U.S. Department of the Interior
Geological Survey

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

Program MQLVEMCPL:
Marquardt inversion of electromagnetic coupling data for a layered halfspace model with complex conductivities

by

James Kauahikaua and Walter L. Anderson

OPEN-FILE REPORT 80-1158

1980

CONTENTS.

DISCLAIMER 3

INTRODUCTION 4

PARAMETERS AND DATA REQUIRED 5

PROGRAM FILES 5

DETAIL PARAMETER AND DATA DEFINITIONS 6

$parms parameters 6

$init parameters 13

A NOTE ON COMPLEX CONDUCTIVITIES 14

DATA MATRIX NOTES 15

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING 16

SPECIAL OBJECT FORMAT PHRASES 17

MULTICS OPERATING INSTRUCTIONS 17

ERROR MESSAGES 18

DECIPHERING THE PRINTED OUTPUT 19

REFERENCES 22

Appendix 1.-- Source listing 23

Source availability 23

Appendix 2.-- Conversion to other systems 81

Appendix 3.-- Test problem input/output listing 83

DISCLAIMER.

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

---

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

By James Kauahikaua and Walter L. Anderson

INTRODUCTION.

Program MQLVEMCPL (Marquardt-Levenburg Electromagnetic Coupling) is a computer program which can be used to invert frequency-domain, electromagnetic (EM) coupling data for parameters of a layered earth model having either real or complex conductivities. EM coupling data are measurements of mutual impedance between two straight, grounded wires at the earth's surface. Details of the computations are described in Kauahikaua and Anderson (1979) and Anderson (1979a). To simplify the programming, only the collinear dipole-dipole array is considered and the complex conductivities are assumed to be described by a "Cole-Cole" relaxation model (Pelton et al, 1978) whose parameters have been modified to improve the stability of the inversion. An IMSL (International Mathematical and Statistical Library, 1977) derivative-free, Levenburg-Marquardt (Brown and Dennis, 1972), nonlinear least-squares subprogram (ZXSSQ) is used for the inversion. See appendix 2 notes for conversion to other systems where the IMSL library is not available.

The following program options are currently available:

(1) Simultaneous (or joint) inversion of amplitude, phase, real part and/or quadrature part of the EM coupling for measurements made at various frequencies.
(2) Simultaneous inversion of EM coupling data for different dipole spacings.
(3) Determination of either real conductivities or Cole-Cole complex conductivities.
(4) Scaling parameters and observation spaces to constrain the solution space and to reduce round-off effects.
(5) Weighted observations.
(6) Holding certain parameters fixed or constrained.
(7) Object-time format control of reading the observed data matrix.
(8) Using an infinitesimal dipole-dipole coupling model for preliminary inversion (good approximation for dipole-spacings greater than three dipole lengths) before using the more expensive finite-length dipole-dipole model for the final solution.

PARAMETERS AND DATA REQUIRED.

Parameters required by program MQLVEMCPL are input using Fortran namelist read statements with specific names: $parms and $init. Namelist $parms contains parameters which govern the nonlinear least-squares minimization part of the program and namelist $init contains parameters which define the form of the field data and the model and which define the methods by which the models are calculated. Reasonable default values have been incorporated into the program for most parameters and these are used whenever the corresponding parameter is omitted in a namelist input. The input data matrix is read from an optional alternate file (unless overridden) using a Fortran object-time format. Preceding the $parms statement is a required 80 (or less) character title.

The general order of input to program MQLVEMCPL 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 format of the input data matrix. The object format begins with a left parenthesis, "(", placed in col. 1.
4. The data matrix may be inserted here if the alternate-data-file option is not used (see $parms parameter ialt below).
5. $init --non-default parameters--$
6. Subsequent runs using the same data matrix but with different $parms and $init parameters may be executed by repeating steps 1,2,3, and 5 (set $parms istop=0 and make sure that ialt does not equal 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 MQLVEMCPL.

file16 output master print-type disk file--contains complete printable output.

DETAIL PARAMETER AND DATA DEFINITIONS.

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

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

n= Number of observed data points y(i),i=1,...,n, where n<=200. The type of observation is explicitly defined by $init iob.

k= Total number of parameters (1<=k<=min0(20,n)). The value of k must be equal to 2*mm-1+icmplx*3, where $init parameter mm>0 is the number of layers in the model, and icmplx<=mm is the number of layers which will have complex conductivities.
(cref: $init parameters mm,icmplx 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<=3) given in the data matrix (y(i),x(i,J),J=1,m),i=1,n. The value of m must be given as follows:
= 1 when $init parameter iob<=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));
= 3 when $init parameter iob=6 (defines mixed observation types in y(i) via x(i,2) and mixed dipole-dipole separations, nn, via x(i,3)).
(cref: $parms iwt, $init iob, and DATA MATRIX NOTES below for all definitions of x(i,m) used).

ialt= Input data matrix alternate logical unit number (default 10) for reading the data under the object-time format specified in file05. The value of ialt can be any value the operating system supports, but cannot be equal to 6,13, or 16. If ialt=5 is used, then the data matrix

((y(i),x(i,j),j=1,m),i=1,n) will immediately follow the object format on file05.
(cref: $parms n,m, $init iob).

istop= 0 to continue processing after completion of the current problem (i.e., a total restart) with the same data matrix as last used, but by using a revised title, $parms, object-time format, and $init parameters. Note that istop=0 can only be used whenever ialt is not 5 (since file ialt is rewound and read again). Also, all $parms and $init parameters previously used will be assumed, with the exception of array b(j) which must always be given.
= 1 (default) to stop the run after completion of the current problem.
(cref: $parms b,ialt).

iwt= 0 (default) for unweighted observations; i.e., all n observations y(i),i=1,...,n will be weighted unity (with assumed standard deviations equal to 1.0).
= 1 for weighted observations given by the formula wt(i)=1.0/x(i,m+1)**2, where x(i,m+1) is the standard deviation augmented to the data matrix for the given m<=3. Note: wt(i)=1.0 is stored automatically if iwt=0 or when iwt=1 and x(i,m+1)=0.0 (to avoid division by 0).
(cref: $parms n,m, $init iob, and DATA MATRIX NOTES).

* ider= 0 (default) to use analytic derivatives, which calls both forward problem (fcode) and analytic derivative (pcode) subroutines.
= 1 to use estimated derivatives, which calls only subroutine fcode. [If converting to subprogram MARQRT (as in appendix 2), then ider=1 must always be used, since pcode is a dummy routine].
(cref: $parms del).

iprt= 0 (default) for standard abbreviated printout format for each iteration. Note scaled values of parameters b(j) and phi (sum of squares) will be given via parameter scalep.
= 1 for detail printout format for each iteration, which includes the parameter changes from the Marquardt algorithm. [note iprt=1 behaves like iprt=-2, unless converting to subprogram MARQRT].
= -1 (recommended if scalep>0 used) for abbreviated printout format for each iteration with printed unscaled values of b(j) but scaled values of phi.
= -2 same as iprt=-1 but also prints on file06 n-observational lines containing: observed value

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

* niter= Maximum number of iterations allowed before accepting the results as "forced off" (default niter=10). Four different types of convergence tests are possible one of which is termed "forced off", which will occur whenever niter has been reached and one of the other convergence criteria has not been achieved. Using a small niter may be useful to monitor the progress for a large problem, and as an aid for achieving a convenient restarting procedure with the last b-vector as a new initial estimate.
(cref: $parms b and Marquardt (1963) for convergence tests used).

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

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

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

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

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

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

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

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

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

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

* iout= Printout file06 and file16 control.
= 1 (default) for printable 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). In this version of the program, output to file06 has been reduced to take less time to printout on a time-sharing terminal; 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., to use unscaled parameters).
= 1 (recommended for program MQLVEMCPL) 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 MQLVEMCPL, 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., to use unscaled observations y(i)).
= 1 to scale observations y(i) using ln(y(i)), provided y(i)>0 for all i=1,2,...,n.
= 2 to scale observations y(i) using arcsinh(y(i)). This option allows for log-type observation scaling whenever y(i) is positive, negative, or zero for any i in (1,n). A special case automatically occurs whenever sy=2, iob=5, and both amplitude and phase data are included; in this case, the MARQRT subprogram will use ln(amplitude) or arcsinh(phase) accordingly.
(cref: $init iob, $parms n and DATA MATRIX NOTES)

b()= Array of initial guesses for all k-parameters. These values must be supplied greater than zero for program MQLVEMCPL (i.e., positive conductivity parameters 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.

