C  MAIN CALLING PROGRAM; E.G.,                                          00000320
C                                                                        00000330
C      EXTERNAL FCODE,PCODE,SUBZ,SUBEND                                  00000340
C      CALL MARQRT(FCODE,PCODE,SUBZ,SUBEND)                              00000350
C      STOP                                                              00000360
C      END                                                               00000370
C                                                                        00000380
C--THIS IS A MODIFIED VERSION OF 'IBM SHARE PROGRAM NO. 1428'.           00000390
C *** MODIFIED BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY,DENVER, COLORADO  00000400
C  FOR NAMELIST INPUT, IMPROVED ESTIMATED DERIVATIVES,                   00000410
C  MODIFIED MARQUARDT LAMBDA DETERMINATION,                              00000420
C  DATA AND PARAMETER SCALING, WEIGHTED OBSERVATIONS, AND                00000430
C  OTHER CHANGES--ALL DONE IN SINGLE-PRECISION FOR THE                   00000440
C *** HONEYWELL MULTICS SYSTEM ***                                       00000450
C                                                                        00000460
C--SEE SHARE PROGRAM NO. 1428 AND/OR COMMENTS IN SUBPROGRAMS FCODE,      00000470
C  PCODE,SUBZ AND SUBEND BELOW, FOR DETAILS ON CODING THE                00000480
C  REQUIRED SUBROUTINES FCODE,PCODE,SUBZ, AND SUBEND.                    00000490
C                                                                        00000500
C--OPERATING NOTE FOR HONEYWELL MULTICS SYSTEM:  $$$$$$$$$$$$$$$$$$$$  00000510
```

```
C                                                                       00000520
C  TO OBTAIN ON-LINE (INTERACTIVE) PRINTING ON UNIT 6 AND               00000530
C  DEFERRED PRINTING ON UNIT 16, USE MULTICS RUN.EC, I.E.,              00000540
C  'RUN &1' OR 'RUN_EO &1' AND DPRINT '&1.FILE16.LIST' AFTER RUN.       00000550
C                                                                       00000560
C--FOLLOWING CHARACTER STATEMENTS ONLY FOR MULTICS SYSTEM:              00000570
      CHARACTER*5 TITLE                                                 00000580
      CHARACTER*4 FMT                                                   00000590
      INTEGER SCALEP,SCALEY, SP,SY                                      00000600
      DIMENSION FMT(18),PRNT(5),SPRNT(5),TITLE(16)                      00000610
      DIMENSION BS(20),DB(20),BA(20),G(20),IB(19),SA(20),P(20)          00000620
      DIMENSION A(20,20),B(20), BINV(20)                                00000630
      DIMENSION X(200,5),Y(200),WT(200)                                 00000640
      DIMENSION XNU(5),SS(4)                                            00000650
      EQUIVALENCE (X(1,5),WT(1)),(IOUT,IFSS1),(IDER,IWS2),              00000660
     1 (IPRT,IWS3),(NITER,IWS4),(INON,IWS6),(SP,SCALEP),(SY,SCALEY)     00000670
C=======================================================              00000680
      NAMELIST/PARMS/N,K,IP,M,IALT,IDER,IPRT,NITER,INON,NPRNT,          00000690
     1 IB,FF,T,E,TAU,XL,GAMCR,DEL,ZETA,B,IOUT,IWT,ISTOP,                00000700
     2 SCALEP,SCALEY,MODLAM, SP,SY                                      00000710
C=======================================================              00000720
      DATA XNU/1.33,1.78,3.16,10.,100./                                 00000730
C                                                                       00000740
C     MAX NO OF PARAMETERS IS K=20  (NOTE: K=N IS ALLOWED)              00000750
C     MAX NO OF IND VARS IS M=4                                         00000760
C     MAX NO OF OBSERVATIONS IS N=200                                   00000770
C     INTERNAL #IWHER# SWITCH USAGE--                                   00000780
C                  IWHER =-1 MEANS INITIALIZE VIA SUBROUTINE SUBZ.      00000790
C                  IWHER = 0 MEANS START NEW PROBLEM OR END RUN         00000800
C                  IWHER = 1 MEANS GET P(S) AND F                       00000810
C                  IWHER GREATER THAN 1 MEANS GET F ONLY                00000820
C--FOLLOWING CALL TO SUPPRESS EXP-UNDERFLOW MESSAGES                    00000830
C  FOR THE DEC-10 AND OTHER SYSTEMS:  $$$$$$$$$$$$$$$$$$$$             00000840
C//        CALL ERRSET(0)                                               00000850
C** FOR THE HONEYWELL MULTICS SYSTEM, USE (INSTEAD) THE FOLLOWING:      00000860
C   IO DETACH ERROR_OUTPUT                                              00000870
C   IO ATTACH ERROR_OUTPUT DISCARD_                                     00000880
C  (OR-- ON USGS SYS, USE SET_UFL -OFF)                                 00000890
C**                                                                     00000900
C--PRESET GLOBAL PARMS (SOME MAY BE OVERRIDDEN BY $PARMS READ-IN)       00000910
      IP=0                                                              00000920
      N=0                                                               00000930
      K=0                                                               00000940
      M=0                                                               00000950
      NPRNT=0                                                           00000960
      MODLAM=1                                                          00000970
      ISTOP=1                                                           00000980
      IWT=0                                                             00000990
      IALT=10                                                           00001000
      IOUT=1                                                            00001010
      IDER=0                                                            00001020
      IPRT=0                                                            00001030
```

```
      MITER=10                                                          00001040
      INON=1                                                            00001050
      LSCALP=0                                                          00001060
      LSCALY=0                                                          00001070
      FF=4.0E0                                                          00001080
      E=.00005E0                                                        00001090
      TAU=.001E0                                                        00001100
      T=2.0E0                                                           00001110
      DEL=.00001E0                                                      00001120
      ZETA=.1E-30                                                       00001130
      GAMCR=45.0E0                                                      00001140
C                                                                       00001150
   10 GAMMA=0.E0                                                        00001160
      SCALEP=LSCALP                                                     00001170
      SCALEY=LSCALY                                                     00001180
      XLL=0.E0                                                          00001190
      SE=0.0                                                            00001200
      NITER=MITER                                                       00001210
   20 IWHER=0                                                           00001220
      ISS=1                                                             00001230
      INU=4                                                             00001240
      XNUFAC=10.0                                                       00001250
      GO TO 150                                                         00001260
   30 CONTINUE                                                          00001270
      IF (IWHER.GT.0) GO TO 100                                         00001280
      IF (IWHER.EQ.0) GO TO 240                                         00001290
C=======================================================              00001300
C     INITIALIZATION (IWHER=-1, IFSS1=IOUT)                             00001310
      CALL SUBZ (Y,X,BINV,PRNT,NPRNT,N,TITLE,IFSS1)                     00001320
C     ************************************************                  00001330
      IPRNT=NPRNT-1                                                     00001340
      IF(NPRNT.LT.0) IPRNT=IABS(NPRNT)-2                                00001350
C                                                                       00001360
C--NOTE: IPRNT IS A SPECIAL INDEX USED IN SCALEY=2 CASES                00001370
C  TO MIX LOG OR ASINH TYPE SCALING WHEN ABS(X(I,IPRNT))=1. OR NOT 1.   00001380
C  RESPECTIVELY, AND ONLY WHEN IPRNT.GT.1                               00001390
      NPRNT=IABS(NPRNT)                                                 00001400
      IF(SCALEY.EQ.0) GO TO 90                                          00001410
      DO 80 I=1,N                                                       00001420
      IF(SCALEY-1) 90,40,60                                             00001430
   40 IF(Y(I).LE.0.)CALL ERRMSG(30HSOME Y(I).LE.0 AND SCALEY=1...,      00001440
     1 6,6,16)                                                          00001450
   50 Y(I)=ALOG(Y(I))                                                   00001460
      GO TO 80                                                          00001470
   60 IF(IPRNT.LE.1) GO TO 70                                           00001480
      IF(ABS(X(I,IPRNT)).NE.1.0) GO TO 70                               00001490
      IF(Y(I).LE.0.)                                                    00001500
     1CALL ERRMSG(50HSOME Y(I).LE.0 WHEN ABS(X(I,IPRNT))=1 AND SCALEY=2,00001510
     2 10,6,16)                                                         00001520
      GO TO 50                                                          00001530
   70 Y(I)=ASINH(Y(I))                                                  00001540
   80 CONTINUE                                                          00001550
```

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

```
      DO 210 I=1,K                                                          00002080
      IF(B(I).EQ.0.E0) CALL ERRMSG(20HSOME B(I) = 0.0      ,4,6,16)         00002090
      BINV(I)=B(I)                                                          00002100
      IF(SCALEP-1) 210,190,200                                              00002110
  190 IF(B(I).LT.0.0)CALL ERRMSG(30HSOME B(I).LT.0. AND SCALEP=1..,         00002120
     1 6,6,16)                                                              00002130
      B(I)=ALOG(B(I))                                                       00002140
      GO TO 210                                                             00002150
  200 B(I)=ASINH(B(I))                                                      00002160
  210 CONTINUE                                                              00002170
      MAXITR=IWS4                                                           00002180
      MITER=NITER                                                           00002190
      ITER=1                                                                00002200
      WRITE (6,2730)                                                        00002210
      IF (IFSS1.NE.1) GO TO 250                                             00002220
      WRITE (16,2730)                                                       00002230
      GO TO 250                                                             00002240
C=============================================================            00002250
C                  END OF LAST PROBLEM                                      00002260
  220 CALL SUBEND(Y,X,BINV,K,N,TITLE,IOUT)                                  00002270
C     ***************************************************                 00002280
  240 IF(ISTOP.EQ.1.OR.IALT.EQ.5) GO TO 241                                 00002290
C--INITIALIZE FOR NEXT PROB (SAME IALT DATA), SINCE ISTOP=0                 00002300
      GO TO 10                                                              00002310
C--FOLLOWING CLOSE STMT ONLY FOR HONEYWELL MULTICS:                         00002320
  241 CALL CLOSE_FILE('-ALL')                                               00002330
C     STOP                                                                  00002340
      RETURN                                                                00002350
  250 CONTINUE                                                              00002360
      IF (IP.LE.0) GO TO 280                                                00002370
      DO 270 I=1,IP                                                         00002380
      IF (IB(I).GT.0) GO TO 270                                             00002390
      CALL ERRMSG(30HIP.GT.1 BUT SOME IB(I).LE.0...,6,6,16)                 00002400
  270 CONTINUE                                                              00002410
  280 CONTINUE                                                              00002420
      IF (K.GT.10) GO TO 290                                                00002430
C--IBKT=1 MEANS USE UPPER A MATRIX FOR SCRATCH STORAGE                      00002440
C       =2 MEANS USE FILE 13 FOR SCRATCH STORAGE                            00002450
      IBKT=1                                                                00002460
      GO TO 300                                                             00002470
  290 IBKT=2                                                                00002480
  300 XKDB=1.E0                                                             00002490
C--READ OBJECT TIME FORMAT FOR DATA ON FILE IALT.                           00002500
      READ(5,2480) (FMT(I),I=1,18)                                          00002510
      M1=M+IWT                                                              00002520
      DO 310 I=1,N                                                          00002530
      READ(IALT,FMT) Y(I),(X(I,L),L=1,M1)                                   00002540
C--SET UP WTS VIA IWT PARM                                                  00002550
      WT(I)=1.0E0                                                           00002560
      IF(IWT.EQ.1.AND.X(I,M1).NE.0.0) WT(I)=1.0E0/X(I,M1)**2                00002570
  310 CONTINUE                                                              00002580
      IF(IALT.NE.5) REWIND IALT                                             00002590
```

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

```
C                    GET P(J) AND F                                    00003120
      GO TO 30                                                        00003130
  490 IF (IP.LE.0) GO TO 640                                          00003140
      DO 500 II=1,IP                                                  00003150
      IWS=IB(II)                                                      00003160
  500 P(IWS)=0.E0                                                     00003170
      GO TO 640                                                       00003180
C      ..................................................             00003190
C                    THIS IS THE ESTIMATED P(J) ROUTINE              00003200
C  (VIA K.M. BROWN S METHOD)                                          00003210
  510 CONTINUE                                                        00003220
      ISW=1                                                           00003230
      IF(XL.LT.0.1E-3) ISW=2                                          00003240
  520 IWHER=3                                                         00003250
      GO TO 30                                                        00003260
  530 FWS=FINV                                                        00003270
      FSAV=F                                                          00003280
      DO 540 II=1,NPRNT                                               00003290
  540 SPRNT(II)=PRNT(II)                                              00003300
      J=1                                                             00003310
  550 IF (IP.LE.0) GO TO 570                                          00003320
      DO 560 II=1,IP                                                  00003330
      IF ((J-IB(II)).EQ.0) GO TO 610                                  00003340
  560 CONTINUE                                                        00003350
  570 HH=DEL*ABS(BINV(J))                                             00003360
      IF(ISW.EQ.2) HH=1.E3*HH                                         00003370
      IF(HH.LE.5.E-5) HH=5.E-5                                        00003380
      TWS=B(J)                                                        00003390
      TWS1=BINV(J)                                                    00003400
      BINV(J)=TWS1+HH                                                 00003410
      IWHER=4                                                         00003420
      GO TO 30                                                        00003430
  580 B(J)=TWS                                                        00003440
      BINV(J)=TWS1                                                    00003450
      IF(ISW.EQ.1) GO TO 600                                          00003460
C--CENTRAL DIFFERENCES (ISW=2--WHEN XL.LT..1E-3)                      00003470
      FHH=FINV                                                        00003480
      BINV(J)=TWS1-HH                                                 00003490
      IWHER=5                                                         00003500
      GO TO 30                                                        00003510
  590 B(J)=TWS                                                        00003520
      BINV(J)=TWS1                                                    00003530
      P(J)=.5E0*(FHH-FINV)/HH                                         00003540
      GO TO 620                                                       00003550
C--FORWARD DIFFERENCES (ISW=1--WHEN XL.GE..1E-3)                      00003560
  600 P(J)=(FINV-FWS)/HH                                              00003570
      GO TO 620                                                       00003580
  610 P(J)=0.E0                                                       00003590
  620 J=J+1                                                           00003600
      IF ((J-K).LE.0) GO TO 550                                       00003610
      FINV=FWS                                                        00003620
      F=FSAV                                                          00003630
```

```
      DO 630 II=1,NPRNT                                                 00003640
  630 PRNT(II)=SPRNT(II)                                                00003650
C                   END OF ESTIMATED P S ROUTINE                        00003660
C     ..................................................                00003670
C                   NOW, USE THE P(J) TO MAKE PARTIALS MATRIX           00003680
C--SET UP FOR SCALING PARTIAL DERIVATIVES AS SELECTED                   00003690
  640 IF(SCALEP-1) 650,710,730                                          00003700
  650 IF(SCALEY-1) 750,660,690                                          00003710
  660 DEN=1.0E0/FINV                                                    00003720
  670 DO 680 JJ=1,K                                                     00003730
  680 P(JJ)=P(JJ)*DEN                                                   00003740
      GO TO 750                                                         00003750
  690 IF(IPRNT.LE.1) GO TO 700                                          00003760
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 660                              00003770
  700 DEN=1.0E0/SQRT(FINV*FINV+1.0E0)                                   00003780
      GO TO 670                                                         00003790
  710 DO 720 JJ=1,K                                                     00003800
  720 P(JJ)=BINV(JJ)*P(JJ)                                              00003810
      GO TO 650                                                         00003820
  730 DO 740 JJ=1,K                                                     00003830
      DEN=BINV(JJ)+SQRT(BINV(JJ)**2+1.0E0)                              00003840
  740 P(JJ)=0.5E0*(DEN+1.0E0/DEN)*P(JJ)                                 00003850
      GO TO 650                                                         00003860
  750 IF(IBKA.EQ.2) WRITE(13) (P(JJ),JJ=1,K)                            00003870
      DO 760 JJ=1,K                                                     00003880
      G(JJ)=G(JJ)+WT(I)*(Y(I)-F)*P(JJ)                                  00003890
      DO 760 II=JJ,K                                                    00003900
      A(II,JJ)=A(II,JJ)+WT(I)*P(II)*P(JJ)                               00003910
  760 A(JJ,II)=A(II,JJ)                                                 00003920
  770 WS=Y(I)-F                                                         00003930
      IF (IFSS3.LE.0) GO TO 810                                         00003940
C--LAST ITERATION RESULTS AND DATA MATRIX FOR PRINTING                  00003950
      IF(I.GT.1) GO TO 771                                              00003960
      IF(IOUT.EQ.0) GO TO 773                                           00003970
      WRITE(16,430)                                                     00003980
      WRITE(16,2550)                                                    00003990
773   IF(IPRT.LT.-1) WRITE(6,772)                                       00004000
772   FORMAT(/11H -UNSCALED-/3X,1HI,4X,3HOBS,11X,3HCAL,11X,3HRES,       00004010
     1 8X,6HX(I,1))                                                     00004020
771   IF(IPRT.LT.-1) WRITE (6,2700) I,Y(I),F,WS,PRNT(1)                 00004030
      IF(NPRNT.GT.0) GO TO 790                                          00004040
      IF (IFSS1.NE.1) GO TO 780                                         00004050
      WRITE (16,2700) I,Y(I),F,WS                                       00004060
  780 CONTINUE                                                          00004070
      GO TO 810                                                         00004080
790   CONTINUE                                                          00004090
      IF (IFSS1.NE.1) GO TO 800                                         00004100
      PERR=0.0                                                          00004110
      IF(F.NE.0.0) PERR=100.0*WS/ABS(F)                                 00004120
      WRITE (16,2700) I,Y(I),F,WS,PERR,(PRNT(JJ),JJ=1,NPRNT)            00004130
  800 CONTINUE                                                          00004140
  810 WS=Y(I)-F                                                         00004150
```

```
      PHI=PHI+WT(I)*WS*WS                                              00004160
      I=I+1                                                            00004170
      IF (I.LE.N) GO TO 470                                            00004180
C--THIS IN END OF I=1 TO N NON-DO LOOP                                 00004190
      IF(IBKA.NE.2) GO TO 860                                          00004200
C--PRINT UNSCALED PARTIALS SAVED ON FILE 13 (WHEN IBKA=2)              00004210
  820 FORMAT(/20H -UNSCALED PARTIALS-)                                 00004220
      IF(IOUT.EQ.1) WRITE(16,820)                                      00004230
      REWIND 13                                                        00004240
      DO 850 II=1,N                                                    00004250
      READ(13) (SA(JJ),JJ=1,K)                                         00004260
  830 FORMAT(2X,I3,5E18.8)                                             00004270
  840 FORMAT(2X,I3,5E18.8/(5X,5E18.8))                                 00004280
      IF(IOUT.EQ.1.AND.K.NE.5) WRITE(16,840) II,(SA(JJ),JJ=1,K)        00004290
      IF(IOUT.EQ.1.AND.K.EQ.5) WRITE(16,830) II,(SA(JJ),JJ=1,K)        00004300
  850 CONTINUE                                                         00004310
      REWIND 13                                                        00004320
      WRITE(6,430)                                                     00004330
      IF(IOUT.EQ.1) WRITE(16,430)                                      00004340
  860 CONTINUE                                                         00004350
      IF (IP.LE.0) GO TO 890                                           00004360
      DO 880 JJ=1,IP                                                   00004370
      IWS=IB(JJ)                                                       00004380
      DO 870 II=1,K                                                    00004390
      A(IWS,II)=0.E0                                                   00004400
  870 A(II,IWS)=0.E0                                                   00004410
  880 A(IWS,IWS)=1.E0                                                  00004420
  890 IF(IBKA-2) 900,1770,1780                                         00004430
C                   SAVE SQUARE ROOTS OF DIAGONAL ELEMENTS             00004440
  900 DO 910 I=1,K                                                     00004450
  910 SA(I)=SQRT(A(I,I))                                               00004460
      DO 950 I=1,K                                                     00004470
      DO 930 J=1,K                                                     00004480
      WS=SA(I)*SA(J)                                                   00004490
      IF (WS.GT.0.E0) GO TO 920                                        00004500
      A(I,J)=0.E0                                                      00004510
      GO TO 930                                                        00004520
  920 A(I,J)=A(I,J)/WS                                                 00004530
  930 CONTINUE                                                         00004540
      IF (SA(I).GT.0.E0) GO TO 940                                     00004550
      G(I)=0.E0                                                        00004560
      GO TO 950                                                        00004570
  940 G(I)=G(I)/SA(I)                                                  00004580
  950 CONTINUE                                                         00004590
      DO 960 I=1,K                                                     00004600
  960 A(I,I)=1.E0                                                      00004610
      PHIZ=PHI                                                         00004620
C                   WE NOW HAVE PHI ZERO                               00004630
      IF(IBKT-1) 970,980,970                                           00004640
  970 WRITE (13) A                                                     00004650
      REWIND 13                                                        00004660
      GO TO 1000                                                       00004670
```

```
  980 DO 990 II=1,K                                                    00004680
      III=II+10                                                        00004690
      DO 990 JJ=1,K                                                    00004700
  990 A(III,JJ)=A(II,JJ)                                               00004710
C      .....                                                           00004720
 1000 CONTINUE                                                         00004730
      IF (ITCT.GT.0) GO TO 1030                                        00004740
C                   FIRST ITERATION                                    00004750
      IF (XL.GT.0.E0) GO TO 1010                                       00004760
      XL=0.01E0                                                        00004770
 1010 ITCT=1                                                           00004780
      DO 1020 J=1,K                                                    00004790
 1020 BS(J)=B(J)                                                       00004800
C                   BS(J) CORRESPONDS TO PHIZ                          00004810
 1030 IBK1=1                                                           00004820
      WS=N-K+IP                                                        00004830
      IF(N.GT.K) SE=SQRT(PHIZ/WS)                                      00004840
      IF (IFSS3.GT.0) GO TO 1040                                       00004850
      WRITE (6,2560) ITER,PHIZ,SE,XLL,GAMMA,XL                         00004860
      IF (IFSS1.NE.1) GO TO 1320                                       00004870
      WRITE (16,2560) ITER,PHIZ,SE,XLL,GAMMA,XL                        00004880
      GO TO 1320                                                       00004890
 1040   WRITE(6,2490) PHIZ,SE,XL                                       00004900
      IF (IFSS1.NE.1) GO TO 1320                                       00004910
      WRITE (16,2490) PHIZ,SE,XL                                       00004920
      GO TO 1320                                                       00004930
 1050 PHIL=PHI                                                         00004940
C                   WE NOW HAVE PHI(LAMBDA)                            00004950
      DO 1060 J=1,K                                                    00004960
      IF(ABS(DB(J)/(ABS(B(J))+TAU)).GE.E) GO TO 1080                   00004970
 1060 CONTINUE                                                         00004980
      WRITE (6,2680)                                                   00004990
      IF (IFSS1.NE.1) GO TO 1070                                       00005000
      WRITE (16,2680)                                                  00005010
 1070 CONTINUE                                                         00005020
      GO TO 1670                                                       00005030
 1080 IF (IWS4.EQ.0) GO TO 1110                                        00005040
      IF (IWS4.EQ.1) GO TO 1090                                        00005050
      IWS4=IWS4-1                                                      00005060
      ITER=ITER+1                                                      00005070
      GO TO 1110                                                       00005080
 1090 WRITE (6,2690)                                                   00005090
      IF (IFSS1.NE.1) GO TO 1100                                       00005100
      WRITE (16,2690)                                                  00005110
 1100 CONTINUE                                                         00005120
      GO TO 1670                                                       00005130
 1110 XKDB=1.E0                                                        00005140
      IF (PHIL.GT.PHIZ) GO TO 1190                                     00005150
      XLS=XL                                                           00005160
      DO 1120 J=1,K                                                    00005170
      BA(J)=B(J)                                                       00005180
 1120 B(J)=BS(J)                                                       00005190
```

```
      IF (XL.GT..00000001E0) GO TO 1140                                 00005200
      DO 1130 J=1,K                                                     00005210
      B(J)=BA(J)                                                        00005220
 1130 BS(J)=B(J)                                                        00005230
      GO TO 370                                                         00005240
 1140 XL=XL/XNUFAC                                                      00005250
      IBK1=2                                                            00005260
      GO TO 1320                                                        00005270
 1150 PHL4=PHI                                                          00005280
C                   WE NOW HAVE PHI(LAMBDA/XNUFAC)                     00005290
      IF (PHL4.GT.PHIZ) GO TO 1170                                      00005300
      DO 1160 J=1,K                                                     00005310
 1160 BS(J)=B(J)                                                        00005320
      GO TO 370                                                         00005330
 1170 XL=XLS                                                            00005340
C1170 CONTINUE                                                          00005350
      DO 1180 J=1,K                                                     00005360
      BS(J)=BA(J)                                                       00005370
 1180 B(J)=BA(J)                                                        00005380
      GO TO 370                                                         00005390
 1190 IBK1=4                                                            00005400
      XLS=XL                                                            00005410
      XL=XL/XNUFAC                                                      00005420
      DO 1200 J=1,K                                                     00005430
 1200 B(J)=BS(J)                                                        00005440
      GO TO 1320                                                        00005450
 1210 IF (PHI.LE.PHIZ) GO TO 1260                                       00005460
      XL=XLS                                                            00005470
      IBK1=3                                                            00005480
 1220 XL=XL*XNUFAC                                                      00005490
 1230 DO 1240 J=1,K                                                     00005500
 1240 B(J)=BS(J)                                                        00005510
      GO TO 1320                                                        00005520
 1250 PHIT4=PHI                                                         00005530
C                   WE NOW HAVE PHI(XNUFAC*LAMBDA)                     00005540
      IF (PHIT4.GT.PHIZ) GO TO 1280                                     00005550
 1260 DO 1270 J=1,K                                                     00005560
 1270 BS(J)=B(J)                                                        00005570
      GO TO 370                                                         00005580
 1280 IF (GAMMA.GE.GAMCR) GO TO 1220                                    00005590
      XKDB=XKDB/2.E0                                                    00005600
      DO 1290 J=1,K                                                     00005610
      IF(ABS(DB(J)/(ABS(B(J))+TAU)).GE.E) GO TO 1230                    00005620
 1290 CONTINUE                                                          00005630
      DO 1300 J=1,K                                                     00005640
 1300 B(J)=BS(J)                                                        00005650
      MAXITR=MAXITR-1                                                   00005660
      WRITE (6,2740)                                                    00005670
      IF (IFSS1.NE.1) GO TO 1310                                        00005680
      WRITE (16,2740)                                                   00005690
 1310 CONTINUE                                                          00005700
      GO TO 1670                                                        00005710
```

```
C                                                                       00005720
C     ..................................................                00005730
C                  SET UP FOR MATRIX INVERSION                          00005740
 1320 IF(IBKT-1) 1330,1340,1330                                         00005750
 1330 READ (13) A                                                       00005760
      REWIND 13                                                         00005770
      GO TO 1360                                                        00005780
 1340 DO 1350 II=1,K                                                    00005790
      III=II+10                                                         00005800
      DO 1350 JJ=1,K                                                    00005810
 1350 A(II,JJ)=A(III,JJ)                                                00005820
 1360 DO 1370 I=1,K                                                     00005830
 1370 A(I,I)=A(I,I)+XL                                                  00005840
C                  GET INVERSE OF A AND SOLVE FOR DB(J)S                00005850
      IBKM=1                                                            00005860
C     ..................................................                00005870
C                  THIS IS THE MATRIX INVERSION ROUTINE                 00005880
C                  K IS THE SIZE OF THE MATRIX                          00005890
 1380 IF(K.EQ.1) GO TO 1390                                             00005900
      CALL GJR (A,K,ZETA,MSING)                                         00005910
      IF(MSING-1) 1400,1400,1381                                        00005920
1381  CALL ERRMSG(20HSINGULAR MATRIX.....,4,6,16)                       00005930
C--SPECIAL CASE, K=1                                                    00005940
 1390 A(1,1)=1.0/A(1,1)                                                 00005950
 1400 IF(IBKM-1) 1410,1410,1840                                         00005960
C                  END OF MATRIX INVERSION, SOLVE FOR DB(J)             00005970
 1410 DO 1430 I=1,K                                                     00005980
      DB(I)=0.E0                                                        00005990
      DO 1420 J=1,K                                                     00006000
 1420 DB(I)=A(I,J)*G(J)+DB(I)                                           00006010
 1430 DB(I)=XKDB*DB(I)                                                  00006020
      XLL=0.E0                                                          00006030
      DTG=0.E0                                                          00006040
      GTG=0.E0                                                          00006050
      DO 1440 J=1,K                                                     00006060
      DB(J)=DB(J)/SA(J)                                                 00006070
      DTG=DTG+DB(J)*G(J)                                                00006080
      GTG=GTG+G(J)**2                                                   00006090
      B(J)=B(J)+DB(J)                                                   00006100
 1440 XLL=XLL+DB(J)*DB(J)                                               00006110
      KIP=K-IP                                                          00006120
      IF (KIP.EQ.1) GO TO 1480                                          00006130
      CGAM=DTG/SQRT(XLL*GTG)                                            00006140
      JGAM=1                                                            00006150
      IF (CGAM.GT.0.E0) GO TO 1450                                      00006160
      CGAM=ABS(CGAM)                                                    00006170
      JGAM=2                                                            00006180
 1450 GAMMA=57.2957795E0*(1.5707288E0+CGAM*(-0.2121144E0                00006190
     1+CGAM*(0.074261E0-CGAM*                                           00006200
     2.0187293E0)))*SQRT(1.0E0-CGAM)                                    00006210
      IF(JGAM-1) 1460,1490,1460                                         00006220
 1460 GAMMA=180.E0-GAMMA                                                00006230
```

```
      IF (XL.LT.1.0E0) GO TO 1490                                       00006240
      WRITE (6,2670) XL,GAMMA                                           00006250
      IF (IFSS1.NE.1) GO TO 1470                                        00006260
      WRITE (16,2670) XL,GAMMA                                          00006270
 1470 CONTINUE                                                          00006280
      GO TO 1670                                                        00006290
 1480 GAMMA=0.E0                                                        00006300
 1490 XLL=SQRT(XLL)                                                     00006310
      IBK2=1                                                            00006320
      GO TO 1540                                                        00006330
 1500 IF (IFSS3.LE.0) GO TO 1530                                        00006340
      WRITE (6,2500) (DB(J),J=1,K)                                      00006350
      IF (IFSS1.NE.1) GO TO 1510                                        00006360
      WRITE (16,2500) (DB(J),J=1,K)                                     00006370
 1510 CONTINUE                                                          00006380
      WRITE (6,2510) PHI,XL,GAMMA,XLL                                   00006390
      IF (IFSS1.NE.1) GO TO 1520                                        00006400
      WRITE (16,2510) PHI,XL,GAMMA,XLL                                  00006410
 1520 CONTINUE                                                          00006420
C--PRESET XNUFAC--(IF MODLAM=1)                                         00006430
 1530 GO TO (1570,1150,1250,1210),IBK1                                  00006440
C                                                                       00006450
C     ..................................................................00006460
C                   CALCULATE PHI                                       00006470
 1540 I=1                                                               00006480
      DO 1550 JJ=1,K                                                    00006490
 1550 CALL UNSCAL(B(JJ),BINV(JJ),SCALEP)                                00006500
      PHI=0.E0                                                          00006510
      IWHER=2                                                           00006520
      GO TO 30                                                          00006530
 1560 PHI=PHI+WT(I)*(Y(I)-F)**2                                         00006540
      I=I+1                                                             00006550
      IF (I.LE.N) GO TO 30                                              00006560
      GO TO (1500,2290,1770,2200,2220,2240),IBK2                        00006570
C=========================================================              00006580
C--DETERMINE AN EFFECTIVE MARQUARDT LAMBDA FACTOR (XNUFAC)              00006590
C  BASED ON HISTORY OF SUM OF SQUARES STORED IN LATEST SS(4)--           00006600
 1570 IF(MODLAM.EQ.0) GO TO 1050                                        00006610
      SS(ISS)=PHI                                                       00006620
      INU0=INU                                                          00006630
      GO TO (1590,1580,1600,1610),ISS                                   00006640
C--MACHINE FAILURE IF ISS.GT.4 OR ISS.LT.1                              00006650
C--   STOP 4                                                            00006660
 1580 IS1=0                                                             00006670
      IF(SS(2).GT.SS(1)) IS1=1                                          00006680
 1590 ISS=ISS+1                                                         00006690
      GO TO 1660                                                        00006700
 1600 IS2=0                                                             00006710
      IF(SS(3).GT.SS(2)) IS2=1                                          00006720
      IF(IS1.EQ.IS2) GO TO 1590                                         00006730
      INU=INU0-1                                                        00006740
      GO TO 1590                                                        00006750
```

```
 1610 IS3=0                                                              00006760
      IF(SS(4).GT.SS(3)) IS3=1                                           00006770
      IF(IS1.EQ.IS2.AND.IS3.EQ.IS2) GO TO 1620                           00006780
      IF(IS1.EQ.0.AND.IS2.EQ.0.AND.IS3.EQ.1) GO TO 1640                  00006790
      IF(IS1.EQ.1.AND.IS2.EQ.0.AND.IS3.EQ.1) GO TO 1640                  00006800
      IF(IS1.EQ.1.AND.IS2.EQ.1.AND.IS3.EQ.0) GO TO 1640                  00006810
      GO TO 1650                                                         00006820
 1620 IF(IS1.EQ.0) GO TO 1630                                            00006830
      IF(INU0.GE.3) GO TO 1650                                           00006840
      INU=3                                                              00006850
      GO TO 1650                                                         00006860
 1630 IF(INU0.GE.5) GO TO 1650                                           00006870
      INU=INU0+1                                                         00006880
      GO TO 1650                                                         00006890
 1640 IF(INU0.LE.1) GO TO 1650                                           00006900
      INU=INU0-1                                                         00006910
 1650 IS1=IS2                                                            00006920
      IS2=IS3                                                            00006930
      SS(3)=SS(4)                                                        00006940
 1660 XNUFAC=XNU(INU)                                                    00006950
      GO TO 1050                                                         00006960
C                                                                        00006970
C                                                                        00006980
C     ..................................................................00006990
C                  THIS IS THE CONFIDENCE LIMIT CALCULATION              00007000
 1670 ITR=MAXITR-IWS4+1                                                  00007010
      WRITE(6,1680) ITR                                                  00007020
 1680 FORMAT(1X,I4,11H ITERATIONS)                                       00007030
      IF(IFSS1.EQ.1) WRITE(16,1680) ITR                                  00007040
      DO 1690 J=1,K                                                      00007050
      CALL UNSCAL(BS(J),BINV(J),SCALEP)                                  00007060
      BS(J)=BINV(J)                                                      00007070
 1690 B(J)=BS(J)                                                         00007080
      WRITE(6,2520) TITLE                                                00007090
      IF (IFSS1.NE.1) GO TO 1700                                         00007100
      WRITE(16,2520) TITLE                                               00007110
 1700 CONTINUE                                                           00007120
      IBKA=2                                                             00007130
C--UNSCALE BOTH PARAMETER AND OBSERVATION SPACES PRIOR                   00007140
C  TO FINAL STATISTICS ON LAST INTERATION--AND WHERE                     00007150
C  IBKA=2, IFSS3=0..                                                     00007160
C               THIS WILL PRINT OBS,CAL,RES,ETC.                         00007170
C               AND SAVE UNSCALED PARTIALS ON FILE FILE13..              00007180
      IF(IPRT.GE.0) WRITE(6,1710) (BINV(J),J=1,K)                        00007190
 1710 FORMAT(/28H -FINAL UNSCALED PARAMETERS-/(12X,4E17.8))              00007200
      IF(IFSS1.EQ.1.AND.IPRT.GE.0) WRITE(16,1710) (BINV(J),J=1,K)        00007210
      IF(SCALEY.EQ.0) GO TO 1760                                         00007220
      DO 1750 I=1,N                                                      00007230
      IF(SCALEY.NE.1) GO TO 1730                                         00007240
 1720 Y(I)=EXP(Y(I))                                                     00007250
      GO TO 1750                                                         00007260
 1730 IF(IPRNT.LE.1) GO TO 1740                                          00007270
```