For $init icmplx=0, the parameter order must be given as:

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

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

For $init icmplx>0, the parameter order must be

given as:

b(1),b(2),...,b(mc) are the mc=mm+icmplx*3 layer conductivity parameters, including the Cole-Cole parameters (1) , and

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

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

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

iopt= 1 (default) implies strict descent of the sum of squares is desired in the derivative-free Marquardt algorithm (ZXSSQ), with default values used in the input array parm().
= 0 implies strict descent is not necessary (i.e., the "best" or optimum Marquardt parameter used may not yield a strict decreasing sum of squares at each iteration).
= 2 implies strict descent is desired with user parameter choices as given (or assumed) in input array parm().
(cref: $parms parm()).

parm()= array of length 4 required only when iopt=2. The default is parm()=.01,2.,120.,.1, where each element is defined by the corresponding index as follows:
i=1, the initial value of the Marquardt parameter used to scale the diagonal of the approximate Hessian matrix, xJtJ, by the factor (1.0+parm(1)). A small value gives a Newton step, while a large

---

(1) Each Cole-Cole conductivity is described by 4 parameters. They are (in the order anticipated by program MQLVEMCPL through the $parms b() array) the zero-frequency conductivity, chargeability, frequency dependence, and time constant (Pelton et al, 1978; FUNCTION COLE2 in appendix 1). For example, the b() array for a two layer model with the second layer having a complex conductivity would be 'b=sig1,sig2,charge2,frdep2,time2,thick1'.

value gives a steepest descent step. (default parm(1)=.01).

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

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

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

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

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

delta= The third convergence criterion. Convergence is statisfied if the Euclidean norm of the approximate gradient is <= delta. (default delta=0.0).

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

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

$end [end of $parms namelist]

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

iob= Observation-type defined for y(i):
= 1 (default) defines y(i) as the amplitude of the EM coupling (volts/ampere).
= 2 defines y(i) as the phase of the EM coupling, expressed in (-pi*1000,+pi*1000) milliradians.
= 3 defines y(i) as the real part of the EM coupling.
= 4 defines y(i) as the quadrature part of the EM coupling.
(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, =2.0 for phase, =3.0 for real part, or =4.0 for quadrature part of the EM coupling.
(note: for iob=5, m=2 must also be given in $parms).
= 6 defines mixed observation-type frequency soundings at different nn spacings where the i-th observation is given by x(i,2), the same as for iob=5, and the nn spacing is given in x(i,3).
(note: for iob=6, m=3 must also be given in $parms).
(cref: $parms m,b(), $init mm,nn, and DATA MATRIX NOTES).

mm= Number of layers in the model (1<=mm<=10; default mm=1).
Note: make sure $parms k=2*mm-1+icmplx*3.
(cref: $parms k,b(), $init iob,icmplx).

nn= Dipole spacing, separation between closest points of source and receiver wires is given by nn*rl (nn>0, default nn=1, ignored when iob=6).
(cref: $init iob,rl,sl).

sl= Source wire length (meters).

rl= Receiver wire length (meters).

kase= Type of wire model computation:
= 1 (default) for infinitesimal dipole-dipole calculation. If rl=sl, this is a good approximation for nn>=3.
= 2 for finite-wire (double integration) calculation.
(cref: $init fintl1,fintl2, in1,in2, mev1,mev2, nfin).

tol= Tolerance used for acceptance of Hankel transform calculations (default tol=1.e-8).

fintl1= Tolerance for integration along source wire (default fintl1=1.e-4).

fintl2= Tolerance for integration along receiver wire (default fintl2=1.e-6).

in1= Type of integration desired across source wire:
= 1 (default) for adaptive quadrature integration,
= 2 for non-adaptive quadrature integration.

in2= Type of integration desired across receiver wire:
= 1 (default) for adaptive quadrature integration,
= 2 for non-adaptive quadrature integration.

mev1= Maximum number of function evaluations allowed for integration across source wire (default mev1=300).

mev2= Maximum number of function evaluations allowed for integration across receiver wire (default mev2=300).

nfin= the interval in log-space (=0.2/nfin) with which the quintic spline nodes are calculated for the double integration (default nfin=1).

icmplx= Number of layer conductivities in the model which are complex (default icmplx=0). Note: icmplx<=min(4,mm); make sure $parms k=2*mm-1+icmplx*3.
(cref: $parms b(),k and $init mm,lcmplx()).

lcmplx()= array of icmplx-indices (in ascending order) corresponding to the layers whose conductivities are complex, e.g. icmplx=2, lcmplx(1)=1, lcmplx(2)=2 signifies that the conductivity of the first and second layers are to be complex. If icmplx=0, the lcmplx array is ignored.
(cref: $parms b and $init icmplx,mm).

$end [end of $init parameters]

A NOTE ON COMPLEX CONDUCTIVITIES

The complex conductivity function used by program MQLVEMCPL is slightly modified from the Cole-Cole function used by Pelton et. al. (1978). The zero-frequency conductivity, chargeability, and frequency dependence are

the same; however, it was necessary to modify the time constant because of a serious correlation between it and the frequency dependence parameter, c. Therefore, the conversion is

time-constant(here) = time-constant(Pelton)**c

The user may change the functional form of the complex conductivity as long as it still requires four parameters. Such changes should be made to the COMPLEX FUNCTION COLE2 in Appendix 1.

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<=3. The data matrix is read on logical unit ialt (default 10) using an object-time format statement (see any Fortran manual). The number of items read depends on $parms m,iwt and $init iob as previously defined. The various data matrix options are summarized as follows:

(a) Specific observation type, frequency soundings for amplitude, phase, real part, or quadrature part of EM couplings (iob<=4, m=1, and a maximum of 3 items per record):

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

(b) Mixed observation types: amplitude, phase, real part and/or quadrature part of EM couplins frequency soundings (iob=5, m=2, and a maximum of 4 items per record):

1. y(i)= i-th observation (where actual type is defined by x(i,2)).
2. x(i,1)= i-th frequency (x(i,1)>0.0 Hz.).
3. x(i,2)= observation type in y(i); use x(i,2)= 1.0 for amplitude, =2.0 for phase, =3.0 for real part, or =4.0 for quadrature part of EM couplins.
4. x(i,3)= standard deviation of observation i (include only if iwt=1). Note: for joint inversion of amplitude and phase data, a weighted least squares should be used (iwt=1 option) to produce near-equal

magnitudes.

(c) Mixed observation types and dipole spacings: same as (b) type data matrix with the addition of a column specifying the $init dipole spacing parameter nn (iob=6,m=3, and a maximum of 5 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); x(i,2) = 1.0, 2.0, 3.0, or 4.0 as described in (b) above.
4. x(i,3) = nn value to use in computing dipole spacing (x(i,3)>0.0).
5. x(i,4) = standard deviation of observation i (include only if iwt=1).

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

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING.

1. EM coupling amplitude sounding, real conductivities (default iob=1,icmplx=0):

```
example 1.
$parms n=60,k=5,m=1,iprt=-1,sp=1,ialt=5,
 b=.1,.05,.033,100,200$
(2f10.0)
1.98      1.
1.85      1.6
--(etc. for 58 more observations)--
$init mm=3,rl=100,sl=100,nn=2,kase=2$
```

2. EM coupling amplitude sounding, first layer conductivity is complex (default iob=1, note that four numbers are required in the b() array to describe the complex conductivity of the first layer):

```
example 2
$parms n=60,k=8,m=1,iprt=-1,sp=1,ialt=5,
 b=.1,.2,.25,1.,.05,.033,100,200$
(2f10.0)
1.98      1.
1.85      1.6
--(etc. for 58 more observations)--
$init mm=3,icmplx=1,lcmplx=1,rl=100,sl=100,nn=2,kase=2$
```

3. Mixed observation types (iob=5), amplitude and phase of EM coupling, and weighted observations (iwt=1):

```
example 3
$parms n=100,k=5,m=2,iprt=-2,sp=1,iwt=1,ialt=5,
 b=.1,.05,.033,100,200$
(4f10.0)
0.201     1.2       1.        .05
-914.6    1.2       2.        .08
0.2987    4.        1.        .05
-964.0    4.        2.        .09
--(etc. for rest of soundins)--
$init mm=3,iob=5,rl=100,sl=100,nn=2,kase=2$
```

SPECIAL OBJECT FORMAT PHRASES.

One may use special Fortran object formats to skip observations without changing the data matrix. For example, if we wish to use only the phase data in example 3 above, we could set n=50 and use the format (/4f10.0). Similarly, if we wanted only amplitudes to be used in example 3, then the format (4f10.0/) would accomplish the desired result.

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

MULTICS OPERATING INSTRUCTIONS.

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

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

3. Attach "file10" to an input data matrix ascii file if ialt=10 (default) is used. If ialt=5 is selected, then ignore this step, but include the data matrix following the object-time format on "file05"--see examples 1 to 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 MQLVEMCPL by typing: mqlvemcpl

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

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

ERROR MESSAGES.

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

Exponent underflow may occur when the argument is less than 10**-38 on Multics; this is not fatal since 0.0 replaces all underflows. To suppress the underflow messages, the command "set_ufl -off" can be used prior to executing MQLVEMCPL.

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.

DECIPHERING THE PRINTED OUTPUT

Results are printed on logical unit 6 (file06) and on unit 16 (file16). A complete sample output listing of file16 has been included for reference in Appendix 3.

The following table defines additional names or terms used in the printed output files which have not been previously defined as $parms or $init parameters. Names prefixed with a "*" are not used by MQLVEMCPL, but are included here in case the CALL IMSLMQ is replaced by CALL MARQRT as described in Appendix 2. Names prefixed with a "%" apply only to IMSLMQ printout (see IMSL, 1977, documentation of ZXSSQ for details on these output names). Names without any prefix apply to both IMSLMQ and MARQRT printed output.

term	definition
% gradient	the gradient vector (scaled via $parms sp) corresponding to the final solution.
% norm of gradient	length of gradient vector.
% est. sign. digits	estimated number of significant digits of the solution.
% marquardt parameter -or- *lambda	Marquardt (1963) lambda factor to be added to the diagonal of the Jacobian transpose times Jacobian matrix at each iteration (final iteration for IMSLMQ).
% ssq -or- *phi	weighted sum-of-squares residual function defined over the n observations; the value of the function to be minimized by nonlinear least squares.
* iter	major iteration count where 1<=iter<=niter.

* s_e -or- se	standard error of estimate (or weighted root mean square error) defined as se = sqrt(phi/(n-ktip)).
* length	length of the Marquardt adjustment vector.
* gamma	angle (in degrees) between the gradient and Marquardt adjustment vector.
* epsilon test	convergence test passed whenever abs(delta(j)/(tau+abs(b(j)))<e for all j, where delta is the adjustment vector and b is the parameter vector.
* gamma lambda test	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 roundins errors--almost certainly due to the presence of very high correlations among the parameter estimates.
* gamma epsilon test	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 halvins of the increments. When this occurs, the value of phi is presumed minimized whithin the limits of the rounding error.
* -force off	no convergence after niter iterations.
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.
x(i,j)	input x(i,j),j=1,m independent variables (see DATA MATRIX NOTES for specific definitions).

* -unscaled	forced scalep=scaley=0 after the last iteration to produce unscaled statistics on conversence or force off.
* partials -or- % Jacobian (xJtJ)	(*-unscaled;%-scaled) partial derivative Jacobian matrix. The derivatives are the changes in the model calculations with respect to changes in the parameters.
* ptp inverse -or- % xJtJ inverse	inverse of the Jacobian transpose matrix times the Jacobian matrix.
correlation matrix	parameter correlation coefficient matrix derived from the (*) ptp or (%) xJtJ inverse matrix.
std error(J)	parameter standard error defined as error(J)=("-unscaled-"se)*sqrt(ptp(J,J)).
* 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.err/parm	parameter relative error defined as std error(J)/parameter value(J).
rho(0)	zero-frequency resistivity of specified layer.
resistivity(i)	real resistivity of i-th layer.
chargeability	Cole-cole complex resistivity chargeability.
c	Cole-cole complex resistivity frequency-dependence parameter.
tau	Cole-cole complex resistivity time constant.
lyr	layer index for thicknesses and depths.
depth(i)	depth to the bottom of the i-th layer.

ACKNOWLEDGEMENTS

Bruce Smith suggested the writing of this program and offerred many appreciated comments. The review by Dave Campbell significantly improved the documentation.

REFERENCES.

Anderson, W.L., 1979a, Numerical integration of related Hankel transforms of orders 0 and 1 by adaptive digital filtering: Geophysics, v. 7, p. 1287-1305.

_____, 1979b, Program IMSLPW: Marquardt inversion of plane-wave frequency soundings: U.S. Geological Survey Open-File Rept. 79-586, 37 p.

_____, 1979c, Program MARQLOOPS: Marquardt inversion of loop-loop frequency soundings: U.S. Geological Survey Open-File Rept. 79-240, 75 p.

Brown, K.M., and Dennis, J.E., 1972, Derivative free analogues of the Levenburg-Marquardt and Gauss algorithms for nonlinear least squares approximations: Numerische Mathematik, vol. 18, p. 289-297.

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

Kauahikaua, J. and Anderson, W.L., 1979, Computation of Electromagnetic Coupling on a layered halfspace with complex conductivities: U.S. Geological Survey Open-File Rept. 79-1430, 91 p.

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

Pelton, W.H., Ward, S.H., Halloff, P.G., Sill, W.R., and Nelson, P.H., 1978, Mineral discrimination and removal of inductive coupling with multifrequency IP: Geophysics, v. 43, p. 588-609.

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 with beginning line numbers in the following order:

```
      COMPLEX FUNCTION COLE2(F,S0,RM,C,TAU)                                 00000010
      COMPLEX FUNCTION CZEX(B,NEW,R)                                        00000120
      SUBROUTINE EEMCPL(Y,X,PARMS,K,N,TITLE,IOUT)                           00000390
      COMPLEX FUNCTION EX(B2)                                               00001140
      COMPLEX FUNCTION F3(G)                                                00001390
      COMPLEX FUNCTION F7(G)                                                00001470
      COMPLEX FUNCTION F7G(G)                                               00001590
      COMPLEX FUNCTION F12(G)                                               00001710
      COMPLEX FUNCTION F13(G)                                               00001800
      SUBROUTINE FCODE(Y,XM,PARMS,PRNT,F,IF,IDER)                           00001880
      COMPLEX FUNCTION FINFUN(X)                                            00003090
      COMPLEX FUNCTION FINQ(DEL,R,TOL)                                      00003510
      COMPLEX FUNCTION FINQDF(DEL,R1,R2,R3,R4,TOL,NEW)                      00003670
      COMPLEX FUNCTION FUNINT(X)                                            00004700
      SUBROUTINE IEMCPL(DY,XM,B,PRNT,NPRNT,ND,TITLE,IOUT)                   00004860
C*****     MQLVEMCPL     *****                                            00005980
      SUBROUTINE QPOINT(NY,Y,B,C,D,E,F,X1,DELX,XX,YY)                       00006160
      SUBROUTINE QUINT(NY,Y,B,C,D,E,F)                                      00006330
      SUBROUTINE RECUR2(G,V1,F1,L1)                                         00007100
      SUBROUTINE RECURS(G,V1,F1)                                            00007450
      SUBROUTINE SETSPL(FUNC,DEL,RMAX,RMIN)                                 00007760
C--ZQUAD PACKAGE (ZBLOCK,ZQUAD1,ZSUB1,ZSUBA1,ZQUAD2,ZSUB2,ZSUBA2)           00008420
      COMPLEX FUNCTION ZEX(B,NEW,R)                                         00011500
        COMPLEX FUNCTION ZHANKO(X,FUN,TOL,L)                                00011710
        COMPLEX FUNCTION ZLAGHO(X,FUN,TOL,L,NEW)                            00013460
      SUBROUTINE ZQUAD1(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)               00015700
      SUBROUTINE ZQUAD2(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)               00016300
      COMPLEX FUNCTION ZSUB1(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)      00016900
      COMPLEX FUNCTION ZSUB2(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)      00017860
      COMPLEX FUNCTION ZSUBA1(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)     00018820
      COMPLEX FUNCTION ZSUBA2(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)     00019700
      SUBROUTINE POLAR(ZR,ZI,DEG,AMP)                                       00020580
      SUBROUTINE ERRMSG(MSG,MS,I6,I9)                                       00020840
      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                           00021070
      SUBROUTINE SAVER(I,J)                                                 00024490
      SUBROUTINE IMSLMQ(SUBZ,SUBEND)                                        00024680
      SUBROUTINE FPXSSQ(C,N,KIP,F)                                          00027890
      SUBROUTINE LNXSSQ(C,N,KIP,F)                                          00028220
```

Source Availability

The current version of the source code may be obtained by writing directly to the author. A magnetic tape copy of the source code will be sent to requestors to be copied and

returned to the author. This method of releasing the program was selected in order to satisfy requests for the latest updated version. The magnetic tape will be recorded in the following mode (unless otherwise requested):