```
      IF(ABS(X(I,IPRNT)).EQ.1.0) GO TO 1720                          00007280
 1740 Y(I)=SINH(Y(I))                                                00007290
 1750 CONTINUE                                                       00007300
 1760 LSCALP=SCALEP                                                  00007310
      LSCALY=SCALEY                                                  00007320
      SCALEP=0                                                       00007330
      SCALEY=0                                                       00007340
      GO TO 370                                                      00007350
 1770 CONTINUE                                                       00007360
 1780 WS=N-K+IP                                                      00007370
      IF(N.GT.K) SE=SQRT(PHI/WS)                                     00007380
      PHIZ=PHI                                                       00007390
      WRITE (6,2490) PHIZ,SE,XL                                      00007400
      IF (IFSS1.NE.1) GO TO 1790                                     00007410
      WRITE (16,2490) PHIZ,SE,XL                                     00007420
C                                                                    00007430
C                    WE NOW HAVE MATRIX A                            00007440
 1790 IF(IBKT-1) 1800,1810,1800                                      00007450
 1800 WRITE (13) A                                                   00007460
      REWIND 13                                                      00007470
      GO TO 1830                                                     00007480
 1810 DO 1820 II=1,K                                                 00007490
      III=II+10                                                      00007500
      DO 1820 JJ=1,K                                                 00007510
 1820 A(III,JJ)=A(II,JJ)                                             00007520
 1830 IBKM=2                                                         00007530
      GO TO 1380                                                     00007540
C                                                                    00007550
C                    WE NOW HAVE C = A INVERSE                       00007560
 1840 DO 1850 J=1,K                                                  00007570
      IF (A(J,J).LT.0.E0) GO TO 1860                                 00007580
 1850 SA(J)=SQRT(A(J,J))                                             00007590
      GO TO 1870                                                     00007600
 1860 IBOUT=1                                                        00007610
 1870 KST=-4                                                         00007620
      IF (IFSS1.NE.1) GO TO 1880                                     00007630
      WRITE (16,2600)                                                00007640
 1880 KST=KST+5                                                      00007650
      KEND=KST+4                                                     00007660
      IF (KEND.LT.K) GO TO 1890                                      00007670
      KEND=K                                                         00007680
 1890 DO 1910 I=1,K                                                  00007690
      IF (IFSS1.NE.1) GO TO 1900                                     00007700
      WRITE (16,2620) I,(A(I,J),J=KST,KEND)                          00007710
 1900 CONTINUE                                                       00007720
 1910 CONTINUE                                                       00007730
      IF (KEND.LT.K) GO TO 1880                                      00007740
      IF (IBOUT.EQ.0) GO TO 1920                                     00007750
      WRITE (6,2760)                                                 00007760
      IF (IFSS1.NE.1) GO TO 220                                      00007770
      WRITE (16,2760)                                                00007780
      GO TO 220                                                      00007790
```

```
 1920 DO 1940 I=1,K                                                    00007800
      DO 1940 J=1,K                                                    00007810
      WS=SA(I)*SA(J)                                                   00007820
      IF (WS.GT.0.E0) GO TO 1930                                       00007830
      A(I,J)=0.E0                                                      00007840
      GO TO 1940                                                       00007850
 1930 A(I,J)=A(I,J)/WS                                                 00007860
 1940 CONTINUE                                                         00007870
      DO 1950 J=1,K                                                    00007880
 1950 A(J,J)=1.E0                                                      00007890
      IF (IFSS1.NE.1) GO TO 1960                                       00007900
      WRITE (16,2610)                                                  00007910
 1960 CONTINUE                                                         00007920
      KST=-9                                                           00007930
 1970 KST=KST+10                                                       00007940
      KEND=KST+9                                                       00007950
      IF (KEND.LT.K) GO TO 1980                                        00007960
      KEND=K                                                           00007970
 1980 DO 2000 I=1,K                                                    00007980
      IF (IFSS1.NE.1) GO TO 1990                                       00007990
      WRITE (16,2750) I,(A(I,J),J=KST,KEND)                            00008000
 1990 CONTINUE                                                         00008010
 2000 CONTINUE                                                         00008020
      IF (KEND.LT.K) GO TO 1970                                        00008030
C                   GET T*SE*SQRT(C(I,I))                             00008040
      DO 2010 J=1,K                                                    00008050
 2010 SA(J)=SE*SA(J)                                                   00008060
      IF(IBKT-1) 2020,2030,2020                                        00008070
 2020 READ (13) A                                                      00008080
      REWIND 13                                                        00008090
      GO TO 2050                                                       00008100
 2030 DO 2040 II=1,K                                                   00008110
      III=II+10                                                        00008120
      DO 2040 JJ=1,K                                                   00008130
 2040 A(II,JJ)=A(III,JJ)                                               00008140
 2050 CONTINUE                                                         00008150
      WRITE (6,2640)                                                   00008160
      IF (IFSS1.NE.1) GO TO 2060                                       00008170
      WRITE (16,2630)                                                  00008180
 2060 CONTINUE                                                         00008190
      WS=K-IP                                                          00008200
      DO 2120 J=1,K                                                    00008210
      IF (IP.LE.0) GO TO 2080                                          00008220
      DO 2070 I=1,IP                                                   00008230
      IF (J.EQ.IB(I)) GO TO 2100                                       00008240
 2070 CONTINUE                                                         00008250
C                                                                      00008260
C--COMPUTE STD.ERR, CONF. LIMITS, AND STD.ERR/PARM.                    00008270
C                                                                      00008280
 2080 HJTD=SQRT(WS*FF)*SA(J)                                           00008290
      STE=SA(J)                                                        00008300
      TWS=STE*T                                                        00008310
```

```
      OPL=BINV(J)-TWS                                                   00008320
      OPU=BINV(J)+TWS                                                   00008330
      SPL=BINV(J)-HJTD                                                  00008340
      SPU=BINV(J)+HJTD                                                  00008350
      HJTD=0.0                                                          00008360
      IF(BINV(J).NE.0.0) HJTD=STE/BINV(J)                               00008370
      WRITE (6,2720) J,STE,OPL,OPU,HJTD                                 00008380
      IF (IFSS1.NE.1) GO TO 2090                                        00008390
      WRITE (16,2720) J,STE,OPL,OPU,SPL,SPU,HJTD                        00008400
 2090 CONTINUE                                                          00008410
      GO TO 2120                                                        00008420
 2100 WRITE (6,2570) J                                                  00008430
      IF (IFSS1.NE.1) GO TO 2110                                        00008440
      WRITE (16,2570) J                                                 00008450
 2110 CONTINUE                                                          00008460
 2120 CONTINUE                                                          00008470
C                    NONLINEAR CONFIDENCE LIMIT                        00008480
      IF (IWS6.EQ.1.OR.N.EQ.K) GO TO 220                                00008490
      WS=K-IP                                                           00008500
      WS1=N-K+IP                                                        00008510
      PKN=WS/WS1                                                        00008520
      PC=PHIZ*(1.E0+FF*PKN)                                             00008530
      WRITE (6,2650) PC                                                 00008540
      IF (IFSS1.NE.1) GO TO 2130                                        00008550
      WRITE (16,2650) PC                                                00008560
 2130 CONTINUE                                                          00008570
      WRITE (6,2660)                                                    00008580
      IF (IFSS1.NE.1) GO TO 2140                                        00008590
      WRITE (16,2660)                                                   00008600
 2140 CONTINUE                                                          00008610
      IFSS3=1                                                           00008620
C--       NON- DO LOOP J=1,K                                           00008630
C  (SINCE CONTROL JUMPS OUT AND BACK INSIDE LOOP)                       00008640
      J=1                                                               00008650
 2150 IBKP=1                                                            00008660
      DO 2160 JJ=1,K                                                    00008670
 2160 B(JJ)=BS(JJ)                                                      00008680
      IF (IP.LE.0) GO TO 2180                                           00008690
      DO 2170 JJ=1,IP                                                   00008700
      IF (J.EQ.IB(JJ)) GO TO 2380                                       00008710
 2170 CONTINUE                                                          00008720
 2180 DD=-1.E0                                                          00008730
      IBKN=1                                                            00008740
 2190 D=DD                                                              00008750
      B(J)=BS(J)+D*SA(J)                                                00008760
      IBK2=4                                                            00008770
      GO TO 1540                                                        00008780
 2200 PHI1=PHI                                                          00008790
      IF (PHI1.GE.PC) GO TO 2230                                        00008800
 2210 D=D+DD                                                            00008810
      IF (D/DD.GE.5.E0) GO TO 2420                                      00008820
      B(J)=BS(J)+D*SA(J)                                                00008830
```

```
      IBK2=5                                                                    00008840
      GO TO 1540                                                                00008850
 2220 PHID=PHI                                                                  00008860
      IF (PHID.LT.PC) GO TO 2210                                                00008870
      IF (PHID.GE.PC) GO TO 2250                                                00008880
 2230 D=D/2.E0                                                                  00008890
      IF (D/DD.LE..001E0) GO TO 2420                                            00008900
      B(J)=BS(J)+D*SA(J)                                                        00008910
      IBK2=6                                                                    00008920
      GO TO 1540                                                                00008930
 2240 PHID=PHI                                                                  00008940
      IF (PHID.GT.PC) GO TO 2230                                                00008950
 2250 XK1=PHIZ/D+PHI1/(1.E0-D)+PHID/(D*(D-1.E0))                                00008960
      XK2=-(PHIZ*(1.E0+D)/D+D/(1.E0-D)*PHI1+PHID/(D*(D-1.E0)))                  00008970
      XK3=PHIZ-PC                                                               00008980
      BC=(SQRT(XK2*XK2-4.E0*XK1*XK3)-XK2)/(2.E0*XK1)                            00008990
      IF(IBKN-1) 2260,2260,2270                                                 00009000
 2260 B(J)=BS(J)-SA(J)*BC                                                       00009010
      GO TO 2280                                                                00009020
 2270 B(J)=BS(J)+SA(J)*BC                                                       00009030
 2280 IBK2=2                                                                    00009040
      GO TO 1540                                                                00009050
 2290 IF(IBKN-1) 2300,2300,2310                                                 00009060
 2300 IBKN=2                                                                    00009070
      DD=1.E0                                                                   00009080
      BL=B(J)                                                                   00009090
      PL=PHI                                                                    00009100
      GO TO 2190                                                                00009110
 2310 BU=B(J)                                                                   00009120
      PU=PHI                                                                    00009130
      GO TO (2320,2340,2360,2400), IBKP                                         00009140
 2320 WRITE (6,2620) J,BL,PL,BU,PU                                              00009150
      IF (IFSS1.NE.1) GO TO 2330                                                00009160
      WRITE (16,2620) J,BL,PL,BU,PU                                             00009170
 2330 CONTINUE                                                                  00009180
      GO TO 2470                                                                00009190
 2340 WRITE (6,2590) J,BU,PU                                                    00009200
      IF (IFSS1.NE.1) GO TO 2350                                                00009210
      WRITE (16,2590) J,BU,PU                                                   00009220
 2350 CONTINUE                                                                  00009230
      GO TO 2470                                                                00009240
 2360 WRITE (6,2620) J,BL,PL                                                    00009250
      IF (IFSS1.NE.1) GO TO 2370                                                00009260
      WRITE (16,2620) J,BL,PL                                                   00009270
 2370 CONTINUE                                                                  00009280
      GO TO 2470                                                                00009290
 2380 WRITE (6,2570) J                                                          00009300
      IF (IFSS1.NE.1) GO TO 2390                                                00009310
      WRITE (16,2570) J                                                         00009320
 2390 CONTINUE                                                                  00009330
      GO TO 2470                                                                00009340
 2400 WRITE (6,2580) J                                                          00009350
```

```
      IF (IFSS1.NE.1) GO TO 2410                                        00009360
      WRITE (16,2580) J                                                 00009370
 2410 CONTINUE                                                          00009380
      GO TO 2470                                                        00009390
 2420 IF(IBKN-1) 2430,2430,2440                                         00009400
C                   DELETE LOWER PRINT                                  00009410
 2430 IBKP=2                                                            00009420
      GO TO 2290                                                        00009430
 2440 IF(IBKP-1) 2450,2450,2460                                         00009440
C                   DELETE UPPER PRINT                                  00009450
 2450 IBKP=3                                                            00009460
      GO TO 2290                                                        00009470
C                   LOWER IS ALREADY DELETED, SO DELETE BOTH            00009480
 2460 IBKP=4                                                            00009490
      GO TO 2290                                                        00009500
C--END OF NON- DO LOOP J=1,K                                            00009510
 2470 J=J+1                                                             00009520
      IF(J.LE.K) GO TO 2150                                             00009530
      GO TO 220                                                         00009540
C     ..................................................................00009550
 2480 FORMAT(18A4)                                                      00009560
 2490   FORMAT(/13X,4H PHI,14X,4H S E,9X,7H LAMBDA/5X,2E18.8,E13.3)     00009570
 2500 FORMAT (/12H INCREMENTS ,4E17.8/(12X,4E17.8))                     00009580
 2510 FORMAT (13X,4H PHI10X,7H LAMBDA6X,7H GAMMA 6X,7H LENGTH/5X,E18.8,300009590
     1E13.3)                                                            00009600
 2520 FORMAT(16H1M A R Q R T  --,5X,16A5)                               00009610
 2530   FORMAT(/5H N = ,I4,8X,4HK = ,I3,9X,5HIP = ,I3,8X,4HM = ,I2,10X, 00009620
     1 6HGAMCR=,E9.3/5H DEL=,E10.3,2X,9HMODLAM = ,I1,6X,3HFF=,E10.3,3X, 00009630
     2 2HT=,E10.3,4X,2HE=,E10.3/5H TAU=,E10.3,2X,3HXL=,E10.3,3X,        00009640
     3 5HZETA=,E10.3,8H IALT = ,I2,7X,8HISTOP = ,I1/7H IWT = ,I1,9X,    00009650
     4 7HIDER = ,I1,8X,7HIPRT = ,I2,7X,8HNITER = ,I4,4X,7HINON = ,I1/   00009660
     5 8H IOUT = ,I2,7X,                                                00009670
     6 8HNPRNT = ,I1,7X,9HSCALEP = ,I1,6X,9HSCALEY = ,I1/)              00009680
 2540 FORMAT (/12H PARAMETERS ,4E17.8/(12X,4E17.8))                     00009690
 2550   FORMAT(3X,1HI,4X,8HOBS.Y(I),6X,3HCAL,11X,3HRES,8X,8H%RES.ERR,6X,00009700
     1 6HX(I,1),8X,6HX(I,2),8X,6HX(I,3),8X,6HX(I,4),8X,6HX(I,5))        00009710
 2560   FORMAT(/1X,4HITER,8X,4H PHI,14X,4H S E,11X,7H LENGTH,6X,        00009720
     1 7H GAMMA ,6X,7H LAMBDA/1X,I4,2E18.8,3E13.3)                      00009730
 2570 FORMAT (2X,I3,20H PARAMETER NOT USED )                            00009740
 2580 FORMAT (2X,I3,12H NONE FOUND )                                    00009750
 2590 FORMAT (2X,I3,36X,2E18.8)                                         00009760
 2600 FORMAT (1H /13H PTP INVERSE )                                     00009770
 2610 FORMAT (1H /30H PARAMETER CORRELATION MATRIX )                    00009780
 2620 FORMAT (2X,I3,5E18.8)                                             00009790
 2630   FORMAT(//4X,13HPARAMETER STD,17X,15HONE - PARAMETER,21X,        00009800
     1 14H SUPPORT PLANE/11X,6H ERROR,12X,6H LOWER,12X,6H UPPER,12X,    00009810
     2 6H LOWER,12X,6H UPPER,10X,14HSTD.ERROR/PARM)                     00009820
 2640   FORMAT(//4X,13HPARAMETER STD,17X,15HONE - PARAMETER/11X,        00009830
     1 6H ERROR,12X,6H LOWER,12X,6H UPPER,10X,14HSTD.ERROR/PARM)        00009840
 2650 FORMAT (//30H  NONLINEAR CONFIDENCE LIMITS //13H PHI CRITICAL,    00009850
     1 E15.8)                                                           00009860
 2660 FORMAT (1H /6H  PARA6X,8H LOWER B8X,10H LOWER PHI10X,8H UPPER B8X,00009870
```

```
     110H UPPER PHI)                                                    00009880
 2670 FORMAT (/19H -GAMMA LAMBDA TEST,5X,2E13.3)                        00009890
 2680 FORMAT (/15H -EPSILON TEST )                                      00009900
 2690 FORMAT (/12H -FORCE OFF )                                         00009910
 2700 FORMAT(1X,I3,2E14.6,E11.3,6E14.6)                                 00009920
 2720 FORMAT (2X,I3,6E18.8)                                             00009930
 2730 FORMAT (1H )                                                      00009940
 2740 FORMAT (/20H -GAMMA EPSILON TEST)                                 00009950
 2750 FORMAT (3X,I5,2X,10F10.4)                                         00009960
 2760 FORMAT (/27H  NEGATIVE DIAGONAL ELEMENT)                          00009970
      END                                                               00009980

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

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

      SUBROUTINE UNSCAL(BIN,BOUT,SCALEP)                                00010650
C// MODIFIED TO TRAP ERRORS >10**38 ON MULTICS                         00010660
C--UNSCALE PARMETER BIN TO BOUT VIA SCALEP                             00010670
      INTEGER SCALEP                                                    00010680
      IF(SCALEP-1) 10,20,30                                             00010690
   10 BOUT=BIN                                                          00010700
      GO TO 40                                                          00010710
   20 IF(BIN.GT.88.028) GO TO 99                                        00010720
      BOUT= EXP_(BIN)                                                   00010730
      GO TO 40                                                          00010740
   30 BOUT= SINH(BIN)                                                   00010750
   40 RETURN                                                            00010760
   99 WRITE(6,699) BIN                                                  00010770
      WRITE(16,699) BIN                                                 00010780
699   FORMAT('0"UNSCAL" ARG=',E16.8,' >88.028 FOR EXP_() ON MULTICS'/   00010790
     & ' --CHECK ALL $PARMS AND DATA --IF OK, THEN--'/                   00010800
     & ' --TRY RESTARTING WITH DIFFERENT SCALING OPTION(S) --OR--'/      00010810
     & ' --RESTART WITH BETTER "GUESSED" STARTING PARAMETERS.')         00010820
      CALL CLOSE_FILE('-ALL')                                           00010830
      STOP                                                              00010840
      END                                                               00010850

      REAL FUNCTION ASINH(X)                                            00010860
C--INVERSE HYPERBOLIC SIN FUNCTION                                      00010870
C                                                                       00010880
```

```
      REAL*8 X2                                                          00010890
      X2=X                                                               00010900
      ASINH=DLOG(X2+DSQRT(X2*X2+1.0D0))                                  00010910
      RETURN                                                             00010920
      END                                                                00010930

      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                    00010940
C--ERROR MESSAGE WRITE ROUTINE AND STOP, WHERE--                         00010950
C                                                                        00010960
C       MSG=    ANY MULTIPLE OF 5 CHARACTERS--MAX. OF 120               00010970
C               (USE NH-------- FORM FOR ANSI COMPATIBILITY)            00010980
C       M5=     NO.CHARS IN MSG/5 (REMAINDER MUST BE 0) 1.LE.M5.LE.24   00010990
C       I6=     1ST UNIT  FOR WRITE(I6, ) MSG -- USUALLY I6=6 FOR LPT.  00011000
C               IF I6.LE.0 UNIT I6 IGNORED.                             00011010
C       I9=     2ND UNIT  FOR WRITE(I9, ) MSG --                        00011020
C               IF I9.LE.0, UNIT I9 IGNORED.                            00011030
C--MESSAGE WRITTEN IN FORM--                                             00011040
C       /ERROR--MSG HERE                                                00011050
C                                                                        00011060
      DIMENSION MSG(30)                                                  00011070
      J=5*M5                                                             00011080
      K=J/4+MOD(J,4)                                                     00011090
      IF(I6.GT.0) WRITE(I6,10) (MSG(I),I=1,K)                            00011100
   10 FORMAT(/8H ERROR--,30A4)                                           00011110
      IF(I9.GT.0) WRITE(I9,10) (MSG(I),I=1,K)                            00011120
      CALL CLOSE_FILE('-ALL')                                            00011130
C                                                                        00011140
      STOP                                                               00011150
      END                                                                00011160