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

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

```
      COMPLEX FUNCTION COLE2(F,SO,RM,C,TAU)                              00000010
      COMPLEX Z,ONE                                                      00000020
      DATA ONE,HAFPI/(1.0,0.0),1.570796327/                              00000030
      OMEGA=6.28318531*F                                                 00000040
      ARG=HAFPI*C                                                        00000050
      Z=CMPLX(COS(ARG),SIN(ARG))                                         00000060
      R=(OMEGA**C)*TAU                                                   00000070
      Z=ONE/(ONE+Z*R)                                                    00000080
      COLE2=SO/(ONE-RM*(ONE-Z))                                          00000090
      RETURN                                                             00000100
      END                                                                00000110

      COMPLEX FUNCTION CZEX(B,NEW,R)                                     00000120
C--CZEX COMPUTES THE P(R) TERM WHICH IS                                  00000130
C  DOUBLE INTEGRATED OVER FINITE LIMITS.                                 00000140
C  IT IS PART OF THE EQUATION FOR THE                                    00000150
C  ELECTRIC FIELD OF AN ELECTRIC DIPOLE.                                 00000160
C                                                                        00000170
C       B   INDUCTION NUMBER                                            00000180
C       R   DISTANCE                                                    00000190
C       NEW CONTROLS ZLAGH0 INTEGRATION                                 00000200
C                                                                        00000210
C--CZEX IS IDENTICAL TO ZEX EXCEPT THAT                                  00000220
C  IT ALLOWS FOR COMPLEX CONDUCTIVITIES                                  00000230
C  WHICH ARE COMPUTED BY USER-DEFINED COMPLEX ROUTINE                    00000240
C  SIGMA                                                                 00000250
      COMPLEX ZLAGH0,TWODEL3,ONE,SIGMA,SIGMA1,CB,CK                      00000260
      EXTERNAL F3                                                        00000270
      COMMON/CFLAG/CK(10),ICMPLX                                         00000280
      COMMON/PARM/ISTEP,A1,A2,A3,SIG1,A5,M,TOL                           00000290
      COMMON/CONST/DEL,DEL2,TWODEL3                                      00000300
      DATA ONE/(1.0,0.0)/                                                00000310
      CZEX=CMPLX(0.0,0.0)                                                00000320
      CB=CMPLX(B,B)*CSQRT(CK(1))                                         00000330
      IF(M.EQ.1) GO TO 2                                                 00000340
      CZEX=ZLAGH0(ALOG(B),F3,TOL,LW,NEW)/B                               00000350
    2 CZEX=TWODEL3*CZEX+(ONE-(ONE+CB)*CEXP(-CB))/R**3                    00000360
      RETURN                                                             00000370
      END                                                                00000380

      SUBROUTINE EEMCPL(Y,X,PARMS,K,N,TITLE,IOUT)                        00000390
C                                                                        00000400
C  TERMINATION ROUTINE FOR NONLINEAR LEAST SQUARES INVERSION             00000410
C  OF EM-COUPLING DATA                                                   00000420
C                                                                        00000430
      CHARACTER*5 TITLE                                                  00000440
      COMPLEX CK,Z                                                       00000450
      DIMENSION Y(1),X(200,5),PARMS(1),TITLE(16)                         00000460
      COMMON/EM/IOB,RL,FINTL1,IN1,MEV1,LCMPLX(4),NSIG,R(4),Z,KASE        00000470
      COMMON/CFLAG/CK(10),ICMPLX                                         00000480
      WRITE(6,10) TITLE                                                  00000490
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                                   00000500
```

```
 10 FORMAT(12H1E M C P L--,5X,16A5//                                    00000510
   1 28H FINAL UNSCALED PARAMETERS--/)                                  00000520
    MM=(K+1)/2                                                          00000530
    M2=MM+1                                                             00000540
    IF(ICMPLX.GT.0) GO TO 100                                           00000550
    WRITE(6,15)                                                         00000560
    IF(IOUT.EQ.1) WRITE(16,15)                                          00000570
 15 FORMAT(32X,11HRESISTIVITY,2X,3HLYR,10X,5HDEPTH/)                    00000580
    DO 30 I=1,MM                                                        00000590
         RES=1./PARMS(I)                                                00000600
         WRITE(6,20) I,PARMS(I),RES                                     00000610
         IF(IOUT.EQ.1) WRITE(16,20) I,PARMS(I),RES                      00000620
 20      FORMAT(5X,I3,4X,5E16.8)                                        00000630
 30      CONTINUE                                                       00000640
    IF(K.EQ.1) RETURN                                                   00000650
    D=0.0                                                               00000660
    DO 50 I=M2,K                                                        00000670
         D=D+PARMS(I)                                                   00000680
         L=I-MM                                                         00000690
         WRITE(6,40) I,PARMS(I),L,D                                     00000700
         IF(IOUT.EQ.1) WRITE(16,40) I,PARMS(I),L,D                      00000710
 40      FORMAT(5X,I3,4X,E16.8,16X,I3,4X,E16.8)                         00000720
 50      CONTINUE                                                       00000730
    RETURN                                                              00000740
100 WRITE(6,110)                                                        00000750
    IF(IOUT.EQ.1) WRITE(16,110)                                         00000760
110 FORMAT(34X,6HRHO(0),6X,13HCHARGEABILITY,8X,1HC,14X,3HTAU/)          00000770
    KOUNT=0                                                             00000780
    MM=(K+1-3*ICMPLX)/2                                                 00000790
    DO 130 I=1,MM                                                       00000800
         IFLAG=0                                                        00000810
         SIG=PARMS(KOUNT*4+I-KOUNT)                                     00000820
         DO 120 J=1,ICMPLX                                              00000830
              IF(LCMPLX(J).NE.I) GO TO 120                              00000840
              IFLAG=1                                                   00000850
              KOUNT=KOUNT+1                                             00000860
              INDEX=(KOUNT-1)*4+I-KOUNT                                 00000870
              SIG=PARMS(INDEX+1)                                        00000880
              RM=PARMS(INDEX+2)                                         00000890
              C=PARMS(INDEX+3)                                          00000900
              TAU=PARMS(INDEX+4)                                        00000910
              GO TO 125                                                 00000920
120           CONTINUE                                                  00000930
125      RES=1./SIG                                                     00000940
         IF(IFLAG.EQ.0) WRITE(6,20) I,SIG,RES                           00000950
         IF(IOUT.EQ.1.AND.IFLAG.EQ.0) WRITE(16,20) I,SIG,RES            00000960
         IF(IFLAG.EQ.1) WRITE(6,20) I,SIG,RES,RM,C,TAU                  00000970
         IF(IOUT.EQ.1.AND.IFLAG.EQ.1) WRITE(16,20) I,SIG,RES,RM,C,TAU   00000980
130      CONTINUE                                                       00000990
    IF(K.EQ.4) RETURN                                                   00001000
    WRITE(6,140)                                                        00001010
    IF(IOUT.EQ.1) WRITE(16,140)                                         00001020
```

```
  140 FORMAT(/45X,3HLYR,10X,5HDEPTH/)                                    00001030
      M2=NSIG+1                                                          00001040
      D=0.0                                                              00001050
      DO 150 I=M2,K                                                      00001060
           D=D+PARMS(I)                                                  00001070
           L=I-M2+1                                                      00001080
           WRITE(6,40) I,PARMS(I),L,D                                    00001090
           IF(IOUT.EQ.1) WRITE(16,40) I,PARMS(I),L,D                     00001100
  150      CONTINUE                                                      00001110
      RETURN                                                             00001120
      END                                                                00001130

      COMPLEX FUNCTION EX(B2)                                            00001140
      COMPLEX ZHANK0,ZHANK1,XX,YY,CB,CK                                  00001150
      COMMON/CFLAG/CK(10),ICMPLX                                         00001160
      COMMON/PARM/IS,X,Y,R,SIG1,BNYQ,M,TOL                              00001170
      COMMON/FX/XR2                                                      00001180
C  NOTE:  XR2 PASSED TO F11                                              00001190
      EXTERNAL F7,F11                                                    00001200
      B=SQRT(B2)                                                         00001210
      DEL=R/B                                                            00001220
      DEL2=DEL*DEL                                                       00001230
      R2=R*R                                                             00001240
      XR2=X*X/R2                                                         00001250
      BL=ALOG(B)                                                         00001260
      CB=CMPLX(B,B)                                                      00001270
      IF(ICMPLX.GT.0) CB=CB*CSQRT(CK(1))                                 00001280
      EX=CMPLX(0.0,0.0)                                                  00001290
      IF(M.EQ.1) GO TO 10                                                00001300
      XX=ZHANK1(BL,F7,TOL,LW)/B                                          00001310
      YY=ZHANK0(BL,F11,TOL,LW)/B                                         00001320
      EX=(1.-2.*XR2)*XX/R-YY/DEL                                         00001330
   10 EX=EX-CMPLX(0.0,DEL2/(R2*R))*(CMPLX(1.-3.*Y*Y/R2,0.0)              00001340
     1 +(CMPLX(1.0,0.0)+CB)*CEXP(-CB))                                   00001350
      EX=EX*CMPLX(0.0,1./(6.2831853*SIG1*DEL2))                          00001360
      RETURN                                                             00001370
      END                                                                00001380

      COMPLEX FUNCTION F3(G)                                             00001390
      COMPLEX V1,F1,C,ONE                                                00001400
      DATA ONE/(1.0,0.0)/                                                00001410
      CALL RECURS(G,V1,F1)                                               00001420
      C=G                                                                00001430
      F3=(V1*C*(ONE-F1))/((C+V1*F1)*(C+V1))                              00001440
      RETURN                                                             00001450
      END                                                                00001460

      COMPLEX FUNCTION F7(G)                                             00001470
      COMPLEX V1,F1,L1,I1,ONE,TWO,C,FN7                                  00001480
      COMMON/SAVE7/FN7(283),G7(283),N7                                   00001490
      DATA I1,ONE,TWO/(0.0,1.0),(1.0,0.0),(2.0,0.0)/                     00001500
      CALL RECUR2(G,V1,F1,L1)                                            00001510
```

```
      C=G                                                                00001520
      N7=N7+1                                                            00001530
      FN7(N7)=I1*V1*(L1-ONE)                                             00001540
      G7(N7)=G                                                           00001550
      F7=FN7(N7)+(TWO*V1*(ONE-F1))/((C+V1*F1)*(C+V1))                    00001560
      RETURN                                                             00001570
      END                                                                00001580

      COMPLEX FUNCTION F7G(G)                                            00001590
C--KERNEL OF HANKEL TRANSFORM USED BY                                    00001600
C  ROUTINES FINQ AND SCHCOPL                                             00001610
C--CALLS RECUR2                                                          00001620
      COMPLEX V1,F1,L1,I1,ONE,TWO,C                                      00001630
      DATA I1,ONE,TWO/(0.0,1.0),(1.0,0.0),(2.0,0.0)/                     00001640
      CALL RECUR2(G,V1,F1,L1)                                            00001650
      C=G                                                                00001660
      F7G=I1*V1*(L1-ONE)+(TWO*V1*(ONE-F1))/((C+V1*F1)*(C+V1))            00001670
      F7G=F7G/G                                                          00001680
      RETURN                                                             00001690
      END                                                                00001700

      COMPLEX FUNCTION F12(G)                                            00001710
      COMPLEX V1,F1,L1,I1,ONE,TWO,C                                      00001720
      DATA I1,ONE,TWO/(0.0,1.0),(1.0,0.0),(2.0,0.0)/                     00001730
      CALL RECUR2(G,V1,F1,L1)                                            00001740
      C=G                                                                00001750
      F12=I1*V1*(L1-ONE)+(TWO*V1*(ONE-F1))/((C+V1*F1)*(C+V1))            00001760
      F12=C*F12                                                          00001770
      RETURN                                                             00001780
      END                                                                00001790

      COMPLEX FUNCTION F13(G)                                            00001800
      COMPLEX I1,C,V1,F1,L1,ONE                                          00001810
      DATA I1,ONE/(0.0,1.0),(1.0,0.0)/                                   00001820
      CALL RECUR2(G,V1,F1,L1)                                            00001830
      C=G                                                                00001840
      F13=C*I1*V1*(L1-ONE)                                               00001850
      RETURN                                                             00001860
      END                                                                00001870

      SUBROUTINE FCODE(Y,XM,PARMS,PRNT,F,IF,IDER)                        00001880
C                                                                        00001890
C  FORWARD MODELLING ROUTINE PATTERNED AFTER PROGRAM                     00001900
C  EMCUPL, WITH A FEW IMPORTANT DIFFERENCES                              00001910
C    FOR KASE=1, A TRUE DIPOLAR ELECTRIC FIELD                           00001920
C      IS CALCULATED                                                     00001930
C    FOR KASE=2, THE FINITE WIRE MODEL IS USED.                          00001940
C                                                                        00001950
C  WRITTEN BY JKAUAHIKAUA, USGS, SEPTEMBER, 1978                         00001960
C                                                                        00001970
      DIMENSION Y(200),XM(200,5),PRNT(5),SO(4),RM(4),C(4),TAU(4),        00001980
     1 PARMS(1),OLDB(20)                                                 00001990
```

```
      COMPLEX CK,Z,ZSUB1,ZSUBA1,ZERR1,EX,COLE2,FINQDF                        00002000
      EXTERNAL CZEX,ZEX,FINFUN                                               00002010
      COMMON/CFLAG/CK(10),NCMPLX                                             00002020
      COMMON/EM/IOB,RL,FINTL1,IN1,MEV1,LCMPLX(4),NSIG,R1,R2,R3,R4,Z,KASE00002030
      COMMON/MODEL/RK(10),DD(9),M                                            00002040
      COMMON/THICK/D(9)                                                      00002050
      COMMON/PARM/IS,A1,A2,A3,SSIG1,A5,NLYR,TOL                              00002060
      COMMON/ENDS/XXM,YYM,XXN,YYN,DS                                         00002070
      COMMON/FIN/RMAX,RMIN,R,HFL,SIG1,XX,YY                                  00002080
C                                                                            00002090
      IF(IF.EQ.1) OLDF=0.0                                                   00002100
      IF(IF.EQ.1) OLDN=0.0                                                   00002110
      IF(IDER.EQ.0) GO TO 400                                                00002120
      NPARMS=2*M-1                                                           00002130
      IF(NCMPLX.GT.0) NPARMS=NSIG+(M-1)                                      00002140
      DO 397 I=1,NPARMS                                                      00002150
           IF(PARMS(I).NE.OLDB(I)) GO TO 398                                 00002160
  397      CONTINUE                                                          00002170
      GO TO 400                                                              00002180
  398 DO 399 I=1,NPARMS                                                      00002190
  399      OLDB(I)=PARMS(I)                                                  00002200
      GO TO 401                                                              00002210
  400 IF(XM(IF,1).EQ.OLDF.AND.IOB.LE.5) GO TO 200                            00002220
      IF(XM(IF,1).EQ.OLDF.AND.XM(IF,3).EQ.OLDN.AND.IOB.EQ.6) GO TO 200       00002230
      IF(XM(IF,1).EQ.OLDF) GO TO 100                                         00002240
  401 OLDF=XM(IF,1)                                                          00002250
      IF(IOB.EQ.6) OLDN=XM(IF,3)                                             00002260
      NEW=1                                                                  00002270
      IF(NCMPLX.GT.0) GO TO 2                                                00002280
      SIG1=PARMS(1)                                                          00002290
      SSIG1=SIG1                                                             00002300
      M1=M-1                                                                 00002310
      DO 1 I=1,M1                                                            00002320
           RK(I)=PARMS(I)/SIG1                                               00002330
    1      D(I)=PARMS(M+I)                                                   00002340
      RK(M)=PARMS(M)/SIG1                                                    00002350
      GO TO 10                                                               00002360
    2 KOUNT=0                                                                00002370
      DO 4 I=1,M                                                             00002380
           D(I)=PARMS(NSIG+I)                                                00002390
           CK(I)=PARMS(KOUNT*4+I-KOUNT)                                      00002400
           DO 3 J=1,NCMPLX                                                   00002410
                IF(LCMPLX(J).NE.I) GO TO 3                                   00002420
                KOUNT=KOUNT+1                                                00002430
                INDEX=(KOUNT-1)*4+I-KOUNT                                    00002440
                SO(KOUNT)=PARMS(INDEX+1)                                     00002450
                RM(KOUNT)=PARMS(INDEX+2)                                     00002460
                C(KOUNT)=PARMS(INDEX+3)                                      00002470
                TAU(KOUNT)=PARMS(INDEX+4)                                    00002480
                GO TO 4                                                      00002490
    3           CONTINUE                                                     00002500
    4      CONTINUE                                                          00002510
```

```
      SIG1=CK(1)                                                          00002520
      DO 5 I=1,NCMPLX                                                     00002530
           IF(LCMPLX(I).NE.1) GO TO 5                                     00002540
           SIG1=SO(I)                                                     00002550
    5      CONTINUE                                                      00002560
      DO 6 I=1,M                                                          00002570
    6      CK(I)=CK(I)/SIG1                                              00002580
      SSIG1=SIG1                                                          00002590
   10 XB=SQRT(1./(SIG1*3.9478417E-6))                                     00002600
      CON=-1./(SIG1*6.283185308)                                          00002610
      DEL=XB/SQRT(OLDF)                                                   00002620
      DO 20 I=1,M1                                                        00002630
   20      DD(I)=2.*D(I)/DEL                                             00002640