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

```
      GO TO 40                                                          00011390
   30 RAD=PI+PV                                                         00011400
   40 AMP=SQRT(ZR*ZR+ZI*ZI)                                             00011410
      PHZ180=57.29577951*RAD                                            00011420
      IF(PHZ180.GT.180.0) PHZ180=PHZ180-360.0                           00011430
      RETURN                                                            00011440
      END                                                               00011450

      SUBROUTINE RECUR1(G,V1,F1)                                        00011460
C--BACKWARD RECURRENCE FOR COMPLEX V1,F1 GIVEN REAL*4 ARGUMENT G AND:   00011470
COMMON/MODEL/ PARAMETERS:                                               00011480
C     K(10) = NORMALIZED CONDUCTIVITY ARRAY (M VALUES,WHERE K(1)=1.0).  00011490
C     D(9)  = LAYER THICKNESS ARRAY (M-1 VALUES) D=2*THICKNESS/DEL.     00011500
C     M     = NUMBER LAYERS (M.GE.1.AND.M.LE.10)                        00011510
C             SPECIAL CASE WHEN M=1 (HOMOGENEOUS--D IGNORED)            00011520
C                                                                       00011530
C--NOTE: G,K,D ARE REAL*4                                               00011540
C                                                                       00011550
C                                                                       00011560
      COMMON/MODEL/K,D,M                                                00011570
      REAL*4 K(10),D(9)                                                 00011580
      COMPLEX C,VM,V1,F1,EVD,ONE                                        00011590
      DATA ONE/(1.0,0.0)/                                               00011600
      F1=ONE                                                            00011610
      G2=G*G                                                            00011620
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                      00011630
      IF(M.EQ.1) GO TO 2                                                00011640
      J=M-1                                                             00011650
    1 V1=CSQRT(CMPLX(G2,2.0*K(J)))                                      00011660
      EVD=CEXP(-V1*D(J))                                                00011670
      C=(ONE-EVD)/(ONE+EVD)                                             00011680
      F1=(VM*F1+V1*C)/(V1+VM*F1*C)                                      00011690
      IF(J.EQ.1) GO TO 3                                                00011700
      J=J-1                                                             00011710
      VM=V1                                                             00011720
      GO TO 1                                                           00011730
    2 V1=VM                                                             00011740
    3 RETURN                                                            00011750
      END                                                               00011760

      SUBROUTINE RECURF(G,DEL,SIG1,V1,F1,PF1,JJ)                        00011770
C--GET PF1=PARTIAL OF F1 W/R PARM. JJ, EVALUATED AT                     00011780
C  THE GIVEN G,DEL, AND SIG1 (OTHER MODEL PARMS IN COMMON/MODEL/)       00011790
C  ALSO GIVEN ARE V1,F1 AS IN RECUR1.                                   00011800
C                                                                       00011810
      IMPLICIT COMPLEX (A-H,O-Z)                                        00011820
      REAL   K,D,G,G2,DEL,SIG1                                          00011830
      COMMON/MODEL/K,D,M                                                00011840
      DIMENSION K(10),D(9)                                              00011850
      DATA ONE,ZERO,CI/(1.0,0.0),(0.0,0.0),(0.0,1.0)/                   00011860
      TWODEL=CMPLX(2.0/DEL,0.0)                                         00011870
      JJM=JJ-M                                                          00011880
```

```
      FM=ONE                                                          00011890
      PF1=ZERO                                                        00011900
   30 G2=G*G                                                          00011910
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                    00011920
   50 IF(M.EQ.1) GO TO 150                                            00011930
C--INITIALIZE PARTIAL INDEX J=M-1 (NUM. INDEX)                        00011940
      J=M-1                                                           00011950
C--LOOP ON J INDEX                                                    00011960
   70 V1=CSQRT(CMPLX(G2,2.0*K(J)))                                    00011970
      EVD=CEXP(-V1*D(J))                                              00011980
   90 EVD1=ONE+EVD                                                    00011990
      E1=(ONE-EVD)/EVD1                                               00012000
      E11=ONE+E1                                                      00012010
      T=VM*FM                                                         00012020
      DEN=V1+T*E1                                                     00012030
      F1=(T+V1*E1)/DEN                                                00012040
      IF(JJ.LE.1) GO TO 100                                           00012050
C--RECUR FOR PF1 W/R DIST                                             00012060
      EMD1=ZERO                                                       00012070
      IF(JJM.EQ.J) EMD1=(TWODEL*V1*EVD*E11)/EVD1                      00012080
      PF1=((VM*PF1+V1*EMD1)-F1*VM*(FM*EMD1+E1*PF1))/DEN               00012090
      GO TO 140                                                       00012100
C--RECUR FOR PF1 W/R SIGMA                                            00012110
  100 VMS=ZERO                                                        00012120
      VMS1=ZERO                                                       00012130
      EMS1=ZERO                                                       00012140
      IF(JJ.EQ.1) GO TO 110                                           00012150
      IF(J+1.EQ.JJ) VMS=CI/(SIG1*VM)                                  00012160
      IF(J.EQ.JJ) VMS1=CI/(SIG1*V1)                                   00012170
      GO TO 120                                                       00012180
  110 IF(M.GT.1) VMS=-CI*K(J+1)/(SIG1*VM)                             00012190
      IF(J.GT.1) VMS1=-CI*K(J)/(SIG1*V1)                              00012200
  120 IF(JJ.NE.J) GO TO 130                                           00012210
      IF(J.EQ.1) EMS1=(EVD*V1*D(1)*E11)/(2.0*SIG1*EVD1)               00012220
      IF(J.GT.1) EMS1=(D(J)*EVD*VMS1*E11)/EVD1                        00012230
  130 PF1=((FM*VMS+VM*PF1+V1*EMS1+E1*VMS1)-F1*                        00012240
     1(VMS1+VM*(FM*EMS1+E1*PF1)+FM*E1*VMS))/DEN                       00012250
  140 IF(J.EQ.1) GO TO 180                                            00012260
      J=J-1                                                           00012270
      VM=V1                                                           00012280
      FM=F1                                                           00012290
      GO TO 70                                                        00012300
C--SPECIAL CASE M=1 (HOMOGENEOUS EARTH)                               00012310
  150 F1=FM                                                           00012320
      V1=VM                                                           00012330
      J=1                                                             00012340
      EVD=ZERO                                                        00012350
      GO TO 90                                                        00012360
  180 RETURN                                                          00012370
      END                                                             00012380
```

```
      SUBROUTINE RECURS(G,V1,F1)                                        00012390
      COMPLEX ESAV(80,9),VSAV(80,10)                                    00012400
      REAL GSAV(80)                                                     00012410
      COMMON/SAV/GSAV,ESAV,VSAV,ISAV,LSAV                               00012420
      COMMON/MODEL/K,D,M                                                00012430
      REAL   K(10),D(9)                                                 00012440
      COMPLEX C,VM,V1,F1,EVD,ONE,T                                      00012450
      DATA ONE/(1.0,0.0)/                                               00012460
      F1=ONE                                                            00012470
      G2=G*G                                                            00012480
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                      00012490
      NSAV=0                                                            00012500
      IF(ISAV.GT.0.AND.ISAV.LE.80) NSAV=1                               00012510
      IF(NSAV.NE.0) VSAV(ISAV,M)=VM                                     00012520
      IF(M.EQ.1) GO TO 30                                               00012530
      J=M-1                                                             00012540
   10 V1=CSQRT(CMPLX(G2,2.0*K(J)))                                      00012550
      EVD=CEXP(-V1*D(J))                                                00012560
      IF(NSAV.EQ.0) GO TO 20                                            00012570
      VSAV(ISAV,J)=V1                                                   00012580
      ESAV(ISAV,J)=EVD                                                  00012590
   20 C=(ONE-EVD)/(ONE+EVD)                                             00012600
      T=VM*F1                                                           00012610
      F1=(T+V1*C)/(V1+T*C)                                              00012620
      IF(J.EQ.1) GO TO 40                                               00012630
      J=J-1                                                             00012640
      VM=V1                                                             00012650
      GO TO 10                                                          00012660
   30 V1=VM                                                             00012670
   40 RETURN                                                            00012680
      END                                                               00012690

      COMPLEX FUNCTION F3(G)                                            00012700
      COMPLEX V1,F1,C,ONE                                               00012710
      DATA ONE/(1.0,0.0)/                                               00012720
      CALL RECUR1(G,V1,F1)                                              00012730
      C=G                                                               00012740
      F3=(V1*C*(ONE-F1))/((C+V1*F1)*(C+V1))                             00012750
      RETURN                                                            00012760
      END                                                               00012770

      COMPLEX FUNCTION F4(G)                                            00012780
C--F4=G*F3(G)--SEE SUBPROGRAM F3.                                       00012790
      COMPLEX F3                                                        00012800
      F4=G*F3(G)                                                        00012810
      RETURN                                                            00012820
      END                                                               00012830

      COMPLEX FUNCTION GF4(G)                                           00012840
C--KERNEL G*F4(G)                                                       00012850
      COMPLEX F4                                                        00012860
      GF4=G*F4(G)                                                       00012870
```

```
      RETURN                                                            00012880
      END                                                               00012890

      COMPLEX FUNCTION FINITE(FUNC,BFIN)                                00012900
C--COMPUTE FINITE INTEGRAL OVER (-L,L) OF COMPLEX FIELD FUNCTION        00012910
C  BY LAG-CONVOLUTION AND QUINTIC SPLINE INTERPOLATION PRIOR TO         00012920
C  AUTOMATIC INTEGRATION BY SUBR 'CANC4'.                               00012930
C  'FINITE' CALLS 'FUNC' (WHICH CALLS 'ZLAGH1 OR ZLAGH0'), 'QUINT', AND 00012940
C  'CANC4' (WHICH CALLS 'FUNINT' AND 'QPOINT').                         00012950
C                                                                       00012960
C  PARAMETERS:                                                          00012970
C                                                                       00012980
C  FUNC =  EXTERNAL DECLARED COMPLEX FUNCTION DEFINING THE DIPOLE FIELD 00012990
C          FUNCTION WITH CALLING SEQ: FUNC(B,NEW,R), WHERE              00013000
C           B = ANY IND. NO.                                            00013010
C           NEW = 1 FIRST TIME, 0 OTHERWISE (REF: ZLAGH1 OR ZLAGH0)     00013020
C           R = B*DEL FOR ANY B OR DEL (SKIN DEPTH).                    00013030
C  BFIN =  FIXED IND. NO. FOR THE FINITE INTEGRAL (BFIN.GT.0).          00013040
C                                                                       00013050
C--COMMON PARMAMETERS (INPUT) REQUIRED:                                 00013060
C                                                                       00013070
C  HAKTOL = REQUESTED HANKEL TRANSFORM (ZLAG) TOLERANCE.                00013080
C           USE HAKTOL.LE.1.E-6*EPS, EPS=ACTUAL HANKEL REL. ERROR.      00013090
C  FINTOL = REQUESTED FINITE INTEGRAL (CANC4) TOLERANCE.                00013100
C           USE FINTOL.LE.1.E-3*EP, EP=ACTUAL FINITE REL. ERROR.        00013110
C  INTYPE = INTEGRATION TYPE FOR CANC4 (NORMALLY, INTYPE=2 OR 4).       00013120
C  NFIN   = 1 TO USE 1-PASS ZLAG, =2 FOR 2-PASS, ETC.                   00013130
C           NOTE: NFIN.GT.1 TAKES 'NFIN TIMES' AS LONG TO RUN, BUT      00013140
C           WILL GIVE ADDITIONAL ACCURACY, IF NEEDED.                   00013150
C  MEV    = MAX. FUNCT EVAL"S FOR CANC4 (NORMALLY MEV>300)              00013160
C  R1     = MAX SPACING FROM WIRE END TO RECEIVER POINT (XX,YY)         00013170
C         = SQRT((XX+L)**2+YY*YY)                                       00013180
C  R2     = MIN SPACING FROM WIRE END TO RECEIVER POINT (XX,YY)         00013190
C         = SQRT((XX-L)**2+YY*YY)                                       00013200
C  R0     = SPACING FROM WIRE CENTER TO RECEIVER POINT (XX,YY)          00013210
C         = SQRT(XX*XX+YY*YY)                                           00013220
C  D(9)   = THICKNESS OF M-LAYERS IN MODEL. (METERS)                    00013230
C  K(10)  = CONDUCTIVITY RATIO SIG(I)/SIG(1)                            00013240
C         FOR I=1,M                                                     00013250
C  M      = NO. LAYERS IN MODEL (M.GE.1.AND.M.LT.10)                    00013260
C                                                                       00013270
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW           00013280
      COMMON/SPLN80/FDR(80),AR(80),BR(80),CR(80),DR(80),ER(80),         00013290
     & FDI(80),AI(80),BI(80),CI(80),DI(80),EI(80),RLM1,DELRLM,NB        00013300
      COMMON/FIN/R1,R2,R0,L,SIG1,X,Y                                    00013310
      COMMON/THICK/D(9)                                                 00013320
      COMMON/MODEL/K(10),DD(9),M                                        00013330
      COMMON/CONST/DEL,DEL2,Z2DEL3                                      00013340
      REAL L,K                                                          00013350
      COMPLEX FUNC,ESUM,FD,FUNINT,CANC4,Z2DEL3                          00013360
      EXTERNAL FUNINT                                                   00013370
C   ISIZE IS THE MAXIMUM POSSIBLE NUMBER OF NODES IN QUINTIC SPLINE.    00013380
```

```
      DATA ISIZE/80/                                                    00013390
      DEL=R0/BFIN                                                       00013400
      DEL2=DEL*DEL                                                      00013410
      Z2DEL3=CMPLX(0.0,2./(DEL2*DEL))                                   00013420
      M1=M-1                                                            00013430
      DO 1 I=1,M1                                                       00013440
    1 DD(I)=2.*D(I)/DEL                                                 00013450
      BMAX=R1/DEL                                                       00013460
      BMIN=R2/DEL                                                       00013470
      IF(X.LE.L) BMIN=Y/DEL                                             00013480
      NB=AINT(5.*ALOG(BMAX/BMIN))+2                                     00013490
      NB=MAX0(NB,3)                                                     00013500
      X0=ALOG(BMIN)+NB*0.2                                              00013510
      NB=NB+3                                                           00013520
      NRMAX=ISIZE/NB                                                    00013530
      IF(NFIN.LE.NRMAX) GO TO 3                                         00013540
      IF(NRMAX.GT.0.0) GO TO 2                                          00013550
      PRINT, 'ERROR IN FINITE: INSUFFICIENT SPLINE NODES'               00013560
      STOP                                                              00013570
    2 NFIN=NRMAX                                                        00013580
      PRINT, 'ERROR IN FINITE: NFIN TOO LARGE, RESET TO ',NFIN          00013590
3     DELRLM=.2/FLOAT(NFIN)                                             00013600
      X0=X0-DELRLM                                                      00013610
      DO 5 ITIME=1,NFIN                                                 00013620
      NEW=1                                                             00013630
      X0=X0+DELRLM                                                      00013640
      DO 5 J=1,NB                                                       00013650
      I=(NB+1)-J                                                        00013660
      I=NFIN*(I-1)+ITIME                                                00013670
      XX=X0-0.2*J                                                       00013680
      BM=EXP(XX)                                                        00013690
      RM=BM*DEL                                                         00013700
      IF(I.EQ.1) RLM1=ALOG(RM)                                          00013710
      FD=FUNC(BM,NEW,RM)                                                00013720
      FDR(I)=REAL(FD)                                                   00013730
      FDI(I)=AIMAG(FD)                                                  00013740
    5 NEW=0                                                             00013750
      NB=NFIN*NB                                                        00013760
      CALL QUINT(NB,FDR,AR,BR,CR,DR,ER)                                 00013770
      CALL QUINT(NB,FDI,AI,BI,CI,DI,EI)                                 00013780
      IF(X.LT.L) GO TO 8                                                00013790
      FINITE=CANC4(X-L,X+L,FINTOL,NEV,INTYPE,FUNINT,MEV,ESUM)           00013800
      GO TO 10                                                          00013810
    8 FINITE=2.*CANC4(0.,ABS(X-L),FINTOL,NEV,INTYPE,FUNINT,MEV,ESUM)    00013820
      IF(X.EQ.0.0) GO TO 10                                             00013830
      FINITE=FINITE+CANC4(ABS(X-L),X+L,FINTOL,NEV,INTYPE,FUNINT,MEV,    00013840
     & ESUM)                                                            00013850
   10 RETURN                                                            00013860
      END                                                               00013870

       COMPLEX FUNCTION FINHX(B)                                        00013880
C-- HX FIELD FOR FINITE WIRE (L.GT.0) AND GROUND (H=0) CASE.            00013890
```

```
      COMMON/FIN/R1,R2,R,L,SIG1,X,Y                                    00013900
      COMMON/THICK/D(9)                                                00013910
      COMMON/MODEL/K(10),DD(9),M                                       00013920
      REAL L,K                                                         00013930
      DOUBLE PRECISION B8                                              00013940
      COMPLEX F31,F32,ZIK1,ZIK2                                        00013950
      DEL=R/B                                                          00013960
      FINHX=CMPLX(0.0,0.0)                                             00013970
      IF(M.EQ.1) GO TO 12                                              00013980
      B1=R1/DEL                                                        00013990
      B2=R2/DEL                                                        00014000
      M1=M-1                                                           00014010
      DO 1 I=1,M1                                                      00014020
1     DD(I)=2.*D(I)/DEL                                                00014030
      CALL FINF3(B1,B2,F31,F32)                                        00014040
      FINHX=CMPLX(1./DEL,0.0)*(F31/R1-F32/R2)                          00014050
12    B8=0.7071067811865475D0/DEL                                      00014060
      CALL IK1(B8*DBLE(R1),ZIK1)                                       00014070
      CALL IK1(B8*DBLE(R2),ZIK2)                                       00014080
      FINHX=FINHX+(ZIK1/R1**2-ZIK2/R2**2)                              00014090
      FINHX=-FINHX*CMPLX(Y*2.0,0.0)                                    00014100
      RETURN                                                           00014110
      END                                                              00014120

      COMPLEX FUNCTION FINHY(B)                                        00014130
C-- HY FIELD FOR FINITE WIRE (L.GT.0) AND GROUND (H=0) CASE.           00014140
      EXTERNAL ZHY                                                     00014150
      COMMON/FIN/R1,R2,R,L,SIG1,X,Y                                    00014160
      COMMON/CONST/DEL,DEL2,Z                                          00014170
      COMMON/PASS/FINHX                                                00014180
      COMMON/MODEL/RK(10),DD(9),M                                      00014190
      COMPLEX FINITE,F31,F32,FINHX,Z,ZIK1,ZIK2                         00014200
      DOUBLE PRECISION B8                                              00014210
      REAL L                                                           00014220
      DEL=R/B                                                          00014230
      DEL2=DEL*DEL                                                     00014240
      FINHY=FINITE(ZHY,B)                                              00014250
      FINHX=CMPLX(0.0,0.0)                                             00014260
      IF(M.EQ.1) GO TO 10                                              00014270
      B1=R1/DEL                                                        00014280
      B2=R2/DEL                                                        00014290
      CALL FINF3(B1,B2,F31,F32)                                        00014300
      FINHY=FINHY-((X+L)*F31/R1-(X-L)*F32/R2)*CMPLX(1./DEL,0.0)        00014310
10    B8=0.7071067811865475D0/DEL                                      00014320
      R12=R1*R1                                                        00014330
      R22=R2*R2                                                        00014340
      CALL IK1(B8*DBLE(R1),ZIK1)                                       00014350
      ZIK1=ZIK1/R12                                                    00014360
      CALL IK1(B8*DBLE(R2),ZIK2)                                       00014370
      ZIK2=ZIK2/R22                                                    00014380
      FINHY=-(FINHY-((X+L)*ZIK1-(X-L)*ZIK2))*CMPLX(2.0,0.0)            00014390
      IF(X*Y.EQ.0.0) GO TO 30                                          00014400
```

```
      IF(M.EQ.1) GO TO 15                                                00014410
      FINHX=(F31/R1-F32/R2)*CMPLX(1./DEL,0.0)                            00014420
15    FINHX=-(FINHX+ZIK1-ZIK2)*CMPLX(Y*2.0,0.0)                          00014430
30    RETURN                                                             00014440
      END                                                                00014450

      SUBROUTINE SETRHO(X)                                               00014460
C** SP-VERSION (FOR EPS IN COMMON/SHARE/) **                             00014470
C--SET RHO-DEPENDENT CONSTANTS IN COMMON/SHARE/  WHERE                   00014480
C  PARAMETER                                                             00014490
C     X     = REAL*4 ARGUMENT..NOTE: X-XX DISPLACEMENT USED IN RHO IF    00014500
C             L>0; ELSE (L=0) X IS DUMMY PARM AND WHERE RHO IS GIVEN IN  00014510
C             COMMON/SHARE/--SEE COMMON STATEMENT BELOW--                00014520
C                                                                        00014530
      REAL   EPS                                                         00014540
      REAL   L                                                           00014550
      COMMON/SHARE/                                                      00014560
     * EPS,                                                             00014570
     * C2,C3,C4,                                                        00014580
     * XX,YY,YY2,RHO,RHO2,DELRHO,B,                                     00014590
     * L,DEL,DEL2,                                                      00014600
     * METHOD,NZ,NW                                                     00014610
      IF(L.EQ.0.0) GO TO 1                                               00014620
      XXX2=(X-XX)**2                                                     00014630
    2 RHO2=XXX2+YY2                                                      00014640
      RHO=SQRT(RHO2)                                                     00014650
      DELRHO=DEL*RHO                                                     00014660
      IF(DEL.NE.0.0) B=RHO/DEL                                           00014670
    3 RETURN                                                             00014680
    1 XXX2=XX**2                                                         00014690
      GO TO 2                                                            00014700
      END                                                                00014710

      COMPLEX FUNCTION ZHY(B,NEW,R)                                      00014720
C-- HY FUNC FOR GROUNDED DEL. DIPOLE USING LAG-CONVOLUTION              00014730
C  SUBR 'ZLAGHO' FOR GIVEN IND.NO. B.GT.0.                               00014740
C--NOTE: THIS IS JUST THE HANKEL TRANSFORM TERM OF HY USED IN            00014750
C  SUBR 'FINHY'.                                                         00014760
C  SEE FINHY SUBR FOR HALF-SPACE TERMS, ETC.                             00014770
C                                                                        00014780
      COMPLEX ZLAGHO,Z,I1K1,IKDIF,ERRFIN                                 00014790
      DOUBLE PRECISION B8                                                00014800
      EXTERNAL F4                                                        00014810
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEVFIN,MEVFIN,ERRFIN,LW    00014820
      COMMON/MODEL/RK(10),DD(9),M                                        00014830
      COMMON/CONST/DEL,DEL2,Z                                            00014840
      ZHY=CMPLX(0.0,0.0)                                                 00014850
      IF(M.EQ.1) GO TO 2                                                 00014860
      ZHY=ZLAGHO(ALOG(B),F4,HAKTOL,LW,NEW)/B                             00014870
2     B8=0.70710678118654D0*B                                            00014880
      CALL IKS(B8,I1K1,IKDIF)                                            00014890
      ZHY=ZHY*CMPLX(1./DEL2,0.0)-(IKDIF-2.*I1K1)/R**2                    00014900
```

```
      RETURN                                                            00014910
      END                                                               00014920

        COMPLEX FUNCTION ZLAGHO(X,FUN,TOL,L,NEW)                        00014930
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE               00014940
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*J0(G*B)*DG' DEFINED AS THE    00014950
C  COMPLEX HANKEL TRANSFORM OF ORDER 0 AND ARGUMENT X(=ALOG(B))         00014960
C  BY CONVOLUTION FILTERING WITH COMPLEX FUNCTION 'FUN'--AND            00014970
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....       00014980
C                                                                       00014990
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.                    00015000
C                                                                       00015010
C--PARAMETERS:                                                          00015020
C                                                                       00015030
C     * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE HANKEL TRANSFORM 00015040
C                'ZLAGHO' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,  00015050
C                SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT, 00015060
C                THEN SUBPROGRAM 'ZHANKO' IS ADVISED FOR GENERAL USE.   00015070
C                (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).           00015080
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)   00015090
C                OF A REAL ARGUMENT G.                                  00015100
C              NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN  00015110
C                  CALLING PROGRAM AND IN SUBPROGRAM FUN.               00015120
C                  THE COMPLEX FUNCTION FUN SHOULD BE A MONOTONE        00015130
C                  DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00015140
C                FOR REAL-ONLY FUNCTIONS, SUBPROGRAM 'RLAGHO' IS ADVISED;00015150
C                HOWEVER, TWO REAL-FUNCTIONS F1(G),F2(G) MAY BE         00015160
C                INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G))00015170
C       TOL=       REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,    00015180
C                  IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00015190
C                  THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,     00015200
C                  TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON      00015210
C                  THE FUNCTION FUN AND PARAMETER X...IN GENERAL,       00015220
C                A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY' 00015230
C                BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00015240
C                RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00015250
C                APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,  00015260
C                ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...  00015270
C       L=         RESULTING NO. FILTER WTS. USED IN THE VARIABLE       00015280
C                  CONVOLUTION (L DEPENDS ON TOL AND FUN).              00015290
C                MIN.L=20 AND MAX.L=193--WHICH COULD                    00015300
C                  OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING 00015310
C                  VERY FAST...                                         00015320
C     * NEW=     1 IS NECESSARY 1ST TIME OR BRAND NEW X.                00015330
C                0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20       00015340
C                  IS ASSUMED INTERNALLY BY THIS ROUTINE.               00015350
C                  NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL             00015360
C                  STILL ASSUME X=(LAST X)-0.20 ANYWAY...               00015370
C                  IT IS THE USERS RESPONSIBILITY TO NORMALIZE          00015380
C                  BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).00015390
C                THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT     00015400
C                  TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A           00015410
```

```
C                 SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00015420
C                 ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...         00015430
C                 THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED            00015440
C                 KERNELS WILL BE USED IN THE LAGGED CONVOLUTION        00015450
C                 WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS    00015460
C                 WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)            00015470
C                                                                       00015480
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZLAGHO; THE HANKEL 00015490
C  TRANSFORM IS THEN ZLAGHO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM   00015500
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...          00015510
C                                                                       00015520
C--USAGE-- 'ZLAGHO' IS CALLED AS FOLLOWS:                               00015530
C     ...                                                               00015540
C     COMPLEX Z,ZLAGHO,ZF                                               00015550
C     EXTERNAL ZF                                                       00015560
C     ...                                                               00015570
C     Z=ZLAGHO(ALOG(B),ZF,TOL,L,NEW)/B                                  00015580
C     ...                                                               00015590
C     END                                                               00015600
C     COMPLEX FUNCTION ZF(G)                                            00015610
C     ...USER SUPPLIED CODE...                                          00015620
C     END                                                               00015630
C                                                                       00015640
C--NOTES:                                                               00015650
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM     00015660
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS       00015670
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....                              00015680
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION   00015690
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN           00015700
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,   00015710
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'     00015720
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING  00015730
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).     00015740
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                  00015750
C                 ...                                                   00015760
C                 NB=AINT(5.*ALOG(BMAX/BMIN))+1                         00015770
C                 NB1=NB+1                                              00015780
C                 X0=ALOG(BMAX)+.2                                      00015790
C                 NEW=1                                                 00015800
C                 DO 1 J=1,NB                                           00015810
C                 I=NB1-J                                               00015820
C                 X=X0-.2*J                                             00015830
C                 ARG(I)=EXP(X)                                         00015840
C                 Z(I)=ZLAGHO(X,ZF,TOL,L,NEW)/ARG(I)                    00015850
C     1           NEW=0                                                 00015860
C                 ...                                                   00015870
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),Z(I),I=1,NB FOR      00015880
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,  00015890
C     TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)          00015900
C     SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.               00015910
C     (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY          00015920
C     ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW              00015930
```

```
C     BMAX,BMIN AND BY SETTING NEW=1....                                    00015940
C     (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE    00015950
C                                                                           00015960
C                                                                           00015970
        COMPLEX FUN,C,CMAX,SAVE                                             00015980
        DIMENSION KEY(193),SAVE(193),T(2),TMAX(2)                           00015990
        DIMENSION YT(193),Y1(76),Y2(76),Y3(41)                              00016000
        EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                                 00016010
        EQUIVALENCE (YT(1),Y1(1)),(YT(77),Y2(1)),(YT(153),Y3(1))            00016020
C--JO-EXTENDED FILTER WEIGHT ARRAYS:                                        00016030
       DATA Y1/                                                             00016040
      1 5.8565723E-08, 7.1143477E-11,-7.8395565E-11, 8.7489547E-11,         00016050
      2-8.9007811E-11, 9.8790055E-11,-9.8675347E-11, 1.1118797E-10,         00016060
      3-1.0893474E-10, 1.2543400E-10,-1.1979399E-10, 1.4200767E-10,         00016070
      4-1.3106341E-10, 1.6153229E-10,-1.4238602E-10, 1.8486236E-10,         00016080
      5-1.5315381E-10, 2.1319755E-10,-1.6238115E-10, 2.4824144E-10,         00016090
      6-1.6850378E-10, 2.9243813E-10,-1.6909302E-10, 3.4934366E-10,         00016100
      7-1.6043759E-10, 4.2417082E-10,-1.3690001E-10, 5.2458440E-10,         00016110
      8-8.9946096E-11, 6.6188220E-10,-6.6964033E-12, 8.5276151E-10,         00016120
      9 1.3222770E-10, 1.1219600E-09, 3.5591442E-10, 1.5061956E-09,         00016130
      1 7.0795382E-10, 2.0600379E-09, 1.2535947E-09, 2.8646623E-09,         00016140
      2 2.0904225E-09, 4.0409101E-09, 3.3642886E-09, 5.7687700E-09,         00016150
      3 5.2930786E-09, 8.3164338E-09, 8.2021809E-09, 1.2083635E-08,         00016160
      4 1.2577400E-08, 1.7666303E-08, 1.9143895E-08, 2.5953011E-08,         00016170
      5 2.8983953E-08, 3.8268851E-08, 4.3712685E-08, 5.6590075E-08,         00016180
      6 6.5740136E-08, 8.3864288E-08, 9.8662323E-08, 1.2448811E-07,         00016190
      7 1.4784461E-07, 1.8501974E-07, 2.2129198E-07, 2.7524203E-07,         00016200
      8 3.3094739E-07, 4.0974828E-07, 4.9462868E-07, 6.1030809E-07,         00016210
      9 7.3891802E-07, 9.0939667E-07, 1.1034727E-06, 1.3554600E-06,         00016220
      1 1.6474556E-06, 2.0207696E-06, 2.4591294E-06, 3.0131400E-06/         00016230
       DATA Y2/                                                             00016240
      1 3.6701680E-06, 4.4934101E-06, 5.4770076E-06, 6.7015208E-06,         00016250
      2 8.1726989E-06, 9.9954201E-06, 1.2194425E-05, 1.4909101E-05,         00016260
      3 1.8194388E-05, 2.2239184E-05, 2.7145562E-05, 3.3174088E-05,         00016270
      4 4.0499452E-05, 4.9486730E-05, 6.0421440E-05, 7.3822001E-05,         00016280
      5 9.0141902E-05, 1.1012552E-04, 1.3448017E-04, 1.6428337E-04,         00016290
      6 2.0062570E-04, 2.4507680E-04, 2.9930366E-04, 3.6560582E-04,         00016300
      7 4.4651421E-04, 5.4541300E-04, 6.6612648E-04, 8.1365181E-04,         00016310
      8 9.9374786E-04, 1.2138120E-03, 1.4824945E-03, 1.8107657E-03,         00016320
      9 2.2115938E-03, 2.7012675E-03, 3.2991969E-03, 4.0295817E-03,         00016330
      1 4.9214244E-03, 6.0106700E-03, 7.3405529E-03, 8.9643708E-03,         00016340
      2 1.0946310E-02, 1.3365017E-02, 1.6314985E-02, 1.9910907E-02,         00016350
      3 2.4289325E-02, 2.9612896E-02, 3.6070402E-02, 4.3876936E-02,         00016360
      4 5.3264829E-02, 6.4465091E-02, 7.7664144E-02, 9.2918324E-02,         00016370
      5 1.1000121E-01, 1.2811102E-01, 1.4543025E-01, 1.5832248E-01,         00016380
      6 1.6049224E-01, 1.4170064E-01, 8.8788108E-02,-1.1330934E-02,         00016390
      7-1.5331864E-01,-2.9094670E-01,-2.9084655E-01,-2.9708834E-02,         00016400
      8 3.9009601E-01, 1.7999785E-01,-4.1858139E-01, 1.5317216E-01,         00016410
      9 6.5184953E-02,-1.0751806E-01, 7.8429567E-02,-4.6019124E-02,         00016420
      1 2.5309571E-02,-1.3904823E-02, 7.8187120E-03,-4.5190369E-03/         00016430
       DATA Y3/                                                             00016440
      1 2.6729062E-03,-1.6073718E-03, 9.7715622E-04,-5.9804407E-04,         00016450
```

```
     2 3.6749320E-04,-2.2635296E-04, 1.3960805E-04,-8.6172618E-05,      00016460
     3 5.3212947E-05,-3.2867888E-05, 2.0304203E-05,-1.2543926E-05,      00016470
     4 7.7499633E-06,-4.7882430E-06, 2.9584108E-06,-1.8278645E-06,      00016480
     5 1.1293571E-06,-6.9778174E-07, 4.3113019E-07,-2.6637753E-07,      00016490
     6 1.6458373E-07,-1.0168954E-07, 6.2829807E-08,-3.8819969E-08,      00016500
     7 2.3985272E-08,-1.4819520E-08, 9.1563774E-09,-5.6573541E-09,      00016510
     8 3.4954514E-09,-2.1597005E-09, 1.3343946E-09,-8.2447148E-10,      00016520
     9 5.0941033E-10,-3.1474631E-10, 1.9447072E-10,-1.2015685E-10,      00016530
     1 7.4241055E-11,-4.5871468E-11, 2.8343095E-11,-1.7513137E-11,      00016540
     2 6.9049613E-12/                                                   00016550
C--$$ENDATA                                                             00016560
C                                                                       00016570
      IF(NEW) 10,30,10                                                  00016580
10    LAG=-1                                                            00016590
      XO=-X-26.30455704                                                 00016600
      DO 20 IR=1,193                                                    00016610
20    KEY(IR)=0                                                         00016620
30    LAG=LAG+1                                                         00016630
      ZLAGHO=(0.0,0.0)                                                  00016640
      CMAX=(0.0,0.0)                                                    00016650
      L=0                                                               00016660
      ASSIGN 110 TO M                                                   00016670
      I=129                                                             00016680
      GO TO 200                                                         00016690
110   TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                                  00016700
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                                  00016710
      I=I+1                                                             00016720
      IF(I.LE.146) GO TO 200                                            00016730
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150                   00016740
      CMAX=TOL*CMAX                                                     00016750
      ASSIGN 120 TO M                                                   00016760
      I=128                                                             00016770
      GO TO 200                                                         00016780
120   IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130       00016790
      I=I-1                                                             00016800
      IF(I.GT.0) GO TO 200                                              00016810
130   ASSIGN 140 TO M                                                   00016820
      I=147                                                             00016830
      GO TO 200                                                         00016840
140   IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190       00016850
      I=I+1                                                             00016860
      IF(I.LE.193) GO TO 200                                            00016870
      GO TO 190                                                         00016880
150   ASSIGN 160 TO M                                                   00016890
      I=1                                                               00016900
      GO TO 200                                                         00016910
160   IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170                         00016920
      I=I+1                                                             00016930
      IF(I.LE.128) GO TO 200                                            00016940
170   ASSIGN 180 TO M                                                   00016950
      I=193                                                             00016960
      GO TO 200                                                         00016970
```

```
180   IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190                          00016980
      I=I-1                                                              00016990
      IF(I.GE.147) GO TO 200                                             00017000
190   RETURN                                                             00017010
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)               00017020
200   LOOK=I+LAG                                                         00017030
      IQ=LOOK/194                                                        00017040
      IR=MOD(LOOK,194)                                                   00017050
      IF(IR.EQ.0) IR=1                                                   00017060
      IROLL=IQ*193                                                       00017070
      IF(KEY(IR).LE.IROLL) GO TO 220                                     00017080
210   C=SAVE(IR)*YT(I)                                                   00017090
      ZLAGH0=ZLAGH0+C                                                    00017100
      L=L+1                                                              00017110
      GO TO M,(110,120,140,160,180)                                      00017120
220   KEY(IR)=IROLL+IR                                                   00017130
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                              00017140
      GO TO 210                                                          00017150
      END                                                                00017160

      COMPLEX FUNCTION CANC4(A1,B1,EP,M,N,FUN,MF,ESUM)                   00017170
C--COMPLEX FUNCTION DEFINITE INTEGRATION BY                              00017180
C     ADAPTIVE QUADRATURE USING NEWTON-COTES NO. 4 (N=1,2,3), OR         00017190
C     AUTOMATIC GAUSSIAN QUADRATURE (N=4,5).....                         00017200