      IF(NCMPLX.EQ.0) GO TO 25                                            00002650
      DO 21 I=1,NCMPLX                                                    00002660
   21      CK(LCMPLX(I))=COLE2(OLDF,SO(I),RM(I),C(I),TAU(I))/SIG1         00002670
   25 IF(KASE.EQ.1) GO TO 170                                             00002680
C                                                                         00002690
      IF(NCMPLX.EQ.0) CALL SETSPL(ZEX,DEL,RMAX,RMIN)                      00002700
      IF(NCMPLX.GT.0) CALL SETSPL(CZEX,DEL,RMAX,RMIN)                     00002710
C                                                                         00002720
  100 IF(KASE.EQ.1) GO TO 170                                             00002730
      IF(IOB.LE.5) GO TO 150                                              00002740
C                                                                         00002750
      XXM=HFL+RL*XM(IF,3)                                                 00002760
      XXN=XXM+RL                                                          00002770
      R1=XXM+HFL                                                          00002780
      R2=XXM-HFL                                                          00002790
      R3=XXN+HFL                                                          00002800
      R4=XXN-HFL                                                          00002810
  150 IF(IN1.EQ.1) Z=ZSUBA1(XXM,XXN,FINTL1,NEV1,ICK,ZERR1,FINFUN,MEV1)    00002820
      IF(IN1.EQ.2) Z=ZSUB1( XXM,XXN,FINTL1,NEV1,ICK,ZERR1,FINFUN,MEV1)    00002830
      IF(MEV1.GE.NEV1-1.AND.ICK.GE.0) GO TO 160                           00002840
      WRITE(6,520) NEV1,MEV1,ICK,F,XXM,XXN                                00002850
  520 FORMAT(40H GAUSS QUADRATURE: COMPUTED INTEGRAL MAY,                 00002860
     1 13H BE ERRONEOUS/2X,3I8,3E16.8)                                    00002870
  160 Z=CON*(Z-FINQDF(DEL,R4,R3,R2,R1,FINTL1,NEW))                        00002880
      IF(NCMPLX.GT.0) Z=Z/CK(1)                                           00002890
      NEW=0                                                               00002900
      GO TO 200                                                           00002910
  170 IF(IOB.EQ.6) R=HFL+(XM(IF,3)+0.5)*RL                                00002920
      Z=EX(R*R/(DEL*DEL))*RL*2.*HFL                                       00002930
      IF(NCMPLX.GT.0) Z=Z/CK(1)                                           00002940
  200 CONTINUE                                                            00002950
      ITYPE=IOB                                                           00002960
      IF(IOB.GT.4) ITYPE=IFIX(XM(IF,2))                                   00002970
      GO TO (210,220,230,240),ITYPE                                       00002980
  210 F=CABS(Z)                                                           00002990
      GO TO 300                                                           00003000
  220 CALL POLAR(REAL(Z),AIMAG(Z),PHZ,AMP)                                00003010
      F=PHZ*17.453293                                                     00003020
      GO TO 300                                                           00003030
```

```
  230 F=REAL(Z)                                                          00003040
      GO TO 300                                                          00003050
  240 F=AIMAG(Z)                                                         00003060
  300 RETURN                                                             00003070
      END                                                                00003080

      COMPLEX FUNCTION FINFUN(X)                                         00003090
C--COMPUTES FINITE INTEGRAL OVER INTERVAL -L,L                           00003100
C  (L PASSED IN THROUGH COMMON AREA FIN) OF                              00003110
C  COMPLEX FUNCTION FUNINT (SPLINE INTERPOLATOR)                        00003120
C  AT FIELD POINT (XX,YY).                                              00003130
C  ASSUMES PRIOR CALL TO SETSPL ...........                             00003140
C--CALLS FUNINT                                                         00003150
C         ZSUBA2 - ADAPTIVE GAUSSIAN INTEGRATION                        00003160
C         ZSUB2  - NON-ADAPTIVE GAUSSIAN INTEGRATION                    00003170
      REAL L                                                             00003180
      EXTERNAL FUNINT                                                    00003190
      COMPLEX ESUM,ZSUBA2,ZSUB2                                          00003200
      COMMON/FIN/R1,R2,R,L,SIG1,XX,YY                                    00003210
      COMMON/ENDS/XM,YM,XN,YN,DS                                         00003220
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW            00003230
C  CHECK TO SEE THAT X IS IN THE RANGE XM -> XN                          00003240
      IF(X.LE.XN.AND.X.GE.XM) GO TO 1                                    00003250
      WRITE(6,100) X                                                     00003260
  100 FORMAT(34H FINFUN: X NOT IN PROPER RANGE, X=,E12.4)                00003270
      STOP                                                               00003280
    1 XX=X                                                               00003290
      YY=YM+DS*(X-XM)                                                    00003300
      IF(ABS(X).LT.L) GO TO 8                                            00003310
      IF(INTYPE.EQ.1) FINFUN=ZSUBA2(X-L,X+L,FINTOL,NEV,ICK,ESUM,FUNINT,  00003320
     1 MEV)                                                              00003330
      IF(INTYPE.EQ.2) FINFUN=ZSUB2(X-L,X+L,FINTOL,NEV,ICK,ESUM,FUNINT,   00003340
     1 MEV)                                                              00003350
      GO TO 10                                                           00003360
    8 XMIN=AMIN1(ABS(X-L),ABS(X+L))                                      00003370
      XMAX=AMAX1(ABS(X-L),ABS(X+L))                                      00003380
      IF(INTYPE.EQ.1) FINFUN=2.*ZSUBA2(0.,XMIN,FINTOL,NEV,ICK,ESUM,      00003390
     1 FUNINT,MEV)                                                       00003400
      IF(INTYPE.EQ.2) FINFUN=2.*ZSUB2(0.,XMIN,FINTOL,NEV,ICK,ESUM,       00003410
     1 FUNINT,MEV)                                                       00003420
      IF(X.EQ.0.0) GO TO 10                                              00003430
      IF(INTYPE.EQ.1) FINFUN=FINFUN+ZSUBA2(XMIN,XMAX,FINTOL,NEV1,ICK,    00003440
     1 ESUM,FUNINT,MEV)                                                  00003450
      IF(INTYPE.EQ.2) FINFUN=FINFUN+ZSUB2(XMIN,XMAX,FINTOL,NEV1,ICK,     00003460
     1 ESUM,FUNINT,MEV)                                                  00003470
      NEV=NEV+NEV1                                                       00003480
   10 RETURN                                                             00003490
      END                                                                00003500

      COMPLEX FUNCTION FINQ(DEL,R,TOL)                                   00003510
C--FINQ CALCULATES THE JO HANKEL TRANSFORM                               00003520
C  (USING DIGITAL FILTER ROUTINE ZHANKO)                                 00003530
```

```
C  REQUIRED BY PROGRAM EMCUPL                                          00003540
C--CALLS ZHANKO, F7G                                                  00003550
      COMMON/MODEL/RK(10),D(9),M                                      00003560
      COMPLEX ZHANKO,ES                                               00003570
      EXTERNAL F7G                                                    00003580
      B=R/DEL                                                         00003590
      FINQ=CMPLX(0.0,0.0)                                             00003600
      IF(M.EQ.1) GO TO 1                                              00003610
      FINQ=ZHANKO(ALOG(B),F7G,TOL,LW)/B                               00003620
      FINQ=FINQ*CMPLX(0.0,1./DEL)                                     00003630
    1 FINQ=FINQ-1./R                                                  00003640
      RETURN                                                          00003650
      END                                                             00003660

      COMPLEX FUNCTION FINQDF(DEL,R1,R2,R3,R4,TOL,NEW)                00003670
C--COMPUTES FINQ AT FOUR SEPARATIONS USING                            00003680
C  PREVIOUS CALCULATED RESULTS IF NEW=0                               00003690
C    DEL   SKIN DEPTH IN FIRST LAYER                                  00003700
C    R1,R2,R3,R4  THE FOUR SEPARATIONS                                00003710
C    NEW  = 1     FIRST CALL FOR A MODEL                              00003720
C         = 0     SUBSEQUENT CALL WHERE PREVIOUS                      00003730
C                 RESULTS MAY BE USED                                 00003740
C                                                                     00003750
C    FINQDF = FINQ(R1)-FIN1(R2) - (FINQ(R3)-FINQ(R4))                 00003760
C                                                                     00003770
C--CALLS FINQ                                                         00003780
      COMPLEX FINQ,Q1,Q2,Q3,Q4,ZERO,QOLD(4)                           00003790
      REAL ROLD(4)                                                    00003800
      INTEGER ISET(4)                                                 00003810
      COMMON/SAVEQ/ROLD,QOLD                                          00003820
      DATA ZERO/(0.0,0.0)/                                            00003830
      FINQDF=ZERO                                                     00003840
      Q1=ZERO                                                         00003850
      Q2=ZERO                                                         00003860
      Q3=ZERO                                                         00003870
      Q4=ZERO                                                         00003880
      DO 1 I=1,4                                                      00003890
    1       ISET(I)=0                                                 00003900
      IF(NEW.EQ.0) GO TO 10                                           00003910
      DO 2 I=1,4                                                      00003920
            ROLD(I)=0E0                                               00003930
    2       QOLD(I)=ZERO                                              00003940
            GO TO 100                                                 00003950
C  CHECK TO SEE IF Q VALUE WAS CALCULATED                             00003960
C  IN PREVIOUS CALL TO FINQDF                                         00003970
   10 DO 20 I=1,4                                                     00003980
            IF(ROLD(I).NE.R1) GO TO 11                                00003990
            Q1=QOLD(I)                                                00004000
            ISET(1)=I+1                                               00004010
            GO TO 20                                                  00004020
   11       IF(ROLD(I).NE.R2) GO TO 12                                00004030
            Q2=QOLD(I)                                                00004040
```

```
            ISET(2)=I+1                                                   00004050
            GO TO 20                                                      00004060
     12     IF(ROLD(I).NE.R3) GO TO 13                                    00004070
            Q3=QOLD(I)                                                    00004080
            ISET(3)=I+1                                                   00004090
            GO TO 20                                                      00004100
     13     IF(ROLD(I).NE.R4) GO TO 20                                    00004110
            Q4=QOLD(I)                                                    00004120
            ISET(4)=I+1                                                   00004130
     20     CONTINUE                                                      00004140
    100 IF(R1.EQ.R2) GO TO 150                                            00004150
        IF(R1.EQ.R3) GO TO 400                                            00004160
    101 IF(ISET(1).GT.0) GO TO 110                                        00004170
        ISET(1)=1                                                         00004180
        Q1=FINQ(DEL,R1,TOL)                                               00004190
    110 IF(ISET(2).GT.0) GO TO 120                                        00004200
        ISET(2)=1                                                         00004210
        Q2=FINQ(DEL,R2,TOL)                                               00004220
    120 FINQDF=Q1-Q2                                                      00004230
    150 IF(R3.EQ.R4) GO TO 300                                            00004240
        IF(ISET(3).GT.0) GO TO 250                                        00004250
        IF(R3.NE.R1.OR.ISET(1).EQ.0) GO TO 155                            00004260
        Q3=Q1                                                             00004270
        GO TO 250                                                         00004280
    155 IF(R3.NE.R2.OR.ISET(2).EQ.0) GO TO 160                            00004290
        Q3=Q2                                                             00004300
        GO TO 250                                                         00004310
    160 Q3=FINQ(DEL,R3,TOL)                                               00004320
        ISET(3)=1                                                         00004330
    250 IF(ISET(4).GT.0) GO TO 290                                        00004340
        IF(R4.NE.R1.OR.ISET(1).EQ.0) GO TO 255                            00004350
        Q4=Q1                                                             00004360
        GO TO 290                                                         00004370
    255 IF(R4.NE.R2.OR.ISET(2).EQ.0) GO TO 260                            00004380
        Q4=Q2                                                             00004390
        GO TO 290                                                         00004400
    260 Q4=FINQ(DEL,R3,TOL)                                               00004410
        ISET(4)=1                                                         00004420
    290 FINQDF=FINQDF-(Q3-Q4)                                             00004430
    300 GO TO 500                                                         00004440
    400 IF(R2.EQ.R4) GO TO 500                                            00004450
        IF(ISET(2).GT.0) GO TO 410                                        00004460
        ISET(2)=1                                                         00004470
        Q2=FINQ(DEL,R2,TOL)                                               00004480
    410 IF(ISET(4).GT.0) GO TO 420                                        00004490
        ISET(4)=1                                                         00004500
        Q4=FINQ(DEL,R4,TOL)                                               00004510
    420 FINQDF=Q4-Q2                                                      00004520
C                                                                         00004530
C  SAVE CALCULATED (ISET(I)=1) VALUES                                     00004540
C                                                                         00004550
    500 IF(ISET(1).NE.1) GO TO 510                                        00004560
```

```
      ROLD(1)=R1                                                        00004570
      QOLD(1)=Q1                                                        00004580
  510 IF(ISET(2).NE.1) GO TO 520                                        00004590
      ROLD(2)=R2                                                        00004600
      QOLD(2)=Q2                                                        00004610
  520 IF(ISET(3).NE.1) GO TO 530                                        00004620
      ROLD(3)=R3                                                        00004630
      QOLD(3)=Q3                                                        00004640
  530 IF(ISET(4).NE.1) GO TO 540                                        00004650
      ROLD(4)=R4                                                        00004660
      QOLD(4)=Q4                                                        00004670
  540 RETURN                                                            00004680
      END                                                               00004690

      COMPLEX FUNCTION FUNINT(X)                                        00004700
C--COMPLEX FUNCTION INTERPOLATION BY QUINTIC SPLINE VIA                 00004710
C  CALL TO 'QPOINT', WHERE THE QUINTIC SPLINE                           00004720
C  COEFFICIENTS AR,BR,CR,DR,ER, AI,BI,CI,DI,EI WERE                     00004730
C  PREVIOUSLY OBTAINED BY SUBR 'QUINT'.                                 00004740
C                                                                       00004750
      DIMENSION SR(80),AR(80),BR(80),CR(80),DR(80),ER(80),              00004760
     & SI(80),AI(80),BI(80),CI(80),DI(80),EI(80)                        00004770
      COMMON/SPLN80/SR,AR,BR,CR,DR,ER,SI,AI,BI,CI,DI,EI,RLM1,DELRLM,NL  00004780
      COMMON/FIN/R1,R2,R0,XL,SIG1,XX,Y                                  00004790
      R=ALOG(SQRT(X*X+Y*Y))                                             00004800
      CALL QPOINT(NL,SR,AR,BR,CR,DR,ER,RLM1,DELRLM,R,YR)                00004810
      CALL QPOINT(NL,SI,AI,BI,CI,DI,EI,RLM1,DELRLM,R,YI)                00004820
      FUNINT=CMPLX(YR,YI)                                               00004830
      RETURN                                                            00004840
      END                                                               00004850

      SUBROUTINE IEMCPL(DY,XM,B,PRNT,NPRNT,ND,TITLE,IOUT)               00004860
C                                                                       00004870
C  INITIALIZATION ROUTINE FOR NONLINEAR LEAST-SQUARES INVERSION         00004880
C  PROGRAM.  INVERSE FOR PROGRAM EMCUPL.                                00004890
C                                                                       00004900
      DIMENSION DY(200),XM(200,5),B(1),PRNT(5),TITLE(16)                00004910
      CHARACTER*5 TITLE                                                 00004920
      COMPLEX CK,ZERR2,Z                                                00004930
      REAL HAKTOL,HAFL                                                  00004940
      INTEGER LCMPLX(4)                                                 00004950
      COMMON/CFLAG/CK(10),ICMPLX                                        00004960
      COMMON/FINERR/HAKTOL,FINTL2,IN2,NFIN,NEV2,MEV2,ZERR2,LW           00004970
      COMMON/FIN/RMAX,RMIN,R0,HAFL,SIG1,XX,YY                           00004980
      COMMON/ENDS/XXM,YYM,XXN,YYN,DS                                    00004990
      COMMON/MODEL/RK(10),DD(9),M                                       00005000
      COMMON/PARM/IS,X,Y,R,SSIG1,A5,NLYR,TOL                            00005010
      COMMON/EM/IOB,RL,FINTL1,IN1,MEV1,LCMPLX,NSIG,R1,R2,R3,R4,Z,KASE   00005020
      EQUIVALENCE (N,NN),(M,MM)                                         00005030
      NAMELIST/INIT/M,MM,SL,RL,TOL,FINTL1,FINTL2,IN1,IN2,               00005040
     1 NFIN,MEV1,MEV2,ICMPLX,KASE,LCMPLX,IOB,N,NN                       00005050
C                                                                       00005060
```

```
      WRITE(6,20) TITLE                                                 00005070
      IF(IOUT.EQ.1) WRITE(16,20) TITLE                                  00005080
   20 FORMAT(12H1E M C P L--,5X,16A5/)                                  00005090
C  PRESET DEFAULTS                                                      00005100
      M=1                                                               00005110
      N=1                                                               00005120
      IOB=1                                                             00005130
      TOL=1.E-8                                                         00005140
      FINTL1=1.E-4                                                      00005150
      FINTL2=1.E-6                                                      00005160
      IN1=1                                                             00005170
      IN2=1                                                             00005180
      NFIN=1                                                            00005190
      MEV1=300                                                          00005200
      MEV2=300                                                          00005210
      KASE=1                                                            00005220
      ICMPLX=0                                                          00005230
C                                                                       00005240
      READ(5,INIT)                                                      00005250
C                                                                       00005260
      HAFL=SL/2.                                                        00005270
C                                                                       00005280
C  THE RANK OF THE TOLERANCES - FINTL1, FINTL2, TOL REFLECTS            00005290
C  THE NESTING OF THE INTEGRALS.  TOL IS THE SMALLEST BECAUSE           00005300
C  IT IS THE INNERMOST.                                                 00005310
C                                                                       00005320
      IF(KASE.EQ.1.AND.TOL.EQ.1.E-8) TOL=FINTL1                         00005330