C  A GENERAL  ROUTINE  IN SINGLE-PRECISION COMPLEX...                    00017210
C     BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.          00017220
C--PARAMETERS--                                                          00017230
C     A1    = LOWER LIMIT OF INTEGRATION (REAL*4)                        00017240
C     B1    = UPPER LIMIT OF INTEGRATION (REAL*4)                        00017250
C     EP    = DESIRED REL. ERROR (REAL*4) IN COMPLEX RESULT 'CANC4'.     00017260
C             (FOR BOTH RE & IM PARTS OF CANC4)                          00017270
C     M     = RESULTING NUMBER OF COMPLEX 'FUN EVALUATIONS'              00017280
C     N     = ERROR TEST TYPE:                                           00017290
C           = 1 FOR ABS. ERROR TEST (NOT GENERALLY RECOMMENDED)          00017300
C         = 2 FOR 'L-ONE' ERROR TEST                                     00017310
C         = 3 FOR 'L-INFINITY' ERROR TEST                                00017320
C         = 4 FOR REL. ERROR TEST USING ADAPTIVE GAUSSIAN QUADRATURE     00017330
C         = 5 FOR REL. ERROR TEST USING NON-ADAPTIVE GAUSSIAN QUAD...    00017340
C     NOTE:  N=1,2,OR 3 USES ADAPTIVE NEWTON-COTES QUADRATURE, AND       00017350
C              N=4 OR 5 USES ADAPTIVE OR NON-ADAPTIVE GAUSS QUADRATURES  00017360
C     FUN   = EXTERNAL COMPLEX FUNCTION NAME (COMPLEX*8)                 00017370
C     MF    = MAX. FUN EVALUATIONS ALLOWED BEFORE ACCEPTING COMPLEX      00017380
C             RESULT 'CANC4' WITH M.GE.MF....                            00017390
C     ESUM  = ACTUAL COMPLEX ERROR ACHIEVED AT EXIT....                  00017400
C--SUBPROGRAMS CALLED:  CQSUBA,CQSUB (WHICH CALLS CQUAD)                 00017410
C     (THESE ARE FOR N=4 OR 5 GAUSSIAN QUADRATURES)                      00017420
C                                                                        00017430
      COMPLEX FUN,ESUM,TSUM,FA, F,X,Z, CQSUBA,CQSUB,                     00017440
     1 F1,FS,F3,FM,F2,FT,F4,FB,FTP,FBP,FMAX,FTST,EST,AEST,EST1,EST2,AEST00017450
     21,AEST2,ABSAR,DELTA,DIFF,DAFT,SUM                                  00017460
      DIMENSION F2(30),F4(30),FTP(30),FBP(30),FTST(5),EST2(30),NRTR(30)  00017470
      DIMENSION AEST2(30),XB(30)                                         00017480
```

```
      DIMENSION FMX(2)                                                  00017490
      EXTERNAL FUN                                                      00017500
      EQUIVALENCE (FMX(1),FMAX)                                         00017510
C--STATEMENT FUNCTION GOES HERE ON HONEYWELL MULTICS SYSTEM             00017520
      F(X)=CMPLX(ABS(REAL(X)),ABS(AIMAG(X)))                            00017530
C     THE PARAMETER SETUP FOR THE INITIAL CALL                          00017540
      IF(N.LE.0)GO TO 210                                               00017550
      IF(N.GT.5)GO TO 211                                               00017560
      GO TO (10,10,10,400,500),N                                        00017570
10      A=A1                                                            00017580
      B=B1                                                              00017590
      EPS=EP*63.0                                                       00017600
      ESUM=(0.0,0.0)                                                    00017610
      TSUM=(0.0,0.0)                                                    00017620
      LVL=1                                                             00017630
      DA=B-A                                                            00017640
      FA=FUN(A)                                                         00017650
      FS=FUN((3.0  *A+B)/4.0  )                                         00017660
      FM=FUN((A+B)*0.5  )                                               00017670
      FT=FUN((A+3.0  *B)/4.0  )                                         00017680
      FB=FUN(B)                                                         00017690
      M=5                                                               00017700
      FMAX=F(FA)                                                        00017710
      FTST(1)=FMAX                                                      00017720
      FTST(2)=F(FS)                                                     00017730
      FTST(3)=F(FM)                                                     00017740
      FTST(4)=F(FT)                                                     00017750
      FTST(5)=F(FB)                                                     00017760
      DO 100 I=2,5                                                      00017770
      IF(FMX(1).GE.REAL(FTST(I)))GO TO 101                              00017780
      FMX(1)=REAL(FTST(I))                                              00017790
  101 IF(FMX(2).GE.AIMAG(FTST(I)))GO TO 100                             00017800
      FMX(2)=AIMAG(FTST(I))                                             00017810
  100 CONTINUE                                                          00017820
      EST=(7.0  *(FA+FB)+32.0  *(FS+FT)+12.0  *FM)*DA/90.0              00017830
      ABSAR=(7.0  *(FTST(1)+FTST(5))+32.0  *(FTST(2)+FTST(4))+12.0  *FTS00017840
     1T(3))*DA/90.0                                                     00017850
      AEST=ABSAR                                                        00017860
C     1=RECUR                                                           00017870
    1 SX=(DA/(2.0  **LVL))/90.0                                         00017880
      F1=FUN((7.0  *A+B)/8.0  )                                         00017890
      F3=FUN((5.0  *A+3.0  *B)/8.0  )                                   00017900
      F2(LVL)=FUN((3.0  *A+5.0  *B)/8.0  )                              00017910
      F4(LVL)=FUN((A+7.0  *B)/8.0  )                                    00017920
      EST1=SX*(7.0  *(FA+FM)+32.0  *(F1+F3)+12.0  *FS)                  00017930
      FBP(LVL)=FB                                                       00017940
      FTP(LVL)=FT                                                       00017950
      XB(LVL)=B                                                         00017960
      EST2(LVL)=SX*(7.0  *(FM+FB)+32.0  *(F2(LVL)+F4(LVL))+12.0  *FT)   00017970
      SUM=EST1+EST2(LVL)                                                00017980
      FTST(1)=F(F1)                                                     00017990
      FTST(2)=F(F2(LVL))                                                00018000
```

```
      FTST(3)=F(F3)                                                     00018010
      FTST(4)=F(F4(LVL))                                                00018020
      FTST(5)=F(FM)                                                     00018030
      AEST1=SX*(7.0  *(F(FA)   +FTST(5))+32.0  *(FTST(1)+FTST(3))+12.0  00018040
     X*F(FS))                                                           00018050
      AEST2(LVL)=SX*(7.0  *(FTST(5)+F(FB)   )+32.0  *(FTST(2)+FTST(4))+100018060
     X2.0*F(FT))                                                        00018070
      ABSAR=ABSAR-AEST+AEST1+AEST2(LVL)                                 00018080
      M=M+4                                                             00018090
      IF(M.GE.MF) GO TO 5                                               00018100
      GO TO (201,200,202),N                                             00018110
  200 DELTA=ABSAR                                                       00018120
      GO TO 205                                                         00018130
  210 WRITE(6,39)                                                       00018140
   39 FORMAT(' CANC4- ERROR RETURN-N.LE.0')                             00018150
      GO TO 999                                                         00018160
  211 WRITE(6,40)                                                       00018170
   40 FORMAT(' CANC4- ERROR RETURN-N.GT.5')                             00018180
      GO TO 999                                                         00018190
  201 DELTA=(1.0,1.0)                                                   00018200
      GO TO 205                                                         00018210
  202 DO 203 I=1,4                                                      00018220
      IF(FMX(1).GE.REAL(FTST(I)))GO TO 2031                             00018230
      FMX(1)=REAL(FTST(I))                                              00018240
 2031 IF(FMX(2).GE.AIMAG(FTST(I)))GO TO 203                             00018250
      FMX(2)=AIMAG(FTST(I))                                             00018260
  203 CONTINUE                                                          00018270
      DELTA=FMAX                                                        00018280
  205 DAFT=EST-SUM                                                      00018290
      DIFF=F(DAFT)                                                      00018300
      DAFT=DAFT/63.0                                                    00018310
      Z=DIFF-EPS*DELTA                                                  00018320
      IF(REAL(Z).LE.0.0.AND.AIMAG(Z).LE.0.0) GO TO 6                    00018330
    3 IF(LVL-30)4,2,2                                                   00018340
    6 IF(LVL-1)2,4,2                                                    00018350
C     2=UP                                                              00018360
    2 A=B                                                               00018370
      ESUM=ESUM+DAFT                                                    00018380
      TSUM=TSUM+SUM                                                     00018390
    9 LVL=LVL-1                                                         00018400
      L=NRTR(LVL)                                                       00018410
      GO TO (11,12),L                                                   00018420
C     11=R1,12=R2                                                       00018430
    4 NRTR(LVL)=1                                                       00018440
      EST=EST1                                                          00018450
      AEST=AEST1                                                        00018460
      FB=FM                                                             00018470
      FT=F3                                                             00018480
      FM=FS                                                             00018490
      FS=F1                                                             00018500
      B=(A+B)/2.0                                                       00018510
      EPS=EPS/2.0                                                       00018520
```

```
    7 LVL=LVL+1                                                    00018530
      GO TO 1                                                      00018540
   11 NRTR(LVL)=2                                                  00018550
      FA=FB                                                        00018560
      FS=F2(LVL)                                                   00018570
      FM=FTP(LVL)                                                  00018580
      FT=F4(LVL)                                                   00018590
      FB=FBP(LVL)                                                  00018600
      B=XB(LVL)                                                    00018610
      EST=EST2(LVL)                                                00018620
      AEST=AEST2(LVL)                                              00018630
      GO TO 7                                                      00018640
   12 EPS=2.0  *EPS                                                00018650
      IF(LVL-1)5,5,9                                               00018660
    5 CANC4=TSUM-ESUM                                              00018670
      GO TO 999                                                    00018680
400   CANC4=CQSUBA(A1,B1,EP,M,ICK,ESUM,FUN,MF)                     00018690
      GO TO 999                                                    00018700
500   CANC4=CQSUB(A1,B1,EP,M,ICK,ESUM,FUN,MF)                      00018710
999      RETURN                                                    00018720
      END                                                          00018730

      SUBROUTINE CQUAD(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)       00018740
C--MODIFIED BY W.L.ANDERSON FOR COMPLEX FUNCTIONS--12/28/73.       00018750
      COMPLEX F,RESULT,FUNCT,FZERO,ACUM                            00018760
      DIMENSION FUNCT(127), P(381), RESULT(8)                      00018770
C THIS SUBROUTINE ATTEMPTS TO CALCULATE THE INTEGRAL OF F(X)       00018780
C OVER THE INTERVAL *A* TO *B* WITH RELATIVE ERROR NOT             00018790
C EXCEEDING *EPSIL*.                                               00018800
C THE RESULT IS OBTAINED USING A SEQUENCE OF 1,3,7,15,31,63,       00018810
C 127, AND 255 POINT INTERLACING FORMULAE(NO INTEGRAND             00018820
C EVALUATIONS ARE WASTED) OF RESPECTIVE DEGREE 1,5,11,23,          00018830
C 47,95,191 AND 383. THE FORMULAE ARE BASED ON THE OPTIMAL         00018840
C EXTENSION OF THE 3-POINT GAUSS FORMULA.  DETAILS OF              00018850
C THE FORMULAE ARE GIVEN IN *THE OPTIMUM ADDITION OF POINTS        00018860
C TO QUADRATURE FORMULAE* BY T.N.L. PATTERSON,MATHS.COMP.          00018870
C VOL 22,847-856,1968.                                             00018880
C                        *** INPUT ***                             00018890
C A        LOWER LIMIT OF INTEGRATION.                             00018900
C B        UPPER LIMIT OF INTEGRATION.                             00018910
C EPSIL    RELATIVE ACCURACY REQUIRED. WHEN THE RELATIVE            00018920
C          DIFFERENCE OF TWO SUCCESSIVE FORMULAE DOES NOT          00018930
C          EXCEED *EPSIL* THE LAST FORMULA COMPUTED IS TAKEN       00018940
C          AS THE RESULT.                                          00018950
C F        F(X) IS THE INTEGRAND.                                  00018960
C                        *** OUTPUT ***                            00018970
C RESULT   THIS ARRAY,WHICH SHOULD BE DECLARED TO HAVE AT          00018980
C          LEAST 8 ELEMENTS, HOLDS THE RESULTS OBTAINED BY         00018990
C          THE 1,3,7, ETC., POINT FORMULAE. THE NUMBER OF          00019000
C          FORMULAE COMPUTED DEPENTS ON *EPSIL*.                   00019010
C K        RESULT(K) HOLDS THE VALUE OF THE INTEGRAL TO THE        00019020
C          SPECIFIED RELATIVE ACCURACY.                            00019030
```

```
C NPTS    NUMBER INTEGRAND EVALUATIONS.                                  00019040
C ICHECK  ON EXIT NORMALLY ICHECK=0. HOWEVER IF CONVERGENCE              00019050
C         TO THE ACCURACY REQUESTED IS NOT ACHIEVED ICHECK=1             00019060
C         ON EXIT.                                                       00019070
C MEV     MAX.ALLOWABLE EVALUATIONS BEFORE ACCEPTING RESULT              00019080
C           WITH NPTS>=MEV...                                            00019090
C ABSCISSAE AND WEIGHTS OF QUADRATURE RULES ARE STACKED IN               00019100
C ARRAY *P* IN THE ORDER IN WHICH THEY ARE NEEDED.                       00019110
      DATA                                                               00019120
     * P( 1),P( 2),P( 3),P( 4),P( 5),P( 6),P( 7),                       00019130
     * P( 8),P( 9),P(10),P(11),P(12),P(13),P(14),                       00019140
     * P(15),P(16),P(17),P(18),P(19),P(20),P(21),                       00019150
     * P(22),P(23),P(24),P(25),P(26),P(27),P(28)/                       00019160
     *  0.77459666924148337704E 00,0.55555555555555555556E 00,          00019170
     *  0.88888888888888888889E 00,0.26848808986833344073E 00,          00019180
     *  0.96049126870802028342E 00,0.10465622602646726519E 00,          00019190
     *  0.43424374934680255800E 00,0.40139741477596222291E 00,          00019200
     *  0.45091653865847414235E 00,0.13441525524378422036E 00,          00019210
     *  0.51603282997079739697E-01,0.20062852937698902103E 00,          00019220
     *  0.99383196321275502221E 00,0.17001719629940260339E-01,          00019230
     *  0.88845923287225699889E 00,0.92927195315124537686E-01,          00019240
     *  0.62110294673722640294E 00,0.17151190913639138079E 00,          00019250
     *  0.22338668642896688163E 00,0.21915685840158749640E 00,          00019260
     *  0.22551049979820668739E 00,0.67207754295990703540E-01,          00019270
     *  0.25807598096176653565E-01,0.10031427861179557877E 00,          00019280
     *  0.84345657393211062463E-02,0.46462893261757986541E-01,          00019290
     *  0.85755920049990351154E-01,0.10957842105592463824E 00/          00019300
      DATA                                                             00019310
     * P(29),P(30),P(31),P(32),P(33),P(34),P(35),                       00019320
     * P(36),P(37),P(38),P(39),P(40),P(41),P(42),                       00019330
     * P(43),P(44),P(45),P(46),P(47),P(48),P(49),                       00019340
     * P(50),P(51),P(52),P(53),P(54),P(55),P(56)/                       00019350
     *  0.99909812496766759766E 00,0.25447807915618744154E-02,          00019360
     *  0.98153114955374010687E 00,0.16446049854387810934E-01,          00019370
     *  0.92965485742974005667E 00,0.35957103307129322097E-01,          00019380
     *  0.83672593816886873550E 00,0.56979509494123357412E-01,          00019390
     *  0.70249620649152707861E 00,0.76879620499003531043E-01,          00019400
     *  0.53131974364437562397E 00,0.93627109981264473617E-01,          00019410
     *  0.33113539325797683309E 00,0.10566989358023480974E 00,          00019420
     *  0.11248894313318662575E 00,0.11195687302095345688E 00,          00019430
     *  0.11275525672076869161E 00,0.33603877148207730542E-01,          00019440
     *  0.12903800100351265626E-01,0.50157139305899537414E-01,          00019450
     *  0.42176304415588548391E-02,0.23231446639910269443E-01,          00019460
     *  0.42877960025007734493E-01,0.54789210527962865032E-01,          00019470
     *  0.12651565562300680114E-02,0.82230079572359296693E-02,          00019480
     *  0.17978551568128270333E-01,0.28489754745833548613E-01/          00019490
      DATA                                                             00019500
     * P(57),P(58),P(59),P(60),P(61),P(62),P(63),                       00019510
     * P(64),P(65),P(66),P(67),P(68),P(69),P(70),                       00019520
     * P(71),P(72),P(73),P(74),P(75),P(76),P(77),                       00019530
     * P(78),P(79),P(80),P(81),P(82),P(83),P(84)/                       00019540
     *  0.38439810249455532039E-01,0.46813554990628012403E-01,          00019550
```

```
*  0.52834946790116519862E-01,0.55978436510476319408E-01,          00019560
*  0.99987288812035761194E 00,0.36322148184553065969E-03,          00019570
*  0.99720625937222195908E 00,0.25790497946856882724E-02,          00019580
*  0.98868475754742947994E 00,0.61155068221172463397E-02,          00019590
*  0.97218287474858179658E 00,0.10498246909621321898E-01,          00019600
*  0.94634285837340290515E 00,0.15406750466559497802E-01,          00019610
*  0.91037115695700429250E 00,0.20594233915912711149E-01,          00019620
*  0.86390793819369047715E 00,0.25869679327214746911E-01,          00019630
*  0.80694053195021761186E 00,0.31073551111687964880E-01,          00019640
*  0.73975604435269475868E 00,0.36064432780782572640E-01,          00019650
*  0.66290966002478059546E 00,0.40715510116944318934E-01,          00019660
*  0.57719571005204581484E 00,0.44914531653632197414E-01,          00019670
*  0.48361802694584102756E 00,0.48564330406673198716E-01/          00019680
 DATA                                                             00019690
* P( 85),P( 86),P( 87),P( 88),P( 89),P( 90),P( 91),               00019700
* P( 92),P( 93),P( 94),P( 95),P( 96),P( 97),P( 98),               00019710
* P( 99),P(100),P(101),P(102),P(103),P(104),P(105),               00019720
* P(106),P(107),P(108),P(109),P(110),P(111),P(112)/               00019730
*  0.38335932419873034692E 00,0.51583253952048458777E-01,          00019740
*  0.27774982202182431507E 00,0.53905499335266063927E-01,          00019750
*  0.16823525155220746498E 00,0.55481404356559363988E-01,          00019760
*  0.56344313046592789972E-01,0.56277699831254301273E-01,          00019770
*  0.56377628360384717388E-01,0.16801938574103865271E-01,          00019780
*  0.64519000501757369228E-02,0.25078569652949768707E-01,          00019790
*  0.21088152457266328793E-02,0.11615723319955134727E-01,          00019800
*  0.21438980012503867246E-01,0.27394605263981432516E-01,          00019810
*  0.63260731936263354422E-03,0.41115039786546930472E-02,          00019820
*  0.89892757840641357233E-02,0.14244877372916774306E-01,          00019830
*  0.19219905124727766019E-01,0.23406777495314006201E-01,          00019840
*  0.26417473395058259931E-01,0.27989218255238159704E-01,          00019850
*  0.18073956444538835782E-03,0.12895240826104173921E-02,          00019860
*  0.30577534101755311361E-02,0.52491234548088591251E-02/          00019870
 DATA                                                             00019880
* P(113),P(114),P(115),P(116),P(117),P(118),P(119),               00019890
* P(120),P(121),P(122),P(123),P(124),P(125),P(126),               00019900
* P(127),P(128),P(129),P(130),P(131),P(132),P(133),               00019910
* P(134),P(135),P(136),P(137),P(138),P(139),P(140)/               00019920
*  0.77033752332797418482E-02,0.10297116957956355524E-01,          00019930
*  0.12934839663607373455E-01,0.15536775555843982440E-01,          00019940
*  0.18032216390391286320E-01,0.20357755058472159467E-01,          00019950
*  0.22457265826816098707E-01,0.24282165203336599358E-01,          00019960
*  0.25791626976024229388E-01,0.26952749667633031963E-01,          00019970
*  0.27740702178279681994E-01,0.28138849915627150636E-01,          00019980
*  0.99998243035489159858E 00,0.50536095207862517625E-04,          00019990
*  0.99959879967191068325E 00,0.37774664632698466027E-03,          00020000
*  0.99831663531840739253E 00,0.93836984854238150079E-03,          00020010
*  0.99572410469840718851E 00,0.16811428654214699063E-02,          00020020
*  0.99149572117810613240E 00,0.25687649437940203731E-02,          00020030
*  0.98537149959852037111E 00,0.35728927835172996494E-02,          00020040
*  0.97714151463970571416E 00,0.46710503721143217474E-02,          00020050
*  0.96663785155841656709E 00,0.58434498758356395076E-02/          00020060
 DATA                                                             00020070
```

```
* P(141),P(142),P(143),P(144),P(145),P(146),P(147),                    00020080
* P(148),P(149),P(150),P(151),P(152),P(153),P(154),                    00020090
* P(155),P(156),P(157),P(158),P(159),P(160),P(161),                    00020100
* P(162),P(163),P(164),P(165),P(166),P(167),P(168)/                    00020110
*  0.95373000642576113641E 00,0.70724899954335554680E-02,             00020120
*  0.93832039777959288365E 00,0.83428387539681577056E-02,             00020130
*  0.92034002547001242073E 00,0.96411777297025366953E-02,             00020140
*  0.89974489977694003664E 00,0.10955733387837901648E-01,             00020150
*  0.87651341448470526974E 00,0.12275830560082770087E-01,             00020160
*  0.85064449476835027976E 00,0.13591571009765546790E-01,             00020170
*  0.82215625436498040737E 00,0.14893641664815182035E-01,             00020180
*  0.79108493379984836143E 00,0.16173218729577719942E-01,             00020190
*  0.75748396638051363793E 00,0.17421930159464173747E-01,             00020200
*  0.72142308537009891548E 00,0.18631848256138790186E-01,             00020210
*  0.68298743109107922809E 00,0.19795495048097499488E-01,             00020220
*  0.64227664250975951377E 00,0.20905851445812023852E-01,             00020230
*  0.59940393024224289297E 00,0.21956366305317824939E-01,             00020240
*  0.55449513263193254887E 00,0.22940964229387748761E-01/             00020250
 DATA                                                                 00020260
* P(169),P(170),P(171),P(172),P(173),P(174),P(175),                    00020270
* P(176),P(177),P(178),P(179),P(180),P(181),P(182),                    00020280
* P(183),P(184),P(185),P(186),P(187),P(188),P(189),                    00020290
* P(190),P(191),P(192),P(193),P(194),P(195),P(196)/                    00020300
*  0.50768775753371660215E 00,0.23854052106038540080E-01,             00020310
*  0.45913001198983233287E 00,0.24690524744487676909E-01,             00020320
*  0.40897982122988867241E 00,0.25445769965464765813E-01,             00020330
*  0.35740383783153215238E 00,0.26115673376706097680E-01,             00020340
*  0.30457644155671404334E 00,0.26696622927450359906E-01,             00020350
*  0.25067873030348317661E 00,0.27185513229624791819E-01,             00020360
*  0.19589750271110015392E 00,0.27579749566481873035E-01,             00020370
*  0.14042423315256017459E 00,0.27877251476613701609E-01,             00020380
*  0.84454040083710883710E-01,0.28076455793817246607E-01,             00020390
*  0.28184648949745694339E-01,0.28176319033016602131E-01,             00020400
*  0.28188814180192358694E-01,0.84009692870519326354E-02,             00020410
*  0.32259500250878684614E-02,0.12539284826474884353E-01,             00020420
*  0.10544076228633167722E-02,0.58078616599775673635E-02,             00020430
*  0.10719490006251933623E-01,0.13697302631990716258E-01/             00020440
 DATA                                                                 00020450
* P(197),P(198),P(199),P(200),P(201),P(202),P(203),                    00020460
* P(204),P(205),P(206),P(207),P(208),P(209),P(210),                    00020470
* P(211),P(212),P(213),P(214),P(215),P(216),P(217),                    00020480
* P(218),P(219),P(220),P(221),P(222),P(223),P(224)/                    00020490
*  0.31630366082226447689E-03,0.20557519893273465236E-02,             00020500
*  0.44946378920320678616E-02,0.71224386864583871532E-02,             00020510
*  0.96099525623638830097E-02,0.11703388747657003101E-01,             00020520
*  0.13208736697529129966E-01,0.13994609127619079852E-01,             00020530
*  0.90372734658751149261E-04,0.64476204130572477933E-03,             00020540
*  0.15288767050877655684E-02,0.26245617274044295626E-02,             00020550
*  0.38516876166398709241E-02,0.51485584789781777618E-02,             00020560
*  0.64674198318036867274E-02,0.77683877779219912200E-02,             00020570
*  0.90161081951956431600E-02,0.10178877529236079733E-01,             00020580
*  0.11228632913408049354E-01,0.12141082601668299679E-01,             00020590
```

```
*  0.12895813488012114694E-01,0.13476374833816515982E-01,          00020600
*  0.13870351089139840997E-01,0.14069424957813575318E-01,          00020610
*  0.25157870384280661489E-04,0.18887326450650491366E-03,          00020620
*  0.46918492424785040975E-03,0.84057143271072246365E-03/          00020630
 DATA                                                             00020640
* P(225),P(226),P(227),P(228),P(229),P(230),P(231),                00020650
* P(232),P(233),P(234),P(235),P(236),P(237),P(238),                00020660
* P(239),P(240),P(241),P(242),P(243),P(244),P(245),                00020670
* P(246),P(247),P(248),P(249),P(250),P(251),P(252)/                00020680
*  0.12843824718970101768E-02,0.17864463917586498247E-02,          00020690
*  0.23355251860571608737E-02,0.29217249379178197538E-02,          00020700
*  0.35362449977167777340E-02,0.41714193769840788528E-02,          00020710
*  0.48205888648512683476E-02,0.54778666939189508240E-02,          00020720
*  0.61379152800413850435E-02,0.67957855048827733948E-02,          00020730
*  0.74468208324075910174E-02,0.80866093647888599710E-02,          00020740
*  0.87109650797320868736E-02,0.93159241280693950932E-02,          00020750
*  0.98977752404874974440E-02,0.10452925722906011926E-01,          00020760
*  0.10978183152658912470E-01,0.11470482114693874380E-01,          00020770
*  0.11927026053019270040E-01,0.12345262372243838455E-01,          00020780
*  0.12722884982732382906E-01,0.13057836688353048840E-01,          00020790
*  0.13348311463725179953E-01,0.13592756614812395910E-01,          00020800
*  0.13789874783240936517E-01,0.13938625738306850804E-01,          00020810
*  0.14038227896908623303E-01,0.14088159516508301065E-01/          00020820
 DATA                                                             00020830
* P(253),P(254),P(255),P(256),P(257),P(258),P(259),                00020840
* P(260),P(261),P(262),P(263),P(264),P(265),P(266),                00020850
* P(267),P(268),P(269),P(270),P(271),P(272),P(273),                00020860
* P(274),P(275),P(276),P(277),P(278),P(279),P(280)/                00020870
*  0.99999759637974846462E 00,0.69379364324108267170E-05,          00020880
*  0.99994399620705437576E 00,0.53275293669780613125E-04,          00020890
*  0.99976049092443204733E 00,0.13575491094922871973E-03,          00020900
*  0.99938033802502358193E 00,0.24921240048299729402E-03,          00020910
*  0.99874561446809511470E 00,0.38974528447328229322E-03,          00020920
*  0.99780535449595727456E 00,0.55429531493037471492E-03,          00020930
*  0.99651414591489027385E 00,0.74028280424450333046E-03,          00020940
*  0.99483150280062100052E 00,0.94536151685852538246E-03,          00020950
*  0.99272134428278861533E 00,0.11674841174299594077E-02,          00020960
*  0.99015137040077015918E 00,0.14049079956551446427E-02,          00020970
*  0.98709252795403406719E 00,0.16561127281544526052E-02,          00020980
*  0.98351865757863272876E 00,0.19197129710138724125E-02,          00020990
*  0.97940628167086268381E 00,0.21944069253638388388E-02,          00021000
*  0.97473445975240266776E 00,0.24789582266575679307E-02/          00021010
 DATA                                                             00021020
* P(281),P(282),P(283),P(284),P(285),P(286),P(287),                00021030
* P(288),P(289),P(290),P(291),P(292),P(293),P(294),                00021040
* P(295),P(296),P(297),P(298),P(299),P(300),P(301),                00021050
* P(302),P(303),P(304),P(305),P(306),P(307),P(308)/                00021060
*  0.96948465950245923177E 00,0.27721957645934509940E-02,          00021070
*  0.96364062156981213252E 00,0.30730184347025783234E-02,          00021080
*  0.95718821610986096274E 00,0.33803979910869203823E-02,          00021090
*  0.95011529752129487656E 00,0.36933779170256508183E-02,          00021100
*  0.94241156519108305981E 00,0.40110687240750233989E-02,          00021110
```

```
*   0.93406843615772578800E 00,0.43326409680929828545E-02,          00021120
*   0.92507893290707565236E 00,0.46573172997568547773E-02,          00021130
*   0.91543758715576504064E 00,0.49843645647655386012E-02,          00021140
*   0.90514035881326159519E 00,0.53130866051870565663E-02,          00021150
*   0.89418456833555902286E 00,0.56428181013844441585E-02,          00021160
*   0.88256884024734190684E 00,0.59729195655081658049E-02,          00021170
*   0.87029305554811390585E 00,0.63027734490857587172E-02,          00021180
*   0.85735831088623215653E 00,0.66317812429018878941E-02,          00021190
*   0.84376688267270860104E 00,0.69593614093904229394E-02/          00021200
 DATA                                                               00021210
* P(309),P(310),P(311),P(312),P(313),P(314),P(315),                 00021220
* P(316),P(317),P(318),P(319),P(320),P(321),P(322),                 00021230
* P(323),P(324),P(325),P(326),P(327),P(328),P(329),                 00021240
* P(330),P(331),P(332),P(333),P(334),P(335),P(336)/                 00021250
*   0.82952219463740140018E 00,0.72849479805538070639E-02,          00021260
*   0.81462878765513741344E 00,0.76079896657190565832E-02,          00021270
*   0.79909229096084140180E 00,0.79279493342948491103E-02,          00021280
*   0.78291939411828301639E 00,0.82443037630328680306E-02,          00021290
*   0.76611781930376009072E 00,0.85565435613076896192E-02,          00021300
*   0.74869629361693660282E 00,0.88641732094824942641E-02,          00021310
*   0.73066452124218126133E 00,0.91667111635607884067E-02,          00021320
*   0.71203315536225203459E 00,0.94636899938300652943E-02,          00021330
*   0.69281376977911470289E 00,0.97546565363174114611E-02,          00021340
*   0.67301883023041847920E 00,0.10039172044056840798E-01,          00021350
*   0.65266166541001749610E 00,0.10316812330947621682E-01,          00021360
*   0.63175643771119423041E 00,0.10587167904885197931E-01,          00021370
*   0.61031811371518640016E 00,0.10849844089337314099E-01,          00021380
*   0.58836243447662541 43E 00,0.11104461134006926537E-01/          00021390
 DATA                                                               00021400
* P(337),P(338),P(339),P(340),P(341),P(342),P(343),                 00021410
* P(344),P(345),P(346),P(347),P(348),P(349),P(350),                 00021420
* P(351),P(352),P(353),P(354),P(355),P(356),P(357),                 00021430
* P(358),P(359),P(360),P(361),P(362),P(363),P(364)/                 00021440
*   0.56590588542365442262E 00,0.11350654315980596602E-01,          00021450
*   0.54296566649831149049E 00,0.11588074033043952568E-01,          00021460
*   0.51955966153745702199E 00,0.11816385890830235763E-01,          00021470
*   0.49570640791876146017E 00,0.12035270785279562630E-01,          00021480
*   0.47142506587165887693E 00,0.12244424981611985899E-01,          00021490
*   0.44673538766202847374E 00,0.12443560190714035263E-01,          00021500
*   0.42165768662616330006E 00,0.12632403643542078765E-01,          00021510
*   0.39621280605761593918E 00,0.12810698163877361967E-01,          00021520
*   0.37042208795007823014E 00,0.12978202239537399286E-01,          00021530
*   0.34430734159943802278E 00,0.13134690091960152836E-01,          00021540
*   0.31789081206847668318E 00,0.13279951743930530650E-01,          00021550
*   0.29119514851824668196E 00,0.13413793085110098513E-01,          00021560
*   0.26424337241092676194E 00,0.13536035934956213614E-01,          00021570
*   0.23705884558982972721E 00,0.13646518102571291428E-01/          00021580
 DATA                                                               00021590
* P(365),P(366),P(367),P(368),P(369),P(370),P(371),                 00021600
* P(372),P(373),P(374),P(375),P(376),P(377),P(378),                 00021610
* P(379),P(380),P(381)/                                            00021620
*   0.20966523824318119477E 00,0.13745093443001896632E-01,          00021630
```

```
     *  0.18208649675925219825E 00,0.13831631909506428676E-01,          00021640
     *  0.15434681148137810869E 00,0.13906019601325461264E-01,          00021650
     *  0.12647058437230196685E 00,0.13968158806516938516E-01,          00021660
     *  0.98482396598119202090E-01,0.14017968039456608810E-01,          00021670
     *  0.70406976042855179063E-01,0.14055382072649964277E-01,          00021680
     *  0.42269164765363603212E-01,0.14080351962553661325E-01,          00021690
     *  0.14093886410782462614E-01,0.14092845069160408355E-01,          00021700
     *  0.14094407090096179347E-01/                                     00021710
      ICHECK = 0                                                       00021720
C CHECK FOR TRIVIAL CASE.                                              00021730
      IF (A.EQ.B) GO TO 70                                             00021740
C SCALE FACTORS.                                                       00021750
      SUM = (B+A)/2.0                                                  00021760
      DIFF = (B-A)/2.0                                                 00021770
C 1-POINT GAUSS                                                        00021780
      FZERO = F(SUM)                                                   00021790
      RESULT(1) = 2.0*FZERO*DIFF                                       00021800
      I = 0                                                            00021810
      IOLD = 0                                                         00021820
      INEW = 1                                                         00021830
      K = 2                                                            00021840
      ACUM = (0.0,0.0)                                                 00021850
      GO TO 30                                                         00021860
   10 IF (K.EQ.8) GO TO 50                                             00021870
      IF(INEW+IOLD.GE.MEV) GO TO 60                                    00021880
      K = K + 1                                                        00021890
      ACUM = (0.0,0.0)                                                 00021900
C CONTRIBUTION FROM FUNCTION VALUES ALREADY COMPUTED.                  00021910
      DO 20 J=1,IOLD                                                   00021920
        I = I + 1                                                      00021930
        ACUM = ACUM + P(I)*FUNCT(J)                                    00021940
   20 CONTINUE                                                         00021950
C CONTRIBUTION FROM NEW FUNCTION VALUES.                               00021960
   30 IOLD = IOLD + INEW                                               00021970
      DO 40 J=INEW,IOLD                                                00021980
        I = I + 1                                                      00021990
        X = P(I)*DIFF                                                  00022000
        FUNCT(J) = F(SUM+X) + F(SUM-X)                                 00022010
        I = I + 1                                                      00022020
        ACUM = ACUM + P(I)*FUNCT(J)                                    00022030
   40 CONTINUE                                                         00022040
      INEW = IOLD + 1                                                  00022050
      I = I + 1                                                        00022060
      RESULT(K) = (ACUM+P(I)*FZERO)*DIFF                               00022070
C CHECK FOR CONVERGENCE.                                               00022080
      IF(ABS(REAL(RESULT(K))-REAL(RESULT(K-1))).LE.EPSIL*              00022090
     $ABS(REAL(RESULT(K))).AND.                                        00022100
     $ ABS(AIMAG(RESULT(K))-AIMAG(RESULT(K-1))).LE.EPSIL*              00022110
     $ABS(AIMAG(RESULT(K)))) GO TO 60                                  00022120
      GO TO 10                                                         00022130
C CONVERGENCE NOT ACHIEVED.                                            00022140
   50 ICHECK = 1                                                       00022150
```

```
C NORMAL TERMINATION.                                                   00022160
   60 NPTS = INEW + IOLD                                                00022170
      RETURN                                                            00022180
C TRIVIAL CASE                                                          00022190
   70 K = 2                                                             00022200
      RESULT(1) = (0.0,0.0)                                             00022210
      RESULT(2) = (0.0,0.0)                                             00022220
      NPTS = 0                                                          00022230
      RETURN                                                            00022240
      END                                                               00022250

      COMPLEX FUNCTION CQSUB(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)  00022260
      COMPLEX RELERR,F,RESULT,ESTIM,COMP                                00022270
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION                00022280
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION                    00022290
C ALGORITHM QUAD, TOGETHER WITH, IF NECESSARY, A NON-                   00022300
C ADAPTIVE SUBDIVISION PROCESS.                                         00022310
C   THE CALL TAKES THE FORM                                             00022320
C           CQSUB(A,B,EPSIL,NPTS,ICHECK,RELERR,F,MEV)                   00022330
C AND CAUSES F(X) TO BE INTEGRATED OVER (A,B) WITH RELATIVE             00022340
C ERROR HOPEFULLY NOT EXCEEDING EPSIL.  SHOULD QUAD CONVERGE            00022350
C (ICHECK=0) THEN QSUB WILL RETURN THE VALUE OBTAINED BY IT             00022360
C OTHERWISE SUBDIVISION WILL BE INVOKED AS A RESCUE                     00022370
C OPERATION IN A NON-ADAPTIVE MANNER. THE ARGUMENT RELERR               00022380
C GIVES A CRUDE ESTIMATE OF THE ACTUAL RELATIVE ERROR                   00022390
C OBTAINED.                                                             00022400
C   THE SUBDIVISION STRATEGY IS AS FOLLOWS                              00022410
C LET THE INTERVAL (A,B) BE DIVIDED INTO 2**N PANELS AT STEP            00022420
C N OF THE SUBDIVISION PROCESS.  QUAD IS APPLIED FIRST TO               00022430
C THE SUBDIVIDED INTERVAL ON WHICH QUAD LAST FAILED TO                  00022440
C CONVERGE AND IF CONVERGENCE IS NOW ACHIEVED THE REMAINING             00022450
C PANELS ARE INTEGRATED. SHOULD A CONVERGENCE FAILURE OCCUR             00022460
C ON ANY PANEL THE INTEGRATION AT THAT POINT IS TERMINATED              00022470
C AND THE PROCEDURE REPEATED WITH N INCREASED BY 1. THE                 00022480
C STRATEGY INSURES THAT POSSIBLY DELINQUENT INTERVALS ARE               00022490
C EXAMINED BEFORE WORK, WHICH LATER MIGHT HAVE TO BE                    00022500
C DISCARDED, IS INVESTED ON WELL BEHAVED PANELS. THE                    00022510
C PROCESS IS COMPLETE WHEN NO CONVERGENCE FAILURE OCCURS ON             00022520
C ANY PANEL AND THE SUM OF THE RESULTS OBTAINED BY QUAD ON              00022530
C EACH PANEL IS TAKEN AS THE VALUE OF THE INTEGRAL.                     00022540
C   THE PROCESS IS VERY CAUTIOUS IN THAT THE SUBDIVISION OF             00022550
C THE INTERVAL (A,B) IS UNIFORM, THE FINENESS OF WHICH IS               00022560
C CONTROLLED BY THE SUCCESS OF QUAD.  IN THIS WAY IT IS                 00022570
C RATHER DIFFICULT FOR A SPURIOUS CONVERGENCE TO SLIP                   00022580
C THROUGH.                                                              00022590
C   THE CONVERGENCE CRITERION OF QUAD IS SLIGHTLY RELAXED               00022600
C IN THAT A PANEL IS DEEMED TO HAVE BEEN SUCCESSFULLY                   00022610
C INTEGRATED IF EITHER QUAD CONVERGES OR THE ESTIMATED                  00022620
C ABSOLUTE ERROR COMMITTED ON THIS PANEL DOES NOT EXCEED                00022630
C EPSIL TIMES THE ESTIMATED ABSOLUTE VALUE OF THE INTEGRAL              00022640
C OVER (A,B).  THIS RELAXATION IS TO TRY TO TAKE ACCOUNT OF             00022650
C A COMMON SITUATION WHERE ONE PARTICULAR PANEL CAUSES                  00022660
```

```
C SPECIAL DIFFICULTY, PERHAPS DUE TO A SINGULARITY OF SOME                00022670
C TYPE.  IN THIS CASE QUAD COULD OBTAIN NEARLY EXACT                      00022680
C ANSWERS ON ALL OTHER PANELS AND SO THE RELATIVE ERROR FOR               00022690
C THE TOTAL INTEGRATION WOULD BE ALMOST ENTIRELY DUE TO THE               00022700
C DELINQUENT PANEL.  WITHOUT THIS CONDITION THE COMPUTATION               00022710
C MIGHT CONTINUE DESPITE THE REQUESTED RELATIVE ERROR BEING               00022720
C ACHIEVED.                                                               00022730
C   THE OUTCOME OF THE INTEGRATION IS INDICATED BY ICHECK.                00022740
C   ICHECK=0  -  CONVERGENCE OBTAINED WITHOUT INVOKING                    00022750
C                SUBDIVISION.  THIS CORRESPONDS TO THE                    00022760
C                DIRECT USE OF QUAD.                                      00022770
C   ICHECK=1  -  RESULT OBTAINED AFTER INVOKING SUBDIVISION.              00022780
C   ICHECK=2  -  AS FOR ICHECK=1 BUT AT SOME POINT THE                    00022790
C                RELAXED CONVERGENCE CRITERION WAS USED.                  00022800
C                THE RISK OF UNDERESTIMATING THE RELATIVE                 00022810
C                ERROR WILL BE INCREASED.  IF NECESSARY,                  00022820
C                CONFIDENCE MAY BE RESTORED BY CHECKING                   00022830
C                EPSIL AND RELERR FOR A SERIOUS DISCREPANCY.              00022840
C   ICHECK NEGATIVE                                                       00022850
C                IF DURING THE SUBDIVISION PROCESS THE                    00022860
C                ALLOWED UPPER LIMIT ON THE NUMBER OF PANELS              00022870
C                THAT MAY BE GENERATED (PRESENTLY 4096) IS                00022880
C                REACHED A RESULT IS OBTAINED WHICH MAY BE                00022890
C                UNRELIABLE BY CONTINUING THE INTEGRATION                 00022900
C                WITHOUT FURTHER SUBDIVISION IGNORING                     00022910
C                CONVERGENCE FAILURES. THIS OCCURRENCE IS                 00022920
C                FLAGGED BY RETURNING ICHECK WITH NEGATIVE                00022930
C                SIGN.                                                    00022940
C  THE RELIABILITY OF THE ALGORITHM WILL DECREASE FOR LARGE               00022950
C  VALUES OF EPSIL.  IT IS RECOMMENDED THAT EPSIL SHOULD                  00022960
C  GENERALLY BE LESS THAN ABOUT 0.001.                                    00022970
      DIMENSION RESULT(8)                                                 00022980
      INTEGER BAD, OUT                                                    00022990
      LOGICAL RHS                                                         00023000
      EXTERNAL F                                                          00023010
      DATA NMAX/4096/                                                     00023020
      CALL CQUAD(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)             00023030
      CQSUB = RESULT(K)                                                   00023040
      RELERR = (0.0,0.0)                                                  00023050
      IF(REAL(CQSUB).NE.0.0.AND.AIMAG(CQSUB).NE.0.0) RELERR=              00023060
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(CQSUB),                00023070
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(CQSUB))                    00023080
C CHECK IF SUBDIVISION IS NEEDED.                                         00023090
      IF (ICHECK.EQ.0) RETURN                                             00023100
C SUBDIVIDE                                                               00023110
      ESTIM=CQSUB*EPSIL                                                   00023120
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))                     00023130
      IC = 1                                                              00023140
      RHS = .FALSE.                                                       00023150
      N = 1                                                               00023160
      H = B - A                                                           00023170
      BAD = 1                                                             00023180
```

```
   10 CQSUB = (0.0,0.0)                                                  00023190
      RELERR = (0.0,0.0)                                                 00023200
      H = H*0.5                                                          00023210
      N = N + N                                                          00023220
C INTERVAL (A,B) DIVIDED INTO N EQUAL SUBINTERVALS.                      00023230
C INTEGRATE OVER SUBINTERVALS BAD TO (BAD+1) WHERE TROUBLE               00023240
C HAS OCCURRED.                                                          00023250
      M1 = BAD                                                           00023260
      M2 = BAD + 1                                                       00023270
      OUT = 1                                                            00023280
      GO TO 50                                                           00023290
C INTEGRATE OVER SUBINTERVALS 1 TO (BAD-1)                               00023300
   20 M1 = 1                                                             00023310
      M2 = BAD - 1                                                       00023320
      RHS = .FALSE.                                                      00023330
      OUT = 2                                                            00023340
      GO TO 50                                                           00023350
C INTEGRATE OVER SUBINTERVALS (BAD+2) TO N.                              00023360
   30 M1 = BAD + 2                                                       00023370
      M2 = N                                                             00023380
      OUT = 3                                                            00023390
      GO TO 50                                                           00023400
C SUBDIVISION RESULT                                                     00023410
   40 ICHECK = IC                                                        00023420
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(CQSUB)),                        00023430
     $ AIMAG(RELERR)/ABS(AIMAG(CQSUB)))                                  00023440
      RETURN                                                             00023450
C INTEGRATE OVER SUBINTERVALS M1 TO M2.                                  00023460
   50 IF (M1.GT.M2) GO TO 90                                             00023470
      DO 80 JJ=M1,M2                                                     00023480
        J = JJ                                                           00023490
C EXAMINE FIRST THE LEFT OR RIGHT HALF OF THE SUBDIVIDED                 00023500
C TROUBLESOME INTERVAL DEPENDING ON THE OBSERVED TREND.                  00023510
        IF (RHS) J = M2 + M1 - JJ                                        00023520
        ALPHA = A + H*(J-1)                                              00023530
        BETA = ALPHA + H                                                 00023540
        CALL CQUAD(ALPHA, BETA, RESULT, M, EPSIL, NF, ICHECK, F,MEV)     00023550
        COMP = (RESULT(M)-RESULT(M-1))                                   00023560
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                       00023570
        NPTS = NPTS + NF                                                 00023580
      IF(NPTS.GE.MEV) GO TO 70                                           00023590
        IF (ICHECK.NE.1) GO TO 70                                        00023600
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                                  00023610
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 100                            00023620
C SUBINTERVAL J HAS CAUSED TROUBLE.                                      00023630
C CHECK IF FURTHER SUBDIVISION SHOULD BE CARRIED OUT.                    00023640
        IF (N.EQ.NMAX) GO TO 60                                          00023650
        BAD = 2*J - 1                                                    00023660
        RHS = .FALSE.                                                    00023670
        IF ((J-2*(J/2)).EQ.0) RHS = .TRUE.                               00023680
        GO TO 10                                                         00023690
   60   IC = -IABS(IC)                                                   00023700
```

```
   70   CQSUB = CQSUB + RESULT(M)                                          00023710
   80 CONTINUE                                                            00023720
      RELERR = RELERR + COMP                                              00023730
   90 GO TO (20,30,40), OUT                                               00023740
C RELAXED CONVERGENCE                                                     00023750
  100 IC = ISIGN(2,IC)                                                    00023760
      GO TO 70                                                            00023770
      END                                                                 00023780

      COMPLEX FUNCTION CQSUBA(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)   00023790
      COMPLEX RELERR,F,RESULT,ESTIM,COMP                                  00023800
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION                  00023810
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION                      00023820
C ALGORITHM QUAD TOGETHER WITH, IF NECESSARY AN ADAPTIVE                  00023830
C SUBDIVISION PROCESS.  IT IS GENERALLY MORE EFFICIENT THAN               00023840
C THE NON-ADAPTIVE ALGORITHM QSUB BUT IS LIKELY TO BE LESS                00023850
C RELIABLE(SEE COMP.J.,14,189,1971).                                      00023860
C   THE CALL TAKES THE FORM                                               00023870
C           CQSUBA(A,B,EPSIL,NPTS,ICHECK,RELERR,F,MEV)                    00023880
C AND CAUSES F(X) TO BE INTEGRATED OVER (A,B) WITH RELATIVE               00023890
C ERROR HOPEFULLY NOT EXCEEDING EPSIL.  SHOULD QUAD CONVERGE              00023900
C (ICHECK=0) THEN QSUBA WILL RETURN THE VALUE OBTAINED BY IT              00023910
C OTHERWISE SUBDIVISION WILL BE INVOKED AS A RESCUE                       00023920
C OPERATION IN AN ADAPTIVE MANNER. THE ARGUMENT RELERR GIVES              00023930
C A CRUDE ESTIMATE OF THE ACTUAL RELATIVE ERROR OBTAINED.                 00023940
C   THE SUBDIVISION STRATEGY IS AS FOLLOWS                                00023950
C AT EACH STAGE OF THE PROCESS AN INTERVAL IS PRESENTED FOR               00023960
C SUBDIVISION (INITIALLY THIS WILL BE THE WHOLE INTERVAL                  00023970
C (A,B)).  THE INTERVAL IS HALVED AND QUAD APPLIED TO EACH                00023980
C SUBINTERVAL.  SHOULD QUAD FAIL ON THE FIRST SUBINTERVAL                 00023990
C THE SUBINTERVAL IS STACKED FOR FUTURE SUBDIVISION AND THE               00024000
C SECOND SUBINTERVAL IMMEDIATELY EXAMINED.  SHOULD QUAD FAIL              00024010
C ON THE SECOND SUBINTERVAL THE SUBINTERVAL IS                            00024020
C IMMEDIATELY SUBDIVIDED AND THE WHOLE PROCESS REPEATED.                  00024030
C EACH TIME A CONVERGED RESULT IS OBTAINED IT IS                          00024040
C ACCUMULATED AS THE PARTIAL VALUE OF THE INTEGRAL. WHEN                  00024050
C QUAD CONVERGES ON BOTH SUBINTERVALS THE INTERVAL LAST                   00024060
C STACKED IS CHOSEN NEXT FOR SUBDIVISION AND THE PROCESS                  00024070
C REPEATED.  A SUBINTERVAL IS NOT EXAMINED AGAIN ONCE A                   00024080
C CONVERGED RESULT IS OBTAINED FOR IT SO THAT A SPURIOUS                  00024090
C CONVERGENCE IS MORE LIKELY TO SLIP THROUGH THAN FOR THE                 00024100
C NON-ADAPTIVE ALGORITHM QSUB.                                            00024110
C   THE CONVERGENCE CRITERION OF QUAD IS SLIGHTLY RELAXED                 00024120
C IN THAT A PANEL IS DEEMED TO HAVE BEEN SUCCESSFULLY                     00024130
C INTEGRATED IF EITHER QUAD CONVERGES OR THE ESTIMATED                    00024140
C ABSOLUTE ERROR COMMITTED ON THIS PANEL DOES NOT EXCEED                  00024150
C EPSIL TIMES THE ESTIMATED ABSOLUTE VALUE OF THE INTEGRAL                00024160
C OVER (A,B).  THIS RELAXATION IS TO TRY TO TAKE ACCOUNT OF               00024170
C A COMMON SITUATION WHERE ONE PARTICULAR PANEL CAUSES                    00024180
C SPECIAL DIFFICULTY, PERHAPS DUE TO A SINGULARITY OF SOME                00024190
C TYPE.  IN THIS CASE QUAD COULD OBTAIN NEARLY EXACT                      00024200
C ANSWERS ON ALL OTHER PANELS AND SO THE RELATIVE ERROR FOR               00024210
```

```
C THE TOTAL INTEGRATION WOULD BE ALMOST ENTIRELY DUE TO THE                00024220
C DELINQUENT PANEL.  WITHOUT THIS CONDITION THE COMPUTATION                00024230
C MIGHT CONTINUE DESPITE THE REQUESTED RELATIVE ERROR BEING                00024240
C ACHIEVED.                                                                00024250
C   THE OUTCOME OF THE INTEGRATION IS INDICATED BY ICHECK.                 00024260
C   ICHECK=0  -  CONVERGENCE OBTAINED WITHOUT INVOKING SUB-                00024270
C                DIVISION.  THIS WOULD CORRESPOND TO THE                   00024280
C                DIRECT USE OF QUAD.                                       00024290
C   ICHECK=1  -  RESULT OBTAINED AFTER INVOKING SUBDIVISION.               00024300
C   ICHECK=2  -  AS FOR ICHECK=1 BUT AT SOME POINT THE                     00024310
C                RELAXED CONVERGENCE CRITERION WAS USED.                   00024320
C                THE RISK OF UNDERESTIMATING THE RELATIVE                  00024330
C                ERROR WILL BE INCREASED.  IF NECESSARY,                   00024340
C                CONFIDENCE MAY BE RESTORED BY CHECKING                    00024350
C                EPSIL AND RELERR FOR A SERIOUS DISCREPANCY.               00024360
C   ICHECK NEGATIVE                                                        00024370
C                IF DURING THE SUBDIVISION PROCESS THE STACK               00024380
C                OF DELINQUENT INTERVALS BECOMES FULL (IT IS               00024390
C                PRESENTLY SET TO HOLD AT MOST 100 NUMBERS)                00024400
C                A RESULT IS OBTAINED BY CONTINUING THE                    00024410
C                INTEGRATION IGNORING CONVERGENCE FAILURES                 00024420
C                WHICH CANNOT BE ACCOMMODATED ON THE STACK.                00024430
C                THIS OCCURRENCE IS FLAGGED BY RETURNING                   00024440
C                ICHECK WITH NEGATIVE SIGN.                                00024450
C  THE RELIABILITY OF THE ALGORITHM WILL DECREASE FOR LARGE                00024460
C  VALUES OF EPSIL.  IT IS RECOMMENDED THAT EPSIL SHOULD                   00024470
C  GENERALLY BE LESS THAN ABOUT 0.001.                                     00024480
      DIMENSION RESULT(8), STACK(100)                                      00024490
      EXTERNAL F                                                           00024500
      DATA ISMAX/100/                                                      00024510
      CALL CQUAD(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)              00024520
      CQSUBA = RESULT(K)                                                   00024530
      RELERR = (0.0,0.0)                                                   00024540
      IF(REAL(CQSUBA).NE.0.0.AND.AIMAG(CQSUBA).NE.0.0) RELERR=             00024550
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(CQSUBA),                00024560
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(CQSUBA))                    00024570
C CHECK IF SUBDIVISION IS NEEDED                                           00024580
      IF (ICHECK.EQ.0) RETURN                                              00024590
C SUBDIVIDE                                                                00024600
      ESTIM=CQSUBA*EPSIL                                                   00024610
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))                      00024620
      RELERR = (0.0,0.0)                                                   00024630
      CQSUBA = (0.0,0.0)                                                   00024640
      IS = 1                                                               00024650
      IC = 1                                                               00024660
      SUB1 = A                                                             00024670
      SUB3 = B                                                             00024680
   10 SUB2 = (SUB1+SUB3)*0.5                                               00024690
      CALL  CQUAD(SUB1, SUB2, RESULT, K, EPSIL, NF, ICHECK, F,MEV)         00024700
      NPTS = NPTS + NF                                                     00024710
      IF(NPTS.GE.MEV) GO TO 50                                             00024720
      COMP = (RESULT(K)-RESULT(K-1))                                       00024730
```

```
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                    00024740
      IF (ICHECK.EQ.0) GO TO 30                                       00024750
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                               00024760
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 70                          00024770
      IF (IS.GE.ISMAX) GO TO 20                                       00024780
C STACK SUBINTERVAL (SUB1,SUB2) FOR FUTURE EXAMINATION                00024790
      STACK(IS) = SUB1                                                00024800
      IS = IS + 1                                                     00024810
      STACK(IS) = SUB2                                                00024820
      IS = IS + 1                                                     00024830
      GO TO 40                                                        00024840
   20 IC = -IABS(IC)                                                  00024850
   30 CQSUBA = CQSUBA + RESULT(K)                                     00024860
      RELERR = RELERR + COMP                                          00024870
   40 CALL CQUAD(SUB2, SUB3, RESULT, K, EPSIL, NF, ICHECK, F,MEV)     00024880
      NPTS = NPTS + NF                                                00024890
      IF(NPTS.GE.MEV) GO TO 50                                        00024900
      COMP = (RESULT(K)-RESULT(K-1))                                  00024910
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                    00024920
      IF (ICHECK.EQ.0) GO TO 50                                       00024930
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                               00024940
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 80                          00024950
C SUBDIVIDE INTERVAL (SUB2,SUB3)                                      00024960
      SUB1 = SUB2                                                     00024970
      GO TO 10                                                        00024980
   50 CQSUBA = CQSUBA + RESULT(K)                                     00024990
      RELERR = RELERR + COMP                                          00025000
      IF(NPTS.GE.MEV) RETURN                                          00025010
      IF (IS.EQ.1) GO TO 60                                           00025020
C SUBDIVIDE THE DELINQUENT INTERVAL LAST STACKED                      00025030
      IS = IS - 1                                                     00025040
      SUB3 = STACK(IS)                                                00025050
      IS = IS - 1                                                     00025060
      SUB1 = STACK(IS)                                                00025070
      GO TO 10                                                        00025080
C SUBDIVISION RESULT                                                  00025090
   60 ICHECK = IC                                                     00025100
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(CQSUBA)),                    00025110
     $ AIMAG(RELERR)/ABS(AIMAG(CQSUBA)))                              00025120
      RETURN                                                          00025130
C RELAXED CONVERGENCE                                                 00025140
   70 IC = ISIGN(2,IC)                                                00025150
      GO TO 30                                                        00025160
   80 IC = ISIGN(2,IC)                                                00025170
      GO TO 50                                                        00025180
      END                                                             00025190

      COMPLEX FUNCTION FUNINT(X)                                      00025200
C--COMPLEX FUNCTION INTERPOLATION BY QUINTIC SPLINE VIA               00025210
C  CALL TO 'QPOINT', WHERE THE QUINTIC SPLINE                         00025220
C  COEFFICIENTS AR,BR,CR,DR,ER, AI,BI,CI,DI,EI WERE                   00025230
C  PREVIOUSLY OBTAINED BY SUBR 'QUINT'.                               00025240
```

```
C                                                                       00025250
      DIMENSION SR(80),AR(80),BR(80),CR(80),DR(80),ER(80),              00025260
     & SI(80),AI(80),BI(80),CI(80),DI(80),EI(80)                        00025270
      COMMON/SPLN80/SR,AR,BR,CR,DR,ER,SI,AI,BI,CI,DI,EI,RLM1,DELRLM,NL  00025280
      COMMON/FIN/R1,R2,RO,XL,SIG1,XX,Y                                  00025290
      R=ALOG(SQRT(X*X+Y*Y))                                             00025300
      CALL QPOINT(NL,SR,AR,BR,CR,DR,ER,RLM1,DELRLM,R,YR)                00025310
      CALL QPOINT(NL,SI,AI,BI,CI,DI,EI,RLM1,DELRLM,R,YI)                00025320
      FUNINT=CMPLX(YR,YI)                                               00025330
      RETURN                                                            00025340
      END                                                               00025350

      SUBROUTINE QUINT(NY,Y,B,C,D,E,F)                                  00025360
C--COMPUTES COEFFICIENTS OF A QUINTIC NATURAL SPLINE S(X) GIVEN         00025370
C  THE ORDINATES Y(I) AT ASSUMED EQUIDISTANT POINTS X(I),I=1 TO NY.     00025380
C                                                                       00025390
C  TRANSLATED FROM ALGOL TO FORTRAN BY                                  00025400
C   W.L. ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.            00025410
C  REF: ACM TRANSACTIONS ON MATH. SOFTWARE, SEPT 1976, V.2, N. 3,       00025420
C       PP.281-289.                                                     00025430
C                                                                       00025440
C  PARAMETERS:                                                          00025450
C                                                                       00025460
C  NY =  NUMBER OF DATA POINTS GIVEN IN Y(NY), NY.GT.2.                 00025470
C  Y()=  ARRAY OF NY GIVEN ORDINATES (DIM.GE.NY).                       00025480
C        Y() POINTS ASSUMED EQUALLY SPACED IN X-DIRECTION.              00025490
C  B,C,D,E,F() = RESULTING ARRAYS (EACH DIM.GE.NY) OF                   00025500
C        QUINTIC SPLINE COEFFICIENTS, WHERE                             00025510
C        FOR ANY XX IN [X(I),X(I+1)):                                   00025520
C        S(XX)=((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I) WITH     00025530
C        T=(XX-X(I))/DELX,  DELX=(X(I+1)-X(I)) FOR ANY I.               00025540
C NOTE:  SEE PROC 'QPOINT' TO EVAL THE QUINTIC SPLINE AFTER             00025550
C           'QUINT' IS CALLED.                                          00025560
C                                                                       00025570
      DIMENSION Y(1),B(1),C(1),D(1),E(1),F(1)                           00025580
      IF(NY.LE.2) GO TO 4                                               00025590
      N=NY-3                                                            00025600
      P=0.0                                                             00025610
      Q=0.0                                                             00025620
      R=0.0                                                             00025630
      S=0.0                                                             00025640
      T=0.0                                                             00025650
      DO 1 I=1,N                                                        00025660
      U=P*R                                                             00025670
      B(I)=1.0/(66.0-U*R-Q)                                             00025680
      R=26.0-U                                                          00025690
      C(I)=R                                                            00025700
      D(I)=Y(I+3)-3.0*(Y(I+2)-Y(I+1))-Y(I)-U*S-Q*T                      00025710
      Q=P                                                               00025720
      P=B(I)                                                            00025730
      T=S                                                               00025740
      S=D(I)                                                            00025750
```

```
1     CONTINUE                                                          00025760
      D(N+2)=0.0                                                        00025770
      N1=N+1                                                            00025780
      D(N1)=0.0                                                         00025790
      DO 2 J=1,N                                                        00025800
      I=N1-J                                                            00025810
      D(I)=(D(I)-C(I)*D(I+1)-D(I+2))*B(I)                               00025820
2     CONTINUE                                                          00025830
      N=NY-1                                                            00025840
      Q=0.0                                                             00025850
      V=D(1)                                                            00025860
      T=V                                                               00025870
      R=V                                                               00025880
      DO 3 I=2,N                                                        00025890
      P=Q                                                               00025900
      Q=R                                                               00025910
      R=D(I)                                                            00025920
      S=T                                                               00025930
      T=P-Q-Q+R                                                         00025940
      F(I)=T                                                            00025950
      U=5.0*(-P+Q)                                                      00025960
      E(I)=U                                                            00025970
      D(I)=10.0*(P+Q)                                                   00025980
      C(I)=0.5*(Y(I+1)+Y(I-1)+S-T)-Y(I)-U                               00025990
      B(I)=0.5*(Y(I+1)-Y(I-1)-S-T)-D(I)                                 00026000
3     CONTINUE                                                          00026010
      F(1)=V                                                            00026020
      E(1)=0.0                                                          00026030
      E(NY)=0.0                                                         00026040
      D(1)=0.0                                                          00026050
      D(NY)=0.0                                                         00026060
      C(1)=C(2)-10.0*V                                                  00026070
      C(NY)=C(NY-1)+10.0*T                                              00026080
      B(1)=Y(2)-Y(1)-C(1)-V                                             00026090
      B(NY)=Y(NY)-Y(NY-1)+C(NY)-T                                       00026100
4     RETURN                                                            00026110
      END                                                               00026120

      SUBROUTINE QPOINT(NY,Y,B,C,D,E,F,X1,DELX,XX,YY)                   00026130
C  GIVEN THE QUINTIC SPLINE COEFF"S B(*),C(*),D(*),E(*),F(*) AS         00026140
C  OBTAINED FROM SUBR 'QUINT', AND GIVEN NY OBS. DATA Y(NY) EQUALLY     00026150
C  SPACED BY DELX STARTING AT X1, THEN 'QPOINT' INTERPOLATES            00026160
C  YY AT ANY XX IN (X1,X1+(NY-1)*DELX).                                 00026170
C                                                                       00026180
      DIMENSION Y(1),B(1),C(1),D(1),E(1),F(1)                           00026190
      XMAX=X1+(NY-1)*DELX                                               00026200
      IF(XX.LT.X1.OR.XX.GT.XMAX) GO TO 2                                00026210
      I=(XX-X1)/DELX+1                                                  00026220
      XI=X1+(I-1)*DELX                                                  00026230
      T=(XX-XI)/DELX                                                    00026240
      YY=((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I)                00026250
1     RETURN                                                            00026260
```

```
2     WRITE(6,3) XX,X1,XMAX                                              00026270
3     FORMAT('0QPOINT ERROR-- XX=',E16.8,' NOT IN CLOSED INTERVAL (',    00026280
     & E16.8,',',E16.8,')')                                              00026290
      GO TO 1                                                            00026300
      END                                                                00026310

      SUBROUTINE KELVIN(X,M,B)                                           00026320
C--COMPUTES M(.LE.8) KELVIN FUNCTIONS (ORDERS 0,1) CONSECUTIVELY STORED  00026330
C  IN ARRAY B(M) WHERE:                                                  00026340
C                                                                        00026350
C     X     = DP-ARGUMENT .GT. 0.0D0 (ASYMPTOTIC FORM USED IF X.GE.8.0)  00026360
C     M     = NUMBER OF B'S TO COMPUTE AS DEFINED BELOW (1.GE.M.LE.8)    00026370
C     B(M)  = COMPUTED DP-FUNCTIONS WHERE B IS DEFINED:                  00026380
C             B(1) = BER(X)  --  ORDER 0                                 00026390
C             B(2) = BEI(X)  --  ORDER 0                                 00026400
C             B(3) = KER(X)  --  ORDER 0                                 00026410
C             B(4) = KEI(X)  --  ORDER 0                                 00026420
C             B(5) = BER1(X)  --  ORDER 1                                00026430
C             B(6) = BEI1(X)  --  ORDER 1                                00026440
C             B(7) = KER1(X)  --  ORDER 1                                00026450
C             B(8) = KEI1(X)  --  ORDER 1                                00026460
C ** ACCURACY GOOD TO AT LEAST 14 FIGURES FOR ALL X **                   00026470
C NOTE: THIS METHOD OF GENERATING MULTIPLE KELVIN FUNCTIONS WAS CHOSEN   00026480
C       TO REDUCE TOTAL CPU-TIME SINCE MOST APPLICATIONS REQUIRE         00026490
C       MULTIPLE FUNCTION USE AND IS THEREFORE ACCOMPLISHED BY ONE CALL. 00026500
C E.G: TO OBTAIN BER(X),BEI(X),KER(X), AND KEI(X):  CALL KELVIN(X,4,B)   00026510
C  IF X OR M OUT OF RANGE, ROUTINE EXITS WITHOUT ACTION.                 00026520
C                                                                        00026530
      IMPLICIT REAL*8 (A-H,O-Z)                                          00026540
      REAL*8 B(8),CN(8),SN(8)                                            00026550
      DATA CN          /.7071067811865475D0,0.D0,-.7071067811865475D0,   00026560
     * -1.D0,-.7071067811865475D0,0.D0,.7071067811865475D0,1.D0/,         00026570
     * SN   /.7071067811865475D0,1.D0,.7071067811865475D0,0.D0,           00026580
     * -.7071067811865475D0,-1.D0,-.7071067811865475D0,0.D0/             00026590
      DATA PI4/.7853981633974483D0/,R22/.7071067811865475D0/,            00026600
     * E/0.5D-14/,                                                       00026610
     * PI1/.31830988618379O7D0/                                          00026620
      IF(M.LT.1.OR.M.GT.8.OR.X.LE.0.0D0) GO TO 9                         00026630
      IF(X.GE.8.0D0) GO TO 8                                             00026640
C--SERIES METHODS (X.GT.0.0.AND.X.LT.8.0D0)                              00026650
      X2=0.5D0*X                                                         00026660
      X4=X2**4                                                           00026670
      T1=-0.25D0*X4                                                      00026680
      S1=T1                                                              00026690
      T2=0.0D0                                                           00026700
      T3=0.0D0                                                           00026710
      T4=0.0D0                                                           00026720
      T15=0.0D0                                                          00026730
      T26=0.0D0                                                          00026740
      T75=0.0D0                                                          00026750
      T86=0.0D0                                                          00026760
      IF(M.EQ.1) GO TO 100                                               00026770
```

```
      T2=X2**2                                                          00026780
      S2=T2                                                             00026790
      IF(M.EQ.2) GO TO 100                                              00026800
      T5=1.5D0                                                          00026810
      S5=T1*T5                                                          00026820
      IF(M.EQ.3) GO TO 100                                              00026830
      T6=1.0D0                                                          00026840
      S6=T2                                                             00026850
      IF(M.EQ.4) GO TO 100                                              00026860
      T3=-0.5D0*X2**3                                                   00026870
      S3=T3                                                             00026880
      T4=X2                                                             00026890
      S4=T4                                                             00026900
      IF(M.LE.6) GO TO 100                                              00026910
      T7=-0.25D0*X2**3                                                  00026920
      S7=2.0D0*T7*T5                                                    00026930
      T8=X2                                                             00026940
      S8=T8                                                             00026950
  100 TK=2.0D0                                                          00026960
  101 TK2=TK+TK                                                         00026970
      TK21=TK2-1.0D0                                                    00026980
      TK22=TK2-2.0D0                                                    00026990
      RK2=1.0D0/TK2                                                     00027000
      RK21=1.0D0/TK21                                                   00027010
      RK22=1.0D0/TK22                                                   00027020
      R1=-X4*(RK21*RK2)**2                                              00027030
      T1=T1*R1                                                          00027040
      S1=S1+T1                                                          00027050
      IF(M.EQ.1) GO TO 200                                              00027060
      R2=-X4*(RK22*RK21)**2                                             00027070
      T2=T2*R2                                                          00027080
      S2=S2+T2                                                          00027090
      IF(M.EQ.2) GO TO 200                                              00027100
      T5=T5+RK21+RK2                                                    00027110
      T15=T1*T5                                                         00027120
      S5=S5+T15                                                         00027130
      IF(M.EQ.3) GO TO 200                                              00027140
      T6=T6+RK22+RK21                                                   00027150
      T26=T2*T6                                                         00027160
      S6=S6+T26                                                         00027170
      IF(M.EQ.4) GO TO 200                                              00027180
      T3=T3*(-X4*(RK22*RK21**2*RK2))                                    00027190
      S3=S3+T3                                                          00027200
      T4=T4*(-X4*RK22**2*RK21/(TK2-3.0D0))                              00027210
      S4=S4+T4                                                          00027220
      IF(M.LE.6) GO TO 200                                              00027230
      T7=T7*R1                                                          00027240
      T75=TK2*T7*T5                                                     00027250
      S7=S7+T75                                                         00027260
      T8=T8*R2                                                          00027270
      T86=TK21*T8*T6                                                    00027280
      S8=S8+T86                                                         00027290
```

```
  200 TK=TK+1.0D0                                                    00027300
      IF(DABS(T1).GT.E.OR.DABS(T2).GT.E.OR.DABS(T15).GT.E.OR.        00027310
     * DABS(T26).GT.E.OR.DABS(T3).GT.E.OR.DABS(T4).GT.E.OR.          00027320
     * DABS(T75).GT.E.OR.DABS(T86).GT.E) GO TO 101                   00027330
      B(1)=1.0D0+S1                                                  00027340
      IF(M.EQ.1) GO TO 9                                             00027350
      B(2)=S2                                                        00027360
      IF(M.EQ.2) GO TO 9                                             00027370
      C=0.11593151565841240D0-DLOG(X)                                00027380
      B(3)=C*B(1)+PI4*B(2)+S5                                        00027390
      IF(M.EQ.3) GO TO 9                                             00027400
      B(4)=C*B(2)-PI4*B(1)+S6                                        00027410
      IF(M.EQ.4) GO TO 9                                             00027420
      B(5)=R22*(S3-S4)                                               00027430
      IF(M.EQ.5) GO TO 9                                             00027440
      B(6)=R22*(S3+S4)                                               00027450
      IF(M.EQ.6) GO TO 9                                             00027460
      S7=C*S3-B(1)/X+PI4*S4+S7                                       00027470
      S8=C*S4-B(2)/X-PI4*S3+S8                                       00027480
      B(7)=R22*(S7-S8)                                               00027490
      IF(M.EQ.7) GO TO 9                                             00027500
      B(8)=R22*(S7+S8)                                               00027510
    9 RETURN                                                         00027520
C--GENERAL ASYMPTOTIC FORM FOR NU=0,1:                               00027530
    8 NU=0                                                           00027540
      X2=R22*X                                                       00027550
      X8=8.0D0*X                                                     00027560
      SX=DSQRT(X)                                                    00027570
      EX2=DEXP(-X2)                                                  00027580
      C1=1.2533141373155000D0*EX2/SX                                 00027590
      C2=1.0D0/(2.5066282746310010D0*SX*EX2+1.0D-38)                 00027600
      MAXK=30                                                        00027610
      IF(X.LT.15.0D0) MAXK=X+X                                       00027620
    1 XNU=NU                                                         00027630
      XMU=4.0D0*XNU                                                  00027640
      ALP=X2+PI4*(XNU+XNU-0.5D0)                                     00027650
      BETA=ALP+PI4                                                   00027660
      CB=DCOS(BETA)                                                  00027670
      CA=DCOS(ALP)                                                   00027680
      SB=DSIN(BETA)                                                  00027690
      SA=DSIN(ALP)                                                   00027700
      N4=4*NU                                                        00027710
      FM=0.0D0                                                       00027720
      FP=0.0D0                                                       00027730
      GM=0.0D0                                                       00027740
      GP=0.0D0                                                       00027750
      TM=1.0D0                                                       00027760
      TP=1.0D0                                                       00027770
      K=1                                                            00027780
    2 TK=K                                                           00027790
      T=(XMU-(TK+TK-1.0D0)**2)/(TK*X8)                               00027800
      TPL=DABS(TP)                                                   00027810
```

```
      TP=-TP*T                                                          00027820
      IF(DABS(TP).GT.TPL) GO TO 21                                      00027830
      TM=TM*T                                                           00027840
      N=MOD(K,8)                                                        00027850
      IF(N.EQ.0) N=8                                                    00027860
      T1=TP*CN(N)                                                       00027870
      FP=FP+T1                                                          00027880
      T2=TM*CN(N)                                                       00027890
      FM=FM+T2                                                          00027900
      T3=TP*SN(N)                                                       00027910
      GP=GP+T3                                                          00027920
      T4=TM*SN(N)                                                       00027930
      GM=GM+T4                                                          00027940
      K=K+1                                                             00027950
      IF(K.GT.MAXK) GO TO 3                                             00027960
      GO TO 2                                                           00027970
   21 FP=FP-T1                                                          00027980
      FM=FM-T2                                                          00027990
      GP=GP-T3                                                          00028000
      GM=GM-T4                                                          00028010
    3 FP=FP+1.0D0                                                       00028020
      FM=FM+1.0D0                                                       00028030
      B(N4+4)=C1*(-FM*SB-GM*CB)                                         00028040
      B(N4+3)=C1*(FM*CB-GM*SB)                                          00028050
      B(N4+2)=C2*(FP*SA-GP*CA)+PI1*B(N4+3)                              00028060
      B(N4+1)=C2*(FP*CA+GP*SA)-PI1*B(N4+4)                              00028070
      IF(NU.EQ.1.OR.M.LE.4) GO TO 9                                     00028080
      NU=1                                                              00028090
      GO TO 1                                                           00028100
      END                                                               00028110

      SUBROUTINE IK1(B8,I1K1)                                           00028120
C--COMPUTE MODIFIED BESSEL FUNCTION PRODUCT I1*K1 FOR                   00028130
C  PARAMETERS                                                           00028140
C  B8 =   DOUBLE PRECISION ARGUMENT (=B/DSQRT(2.0D0) HERE)              00028150
C  I1K1 = I1*K1 COMPLEX RESULT                                          00028160
C                                                                       00028170
C--SUBR KELVIN CALLED AND D.P. USED BEFORE CMPLX"ING                    00028180
C                                                                       00028190
      COMPLEX I1K1,Z,Z2,TERM1                                           00028200
      DOUBLE PRECISION B8,BB(8),Q1,Q2                                   00028210
      IF(B8.GT.20.0D0) GO TO 1                                          00028220
      CALL KELVIN(B8,8,BB)                                              00028230
      Q1=-BB(6)*BB(8)+BB(5)*BB(7)                                       00028240
      Q2= BB(5)*BB(8)+BB(6)*BB(7)                                       00028250
      I1K1=CMPLX(SNGL(Q1),SNGL(Q2))                                     00028260
      RETURN                                                            00028270
    1 B=0.70710678*SNGL(B8)                                             00028280
      Z=CMPLX(B,B)                                                      00028290
      K=-1                                                              00028300
      Z2=4.*Z*Z                                                         00028310
      TERM1=CMPLX(1.0,0.0)                                              00028320
```

```
      I1K1=TERM1                                                        00028330
    2 K=K+2                                                             00028340
      COM=-K/(Z2*(K+1))                                                 00028350
      TERM1=TERM1*COM*(4.-K*K)                                          00028360
      I1K1=I1K1+TERM1                                                   00028370
      IF(CABS(TERM1)/CABS(I1K1).GT.1.E-6) GO TO 2                       00028380
      I1K1=I1K1/(2.*Z)                                                  00028390
      RETURN                                                            00028400
      END                                                               00028410

      SUBROUTINE IKS(B8,I1K1,IKDIF)                                     00028420
C--COMPUTE MODIFIED BESSEL FUNCTION (I & K) SPECIAL COMBINATIONS FOR     00028430
C  PARAMETERS                                                           00028440
C     B8    = DOUBLE PRECISION ARGUMENT (=B/DSQRT(2.D0) HERE)           00028450
C     I1K1  = I1*K1 COMPLEX RESULT                                      00028460
C     IKDIF = 4*I1*K1-(B8*DSQRT(I))*(I0*K1-I1*K0) COMPLEX RESULT DONE IN00028470
C             DP BEFORE CMPLX"ING.                                      00028480
C--SUBROUTINE KELVIN CALLED                                             00028490
C                                                                       00028500
      DOUBLE PRECISION B8,BB(8),BETA,Q1,Q2,R1,R2                        00028510
      COMPLEX I1K1,IKDIF,CAMBDA,DENOM,DENOM1,TERM0,TERM1,TERM11         00028520
      COMPLEX S11,SI0,SI1,SK0,SK1,ONE                                   00028530
      DATA ONE/(1.0,0.0)/                                               00028540
      IF(B8.GT.20.D0) GO TO 10                                          00028550
      CALL KELVIN(B8,8,BB)                                              00028560
      Q1=-BB(6)*BB(8)+BB(5)*BB(7)                                       00028570
      Q2= BB(5)*BB(8)+BB(6)*BB(7)                                       00028580
      I1K1=CMPLX(SNGL(Q1),SNGL(Q2))                                     00028590
      R1=-BB(1)*BB(8)-BB(2)*BB(7) -                                     00028600
     &    BB(6)*BB(3)-BB(5)*BB(4)                                       00028610
      R2=-BB(2)*BB(8)+BB(1)*BB(7) +                                     00028620
     &    BB(5)*BB(3)-BB(6)*BB(4)                                       00028630
      BETA=.7071067811865475D0*B8                                       00028640
      Q1=4.0D0*Q1-BETA*(R1-R2)                                          00028650
      Q2=4.0D0*Q2-BETA*(R1+R2)                                          00028660
      IKDIF=CMPLX(SNGL(Q1),SNGL(Q2))                                    00028670
      RETURN                                                            00028680
   10 B=SNGL(B8/0.7071067811865475D0)                                   00028690
      TOL=1.E-6                                                         00028700
C--FOR LARGE ARGUMENTS, USE ABRAMOWITZ AND STEGUN                       00028710
C         ASYMPTOTIC FORMULAS FOR LARGE ARGUMENTS                       00028720
C         9.7.1 THROUGH 9.7.5, P. 377-378.                              00028730
      CAMBDA=B*CMPLX(1.0,1.0)/2.                                        00028740
      IKDIF=CMPLX(100.,0.)                                              00028750
      ISIGN=1                                                           00028760
      DENOM=8.*CAMBDA                                                   00028770
      DENOM1=(2.*CAMBDA)**2                                             00028780
      NODD=1                                                            00028790
      TERM0=ONE                                                         00028800
      TERM1=ONE                                                         00028810
      TERM11=ONE                                                        00028820
      S11=ONE                                                           00028830
```

```
      SI0=ONE                                                          00028840
      SI1=ONE                                                          00028850
      SK0=ONE                                                          00028860
      SK1=ONE                                                          00028870
    1 NODD2=NODD*NODD                                                  00028880
      OIKDIF=CABS(IKDIF)                                               00028890
      TERM1=TERM1*CMPLX(4.-NODD2,0.)/DENOM                             00028900
      TERM0=TERM0*CMPLX(-FLOAT(NODD2),0.)/DENOM                        00028910
      TERM11=TERM11*CMPLX(NODD*(4.-NODD2)/(NODD+1),0.)/DENOM1          00028920
      ISIGN=-ISIGN                                                     00028930
      S11=S11+ISIGN*TERM11                                             00028940
      SI0=SI0+ISIGN*TERM0                                              00028950
      SI1=SI1+ISIGN*TERM1                                              00028960
      SK0=SK0+TERM0                                                    00028970
      SK1=SK1+TERM1                                                    00028980
      IKDIF=SI0*SK1-SK0*SI1                                            00028990
      NODD=NODD+2                                                      00029000
      IF(ABS(OIKDIF-CABS(IKDIF)).GT.TOL) GO TO 1                       00029010
      I1K1=S11/(CAMBDA*CMPLX(2.,0.))                                   00029020
      IKDIF=CMPLX(4.,0.)*I1K1-IKDIF/CMPLX(2.,0.)                       00029030
      RETURN                                                           00029040
      END                                                              00029050

      SUBROUTINE BESSIK(B,ZBES)                                        00029060
C--SPECIAL MODIFIED BESSEL FUNCTIONS I0,I1,K0,K1 PRODUCTS              00029070
C  ARGUMENT B=RHO/DEL AND WHERE OUTPUT ARRAY ZBES IS                   00029080
C     ZBES(1) = I0*K0                                                  00029090
C     ZBES(2) = I1*K1                                                  00029100
C     ZBES(3) = I0*K1                                                  00029110
C     ZBES(4) = I1*K0                                                  00029120
C     ZBES(5) = 4.*I1*K1-*B*CMPLX(.5,.5)*(I0*K1-I1*K0)                 00029130
C                                                                      00029140
C--SUBROUTINE IKSALL CALLED.                                           00029150
C                                                                      00029160
      REAL*8 BB                                                        00029170
      COMPLEX ZBES(5)                                                  00029180
      BB=.7071067811865475D0*DBLE(B)                                   00029190
      CALL IKSALL(BB,ZBES(1),ZBES(2),ZBES(3),ZBES(4),ZBES(5))          00029200
      RETURN                                                           00029210
      END                                                              00029220

      SUBROUTINE IKSALL(B8,I0K0,I1K1,I0K1,I1K0,IKDIF)                  00029230
C--COMPUTE MODIFIED BESSEL FUNCTION (I & K) PRODUCT COMBINATIONS FOR    00029240
C  PARAMETERS                                                          00029250
C     B8    = DOUBLE PRECISION ARGUMENT (=B/DSQRT(2.D0) HERE)          00029260
C     I0K0  = I0*K0 COMPLEX RESULT                                     00029270
C     I1K1  = I1*K1 COMPLEX RESULT                                     00029280
C     I0K1  = I0*K1 COMPLEX RESULT                                     00029290
C     I1K0  = I1*K0 COMPLEX RESULT                                     00029300
C     IKDIF = 4*I1*K1-(B8*DSQRT(I))*(I0*K1-I1*K0) COMPLEX RESULT DONE IN00029310
C             DP BEFORE CMPLX"ING.                                     00029320
C--SUBROUTINE KELVIN CALLED                                            00029330
```

```
C                                                                       00029340
      DOUBLE PRECISION B8,BB(8),BETA,P1,P2,Q1,Q2,R1,R2                  00029350
      COMPLEX I0K0,I1K1,I0K1,I1K0,IKDIF                                 00029360
      COMPLEX CAMBDA,DENOM,DENOM1,TERM0,TERM1,TERM11,TERM00             00029370
      COMPLEX S00,S11,SI0,SI1,SK0,SK1,ONE                               00029380
      DATA ONE/(1.0,0.0)/                                               00029390
      IF(B8.GT.10D0) GO TO 10                                           00029400
      CALL KELVIN(B8,8,BB)                                              00029410
      P1=-BB(6)*BB(8)+BB(5)*BB(7)                                       00029420
      P2= BB(5)*BB(8)+BB(6)*BB(7)                                       00029430
      I1K1=CMPLX(SNGL(P1),SNGL(P2))                                     00029440
      Q1=-BB(1)*BB(8)-BB(2)*BB(7)                                       00029450
      Q2=BB(1)*BB(7)-BB(2)*BB(8)                                        00029460
      I0K1=CMPLX(SNGL(Q1),SNGL(Q2))                                     00029470
      R1=BB(1)*BB(3)-BB(2)*BB(4)                                        00029480
      R2=BB(2)*BB(3)+BB(1)*BB(4)                                        00029490
      I0K0=CMPLX(SNGL(R1),SNGL(R2))                                     00029500
      R1=BB(6)*BB(3)+BB(5)*BB(4)                                        00029510
      R2=BB(6)*BB(4)-BB(5)*BB(3)                                        00029520
      I1K0=CMPLX(SNGL(R1),SNGL(R2))                                     00029530
      R1=Q1-R1                                                          00029540
      R2=Q2-R2                                                          00029550
      BETA=.7071067811865475D0*B8                                       00029560
      Q1=4.0D0*P1-BETA*(R1-R2)                                          00029570
      Q2=4.0D0*P2-BETA*(R1+R2)                                          00029580
      IKDIF=CMPLX(SNGL(Q1),SNGL(Q2))                                    00029590
      RETURN                                                            00029600
   10 B=SNGL(B8/0.7071067811865475D0)                                   00029610
      TOL=1.E-6                                                         00029620
C--FOR LARGE ARGUMENTS, USE ABRAMOWITZ AND STEGUN                       00029630
C         ASYMPTOTIC FORMULAS FOR LARGE ARGUMENTS                       00029640
C         9.7.1 THROUGH 9.7.5, P. 377-378.                              00029650
      CAMBDA=B*CMPLX(1.0,1.0)/2.                                        00029660
      IKDIF=CMPLX(100.,0.)                                              00029670
      ISIGN=1                                                           00029680
      DENOM=8.*CAMBDA                                                   00029690
      DENOM1=(2.*CAMBDA)**2                                             00029700
      NODD=1                                                            00029710
      TERM0=ONE                                                         00029720
      TERM1=ONE                                                         00029730
      TERM11=ONE                                                        00029740
      TERM00=ONE                                                        00029750
      S00=ONE                                                           00029760
      S11=ONE                                                           00029770
      SI0=ONE                                                           00029780
      SI1=ONE                                                           00029790
      SK0=ONE                                                           00029800
      SK1=ONE                                                           00029810
    1 NODD2=NODD*NODD                                                   00029820
      OIKDIF=CABS(IKDIF)                                                00029830
      TERM1=TERM1*CMPLX(4.-NODD2,0.)/DENOM                              00029840
      TERM0=TERM0*CMPLX(-FLOAT(NODD2),0.)/DENOM                         00029850
```

```
      TERM11=TERM11*CMPLX(NODD*(4.-NODD2)/(NODD+1.),0.)/DENOM1          00029860
      TERM00=TERM00*CMPLX(-FLOAT(NODD*NODD2)/(NODD+1.),0.)/DENOM1       00029870
      ISIGN=-ISIGN                                                      00029880
      S11=S11+ISIGN*TERM11                                              00029890
      S00=S00+ISIGN*TERM00                                              00029900
      SI0=SI0+ISIGN*TERM0                                               00029910
      SI1=SI1+ISIGN*TERM1                                               00029920
      SK0=SK0+TERM0                                                     00029930
      SK1=SK1+TERM1                                                     00029940
      IKDIF=SI0*SK1-SK0*SI1                                             00029950
      NODD=NODD+2                                                       00029960
      IF(ABS(OIKDIF-CABS(IKDIF)).GT.TOL) GO TO 1                        00029970
      DENOM1=ONE/(CAMBDA*CMPLX(2.0,0.0))                                00029980
      I0K0=S00*DENOM1                                                   00029990
      I1K1=S11*DENOM1                                                   00030000
      I0K1=SI0*SK1*DENOM1                                               00030010
      I1K0=SI1*SK0*DENOM1                                               00030020
      IKDIF=CMPLX(4.,0.)*I1K1-IKDIF/CMPLX(2.,0.)                        00030030
      RETURN                                                            00030040
      END                                                               00030050

       SUBROUTINE FINF3(B1,B2,F31,F32)                                  00030060
C--FINF3 FOR MARQHXY (USES ZHANKS)                                      00030070
       COMMON/FINERR/TOL,T,IT,N,NEV,MEV,ES,LW                           00030080
       COMPLEX F31,F32,ZHANKS,ES                                        00030090
       EXTERNAL F3                                                      00030100
       F31=ZHANKS(1,B1,F3,TOL,LW,1)                                     00030110
       IF(B1.EQ.B2) GO TO 10                                            00030120
       F32=ZHANKS(1,B2,F3,TOL,LW,1)                                     00030130
       RETURN                                                           00030140
10     F32=F31                                                          00030150
       RETURN                                                           00030160
       END                                                              00030170

      COMPLEX FUNCTION HX03(X)                                          00030180
C-- HX COMPONENT*4PI FOR GROUND CASE                                    00030190
C  PARAMETER X= REAL*4 ARGUMENT..NOTE--X-XX DISPLACEMENT USED IN        00030200
C  RHO IF L.GT.0 ELSE (L=0) X IS DUMMY PARM AND WHERE RHO IS GIVEN IN   00030210
C  COMMON/SHARE/--PLUS OTHER PARAMETERS IN COMMON/MODEL/ AND /CTL/      00030220
C                                                                       00030230
      REAL   L,K(10),D(9)                                               00030240
      COMPLEX F3,F4,ZHANKS,ZFLD,ZI1,ZI2,TWO,                            00030250
     1 ZI1P(19),ZI2P(19),ZBES(5)                                        00030260
      COMMON/MODEL/K,D,M                                                00030270
      COMMON/SHARE/EPS,C2,C3,C4,XX,YY,YY2,RHO,RHO2,DELRHO,B,            00030280
     1 L,DEL,DEL2,IREST(3)                                              00030290
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                           00030300
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                               00030310
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB                00030320
      EXTERNAL F3,F4                                                    00030330
      DATA TWO/(2.0,0.0)/                                               00030340
      IF(YY.EQ.0.0) GO TO 80                                            00030350
```

```
      IF(L.EQ.0.0) GO TO 10                                          00030360
      XD=X-XX                                                        00030370
      CALL SETRHO(X)                                                 00030380
      C2=XD*YY/(DELRHO**2)                                           00030390
      IF(C2.EQ.0.0) GO TO 80                                         00030400
   10 IF(IOB.LT.5.OR.L.GT.0.0) GO TO 20                              00030410
C--CHECK FOR HX,HY SAVINGS--                                         00030420
      IF(LCOMP.NE.0.AND.(LCOMP.NE.ICOMP)) GO TO 30                   00030430
   20 CALL BESSIK(B,ZBES)                                            00030440
   30 HX03=CMPLX(2.*C2/B**2,0.0)*ZBES(5)                             00030450
      IF(M.EQ.1) GO TO 70                                            00030460
      IF(IOB.LT.5.OR.L.GT.0.0) GO TO 50                              00030470
C--CHECK FOR HX,HY SAVINGS--                                         00030480
      IF(LCOMP.NE.0.AND.(LCOMP.NE.ICOMP).AND.C3.NE.0..AND.C4.NE.0.)  00030490
     1 GO TO 60                                                      00030500
   50 ZI1=ZHANKS(1,B,F3,EPS,LL,1)                                    00030510
      CALL MODIFY(1)                                                 00030520
      ZI2=ZHANKS(0,B,F4,EPS,LL,0)                                    00030530
      IF(L.EQ.0.0) CALL SWAP(1)                                      00030540
   60 HX03=C2*(CMPLX(4./B,0.0)*ZI1-TWO*ZI2)+HX03                     00030550
   70 RETURN                                                         00030560
   80 HX03=(0.0,0.0)                                                 00030570
      GO TO 70                                                       00030580
      END                                                            00030590

      COMPLEX FUNCTION HY03(X)                                       00030600
C-- HY COMPONENT*4PI FOR GROUND CASE                                 00030610
C  PARAMETER X= REAL*4 ARGUMENT..NOTE--X-XX DISPLACEMENT USED IN     00030620
C  RHO IF L.GT.0 ELSE (L=0) X IS DUMMY PARM AND WHERE RHO IS GIVEN IN 00030630
C  COMMON/SHARE/--PLUS OTHER PARAMETERS IS COMMON/MODEL/ AND /CTL/   00030640
C                                                                    00030650
      REAL   L,K(10),D(9)                                            00030660
      COMPLEX F3,F4,ZHANKS,ZFLD,ZI1,ZI2,YR2,ONE,FOUR,                00030670
     1 ZI1P(19),ZI2P(19),ZBES(5)                                     00030680
      COMMON/MODEL/K,D,M                                             00030690
      COMMON/SHARE/EPS,C2,C3,C4,XX,YY,YY2,RHO,RHO2,DELRHO,B,         00030700
     1 L,DEL,DEL2,IREST(3)                                           00030710
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                        00030720
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                            00030730
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB             00030740
      EXTERNAL F3,F4                                                 00030750
      DATA ONE,FOUR/(1.0,0.0),(4.0,0.0)/                             00030760
      IF(L.EQ.0.0) GO TO 10                                          00030770
      XD=X-XX                                                        00030780
      CALL SETRHO(X)                                                 00030790
      XR2=XD*XD/RHO2                                                 00030800
      C3=2.*(XR2-1.)/DEL2                                            00030810
      C4=2.*(1.-2.*XR2)/DELRHO                                       00030820
   10 IF(IOB.LT.5.OR.L.GT.0.0) GO TO 20                              00030830
C--CHECK FOR HX,HY SAVINGS--                                         00030840
      IF(LCOMP.NE.0.AND.(LCOMP.NE.ICOMP)) GO TO 30                   00030850
   20 CALL BESSIK(B,ZBES)                                            00030860
```

```
   30 YR2=YY2/RHO2                                                      00030870
      HY03=CMPLX(2./RHO2,0.0)*((FOUR*YR2-ONE)*ZBES(2)-                  00030880
     & YR2*(FOUR*ZBES(2)-ZBES(5)))                                      00030890
      IF(M.EQ.1) GO TO 70                                               00030900
      IF(IOB.LT.5.OR.L.GT.0.0) GO TO 50                                 00030910
C--CHECK FOR HX,HY SAVINGS--                                            00030920
      IF(LCOMP.NE.0.AND.(LCOMP.NE.ICOMP).AND.C2.NE.0.) GO TO 60         00030930
   50 IF(C4.NE.0.0) ZI1=ZHANKS(1,B,F3,EPS,LL,1)                         00030940
      IF(C3.EQ.0.0) GO TO 60                                            00030950
      IF(C4.NE.0.0) GO TO 55                                            00030960
      ZI2=ZHANKS(0,B,F4,EPS,LL,1)                                       00030970
      GO TO 56                                                          00030980
   55 CALL MODIFY(1)                                                    00030990
      ZI2=ZHANKS(0,B,F4,EPS,LL,0)                                       00031000
   56 IF(L.EQ.0.0) CALL SWAP(1)                                         00031010
   60 HY03=C3*ZI2+C4*ZI1+HY03                                           00031020
   70 RETURN                                                            00031030
      END                                                               00031040

      COMPLEX FUNCTION F3PJ(G)                                          00031050
C--PARTIAL F3 W/R P(JJ),JJ=1,2*MM-1 WHERE                               00031060
C  JJ IS SPECIFIED IN COMMON/PART/JJ                                    00031070
C                                                                       00031080
      COMPLEX V1,F1,C,ZFLD,PF1,ZI1,ZI2,ZI1P(19),ZI2P(19),ZBES(5)        00031090
      COMMON/SHARE/EPS,C2,C3,C4,XX,YY,YY2,RHO,RHO2,DELRHO,BB,           00031100
     1DUML,DEL,DEL2,IREST(3)                                            00031110
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                           00031120
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                               00031130
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB                00031140
      COMMON/PART/JJ,ISEP                                               00031150
      CALL RECURF(G,DEL,SIG1,V1,F1,PF1,JJ)                              00031160
      C=G                                                               00031170
      F3PJ=-PF1*(C*V1)/(C+V1*F1)**2                                     00031180
      RETURN                                                            00031190
      END                                                               00031200

      COMPLEX FUNCTION F4PJ(G)                                          00031210
C--PARTIAL F4 W/R P(JJ) VIA F3PJ(G)                                     00031220
C                                                                       00031230
      COMPLEX F3PJ                                                      00031240
      F4PJ=G*F3PJ(G)                                                    00031250
      RETURN                                                            00031260
      END                                                               00031270

      COMPLEX FUNCTION PHXPJ(X)                                         00031280
C--PARTIAL OF HX (DIPOLE FUNCTION) W/R P(J),J=JTH PARM IN (1,2*MM-1)     00031290
C  X IS DUMMY (IF L=0)  ELSE X-XX USED IN RHO,ETC (IF L.NE.0)           00031300
C  (OTHER PARMS ASSUMED SET IN COMMON S)                                00031310
C                                                                       00031320
      REAL   K(10),D(9),L                                               00031330
      COMPLEX F3PJ,F4PJ,ZHANKS,ZFLD,ZERO,TWO,FOUR,                      00031340
     1 ZI1,ZI2,ZI1P(19),ZI2P(19),ZBES(5),ZLAM                           00031350
```

```
      EXTERNAL F3PJ,F4PJ,GF4                                            00031360
      COMMON/MODEL/K,D,MM                                               00031370
      COMMON/SHARE/EPS,C2,C3,C4,XX,YY,YY2,RHO,RHO2,DELRHO,BB,           00031380
     1 L,DEL,DEL2,IREST(3)                                              00031390
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                           00031400
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                               00031410
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB                00031420
      COMMON/PART/J,ISEP                                                00031430
      DATA ZERO,TWO,FOUR/(0.0,0.0),(2.0,0.0),(4.0,0.0)/                 00031440
      IF(L.EQ.0.0) GO TO 10                                             00031450
      XD=X-XX                                                           00031460
      CALL SETRHO(X)                                                    00031470
      C2=XD*YY/(DELRHO**2)                                              00031480
      IF(C2.EQ.0.0) GO TO 110                                           00031490
   10 IF(J.GT.1) GO TO 60                                               00031500
C--GET PARTIAL W/R SIG1 (SPECIAL CASE J=1 AND SIG1 NOT HELD FIXED).     00031510
      ZI1P(1)=ZERO                                                      00031520
      ZI2P(1)=ZERO                                                      00031530
      IF(MM.EQ.1) GO TO 30                                              00031540
      IF(L.GT.0.0) GO TO 11                                             00031550
      CALL SWAP(-1)                                                     00031560
      CALL MODIFY(1)                                                    00031570
      ZI1P(1)=CMPLX(BB/SIG1,0.0)*ZHANKS(1,BB,GF4,EPS,LL,0)              00031580
      GO TO 12                                                          00031590
   11 ZI1P(1)=CMPLX(BB/SIG1,0.0)*ZHANKS(1,BB,GF4,EPS,LL,1)              00031600
   12 ZI1P(1)=CMPLX(4./BB,0.0)*ZHANKS(1,BB,F3PJ,EPS,LL,1)+ZI1P(1)       00031610
      CALL MODIFY(1)                                                    00031620
      ZI2P(1)=-TWO*ZHANKS(0,BB,F4PJ,EPS,LL,0)                           00031630
   30 IF(IOB.LT.5.OR.L.GT.0.0) GO TO 40                                 00031640
C--CHECK FOR HX,HY SAVINGS--                                            00031650
      IF(LCOMP.NE.0.AND.(LCOMP.NE.ICOMP)) GO TO 50                      00031660
   40 CALL BESSIK(BB,ZBES)                                              00031670
   50 ZLAM=0.5*CMPLX(BB,BB)                                             00031680
      PHXPJ=-(FOUR+ZLAM*ZLAM)*ZBES(2)-ZLAM*                             00031690
     & (TWO*(ZBES(4)-ZBES(3))-ZLAM*ZBES(1))                             00031700
      PHXPJ=C2*(ZI1P(1)+ZI2P(1)+CMPLX(2./(SIG1*BB*BB),0.0)*PHXPJ)       00031710
      GO TO 100                                                         00031720
   60 IF(IOB.LT.5.OR.L.GT.0.0) GO TO 70                                 00031730
C--CHECK FOR HX,HY SAVINGS--                                            00031740
      IF(LCOMP.NE.0.AND.LCOMP.NE.ICOMP.AND.C3.NE.0.AND.C4.NE.0.)        00031750
     1 GO TO 90                                                         00031760
   70 ZI1P(J)=ZHANKS(1,BB,F3PJ,EPS,LL,1)                                00031770
      CALL MODIFY(1)                                                    00031780
      ZI2P(J)=ZHANKS(0,BB,F4PJ,EPS,LL,0)                                00031790
   90 PHXPJ=C2*(FOUR*ZI1P(J)/BB-TWO*ZI2P(J))                            00031800
  100 RETURN                                                            00031810
  110 PHXPJ=(0.0,0.0)                                                   00031820
      GO TO 100                                                         00031830
      END                                                               00031840

      COMPLEX FUNCTION PHYPJ(X)                                         00031850
C--PARTIAL OF HY (DIPOLE FUNCTION) W/R P(J),J=JTH PARM IN (1,2*MM-1)    00031860
```

```
C  X IS DUMMY (IF L=0) ELSE X-XX USED IN RHO,ETC (IF L.NE.0)          00031870
C  (OTHER PARMS ASSUMED SET IN COMMON S)                             00031880
C                                                                    00031890
      REAL K(10),D(9),L                                              00031900
      COMPLEX F3PJ,F4PJ,ZHANKS,ZFLD,                                 00031910
     1 ZI1,ZI2,ZI1P(19),ZI2P(19),ZBES(5),ZLAM,ZERO,TWO,A3,A4         00031920
      EXTERNAL F3PJ,F4PJ,GF4,F4                                      00031930
      COMMON/MODEL/K,D,MM                                            00031940
      COMMON/SHARE/EPS,C2,C3,C4,XX,YY,YY2,RHO,RHO2,DELRHO,BB,        00031950
     1 L,DEL,DEL2,IREST(3)                                          00031960
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                        00031970
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                           00031980
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB            00031990
      COMMON/PART/J,ISEP                                             00032000
      DATA ZERO,TWO/(0.0,0.0),(2.0,0.0)/                             00032010
      IF(L.EQ.0.0) GO TO 10                                          00032020
      XD=X-XX                                                        00032030
      CALL SETRHO(X)                                                 00032040
      XR2=XD*XD/RHO2                                                 00032050
      C3=2.*(XR2-1.)/DEL2                                            00032060
      C4=2.*(1.-2.*XR2)/DELRHO                                       00032070
   10 IF(J.GT.1) GO TO 60                                            00032080
C--GET PARTIAL W/R SIG1 (J=1 AND SIG1 NOT HELD FIXED)                00032090
      ZI1P(1)=ZERO                                                   00032100
      ZI2P(1)=ZERO                                                   00032110
      IF(MM.EQ.1) GO TO 30                                           00032120
      A1=0.5*DEL2*C3                                                 00032130
      A2=0.5*DELRHO*C4                                               00032140
      IF(L.GT.0.0) ZI2=ZHANKS(0,BB,F4,EPS,LL,1)                      00032150
      IF(L.EQ.0.0) CALL SWAP(-1)                                     00032160
      CALL MODIFY(1)                                                 00032170
      ZI1P(1)=CMPLX(-A1*BB/SIG1,0.0)*ZHANKS(1,BB,GF4,EPS,LL,0)+      00032180
     & CMPLX(2.*(A1/SIG1+A2*DEL*BB/RHO),0.0)*ZI2                     00032190
      ZI2P(1)=CMPLX(2.*A2*DEL/RHO,0.0)*ZHANKS(1,BB,F3PJ,EPS,LL,1)    00032200
      CALL MODIFY(1)                                                 00032210
      ZI2P(1)=CMPLX(2.*A1,0.0)*ZHANKS(0,BB,F4PJ,EPS,LL,0)+ZI2P(1)    00032220
   30 IF(IOB.LT.5.OR.L.GT.0.0) GO TO 40                              00032230
C--CHECK FOR HX,HY SAVINGS--                                         00032240
      IF(LCOMP.NE.0.AND.LCOMP.NE.ICOMP) GO TO 50                     00032250
   40 CALL BESSIK(BB,ZBES)                                           00032260
   50 ZLAM=0.5*CMPLX(BB,BB)                                          00032270
      A4=2.*YY2/RHO2                                                 00032280
      A3=2.*A4-1.                                                    00032290
      PHYPJ=(-TWO*A3*ZBES(2)-ZLAM*(A3*(ZBES(4)-ZBES(3))+A4*ZLAM*     00032300
     & (-ZBES(1)+ZBES(2))))/CMPLX(SIG1*RHO2,0.0)                     00032310
      PHYPJ=(ZI1P(1)+ZI2P(1))/DEL2+PHYPJ                             00032320
      GO TO 100                                                      00032330
   60 IF(IOB.LT.5.OR.L.GT.0.0) GO TO 70                              00032340
C--CHECK FOR HX,HY SAVINGS--                                         00032350
      IF(LCOMP.NE.0.AND.LCOMP.NE.ICOMP.AND.C2.NE.0.) GO TO 90        00032360
   70 IF(C4.NE.0.0) ZI1P(J)=ZHANKS(1,BB,F3PJ,EPS,LL,1)               00032370
      IF(C3.EQ.0.0) GO TO 90                                         00032380
```

```
      IF(C4.NE.0.0) GO TO 75                                          00032390
      ZI2P(J)=ZHANKS(0,BB,F4PJ,EPS,LL,1)                              00032400
      GO TO 90                                                        00032410
   75 CALL MODIFY(1)                                                  00032420
      ZI2P(J)=ZHANKS(0,BB,F4PJ,EPS,LL,0)                              00032430
   90 PHYPJ=C3*ZI2P(J)+C4*ZI1P(J)                                     00032440
  100 RETURN                                                          00032450
      END                                                             00032460

      SUBROUTINE PRMHXY                                               00032470
C--PRIMARY HXP AND HYP-FIELDS,YY2,RHO,ETC FOR GIVEN X0,Y0, WHERE      00032480
C  ALL PARAMETERS (IN AND OUT) ARE STORED IN COMMON BLOCKS...          00032490
C                                                                     00032500
      COMPLEX ZFLD,ZI1,ZI2,ZI1P(19),ZI2P(19),ZBES(5)                  00032510
      REAL L                                                          00032520
      COMMON/SHARE/EPS,C2,C3,C4,X0,Y0,YY2,RHO,RHO2,DELRHO,BB,         00032530
     1 L,DEL,DEL2,IREST(3)                                            00032540
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                         00032550
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                             00032560
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB              00032570
      YY2=Y0*Y0                                                       00032580
      CALL SETRHO(0.0)                                                00032590
      IF(L.GT.0.0) GO TO 20                                           00032600
      R1=1.0/(RHO2*RHO2)                                              00032610
      HXP=2.0*X0*Y0*R1                                                00032620
      HYP=-R1*(X0*X0-YY2)                                             00032630
   10 RETURN                                                          00032640
   20 T1=X0-L                                                         00032650
      T2=X0+L                                                         00032660
      R1=1.0/(T1*T1+YY2)                                              00032670
      R2=1.0/(T2*T2+YY2)                                              00032680
      HXP=Y0*(R1-R2)                                                  00032690
      HYP=-(T1*R1-T2*R2)                                              00032700
      GO TO 10                                                        00032710
      END                                                             00032720

      SUBROUTINE MODIFY(N)                                            00032730
C--UTILITY TO MODIFY COMMON/SAVE/ AS FOLLOWS:                         00032740
C  N >0 TO REPLACE FSAVE(I)=FSAVE(I)*(GSAVE(I)**N), I=1,NSAVE.        00032750
C  N <0 TO REPLACE FSAVE(I)=FSAVE(I)/(GSAVE(I)**IABS(N)), I=1,NSAVE.  00032760
C--THIS MAY BE USED IN CONJUNCTION WITH SUBPROGRAM 'ZHANKS' TO         00032770
C  MODIFY SAVED KERNELS WHEN USING NEW=0 (SEE ZHANKS).                00032780
C                                                                     00032790
      COMPLEX FSAVE                                                   00032800
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                         00032810
      IF(N) 5,9,1                                                     00032820
1     IF(N.GT.1) GO TO 3                                              00032830
      DO 2 I=1,NSAVE                                                  00032840
2     FSAVE(I)=FSAVE(I)*CMPLX(GSAVE(I),0.0)                           00032850
      GO TO 9                                                         00032860
3     DO 4 I=1,NSAVE                                                  00032870
4     FSAVE(I)=FSAVE(I)*CMPLX(GSAVE(I)**N,0.0)                        00032880
```

```
      GO TO 9                                                          00032890
5     IF(N.LT.-1) GO TO 3                                              00032900
      DO 6 I=1,NSAVE                                                   00032910
6     FSAVE(I)=FSAVE(I)/CMPLX(GSAVE(I),0.0)                            00032920
9     RETURN                                                           00032930
      END                                                              00032940

      SUBROUTINE SWAP(ICODE)                                           00032950
C--UTILITY TO SWAP COMMON/SAVE/ AS FOLLOWS:                            00032960
C  ICODE =1 TO SWAP COMMON/SAVE/ TO INTERNAL TEMP STORAGE.             00032970
C        =-1 TO RESWAP INTERNAL TEMP STORAGE TO COMMON/SAVE/.          00032980
C                                                                      00032990
C--THIS MAY BE USED IN CONJUNCTION WITH SUBPROGRAM 'ZHANKS' TO USE     00033000
C  DIFFERENT CLASSES OF INTEGRALS.  ALSO, SEE THE UTILITY              00033010
C  SUBROUTINE 'MODIFY'.                                                00033020
C                                                                      00033030
      COMPLEX FSAVE,FSWAP                                              00033040
      DIMENSION FSWAP(283),GSWAP(283)                                  00033050
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                          00033060
      IF(ICODE) 3,1,1                                                  00033070
1     DO 2 I=1,NSAVE                                                   00033080
      FSWAP(I)=FSAVE(I)                                                00033090
2     GSWAP(I)=GSAVE(I)                                                00033100
      NSWAP=NSAVE                                                      00033110
      RETURN                                                           00033120
3     DO 4 I=1,NSWAP                                                   00033130
      FSAVE(I)=FSWAP(I)                                                00033140
4     GSAVE(I)=GSWAP(I)                                                00033150
      NSAVE=NSWAP                                                      00033160
      RETURN                                                           00033170
      END                                                              00033180

      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                      00033190
C======================================================================00033200
C  COMPLEX HANKEL TRANSFORMS OF ORDER 0 OR 1 FOR RELATED (SAVED) KERNELS00033210
C  AND FIXED TRANSFORM ARGUMENT B.GT.0.                                00033220
C                                                                      00033230
C--REF: ANDERSON, W.L., 1979, GEOPHYSICS, VOL. 44, NO. 7, P. 1287-1305. 00033240
C                                                                      00033250
C--SUBPROGRAM ZHANKS EVALUATES THE INTEGRAL FROM 0 TO INFINITY OF      00033260
C  FUN(G)*JN(G*B)*DG, DEFINED AS THE COMPLEX HANKEL TRANSFORM OF       00033270
C  ORDER N (=0 OR 1) AND TRANSFORM ARGUMENT B.GT.0.  THE METHOD IS BY  00033280
C  ADAPTIVE DIGITAL FILTERING OF THE COMPLEX KERNEL FUNCTION FUN,       00033290
C  USING DIRECT AND/OR PREVIOUSLY SAVED KERNEL FUNCTION VALUES.        00033300
C                                                                      00033310
C--PARAMETERS (ALL INPUT, EXCEPT NF)                                   00033320
C                                                                      00033330
C     N      = ORDER (=0 OR 1) OF THE HANKEL TRANSFORM TO BE EVALUATED. 00033340
C     B      = REAL TRANSFORM ARGUMENT B.GT.0.0 OF THE HANKEL TRANSFORM. 00033350
C              IF NEW=0, B IS ASSUMED EQUAL TO THE LAST B USED WHEN NEW=100033360
C              (SEE PARAMETER NEW AND SUBPROGRAM USAGE BELOW).         00033370
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)  00033380
```

```
C              OF A REAL ARGUMENT G.GT.0. THIS REFERENCE MUST BE SUPPLIED00033390
C              EVEN WHEN NEW=0, SINCE THE ADAPTIVE CONVOLUTION           00033400
C              MAY NEED SOME DIRECT FUNCTION CALLS (E.G. IF TOL REDUCED).00033410
C              IF PARAMETERS OTHER THAN G ARE REQUIRED IN FUN, USE COMMON00033420
C              IN THE CALLING PROGRAM AND IN SUBPROGRAM FUN.  BOTH       00033430
C              REAL AND IMAGINARY PARTS OF THE COMPLEX FUNCTION FUN(G)   00033440
C              MUST BE CONTINUOUS BOUNDED FUNCTIONS FOR G.GT.0.0. FOR A  00033450
C              REAL FUNCTION F1(G), FUN=CMPLX(F1(G),0.0) MAY BE USED.    00033460
C              TWO INDEPENDENT REAL-FUNCTIONS F1(G),F2(G) MAY BE         00033470
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)). 00033480
C     TOL    = REQUESTED REAL TRUNCATION TOLERANCE ACCEPTED AT THE FILTER00033490
C              TAILS FOR ADAPTIVE FILTERING.  A TRUNCATION CRITERION IS  00033500
C              DEFINED DURING CONVOLUTION IN A FIXED ABSCISSA RANGE AS   00033510
C              THE MAX. ABSOLUTE CONVOLVED PRODUCT TIMES TOL.  TYPICALLY,00033520
C              TOL.LE.0.00001 WOULD GIVE ABOUT .01 PER CENT ACCURACY     00033530
C              FOR WELL-BEHAVED KERNELS AND MODERATE VALUES OF B.  FOR   00033540
C              VERY LARGE OR SMALL B, A VERY SMALL TOL SHOULD BE USED.   00033550
C              IN GENERAL, DECREASING THE TOLERANCE WOULD PRODUCE HIGHER 00033560
C              ACCURACY IN THE CONVOLUTION SINCE MORE FILTER WEIGHTS ARE 00033570
C              USED (UNLESS EXPONENT UNDERFLOWS OCCUR IN THE KERNEL      00033580
C              EVALUATION -- SEE NOTE (1) BELOW).                        00033590
C              FOR MAXIMUM ACCURACY POSSIBLE, TOL=0.0 MAY BE USED.       00033600
C     NF     = TOTAL NUMBER OF DIRECT FUN CALLS USED DURING CONVOLUTION  00033610
C              FOR ANY VALUE OF NEW (NF IS AN OUTPUT PARAMETER).         00033620
C              NF IS IN THE RANGE 21.LE.NF.LE.283 WHEN NEW=1.  USUALLY,  00033630
C              NF IS MUCH LESS THAN 283 (OR 0) WHEN NEW=0.               00033640
C     NEW   =1 IS REQUIRED FOR THE VERY FIRST CALL TO ZHANKS, OR IF      00033650
C              FORCING DIRECT FUNCTION FUN(G) CALLS, E.G., IF USING      00033660
C              ZHANKS FOR UNRELATED KERNELS.                             00033670
C              NEW=1 INITIALIZES COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE 00033680
C              FOR NSAVE COMPLEX KERNEL VALUES IN FSAVE AND CORRESPONDING00033690
C              REAL ARGUMENTS IN GSAVE FOR THE GIVEN PARAMETER B.        00033700
C     NEW   =0 TO USE RELATED KERNELS (MODIFIED BY USER) CURRENTLY STORED00033710
C              IN COMMON/SAVE/. FUN IS CALLED ONLY IF REQUIRED           00033720
C              DURING THE CONVOLUTION.  ADDITIONAL FUNCTION VALUES WHEN  00033730
C              NEEDED ARE AUTOMATICALLY ADDED TO THE COMMON/SAVE/ BLOCK. 00033740
C                                                                        00033750
C     ******* NOTE THAT IT IS THE USERS RESPONSIBILITY TO MODIFY THE     00033760
C              COMMON FSAVE() VALUES FOR NEW=0 CALLS, EXTERNALLY IN      00033770
C              THE USERS CALLING PROGRAM (SEE SUBPROGRAM USAGE BELOW).   00033780
C                                                                        00033790
C=========================================================================00033800
C--SUBPROGRAM USAGE-- ZHANKS IS CALLED AS FOLLOWS                        00033810
C     ...                                                                00033820
C     COMPLEX Z1,Z2,ZHANKS,FSAVE                                         00033830
C     COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                            00033840
C     EXTERNAL ZF1,ZF2                                                   00033850
C     ...                                                                00033860
C     Z1=ZHANKS(N1,B,ZF1,TOL,NF1,1)                                      00033870
C     DO 1 I=1,NSAVE                                                     00033880
C  C--MODIFY FSAVE IN COMMON/SAVE/ TO OBTAIN RELATED ZF2 FROM ZF1.       00033890
C  C--E.G. FSAVE(I)=GSAVE(I)*FSAVE(I) -- FOR RELATION ZF2(G)=G*ZF1(G)    00033900
```

```
C   1 CONTINUE                                                          00033910
C     Z2=ZHANKS(N2,B,ZF2,TOL,NF2,0)                                     00033920
C     ...                                                               00033930
C     END                                                               00033940
C     COMPLEX FUNCTION ZF1(G)                                           00033950
C     ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF1(G), G.GT.0.    00033960
C     END                                                               00033970
C     COMPLEX FUNCTION ZF2(G)                                           00033980
C     ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF2(G), G.GT.0.    00033990
C     END                                                               00034000
C=====================================================================00034010
C--NOTES                                                               00034020
C       (1).  EXP-UNDERFLOW MAY OCCUR IN EXECUTING THIS SUBPROGRAM.    00034030
C             THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS 00034040
C             EXP-UNDERFLOW TO 0.0.                                     00034050
C       (2).  ANSI FORTRAN (AMERICAN STANDARD X3.9-1966) IS USED, EXCEPT00034060
C             DATA STATEMENTS MAY NEED TO BE CHANGED FOR SOME COMPILERS.00034070
C             TO CONVERT ZHANKS TO THE NEW AMERICAN STANDARD FORTRAN    00034080
C             (X3.9-1978), ADD THE FOLLOWING DECLARATION TO THIS ROUTINE00034090
C             SAVE Y1,ISAVE                                             00034100
C       (3).  THE FILTER ABSCISSA CORRESPONDING TO EACH FILTER WEIGHT   00034110
C             IS GENERATED IN DOUBLE-PRECISION (TO REDUCE ROUND-OFF),   00034120
C             BUT IS USED IN SINGLE-PRECISION IN FUNCTION FUN.          00034130
C       (4).  NO CHECKS ARE MADE ON CALLING PARAMETERS (TO SAVE TIME),  00034140
C             HENCE UNPREDICTABLE RESULTS COULD OCCUR IF ZHANKS         00034150
C             IS CALLED INCORRECTLY (OR IF FUN OR COMMON IS IN ERROR).  00034160
C=====================================================================00034170
C                                                                       00034180
      COMPLEX FUN,C,CMAX,FSAVE                                          00034190
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                           00034200
      DOUBLE PRECISION E,ER,Y1,Y                                        00034210
      DIMENSION T(2),TMAX(2)                                            00034220
      DIMENSION WT0(283),WA0(76),WB0(76),WC0(76),WD0(55),               00034230
     * WT1(283),WA1(76),WB1(76),WC1(76),WD1(55)                        00034240
      EQUIVALENCE (WT0(1),WA0(1)),(WT0(77),WB0(1)),(WT0(153),WC0(1)),   00034250
     * (WT0(229),WD0(1)),(WT1(1),WA1(1)),(WT1(77),WB1(1)),              00034260
     * (WT1(153),WC1(1)),(WT1(229),WD1(1))                              00034270
      EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                               00034280
C-----E=DEXP(.2D0), ER=1.0D0/E                                          00034290
      DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/        00034300
C--J0-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT0 ARRAY)          00034310
      DATA WA0/                                                         00034320
     * 2.1969101E-11, 4.1201161E-09,-6.1322980E-09, 7.2479291E-09,      00034330
     *-7.9821627E-09, 8.5778983E-09,-9.1157294E-09, 9.6615250E-09,      00034340
     *-1.0207546E-08, 1.0796633E-08,-1.1393033E-08, 1.2049873E-08,      00034350
     *-1.2708789E-08, 1.3446466E-08,-1.4174300E-08, 1.5005577E-08,      00034360
     *-1.5807160E-08, 1.6747136E-08,-1.7625961E-08, 1.8693427E-08,      00034370
     *-1.9650840E-08, 2.0869789E-08,-2.1903555E-08, 2.3305308E-08,      00034380
     *-2.4407377E-08, 2.6033678E-08,-2.7186773E-08, 2.9094334E-08,      00034390
     *-3.0266804E-08, 3.2534013E-08,-3.3672072E-08, 3.6408936E-08,      00034400
     *-3.7425022E-08, 4.0787921E-08,-4.1543242E-08, 4.5756842E-08,      00034410
     *-4.6035233E-08, 5.1425075E-08,-5.0893896E-08, 5.7934897E-08,      00034420
```

```
*-5.6086570E-08, 6.5475248E-08,-6.1539913E-08, 7.4301996E-08,      00034430
*-6.7117043E-08, 8.4767837E-08,-7.2583120E-08, 9.7366568E-08,      00034440
*-7.7553611E-08, 1.1279873E-07,-8.1416723E-08, 1.3206914E-07,      00034450
*-8.3217217E-08, 1.5663185E-07,-8.1482581E-08, 1.8860593E-07,      00034460
*-7.3963141E-08, 2.3109673E-07,-5.7243707E-08, 2.8867452E-07,      00034470
*-2.6163525E-08, 3.6808773E-07, 2.7049871E-08, 4.7932617E-07,      00034480
* 1.1407365E-07, 6.3720626E-07, 2.5241961E-07, 8.6373487E-07,      00034490
* 4.6831433E-07, 1.1916346E-06, 8.0099716E-07, 1.6696015E-06,      00034500
* 1.3091334E-06, 2.3701475E-06, 2.0803829E-06, 3.4012978E-06/      00034510
 DATA WBO/                                                        00034520
* 3.2456774E-06, 4.9240402E-06, 5.0005198E-06, 7.1783540E-06,      00034530
* 7.6367633E-06, 1.0522038E-05, 1.1590021E-05, 1.5488635E-05,      00034540
* 1.7510398E-05, 2.2873836E-05, 2.6368006E-05, 3.3864387E-05,      00034550
* 3.9610390E-05, 5.0230379E-05, 5.9397373E-05, 7.4612122E-05,      00034560
* 8.8951409E-05, 1.1094809E-04, 1.3308026E-04, 1.6511335E-04,      00034570
* 1.9895671E-04, 2.4587195E-04, 2.9728181E-04, 3.6629770E-04,      00034580
* 4.4402013E-04, 5.4589361E-04, 6.6298832E-04, 8.1375348E-04,      00034590
* 9.8971624E-04, 1.2132772E-03, 1.4772052E-03, 1.8092022E-03,      00034600
* 2.2045122E-03, 2.6980811E-03, 3.2895354E-03, 4.0238764E-03,      00034610
* 4.9080203E-03, 6.0010999E-03, 7.3216878E-03, 8.9489225E-03,      00034620
* 1.0919448E-02, 1.3340696E-02, 1.6276399E-02, 1.9873311E-02,      00034630
* 2.4233627E-02, 2.9555699E-02, 3.5990069E-02, 4.3791529E-02,      00034640
* 5.3150319E-02, 6.4341372E-02, 7.7506720E-02, 9.2749987E-02,      00034650
* 1.0980561E-01, 1.2791555E-01, 1.4525830E-01, 1.5820085E-01,      00034660
* 1.6058576E-01, 1.4196085E-01, 8.9781222E-02,-1.0238278E-02,      00034670
*-1.5083434E-01,-2.9059573E-01,-2.9105437E-01,-3.7973244E-02,      00034680
* 3.8273717E-01, 2.2014118E-01,-4.7342635E-01, 1.9331133E-01,      00034690
* 5.3839527E-02,-1.1909845E-01, 9.9317051E-02,-6.6152628E-02,      00034700
* 4.0703241E-02,-2.4358316E-02, 1.4476533E-02,-8.6198067E-03/      00034710
 DATA WCO/                                                        00034720
* 5.1597053E-03,-3.1074602E-03, 1.8822342E-03,-1.1456545E-03,      00034730
* 7.0004347E-04,-4.2904226E-04, 2.6354444E-04,-1.6215439E-04,      00034740
* 9.9891279E-05,-6.1589037E-05, 3.7996921E-05,-2.3452250E-05,      00034750
* 1.4479572E-05,-8.9417427E-06, 5.5227518E-06,-3.4114252E-06,      00034760
* 2.1074101E-06,-1.3019229E-06, 8.0433617E-07,-4.9693681E-07,      00034770
* 3.0702417E-07,-1.8969219E-07, 1.1720069E-07,-7.2412496E-08,      00034780
* 4.4740283E-08,-2.7643004E-08, 1.7079403E-08,-1.0552634E-08,      00034790
* 6.5200311E-09,-4.0284597E-09, 2.4890232E-09,-1.5378695E-09,      00034800
* 9.5019040E-10,-5.8708696E-10, 3.6273937E-10,-2.2412348E-10,      00034810
* 1.3847792E-10,-8.5560821E-11, 5.2865474E-11,-3.2664392E-11,      00034820
* 2.0182948E-11,-1.2470979E-11, 7.7057678E-12,-4.7611713E-12,      00034830
* 2.9415274E-12,-1.8170081E-12, 1.1221034E-12,-6.9271067E-13,      00034840
* 4.2739744E-13,-2.6344388E-13, 1.6197105E-13,-9.9147443E-14,      00034850
* 6.0487998E-14,-3.6973097E-14, 2.2817964E-14,-1.4315547E-14,      00034860
* 9.1574735E-15,-5.9567236E-15, 3.9209969E-15,-2.5911739E-15,      00034870
* 1.6406939E-15,-8.8248590E-16, 3.0195409E-16, 2.2622634E-17,      00034880
*-8.0942556E-17,-3.7172363E-17, 1.9299542E-16,-3.3388160E-16,      00034890
* 4.6174116E-16,-5.8627358E-16, 7.2227767E-16,-8.7972941E-16,      00034900
* 1.0211793E-15,-1.0940039E-15, 1.0789555E-15,-9.7089714E-16/      00034910
 DATA WDO/                                                        00034920
* 7.4110927E-16,-4.1700094E-16, 8.5977184E-17, 1.3396469E-16,      00034930
*-1.7838410E-16, 4.8975421E-17, 1.9398153E-16,-5.0046989E-16,      00034940
```

```
     * 8.3280985E-16,-1.1544640E-15, 1.4401527E-15,-1.6637066E-15,      00034950
     * 1.7777129E-15,-1.7322187E-15, 1.5247247E-15,-1.1771155E-15,      00034960
     * 6.9747910E-16,-1.2088956E-16,-4.8382957E-16, 1.0408292E-15,      00034970
     *-1.5220450E-15, 1.9541597E-15,-2.4107448E-15, 2.9241438E-15,      00034980
     *-3.5176475E-15, 4.2276125E-15,-5.0977851E-15, 6.1428456E-15,      00034990
     *-7.3949962E-15, 8.8597601E-15,-1.0515959E-14, 1.2264584E-14,      00035000
     *-1.3949870E-14, 1.5332490E-14,-1.6146782E-14, 1.6084121E-14,      00035010
     *-1.4962523E-14, 1.2794804E-14,-9.9286701E-15, 6.8825809E-15,      00035020
     *-4.0056107E-15, 1.5965079E-15,-7.2732961E-18,-4.0433218E-16,      00035030
     *-6.5679655E-16, 3.3011866E-15,-7.3545910E-15, 1.2394851E-14,      00035040
     *-1.7947697E-14, 2.3774303E-14,-3.0279168E-14, 3.9252831E-14,      00035050
     *-5.5510504E-14, 9.0505371E-14,-1.7064873E-13/                     00035060
C--END OF J0 FILTER WEIGHTS                                             00035070
C                                                                       00035080
C--J1-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT1 ARRAY)          00035090
      DATA WA1/                                                         00035100
     *-4.2129715E-16, 5.3667031E-15,-7.1183962E-15, 8.9478500E-15,      00035110
     *-1.0767891E-14, 1.2362265E-14,-1.3371129E-14, 1.3284178E-14,      00035120
     *-1.1714302E-14, 8.4134738E-15,-3.7726725E-15,-1.4263879E-15,      00035130
     * 6.1279163E-15,-9.1102765E-15, 9.9696405E-15,-9.3649955E-15,      00035140
     * 8.6009018E-15,-8.9749846E-15, 1.1153987E-14,-1.4914821E-14,      00035150
     * 1.9314024E-14,-2.3172388E-14, 2.5605477E-14,-2.6217555E-14,      00035160
     * 2.5057768E-14,-2.2485539E-14, 1.9022752E-14,-1.5198084E-14,      00035170
     * 1.1422464E-14,-7.9323958E-15, 4.8421406E-15,-2.1875032E-15,      00035180
     *-3.2177842E-17, 1.8637565E-15,-3.3683643E-15, 4.6132219E-15,      00035190
     *-5.6209538E-15, 6.4192841E-15,-6.8959928E-15, 6.9895792E-15,      00035200
     *-6.5355935E-15, 5.6125163E-15,-4.1453931E-15, 2.6358827E-15,      00035210
     *-9.5104370E-16, 1.4600474E-16, 5.6166519E-16, 8.2899246E-17,      00035220
     * 5.0032100E-16, 4.3752205E-16, 2.1052293E-15,-9.5451973E-16,      00035230
     * 6.4004437E-15,-2.1926177E-15, 1.1651003E-14, 5.8415433E-16,      00035240
     * 1.8044664E-14, 1.0755745E-14, 3.0159022E-14, 3.3506138E-14,      00035250
     * 5.8709354E-14, 8.1475200E-14, 1.2530006E-13, 1.8519112E-13,      00035260
     * 2.7641786E-13, 4.1330823E-13, 6.1506209E-13, 9.1921659E-13,      00035270
     * 1.3698462E-12, 2.0447427E-12, 3.0494477E-12, 4.5501001E-12,      00035280
     * 6.7870250E-12, 1.0126237E-11, 1.5104976E-11, 2.2536053E-11/      00035290
      DATA WB1/                                                         00035300
     * 3.3617368E-11, 5.0153839E-11, 7.4818173E-11, 1.1161804E-10,      00035310
     * 1.6651222E-10, 2.4840923E-10, 3.7058109E-10, 5.5284353E-10,      00035320
     * 8.2474468E-10, 1.2303750E-09, 1.8355034E-09, 2.7382502E-09,      00035330
     * 4.0849867E-09, 6.0940898E-09, 9.0913020E-09, 1.3562651E-08,      00035340
     * 2.0233058E-08, 3.0184244E-08, 4.5029477E-08, 6.7176304E-08,      00035350
     * 1.0021488E-07, 1.4950371E-07, 2.2303208E-07, 3.3272689E-07,      00035360
     * 4.9636623E-07, 7.4049804E-07, 1.1046805E-06, 1.6480103E-06,      00035370
     * 2.4585014E-06, 3.6677163E-06, 5.4714550E-06, 8.1626422E-06,      00035380
     * 1.2176782E-05, 1.8166179E-05, 2.7099223E-05, 4.0428804E-05,      00035390
     * 6.0307294E-05, 8.9971508E-05, 1.3420195E-04, 2.0021123E-04,      00035400
     * 2.9860417E-04, 4.4545291E-04, 6.6423156E-04, 9.9073275E-04,      00035410
     * 1.4767050E-03, 2.2016806E-03, 3.2788147E-03, 4.8837292E-03,      00035420
     * 7.2596811E-03, 1.0788355E-02, 1.5973323E-02, 2.3612041E-02,      00035430
     * 3.4655327E-02, 5.0608141E-02, 7.2827752E-02, 1.0337889E-01,      00035440
     * 1.4207357E-01, 1.8821315E-01, 2.2996815E-01, 2.5088500E-01,      00035450
     * 2.0334626E-01, 6.0665451E-02,-2.0275683E-01,-3.5772336E-01,      00035460
```

```
     *-1.8280529E-01, 4.7014634E-01, 7.2991233E-03,-3.0614594E-01,      00035470
     * 2.4781735E-01,-1.1149185E-01, 2.5985386E-02, 1.0850279E-02,      00035480
     *-2.2830217E-02, 2.4644647E-02,-2.2895284E-02, 2.0197032E-02/      00035490
      DATA WC1/                                                         00035500
     *-1.7488968E-02, 1.5057670E-02,-1.2953923E-02, 1.1153254E-02,      00035510
     *-9.6138436E-03, 8.2952090E-03,-7.1628361E-03, 6.1882910E-03,      00035520
     *-5.3482055E-03, 4.6232056E-03,-3.9970542E-03, 3.4560118E-03,      00035530
     *-2.9883670E-03, 2.5840861E-03,-2.2345428E-03, 1.9323046E-03,      00035540
     *-1.6709583E-03, 1.4449655E-03,-1.2495408E-03, 1.0805480E-03,      00035550
     *-9.3441130E-04, 8.0803899E-04,-6.9875784E-04, 6.0425624E-04,      00035560
     *-5.2253532E-04, 4.5186652E-04,-3.9075515E-04, 3.3790861E-04,      00035570
     *-2.9220916E-04, 2.5269019E-04,-2.1851585E-04, 1.8896332E-04,      00035580
     *-1.6340753E-04, 1.4130796E-04,-1.2219719E-04, 1.0567099E-04,      00035590
     *-9.1379828E-05, 7.9021432E-05,-6.8334412E-05, 5.9092726E-05,      00035600
     *-5.1100905E-05, 4.4189914E-05,-3.8213580E-05, 3.3045496E-05,      00035610
     *-2.8576356E-05, 2.4711631E-05,-2.1369580E-05, 1.8479514E-05,      00035620
     *-1.5980307E-05, 1.3819097E-05,-1.1950174E-05, 1.0334008E-05,      00035630
     *-8.9364160E-06, 7.7278366E-06,-6.6827083E-06, 5.7789251E-06,      00035640
     *-4.9973715E-06, 4.3215167E-06,-3.7370660E-06, 3.2316575E-06,      00035650
     *-2.7946015E-06, 2.4166539E-06,-2.0898207E-06, 1.8071890E-06,      00035660
     *-1.5627811E-06, 1.3514274E-06,-1.1686576E-06, 1.0106059E-06,      00035670
     *-8.7392952E-07, 7.5573750E-07,-6.5353002E-07, 5.6514528E-07,      00035680
     *-4.8871388E-07, 4.2261921E-07,-3.6546333E-07, 3.1603732E-07/      00035690
      DATA WD1/                                                         00035700
     *-2.7329579E-07, 2.3633470E-07,-2.0437231E-07, 1.7673258E-07,      00035710
     *-1.5283091E-07, 1.3216174E-07,-1.1428792E-07, 9.8831386E-08,      00035720
     *-8.5465227E-08, 7.3906734E-08,-6.3911437E-08, 5.5267923E-08,      00035730
     *-4.7793376E-08, 4.1329702E-08,-3.5740189E-08, 3.0906612E-08,      00035740
     *-2.6726739E-08, 2.3112160E-08,-1.9986424E-08, 1.7283419E-08,      00035750
     *-1.4945974E-08, 1.2924650E-08,-1.1176694E-08, 9.6651347E-09,      00035760
     *-8.3580023E-09, 7.2276490E-09,-6.2501673E-09, 5.4048822E-09,      00035770
     *-4.6739154E-09, 4.0418061E-09,-3.4951847E-09, 3.0224895E-09,      00035780
     *-2.6137226E-09, 2.2602382E-09,-1.9545596E-09, 1.6902214E-09,      00035790
     *-1.4616324E-09, 1.2639577E-09,-1.0930164E-09, 9.4519327E-10,      00035800
     *-8.1736202E-10, 7.0681930E-10,-6.1122713E-10, 5.2856342E-10,      00035810
     *-4.5707937E-10, 3.9526267E-10,-3.4180569E-10, 2.9557785E-10,      00035820
     *-2.5560176E-10, 2.2103233E-10,-1.9113891E-10, 1.6528994E-10,      00035830
     *-1.4294012E-10, 1.2361991E-10,-8.2740936E-11/                     00035840
C--END OF J1 FILTER WEIGHTS                                             00035850
C                                                                       00035860
      NONE=0                                                            00035870
      IF(NEW.EQ.0) GO TO 100                                            00035880
      NSAVE=0                                                           00035890
C-----INITIALIZE KERNEL ABSCISSA GENERATION FOR GIVEN B                 00035900
      Y1=0.73588526614797944600D0/DBLE(B)                               00035910
  100 ZHANKS=(0.0,0.0)                                                  00035920
      CMAX=(0.0,0.0)                                                    00035930
      NF=0                                                              00035940
      Y=Y1                                                              00035950
C-----BEGIN RIGHT-SIDE CONVOLUTION AT WEIGHT 131 (EITHER NEW=1 OR 0)    00035960
      ASSIGN 110 TO M                                                   00035970
      I=131                                                             00035980
```

```
      Y=Y*E                                                          00035990
      GO TO 200                                                      00036000
  110 TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                               00036010
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                               00036020
      I=I+1                                                          00036030
      Y=Y*E                                                          00036040
      IF(I.LE.149) GO TO 200                                         00036050
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) NONE=1                   00036060
C-----ESTABLISH TRUNCATION CRITERION (CMAX=CMPLX(TMAX(1),TMAX(2))    00036070
      CMAX=TOL*CMAX                                                  00036080
      ASSIGN 120 TO M                                                00036090
      GO TO 200                                                      00036100
C-----CHECK FOR FILTER TRUNCATION AT RIGHT END                       00036110
  120 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130    00036120
      I=I+1                                                          00036130
      Y=Y*E                                                          00036140
      IF(I.LE.283) GO TO 200                                         00036150
  130 Y=Y1                                                           00036160
C-----CONTINUE WITH LEFT-SIDE CONVOLUTION AT WEIGHT 130              00036170
      ASSIGN 140 TO M                                                00036180
      I=130                                                          00036190
      GO TO 200                                                      00036200
C-----CHECK FOR FILTER TRUNCATION AT LEFT END                        00036210
  140 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2).AND.          00036220
     * NONE.EQ.0) GO TO 190                                         00036230
      I=I-1                                                          00036240
      Y=Y*ER                                                         00036250
      IF(I.GT.0) GO TO 200                                           00036260
C-----RETURN WITH ISAVE=1 PRESET FOR POSSIBLE NEW=0 USE.             00036270
  190 ISAVE=1                                                        00036280
C-----NORMALIZE BY B TO ACCOUNT FOR INTEGRATION RANGE CHANGE         00036290
      ZHANKS=ZHANKS/B                                                00036300
      RETURN                                                         00036310
C-----SAVE/RETRIEVE PSEUDO-SUBROUTINE (CALL FUN ONLY WHEN NECESSARY) 00036320
  200 G=SNGL(Y)                                                      00036330
      IF(NEW) 300,210,300                                            00036340
  210 IF(ISAVE.GT.NSAVE) GO TO 300                                   00036350
      ISAVE0=ISAVE                                                   00036360
  220 IF(G.EQ.GSAVE(ISAVE)) GO TO 240                                00036370
      ISAVE=ISAVE+1                                                  00036380
      IF(ISAVE.LE.NSAVE) GO TO 220                                   00036390
      ISAVE=ISAVE0                                                   00036400
C-----G NOT IN COMMON/SAVE/----- EVALUATE FUN.                       00036410
      GO TO 300                                                      00036420
C-----G FOUND IN COMMON/SAVE/----- USE FSAVE AS GIVEN.               00036430
  240 C=FSAVE(ISAVE)                                                 00036440
      ISAVE=ISAVE+1                                                  00036450
C-----SWITCH ON ORDER N                                              00036460
  250 IF(N) 270,260,270                                              00036470
  260 C=C*WT0(I)                                                     00036480
      GO TO 280                                                      00036490
  270 C=C*WT1(I)                                                     00036500
```

```
  280 ZHANKS=ZHANKS+C                                                    00036510
      GO TO M,(110,120,140)                                              00036520
C-----DIRECT FUN EVALUATION (AND ADD TO END OF COMMON/SAVE/)             00036530
  300 NSAVE=NSAVE+1                                                      00036540
      C=FUN(G)                                                           00036550
      NF=NF+1                                                            00036560
      FSAVE(NSAVE)=C                                                     00036570
      GSAVE(NSAVE)=G                                                     00036580
      GO TO 250                                                          00036590
      END                                                                00036600

      SUBROUTINE FCODE(Y,X,B,PRNT,F,I,IDER)                              00036610
C--FUNCTION EVALUATION FOR NORMALIZED HX OR HY                           00036620
C                                                                        00036630
C--PARAMETERS--                                                          00036640
C                                                                        00036650
C       Y=      OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)               00036660
C       X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)           00036670
C       B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)               00036680
C       PRNT=   WORK AND PRINT ARRAY (DIM. 5)                            00036690
C       F=      OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. I    00036700
C       I=      OBSERVATION NO. TO EVAL. F (1<=I<=N)                     00036710
C       IDER=   0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED)  00036720
C               1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED)  00036730
C                                                                        00036740
      REAL Y(1),X(200,5),B(1),PRNT(5),F,EPS                              00036750
      COMPLEX HX03,HY03,CERR,CANC4,ZFLD,ZI1,ZI2,ZI1P(19),ZI2P(19),       00036760
     1 ZBES(5),ERRFIN,FINHX,FINHY                                        00036770
      REAL K(10),D(9),L                                                  00036780
      COMMON/MODEL/K,D,MM                                                00036790
      COMMON/SHARE/EPS,C2,C3,C4,XX,YY,YY2,RHO,RHO2,DELRHO,BB,            00036800
     1 L,DEL,DEL2,METHOD,IREST(2)                                        00036810
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                            00036820
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                                00036830
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB                 00036840
      COMMON/FIN/R1,R2,R0,XLEN,FILL,XFIN,YFIN                            00036850
      COMMON/THICK/DIN(9)                                                00036860
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEVFIN,MEVFIN,ERRFIN,LW    00036870
      COMMON/PART/JJ,ISEP                                                00036880
      EXTERNAL HX03,HY03                                                 00036890
      EQUIVALENCE (DUM,NEV)                                              00036900
      DATA FREQL/0.0/,LCOMP/0/                                           00036910
      IF(L.GT.0.0.AND.METHOD.EQ.0.AND.IDER.EQ.0)                         00036920
     & CALL ERRMSG('IDER=0 NOT AVAILABLE WHEN METHOD=0 AND L>0.0 ',      00036930
     & 9,6,16)                                                           00036940
      IF(I.GT.1.OR.MM.EQ.1) GO TO 20                                     00036950
      DO 10 J=2,MM                                                       00036960
      IF(B(J).EQ.B(J-1)) CALL ERRMSG(20HSOME SIG(J)=SIG(J-1),4,6,16)     00036970
   10 CONTINUE                                                           00036980
   20 DO 30 J=1,5                                                        00036990
   30 PRNT(J)=X(I,J)                                                     00037000
      IF(IOB.NE.6) GO TO 40                                              00037010
```

```
      ISEP=0                                                          00037020
      IF(XX.NE.PRNT(2).OR.YY.NE.PRNT(3)) ISEP=1                       00037030
   40 CONTINUE                                                        00037040
      FREQ=PRNT(1)                                                    00037050
      IF(I.EQ.1.OR.IDER.NE.0.OR.FREQ.NE.FREQL) GO TO 50               00037060
      JUMP=1                                                          00037070
      IF(IOB.EQ.5) GO TO 220                                          00037080
      IF(IOB.EQ.6.AND.ISEP.EQ.0) GO TO 240                            00037090
   50 JUMP=0                                                          00037100
      SIG1=B(1)                                                       00037110
      DEL2=1.0/(39.47841762E-7*SIG1*FREQ)                             00037120
      DEL=SQRT(DEL2)                                                  00037130
      IF(IOB.NE.6.OR.ISEP.EQ.0) GO TO 60                              00037140
      XX=PRNT(2)                                                      00037150
      YY=PRNT(3)                                                      00037160
      CALL PRMHXY                                                     00037170
      GO TO 70                                                        00037180
   60 IF(L.EQ.0.0) CALL SETRHO(DUM)                                   00037190
   70 IF(I.EQ.1) IEPS=IEPS+1                                          00037200
      IF(IEPS.EQ.NEPS) EP=EPS                                         00037210
      IF(L.GT.0.0) GO TO 80                                           00037220
      C2=XX*YY/(DELRHO**2)                                            00037230
      XR2=XX*XX/RHO2                                                  00037240
      C3=2.*(XR2-1.)/DEL2                                             00037250
      C4=2.*(1.-2.*XR2)/DELRHO                                        00037260
   80 IF(MM.EQ.1) GO TO 100                                           00037270
      DO 90 J=1,M1                                                    00037280
      K(J)=B(J)/SIG1                                                  00037290
      DIN(J)=B(J+MM)                                                  00037300
   90 D(J)=2.0*B(J+MM)/DEL                                            00037310
  100 K(MM)=B(MM)/SIG1                                                00037320
      ICOMP=0                                                         00037330
      IF(IOB.EQ.5) GO TO 220                                          00037340
      IF(IOB.EQ.6) GO TO 240                                          00037350
      LCOMP=0                                                         00037360
      ICOMP=1                                                         00037370
      IF(IOB.LT.0) ICOMP=-1                                           00037380
C                                                                     00037390
C--GET COMPLEX NORMALIZED FUNCTION (HX/HXP OR HY/HYP)                 00037400
  110 IF(L.GT.0.0) GO TO 130                                          00037410
      IF(ICOMP.EQ.1) ZFLD=HX03(DUM)/HXP                               00037420
      IF(ICOMP.NE.1) ZFLD=HY03(DUM)/HYP                               00037430
  120 IF(ICOMP.EQ.1) ZFLD=B(M2)*ZFLD                                  00037440
      IF(ICOMP.NE.1) ZFLD=B(M2P1)*ZFLD                                00037450
      IF(JUMP.EQ.1) GO TO (170,180,190,200),IOBS                      00037460
      GO TO (170,180,190,200,220,240),IIOB                            00037470
  130 IF(METHOD.NE.0) GO TO 131                                       00037480
C--FINITE WIRE METHOD=0 (L>0.0)                                       00037490
      XLEN=L                                                          00037500
      XFIN=XX                                                         00037510
      YFIN=YY                                                         00037520
      RO=SQRT(XX*XX+YY2)                                              00037530
```

```
      R1=SQRT((XX+L)**2+YY2)                                                00037540
      R2=SQRT((XX-L)**2+YY2)                                                00037550
      BB=R0/DEL                                                             00037560
      IF(ICOMP.EQ.1) ZFLD=FINHX(BB)/HXP                                     00037570
      IF(ICOMP.NE.1) ZFLD=FINHY(BB)/HYP                                     00037580
      NEV=NEVFIN/2                                                          00037590
      CERR=ERRFIN                                                           00037600
      GO TO 150                                                             00037610
  131 IF(ICOMP.EQ.1) GO TO 140                                              00037620
      IF(XX.EQ.0.0)ZFLD=2.*CANC4(0.0,L,EP,NEV,IERR,HY03,MEV,CERR)/HYP       00037630
      IF(XX.NE.0.0)ZFLD=CANC4(-L,L,EP,NEV,IERR,HY03,MEV,CERR)/HYP           00037640
      GO TO 150                                                             00037650
  140 ZFLD=-CANC4(-L,L,EP,NEV,IERR,HX03,MEV,CERR)/HXP                       00037660
  150 IF(NEV.LT.MEV.OR.IER.EQ.0.OR.                                         00037670
     1 (REAL(CERR).LE.EP.AND.AIMAG(CERR).LE.EP)) GO TO 120                  00037680
      WRITE(16,160) NEV,CERR,ZFLD,ICOMP,FREQ,BB,I,EP                        00037690
  160 FORMAT(/45H WARNING--EP ACCURACY NOT ACHIEVED IN FCODE--/              00037700
     1 5H NEV=,I5,6H CERR=,2E12.5,6H ZFLD=,2E12.5,7H ICOMP=,I2/             00037710
     2 6H FREQ=,E12.5,4H BB=,E12.5,3H I=,I5,4H EP=,E12.5)                   00037720
      WRITE(6,160) NEV,CERR,ZFLD,ICOMP,FREQ,BB,I,EP                         00037730
      IERR=IERR+1                                                           00037740
      IF(IERR.LE.5) GO TO 120                                               00037750
      IERR=2                                                                00037760
      IER=0                                                                 00037770
      GO TO 120                                                             00037780
  170 F=CABS(ZFLD)                                                          00037790
      AMP=F                                                                 00037800
      GO TO 210                                                             00037810
  180 CALL POLAR2(ZFLD,AMP,PHZ)                                             00037820
      F=PHZ                                                                 00037830
      GO TO 210                                                             00037840
  190 F=REAL(ZFLD)                                                          00037850
      GO TO 210                                                             00037860
  200 F=AIMAG(ZFLD)                                                         00037870
  210 RETURN                                                                00037880
  220 IOBS=PRNT(2)                                                          00037890
  230 FREQL=FREQ                                                            00037900
      LCOMP=ICOMP                                                           00037910
      ICOMP=1                                                               00037920
      IF(IOBS.LT.0) ICOMP=-1                                                00037930
      IOBS=IABS(IOBS)                                                       00037940
      IF(ICOMP.EQ.LCOMP) GO TO (170,180,190,200),IOBS                       00037950
      GO TO 110                                                             00037960
  240 IOBS=PRNT(4)                                                          00037970
      GO TO 230                                                             00037980
      END                                                                   00037990

      SUBROUTINE PCODE(P,X,B,PRNT,F,I,IP,IB)                                00038000
C--ANALYTIC PARTIALS OF HX/HXP OR HY/HYP W/R PARAMETERS IN B AND IN COMM00038010
C                                                                           00038020
C  (PCODE ONLY CALLED BY MARQRT IF IDER=0--DEFAULT)                         00038030
C                                                                           00038040
```

```
C--PARAMETERS--                                                        00038050
C                                                                      00038060
C       P=      OUTPUT PARTIAL DERIVATIVE ARRAY (DIM. K)               00038070
C               EVALUATED FOR GIVEN X(I,),B(K) AT OBS. I               00038080
C       X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)         00038090
C       B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)             00038100
C       PRNT=   WORK AND PRINT ARRAY (DIM. 5)                          00038110
C       F=      LAST FUNCTION VALUE FROM FCODE AT GIVEN I.             00038120
C               F MAY OR MAY NOT BE NEEDED--BUT AVAILABLE ANYWAY.      00038130
C       I=      OBSERVATION NO. TO EVAL. P ARRAY (1<=I<=N)             00038140
C       IP=     NO. PARAMETERS HELD FIXED (IF ANY--IF NONE IP=0).      00038150
C       IB=     ARRAY OF PARAMETER INDICES HELD FIXED IF IP.GT.0       00038160
C               (DIM. 19).                                             00038170
C                                                                      00038180
      INTEGER IB(1)                                                    00038190
      REAL P(1),X(200,5),B(1),PRNT(5),F,EPS                            00038200
      REAL   K(10),D(9),L                                              00038210
      COMPLEX Z(19),CERR,CANC4,ZFLD,ZI1,ZI2,ZI1P(19),ZI2P(19),         00038220
     1 ZBES(5),PHXPJ,PHYPJ,ZFLD2                                       00038230
      COMMON/MODEL/K,D,MM                                              00038240
      COMMON/SHARE/EPS,C2,C3,C4,XX,YY,YY2,RHO,RHO2,DELRHO,BB,          00038250
     1 L,DEL,DEL2,METHOD,IREST(2)                                      00038260
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                          00038270
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                              00038280
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB               00038290
      COMMON/PART/JJ,ISEP                                              00038300
      EXTERNAL PHXPJ,PHYPJ                                             00038310
      EQUIVALENCE (DUM,NEV)                                            00038320
      DATA FREQLL/0.0/                                                 00038330
C--GET PARTIALS W/R SIGMA(JJ), JJ IN (1,MM), OR                        00038340
C  W/R DIST(JJ-MM), JJ IN (MM+1,2*MM-1).                               00038350
C  (UNLESS PARM HELD FIXED VIA IP,IB())                                00038360
      DO 190 J=1,M2P1                                                  00038370
      P(J)=0.0E0                                                       00038380
      JJ=J                                                             00038390
      IF(IP.LE.0) GO TO 20                                             00038400
      DO 10 II=1,IP                                                    00038410
      IF(IB(II).EQ.J) GO TO 190                                        00038420
   10 CONTINUE                                                         00038430
   20 IF(FREQ.NE.FREQLL.OR.I.EQ.1) GO TO 30                            00038440
      JUMP=1                                                           00038450
      IF(IOB.EQ.5) GO TO 150                                           00038460
      IF(IOB.EQ.6.AND.ISEP.EQ.0) GO TO 170                             00038470
   30 JUMP=0                                                           00038480
   40 IF(J.GE.M2) GO TO 81                                             00038490
      IF(L.EQ.0.0) GO TO 90                                            00038500
      IF(ICOMP.EQ.1) GO TO 50                                          00038510
      IF(XX.EQ.0.0) Z(J)=2.*CANC4(0.,L,EP,NEV,IERR,PHYPJ,MEV,CERR)     00038520
     1 /HYP                                                            00038530
      IF(XX.NE.0.0) Z(J)=CANC4(-L,L,EP,NEV,IERR,PHYPJ,MEV,CERR)        00038540
     1  /HYP                                                           00038550
      GO TO 60                                                         00038560
```

```
   50 Z(J)=-CANC4(-L,L,EP,NEV,IERR,PHXPJ,MEV,CERR)/HXP                    00038570
   60 IF(NEV.LT.MEV.OR.IER.EQ.0.OR.                                       00038580
     1 (REAL(CERR).LE.EP.AND.AIMAG(CERR).LE.EP)) GO TO 100                00038590
   70 WRITE(16,80) NEV,CERR,J,Z(J),ICOMP,FREQ,BB,I,EP                     00038600
   80 FORMAT(/45H WARNING--EP ACCURACY NOT ACHIEVED IN PCODE--/            00038610
     1 5H NEV=,I5,6H CERR=,2E12.5,3H Z(,I2,2H)=,2E12.5,                   00038620
     2 7H ICOMP=,I2/                                                      00038630
     3 6H FREQ=,E12.5,4H BB=,E12.5,3H I=,I5,4H EP=,E12.5)                 00038640
      WRITE(6,80) NEV,CERR,J,Z(J),ICOMP,FREQ,BB,I,EP                      00038650
      IERR=IERR+1                                                         00038660
      IF(IERR.LE.5) GO TO 100                                             00038670
      IERR=2                                                              00038680
      IER=0                                                               00038690
      GO TO 100                                                           00038700
   81 IF(J.EQ.M2.AND.ICOMP.NE.1) GO TO 190                                00038710
      IF(J.EQ.M2P1.AND.ICOMP.EQ.1) GO TO 190                              00038720
      Z(J)=ZFLD/B(J)                                                      00038730
      ZFLD2=ZFLD                                                          00038740
      ZFLD=ZFLD/B(J)                                                      00038750
      GO TO 105                                                           00038760
   90 IF(ICOMP.EQ.1) Z(J)=PHXPJ(DUM)/HXP                                  00038770
      IF(ICOMP.NE.1) Z(J)=PHYPJ(DUM)/HYP                                  00038780
  100 IF(ICOMP.EQ.1) Z(J)=B(M2)*Z(J)                                      00038790
      IF(ICOMP.NE.1) Z(J)=B(M2P1)*Z(J)                                    00038800
  105 IF(JUMP.EQ.1) GO TO (110,120,130,140),IOBS                          00038810
      GO TO (110,120,130,140,150,170),IIOB                                00038820
  110 PP=(REAL(ZFLD)*REAL(Z(J))+AIMAG(ZFLD)*AIMAG(Z(J)))/AMP              00038830
      GO TO 180                                                           00038840
  120 PP=57.29577951*(REAL(ZFLD)*AIMAG(Z(J))-AIMAG(ZFLD)*REAL(Z(J)))/     00038850
     1 (AMP*AMP)                                                          00038860
      GO TO 180                                                           00038870
  130 PP=REAL(Z(J))                                                       00038880
      GO TO 180                                                           00038890
  140 PP=AIMAG(Z(J))                                                      00038900
      GO TO 180                                                           00038910
  150 IOBS=IABS(IFIX(PRNT(2)))                                            00038920
  160 IF(JUMP.EQ.0) GO TO (110,120,130,140),IOBS                          00038930
      GO TO 40                                                            00038940
  170 IOBS=IABS(IFIX(PRNT(4)))                                            00038950
      GO TO 160                                                           00038960
  180 P(J)=PP                                                             00038970
      IF(J.GE.M2) ZFLD=ZFLD2                                              00038980
  190 CONTINUE                                                            00038990
      IF(IOB.GE.5) FREQLL=FREQ                                            00039000
      RETURN                                                              00039010
      END                                                                 00039020

      SUBROUTINE SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)                      00039030
C--INITIALIZATION ROUTINE                                                 00039040
C                                                                         00039050
C  SUBZ IS CALLED BY MARQRT AFTER THE DATA Y(I),X(I,5) ARE READ--         00039060
C  SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL $INIT                    00039070
```

```
C  PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE...                00039080
C                                                                           00039090
C--PARAMETERS--                                                             00039100
C                                                                           00039110
C       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.                             00039120
C       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY                 00039130
C               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES                  00039140
C               PRINTED BY PGM MARQRT...                                    00039150
C       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)                       00039160
C       TITLE=  ALPHA TITLE ARRAY READ IN BY PGM MARQRT.                    00039170
C       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED                         00039180
C               0 IF ONLY UNIT 6 PRINT FILE USED.                           00039190
C                                                                           00039200
C--FOLLOWING CHARACTER STMT ONLY FOR HONEYWELL MULTICS SYS:                  00039210
      CHARACTER*5 TITLE(16)                                                 00039220
      COMPLEX ZFLD,ZI1,ZI2,ZI1P(19),ZI2P(19),ZBES(5),ERRFIN                 00039230
      REAL    Y(1),X(200,5),B(1),PRNT(1),EPS                                00039240
      REAL    K(10),D(9),L                                                  00039250
      COMMON/MODEL/K,D,MM                                                   00039260
      COMMON/SHARE/EPS,C2,C3,C4,X0,Y0,YY2,RHO,RHO2,DELRHO,BB,               00039270
     1 L,DEL,DEL2,METHOD,IREST(2)                                           00039280
      COMMON/CTL/ZBES,ZI1,ZI2,ZI1P,ZI2P,ZFLD,                               00039290
     1 AMP,SIG1,HXP,HYP,FREQ,LCOMP,ICOMP,                                   00039300
     2 EP,NEPS,IEPS,IOB,M1,M21,M2,M2P1,IER,IERR,MEV,IIOB                    00039310
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEVFIN,MEVFIN,ERRFIN,LW       00039320
      NAMELIST/INIT/IOB,MM,X0,Y0,METHOD,EPS,EP,NEPS,L,IER,MEV               00039330
        DATA ISUBZ/0/                                                       00039340
        IF(ISUBZ.NE.0) GO TO 10                                             00039350
C--PRESET                                                                   00039360
        ISUBZ=1                                                             00039370
      MM=1                                                                  00039380
      MEV=300                                                               00039390
      IER=2                                                                 00039400
      L=0.0                                                                 00039410
      IOB=1                                                                 00039420
      X0=0.0                                                                00039430
      Y0=0.0                                                                00039440
      METHOD=0                                                              00039450
      NFIN=1                                                                00039460
      EPS=.1E-3                                                             00039470
   10 EP=.1E-2                                                              00039480
      NEPS=10                                                               00039490
      READ(5,INIT)                                                          00039500
      WRITE(6,20) TITLE                                                     00039510
   20 FORMAT(17H1M A R Q H X Y --,5X,16A5/)                                  00039520
      IF(IOUT.EQ.1) WRITE(16,20) TITLE                                      00039530
      WRITE(6,30) IOB,MM,X0,Y0,L,METHOD,IER,MEV,NFIN,EPS,EP,NEPS            00039540
      IF(IOUT.EQ.1)                                                         00039550
     1 WRITE(16,30) IOB,MM,X0,Y0,L,METHOD,IER,MEV,NFIN,EPS,EP,NEPS          00039560
   30   FORMAT(7H IOB = ,I2,8X,5HMM = ,I2,9X,3HX0=,E12.5,4H Y0=,E12.5,       00039570
     1 3H L=,E12.5/10H METHOD = ,I1,6X,                                     00039580
     2 6HIER = ,I1,9X,6HMEV = ,I5,5X,5HNFIN=,I3/5H EPS=,E11.5,              00039590
```

```
     3 4H EP=,E11.5,2X,7HNEPS = ,I4)                                    00039600
C--TEST $INIT PARMS                                                     00039610
      IF(MM.LT.1.OR.MM.GT.9.OR.(X0.EQ.0.0.AND.IOB.GT.0.AND.IOB.LT.5)    00039620
     1 .OR.Y0.EQ.0.0.OR.                                                00039630
     2 EP.LT.EPS.OR.L.LT.0.0.OR.                                        00039640
     3 IER.LT.0.OR.IER.GT.5.OR.                                         00039650
     4 METHOD.LT.0.OR.METHOD.GT.2.OR.                                   00039660
     5 IOB.EQ.0.OR.IOB.LT.-4.OR.IOB.GT.6.OR.NEPS.LT.1)                  00039670
     6 CALL ERRMSG(30HSOME $INIT PARMS OUT OF RANGE ,6,6,16)            00039680
      IIOB=IABS(IOB)                                                    00039690
C--TEST X(I, ) DATA FOR GIVEN IOB BEFORE PROCEEDING--                   00039700
      DO 60 I=1,N                                                       00039710
      IF(X(I,1).LE.0.0) CALL ERRMSG(                                    00039720
     1 21HSOME FREQ=X(I,1).LE.0,5,6,16)                                 00039730
      IF(IIOB-5) 60,40,50                                               00039740
   40 J=IFIX(X(I,2))                                                    00039750
      IF(J.LT.-4.OR.J.GT.4.OR.J.EQ.0)                                   00039760
     & CALL ERRMSG(                                                     00039770
     140HSOME IOBS=X(I,2) OUT OF RANGE WHEN IOB=5,8,6,16)               00039780
      GO TO 60                                                          00039790
   50 J=IFIX(X(I,4))                                                    00039800
      IF(J.LT.-4.OR.J.GT.4.OR.J.EQ.0) CALL ERRMSG(                      00039810
     1 41HSOME IOBS=X(I,4) OUT OF RANGE WHEN IOB=6 ,9,6,16)             00039820
      IF(J.GT.0.AND.X(I,2).EQ.0.0.OR.X(I,3).EQ.0.0)                     00039830
     1 CALL ERRMSG(                                                     00039840
     2 57HSOME X0=X(I,2) OR Y0=X(I,3)=0 WHEN X(I,4).GT.0.AND.IOB=6 ,    00039850
     3 12,6,16)                                                         00039860
   60 CONTINUE                                                          00039870
C--PRESET SOME GLOBAL CONSTANTS                                         00039880
      IERR=IER                                                          00039890
      IF(IER.EQ.0) IERR=2                                               00039900
      HAKTOL=1.0E-6*EPS                                                 00039910
      FINTOL=1.0E-3*EP                                                  00039920
      INTYPE=IERR                                                       00039930
      MEVFIN=2*MEV                                                      00039940
      IF(IOB.EQ.6) GO TO 150                                            00039950
      CALL PRMHXY                                                       00039960
   70 WRITE(6,80) RHO,HXP,HYP                                           00039970
      IF(IOUT.EQ.1) WRITE(16,80) RHO,HXP,HYP                            00039980
   80 FORMAT(///40H RECEIVER-TRANSMITTER SEPARATION (RHO) =,E12.5/      00039990
     1 /29H PRIMARY FIELDS (HXP X 4PI) =,E12.5,15H  (HYP X 4PI) =,      00040000
     2 E12.5///18H PARAMETER ORDER--/)                                  00040010
   90 M1=MM-1                                                           00040020
      M2=2*MM                                                           00040030
      M21=M2-1                                                          00040040
      M2P1=M2+1                                                         00040050
      WRITE(6,100) (I,I,I=1,MM)                                         00040060
      IF(IOUT.EQ.1) WRITE(16,100) (I,I,I=1,MM)                          00040070
  100 FORMAT(5X,I3,6X,6HSIGMA(,I3,1H))                                  00040080
      IF(MM.EQ.1) GO TO 132                                             00040090
      DO 110 I=1,M1                                                     00040100
      J=MM+I                                                            00040110
```

```
      IF(IOUT.EQ.1) WRITE(16,120) J,I                                    00040120
  110 WRITE(6,120) J,I                                                   00040130
  120 FORMAT(5X,I3,6X,6HTHICK(,I3,1H))                                   00040140
  132 WRITE(6,131) M2,M2                                                 00040150
  131 FORMAT(5X,I3,10X,2HB(,I3,25H)  HX/HXP SHIFT PARAMETER)             00040160
      IF(IOUT.EQ.1) WRITE(16,131) M2,M2                                  00040170
      WRITE(6,133) M2P1,M2P1                                             00040180
  133 FORMAT(5X,I3,10X,2HB(,I3,25H)  HY/HYP SHIFT PARAMETER)             00040190
      IF(IOUT.EQ.1) WRITE(16,133) M2P1,M2P1                              00040200
C--X(I,1)=FREQ, X(I,2)=IOB TYPE (IF IOB=5), X(I,M+1)=STD.DEV. (IF IWT=1)00040210
C  NOTE-- M=1 REQUIRED IN PGM MARQRT WHEN -4<=IOB<=4, AND               00040220
C  M=2 IS NECESSARY WHEN IOB=5...                                       00040230
C  ALSO, M=4 IS NECESSARY WHEN IOB=6...                                 00040240
  130 NPRNT=2                                                           00040250
      IF(IOB.EQ.5) NPRNT=3                                              00040260
      IF(IOB.EQ.6) NPRNT=5                                              00040270
      IEPS=0                                                            00040280
  140 RETURN                                                            00040290
  150 WRITE(6,160)                                                      00040300
      IF(IOUT.EQ.1) WRITE(16,160)                                       00040310
  160 FORMAT(///18H PARAMETER ORDER--/)                                 00040320
      GO TO 90                                                          00040330
      END                                                               00040340

      SUBROUTINE SUBEND(Y,X,B,K,N,TITLE,IOUT)                           00040350
C-- 'MARQHXY' TERMINATION ROUTINE (CALLED ONCE BY MARQRT)               00040360
C  (PARAMETERS SAME AS IN SUBROUTINE FCODE,PCODE, OR SUBZ)              00040370
C  B= FINAL SOLUTION VECTOR OBTAINED BY PGM MARQRT.                     00040380
C                                                                       00040390
C--FOLLOWING CHARACTER STMT. ONLY FOR HONEYWELL MULTICS SYS:            00040400
      CHARACTER*5 TITLE(16)                                             00040410
      REAL Y(1),X(200,5),B(1)                                           00040420
      WRITE(6,10) TITLE                                                 00040430
   10 FORMAT(17H1M A R Q H X Y --,5X,16A5//                             00040440
     1 28H FINAL UNSCALED PARAMETERS--,10X,11HRESISTIVITY,11X,5HDEPTH/) 00040450
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                                  00040460
      MM=(K-1)/2                                                        00040470
      DO 30 I=1,MM                                                      00040480
      R=1.0/B(I)                                                        00040490
      WRITE(6,20) I,B(I),I,R                                            00040500
   20 FORMAT(5X,I3,4X,E16.8,2X,I3,1X,E16.8)                             00040510
      IF(IOUT.EQ.1) WRITE(16,20) I,B(I),I,R                             00040520
   30 CONTINUE                                                          00040530
      IF(K.LE.3) GO TO 52                                               00040540
      M2=MM+1                                                           00040550
      K1=K-2                                                            00040560
      D=0.0                                                             00040570
      DO 50 I=M2,K1                                                     00040580
      D=D+B(I)                                                          00040590
      L=I-MM                                                            00040600
      WRITE(6,40) I,B(I),L,D                                            00040610
   40 FORMAT(5X,I3,4X,E16.8,24X,I3,1X,E16.8)                            00040620
```

```
      IF(IOUT.EQ.1) WRITE(16,40) I,B(I),L,D                          00040630
50    CONTINUE                                                       00040640
52    K1=K-1                                                         00040650
      DO 53 I=K1,K                                                   00040660
      WRITE(6,51) I,B(I)                                             00040670
51    FORMAT(5X,I3,4X,E16.8)                                         00040680
      IF(IOUT.EQ.1) WRITE(16,51) I,B(I)                              00040690
53    CONTINUE                                                       00040700
60    RETURN                                                         00040710
      END                                                            00040720
```

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)
PRINT...	(replace by WRITE(6,)... if necessary)

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

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

Appendix 3.-- Test problem input/output listing

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

file05

```
test2x_hxy_reim
$parms n=36,k=5,m=2,sp=1,sy=2,iprt=-1,
 b=.015,1.5,250,2.5,2.5$
(2e16.8,f10.0)
$init mm=2,x0=193.65,y0=50,iob=5,eps=.1e-5$
```

file10

```
 0.19988102e+01  0.10000000e+01        3.
 0.19487382e+01  0.10000000e+01       -3.
-0.35902090e-02  0.10000000e+01        4.
-0.46805422e-01  0.10000000e+01       -4.
 0.19958991e+01  0.31622777e+01        3.
 0.19064268e+01  0.31622777e+01       -3.
-0.79093666e-02  0.31622777e+01        4.
-0.61334500e-01  0.31622777e+01       -4.
 0.19898521e+01  0.10000000e+02        3.
 0.18580637e+01  0.10000000e+02       -3.
-0.15008689e-01  0.10000000e+02        4.
-0.69438564e-01  0.10000000e+02       -4.
 0.19816850e+01  0.31622777e+02        3.
 0.18152467e+01  0.31622777e+02       -3.
-0.29186640e-01  0.31622777e+02        4.
-0.85370520e-01  0.31622777e+02       -4.
 0.19713796e+01  0.99999999e+02        3.
 0.17722171e+01  0.99999999e+02       -3.
-0.70290070e-01  0.99999999e+02        4.
-0.15272446e+00  0.99999999e+02       -4.
 0.19369595e+01  0.31622776e+03        3.
 0.16537876e+01  0.31622776e+03       -3.
-0.19523958e+00  0.31622776e+03        4.
-0.35100506e+00  0.31622776e+03       -4.
 0.17419008e+01  0.99999998e+03        3.
 0.12331874e+01  0.99999998e+03       -3.
-0.48177406e+00  0.99999998e+03        4.
-0.60865190e+00  0.99999998e+03       -4.
 0.11613263e+01  0.31622776e+04        3.
 0.65861968e+00  0.31622776e+04       -3.
-0.71683308e+00  0.31622776e+04        4.
-0.54685562e+00  0.31622776e+04       -4.
 0.55951296e+00  0.99999998e+04        3.
 0.33628892e+00  0.99999998e+04       -3.
-0.52484126e+00  0.99999998e+04        4.
-0.32504556e+00  0.99999998e+04       -4.
```

```
m a r q h x y --      test2x_hxy_reim

iob =  5         mm =  2          x0= 0.19365e+03 y0= 0.50000e+02 l= 0.00000e+00
method = 0       ier = 2          mev =   300     nfin=  1
eps=0.10000e-05 ep=0.10000e-02  neps =   10


receiver-transmitter separation (rho) = 0.20000e+03

primary fields (hxp x 4pi) = 0.12103e-04  (hyp x 4pi) =-0.21875e-04


parameter order--

      1       sigma(  1)
      2       sigma(  2)
      3       thick(  1)
      4           b(  4)  hx/hxp shift parameter
      5           b(  5)  hy/hyp shift parameter
```

```
m a r q r t  --     test2x_hxy_reim

n =   36        k =   5         ip =   0        m =  2          gamcr=0.450e+02
del= 0.100e-04  modlam = 1      ff= 0.400e+01   t= 0.200e+01    e= 0.500e-04
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 = 3       scalep = 1      scaley = 2

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

parameters     0.15000000e-01   0.15000000e+01   0.25000001e+03   0.25000000e+01
               0.25000000e+01

iter          phi                s e              length       gamma          lambda
   1    0.10153034e+01    0.18097437e+00    0.000e+00    0.000e+00    0.100e-01

parameters     0.19084261e-01   0.28287086e+01   0.19507885e+03   0.20046226e+01
               0.20037439e+01

iter          phi                s e              length       gamma          lambda
   2    0.27852019e-02    0.94786719e-02    0.787e+00    0.400e+02    0.100e-02

parameters     0.20003621e-01   0.21897372e+01   0.19810882e+03   0.20003360e+01
               0.19999351e+01

iter          phi                s e              length       gamma          lambda
   3    0.39935085e-04    0.11350016e-02    0.261e+00    0.592e+02    0.100e-03

parameters     0.19999182e-01   0.20001884e+01   0.20001694e+03   0.19999786e+01
               0.19999494e+01

iter          phi                s e              length       gamma          lambda
   4    0.31665105e-08    0.10106706e-04    0.910e-01    0.477e+02    0.100e-04

parameters     0.20000017e-01   0.20000603e+01   0.19999807e+03   0.20000005e+01
               0.19999998e+01

iter          phi                s e              length       gamma          lambda
   5    0.58219410e-10    0.13704180e-05    0.125e-03    0.705e+02    0.100e-06

parameters     0.19999999e-01   0.19998865e+01   0.19999987e+03   0.19999999e+01
               0.19999989e+01

iter          phi                s e              length       gamma          lambda
   6    0.33198426e-10    0.10348512e-05    0.873e-04    0.461e+02    0.100e-08

-epsilon test
   6 iterations
```

```
m a r q r t  --     test2x_hxy_reim

parameters     0.19999999e-01   0.19998865e+01   0.19999987e+03   0.19999999e+01
               0.19999989e+01

-unscaled-
   i     obs.y(i)        cal            res          %res.err      x(i,1)        x(i,2)        x(i,3)        x(i,4)
   1  0.199881e+01  0.199881e+01 -0.447e-07 -0.223650e-05  0.100000e+01  0.300000e+01  0.000000e+00
   2  0.194874e+01  0.194874e+01 -0.641e-06 -0.328802e-04  0.100000e+01 -0.300000e+01  0.000000e+00
   3 -0.359021e-02 -0.359122e-02  0.101e-05  0.280583e-01  0.100000e+01  0.400000e+01  0.000000e+00
   4 -0.468054e-01 -0.468045e-01 -0.911e-06 -0.194703e-02  0.100000e+01 -0.400000e+01  0.000000e+00
   5  0.199590e+01  0.199590e+01  0.924e-06  0.462885e-04  0.316228e+01  0.300000e+01  0.000000e+00
   6  0.190643e+01  0.190643e+01 -0.118e-05 -0.617486e-04  0.316228e+01 -0.300000e+01  0.000000e+00
   7 -0.790937e-02 -0.791036e-02  0.994e-06  0.125682e-01  0.316228e+01  0.400000e+01  0.000000e+00
   8 -0.613345e-01 -0.613338e-01 -0.662e-06 -0.107886e-02  0.316228e+01 -0.400000e+01  0.000000e+00
   9  0.198985e+01  0.198985e+01  0.180e-05  0.906119e-04  0.100000e+02  0.300000e+01  0.000000e+00
  10  0.185806e+01  0.185806e+01 -0.109e-05 -0.585440e-04  0.100000e+02 -0.300000e+01  0.000000e+00
  11 -0.150987e-01 -0.150990e-01  0.272e-06  0.181267e-02  0.100000e+02  0.400000e+01  0.000000e+00
  12 -0.694386e-01 -0.694386e-01  0.112e-07  0.160946e-04  0.100000e+02 -0.400000e+01  0.000000e+00
  13  0.198168e+01  0.198168e+01  0.167e-05  0.842178e-04  0.316228e+02  0.300000e+01  0.000000e+00
  14  0.181525e+01  0.181525e+01 -0.492e-06 -0.270893e-04  0.316228e+02 -0.300000e+01  0.000000e+00
  15 -0.291866e-01 -0.291856e-01 -0.102e-05 -0.347822e-02  0.316228e+02  0.400000e+01  0.000000e+00
  16 -0.853705e-01 -0.853713e-01  0.791e-06  0.926181e-03  0.316228e+02 -0.400000e+01  0.000000e+00
  17  0.197136e+01  0.197138e+01 -0.179e-06 -0.907050e-05  0.100000e+03  0.300000e+01  0.000000e+00
  18  0.177222e+01  0.177222e+01  0.283e-06  0.159756e-04  0.100000e+03 -0.300000e+01  0.000000e+00
  19 -0.702901e-01 -0.702864e-01 -0.372e-05 -0.528691e-02  0.100000e+03  0.400000e+01  0.000000e+00
  20 -0.152724e+00 -0.152723e+00 -0.128e-05 -0.839100e-03  0.100000e+03 -0.400000e+01  0.000000e+00
  21  0.193696e+01  0.193696e+01 -0.428e-05 -0.220791e-03  0.316229e+03  0.300000e+01  0.000000e+00
  22  0.165379e+01  0.165379e+01 -0.313e-06 -0.189217e-04  0.316228e+03 -0.300000e+01  0.000000e+00
  23 -0.195240e+00 -0.195238e+00 -0.113e-05 -0.581008e-03  0.316228e+03  0.400000e+01  0.000000e+00
  24 -0.351005e+00 -0.351004e+00 -0.630e-06 -0.179364e-03  0.316228e+03 -0.400000e+01  0.000000e+00
  25  0.174190e+01  0.174190e+01  0.715e-06  0.410618e-04  0.100000e+04  0.300000e+01  0.000000e+00
  26  0.123319e+01  0.123319e+01  0.447e-06  0.362504e-04  0.100000e+04 -0.300000e+01  0.000000e+00
  27 -0.481774e+00 -0.481776e+00  0.235e-05  0.488688e-03  0.100000e+04  0.400000e+01  0.000000e+00
  28 -0.608652e+00 -0.608651e+00 -0.574e-06 -0.942567e-04  0.100000e+04 -0.400000e+01  0.000000e+00
  29  0.116133e+01  0.116133e+01 -0.268e-06 -0.230961e-04  0.316228e+04  0.300000e+01  0.000000e+00
  30  0.658620e+00  0.658619e+00  0.432e-06  0.656121e-04  0.316228e+04 -0.300000e+01  0.000000e+00
  31 -0.716833e+00 -0.716833e+00 -0.425e-06 -0.592444e-04  0.316228e+04  0.400000e+01  0.000000e+00
  32 -0.546856e+00 -0.546855e+00 -0.209e-06 -0.381483e-04  0.316228e+04 -0.400000e+01  0.000000e+00
  33  0.559513e+00  0.559513e+00  0.149e-07  0.266324e-05  0.100000e+05  0.300000e+01  0.000000e+00
  34  0.336289e+00  0.336289e+00  0.171e-06  0.509572e-04  0.100000e+05 -0.300000e+01  0.000000e+00
  35 -0.524841e+00 -0.524841e+00 -0.447e-07 -0.851753e-05  0.100000e+05  0.400000e+01  0.000000e+00
  36 -0.325046e+00 -0.325045e+00 -0.164e-06 -0.504277e-04  0.100000e+05 -0.400000e+01  0.000000e+00

-unscaled partials-
   1   -0.98324001e-03   -0.70699443e-03    0.87224474e-05    0.99940517e+00    0.00000000e+00
   2    0.20211623e+01   -0.15181468e-01    0.18929832e-03    0.00000000e+00    0.97436994e+00
   3   -0.36619371e-01   -0.10017511e-02    0.32315393e-04   -0.17956084e-02    0.00000000e+00
   4    0.14721401e+01   -0.63959789e-02    0.27400627e-03    0.00000000e+00   -0.23402268e-01
   5   -0.50427952e-02   -0.18695066e-02    0.39379099e-04    0.99794912e+00    0.00000000e+00
   6    0.35058553e+01   -0.20009365e-01    0.47029676e-03    0.00000000e+00    0.95321449e+00
   7   -0.11282019e+00   -0.12828907e-02    0.73988811e-04   -0.39551807e-02    0.00000000e+00
   8    0.13523127e+01   -0.18612448e-02    0.42195511e-03    0.00000000e+00   -0.30666936e-01
   9   -0.20861430e-01   -0.30347357e-02    0.12128647e-03    0.99492519e+00    0.00000000e+00
  10    0.47866060e+01   -0.18649118e-01    0.88436525e-03    0.00000000e+00    0.92903288e+00
  11   -0.34288906e+00   -0.54717879e-03    0.12056096e-03   -0.75044810e-02    0.00000000e+00
  12    0.39549966e-01    0.44415002e-02    0.45028189e-03    0.00000000e+00   -0.34719307e-01
  13   -0.83905160e-01   -0.29079246e-02    0.24762425e-03    0.99984169e+00    0.00000000e+00
  14    0.47683265e+01   -0.12720029e-01    0.12774321e-02    0.00000000e+00    0.90762408e+00
  15   -0.10392413e+01    0.76442611e-03    0.11748840e-03   -0.14592813e-01    0.00000000e+00
```

16	-0.30897405e+01	0.77454575e-02	0.24581354e-03	0.00000000e+00	-0.42685679e-01
17	-0.45444414e+00	-0.16932984e-02	0.35371194e-03	0.98568992e+00	0.00000000e+00
18	0.14717034e+01	-0.52320637e-02	0.14342632e-02	0.00000000e+00	0.88610888e+00
19	-0.31423824e+01	0.14579152e-02	0.14959256e-04	-0.35143179e-01	0.00000000e+00
20	-0.88465329e+01	0.76576153e-02	-0.28830302e-03	0.00000000e+00	-0.76361634e-01
21	-0.31857264e+01	-0.21089474e-03	0.29726317e-03	0.96848192e+00	0.00000000e+00
22	-0.10289088e+02	0.89324522e-03	0.72470024e-03	0.00000000e+00	0.82689439e+00
23	-0.97794977e+01	0.13002171e-02	-0.21941326e-03	-0.97619230e-01	0.00000000e+00
24	-0.15165902e+02	0.45758025e-02	-0.10864710e-02	0.00000000e+00	-0.17550231e+00
25	-0.16238038e+02	0.45790764e-03	-0.11434795e-03	0.87095008e+00	0.00000000e+00
26	-0.26559969e+02	0.11841594e-02	-0.51839372e-03	0.00000000e+00	0.61659381e+00
27	-0.14938942e+02	0.13582394e-03	-0.29869611e-03	-0.24088822e+00	0.00000000e+00
28	-0.56460463e+01	-0.28678413e-03	-0.31500079e-03	0.00000000e+00	-0.30432583e+00
29	-0.31250369e+02	-0.44946366e-04	0.89221082e-05	0.58066332e+00	0.00000000e+00
30	-0.20334594e+02	-0.87585275e-04	0.56106976e-04	0.00000000e+00	0.32930980e+00
31	-0.13408090e+01	-0.30843700e-04	0.47436985e-04	-0.35841635e+00	0.00000000e+00
32	0.89675951e+01	0.20093635e-05	0.53502089e-04	0.00000000e+00	-0.27342786e+00
33	-0.16925842e+02	0.54567657e-07	-0.78814387e-06	0.27975648e+00	0.00000000e+00
34	-0.87445517e+01	-0.32[illegible]2781e-06	-0.22514910e-06	0.00000000e+00	0.16814447e+00
35	0.13070493e+02	-0.64[illegible]5825e-06	0.65633971e-06	-0.26242063e+00	0.00000000e+00
36	0.81278172e+01	-0.52938952e-06	0.95483280e-06	0.00000000e+00	-0.16252279e+00

-unscaled-

phi	s e	lambda
0.57873324e-10	0.13663387e-05	0.100e-08

gtp inverse

1	0.50076640e-03	0.53534477e+00	-0.66234632e+01	0.43275620e-02	0.12743178e-01
2	0.53534479e+00	0.22718673e+04	-0.43145912e+04	0.66088848e+01	0.31293334e+02
3	-0.66234627e+01	-0.43145900e+04	0.32485865e+06	-0.86046941e+02	-0.31866170e+03
4	0.43275620e-02	0.66088843e+01	-0.86046946e+02	0.18080113e+00	0.15535608e+00
5	0.12743178e-01	0.31293332e+02	-0.31866172e+03	0.15535608e+00	0.81890342e+00

parameter correlation matrix

1	1.0000	0.5019	-0.5193	0.4548	0.6293
2	0.5019	1.0000	-0.1588	0.3261	0.7255
3	-0.5193	-0.1588	1.0000	-0.3550	-0.6178
4	0.4548	0.3261	-0.3550	1.0000	0.4037
5	0.6293	0.7255	-0.6178	0.4037	1.0000

	parameter std error	one - parameter lower	one - parameter upper	support plane lower	support plane upper	std.error/parm
1	0.30575668e-07	0.19999938e-01	0.20000060e-01	0.19999862e-01	0.20000136e-01	0.15287835e-05
2	0.65125316e-04	0.19997563e+01	0.20000167e+01	0.19995953e+01	0.20001778e+01	0.32564505e-04
3	0.77876350e-03	0.19999831e+03	0.20000143e+03	0.19999639e+03	0.20000335e+03	0.38938200e-05
4	0.58097700e-06	0.19999987e+01	0.20000010e+01	0.19999973e+01	0.20000025e+01	0.29048852e-06
5	0.12364447e-05	0.19999964e+01	0.20000014e+01	0.19999934e+01	0.20000044e+01	0.61822270e-06