      IF(M.GT.10.OR.M.LE.0.OR.KASE.GT.2.OR.KASE.LE.0.OR.                00005340
     1 ICMPLX.GT.4.OR.ICMPLX.LT.0.OR.IOB.GT.6.OR.IOB.LE.0)              00005350
     2 CALL ERRMSG("SOME $INIT OUT OF RANGE.",5,6,16)                   00005360
      MINN=N                                                            00005370
      MAXN=N                                                            00005380
      IF(IOB.LE.5) GO TO 7                                              00005390
      MINN=100                                                          00005400
      MAXN=0                                                            00005410
      DO 6 I=1,ND                                                       00005420
           IF(XM(I,2).GT.4.OR.XM(I,2).LE.0)                             00005430
     1     CALL ERRMSG("SOME XM(I,2) OUT OF RANGE",5,6,16)              00005440
           IF(IOB.EQ.5) GO TO 6                                         00005450
           IF(XM(I,3).LE.0) CALL ERRMSG("SOME XM(I,3).LE.0 ",4,6,16)    00005460
           MINN=MIN0(MINN,IFIX(XM(I,3)))                                00005470
           MAXN=MAX0(MAXN,IFIX(XM(I,3)))                                00005480
    6      CONTINUE                                                     00005490
      GO TO 8                                                           00005500
    7 IF(N.LE.0.AND.IOB.LT.6) CALL ERRMSG("N .LE. 0  ",2,6,16)          00005510
C                                                                       00005520
C  DEFINE EQUIVALENT AND DUMMY COMMON PARAMETERS                        00005530
C                                                                       00005540
    8 NLYR=M                                                            00005550
      HAKTOL=TOL                                                        00005560
      YYM=0.0                                                           00005570
      YYN=0.0                                                           00005580
```

```
      DS=0.0                                                                00005590
      YY=0.0                                                                00005600
      Y=0.0                                                                 00005610
      IS=0                                                                  00005620
      NSIG=ICMPLX*4+(M-ICMPLX)                                              00005630
      TWL=2.*HAFL                                                           00005640
      RMAX=TWL+(MAXN+1)*RL                                                  00005650
      RMIN=MINN*RL                                                          00005660
      XXM=HAFL+N*RL                                                         00005670
      XXN=XXM+RL                                                            00005680
      R1=XXM+HAFL                                                           00005690
      R2=XXM-HAFL                                                           00005700
      R3=XXN+HAFL                                                           00005710
      R4=XXN-HAFL                                                           00005720
      R0=HAFL+(0.5+FLOAT(MINN))*RL                                          00005730
      R=R0                                                                  00005740
      XX=R0                                                                 00005750
      X=R                                                                   00005760
C                                                                           00005770
C  WRITE OUT $INIT PARAMETERS                                               00005780
C                                                                           00005790
      WRITE(6,40) M,IOB,ICMPLX,KASE,SL,RL,TOL                               00005800
      IF(IOUT.EQ.1) WRITE(16,40) M,IOB,ICMPLX,KASE,SL,RL,TOL                00005810
   40 FORMAT(/" M="I5,9X,"IOB=",I5,7X,"ICMPLX=",I5,4X,"KASE=",I5,6X/        00005820
     1 " SL=",E12.3,1X,"RL=",E12.3,1X,"TOL=",E11.3/)                        00005830
      IF(ICMPLX.EQ.0) GO TO 43                                              00005840
      WRITE(6,41) (LCMPLX(I),I=1,ICMPLX)                                    00005850
      IF(IOUT.EQ.1) WRITE(16,41) (LCMPLX(I),I=1,ICMPLX)                     00005860
   41 FORMAT(" LCMPLX=",4I5/)                                               00005870
   43 IF(KASE.EQ.1) GO TO 50                                                00005880
      WRITE(6,45) FINTL1,IN1,MEV1,NFIN,FINTL2,IN2,MEV2                      00005890
      IF(IOUT.EQ.1) WRITE(16,45) FINTL1,IN1,MEV1,NFIN,FINTL2,IN2,MEV2       00005900
   45 FORMAT(" FINTL1=",E8.3,1X,"IN1=",I5,7X,"MEV1=",I5,6X,                 00005910
     1 "NFIN=",I5/" FINTL2=",E8.3,1X,"IN2=",I5,7X,"MEV2=",I5/)              00005920
   50 WRITE(6,55)                                                           00005930
      IF(IOUT.EQ.1) WRITE(16,55)                                            00005940
   55 FORMAT(/)                                                             00005950
      RETURN                                                                00005960
      END                                                                   00005970

C*****     MQLVEMCPL     *****                                            00005980
C  NONLINEAR LEAST-SQUARES INVERSION PROGRAM FOR EM                         00005990
C  COUPLING DATA OBSERVED IN THE COLLINEAR DIPOLE-DIPOLE OR BIPOLE-         00006000
C  DIPOLE MODE.  MINIMIZATION OF THE SUM OF SQUARES IS DONE VIA THE         00006010
C  MARQUARDT-LEVENBURG METHOD.  ANALYTICAL DERIVATIVES ARE NOT              00006020
C  NEEDED.                                                                  00006030
C                                                                           00006040
      EXTERNAL IEMCPL,EEMCPL                                                00006050
      CALL INITREF('>UDD>EMODL_INV>JKAUAHIKAUA>ARCHIVES','EMCUPL_OLIB',     00006060
     1 'FCODE')                                                             00006070
      CALL IMSLMQ(IEMCPL,EEMCPL)                                            00006080
C                                                                           00006090
```

```
C  FCODE IS PASSED TO IMSLMQ BY BEING NAMED FCODE                          00006100
C                                                                         00006110
      STOP                                                                00006120
      END                                                                 00006130

      SUBROUTINE QPOINT(NY,Y,B,C,D,E,F,X1,DELX,XX,YY)                     00006140
C  GIVEN THE QUINTIC SPLINE COEFF'S B(*),C(*),D(*),E(*),F(*) AS           00006150
C  OBTAINED FROM SUBR 'QUINT', AND GIVEN NY OBS. DATA Y(NY) EQUALLY       00006160
C  SPACED BY DELX STARTING AT X1, THEN 'QPOINT' INTERPOLATES              00006170
C  YY AT ANY XX IN (X1,X1+(NY-1)*DELX).                                   00006180
C                                                                         00006190
      DIMENSION Y(1),B(1),C(1),D(1),E(1),F(1)                             00006200
      XMAX=X1+(NY-1)*DELX                                                 00006210
      IF(XX.LT.X1.OR.XX.GT.XMAX) GO TO 2                                  00006220
      I=(XX-X1)/DELX+1                                                    00006230
      XI=X1+(I-1)*DELX                                                    00006240
      T=(XX-XI)/DELX                                                      00006250
      YY=((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I)                  00006260
1     RETURN                                                              00006270
2     WRITE(6,3) XX,X1,XMAX                                               00006280
3     FORMAT('0QPOINT ERROR-- XX=',E16.8,' NOT IN CLOSED INTERVAL (',     00006290
     & E16.8,',',E16.8,')')                                               00006300
      GO TO 1                                                             00006310
      END                                                                 00006320

      SUBROUTINE QUINT(NY,Y,B,C,D,E,F)                                    00006330
C--COMPUTES COEFFICIENTS OF A QUINTIC NATURAL SPLINE S(X) GIVEN           00006340
C  THE ORDINATES Y(I) AT ASSUMED EQUIDISTANT POINTS X(I),I=1 TO NY.       00006350
C                                                                         00006360
C  TRANSLATED FROM ALGOL TO FORTRAN BY                                    00006370
C   W.L. ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.              00006380
C  REF: ACM TRANSACTIONS ON MATH. SOFTWARE, SEPT 1976, V.2, N. 3,         00006390
C       PP.281-289.                                                       00006400
C                                                                         00006410
C  PARAMETERS:                                                            00006420
C                                                                         00006430
C  NY =  NUMBER OF DATA POINTS GIVEN IN Y(NY), NY.GT.2.                   00006440
C  Y()=  ARRAY OF NY GIVEN ORDINATES (DIM.GE.NY).                         00006450
C        Y() POINTS ASSUMED EQUALLY SPACED IN X-DIRECTION.                00006460
C  B,C,D,E,F() = RESULTING ARRAYS (EACH DIM.GE.NY) OF                     00006470
C        QUINTIC SPLINE COEFFICIENTS, WHERE                               00006480
C        FOR ANY XX IN [X(I),X(I+1)):                                     00006490
C        S(XX)=((((F(I)*T+E(I))*T+D(I))*T+C(I))*T+B(I))*T+Y(I) WITH       00006500
C        T=(XX-X(I))/DELX,  DELX=(X(I+1)-X(I)) FOR ANY I.                 00006510
C NOTE:  SEE PROC 'QPOINT' TO EVAL THE QUINTIC SPLINE AFTER              00006520
C           'QUINT' IS CALLED.                                            00006530
C                                                                         00006540
      DIMENSION Y(1),B(1),C(1),D(1),E(1),F(1)                             00006550
      IF(NY.LE.2) GO TO 4                                                 00006560
      N=NY-3                                                              00006570
      P=0.0                                                               00006580
      Q=0.0                                                               00006590
```

```
      R=0.0                                                          00006600
      S=0.0                                                          00006610
      T=0.0                                                          00006620
      DO 1 I=1,N                                                     00006630
      U=P*R                                                          00006640
      B(I)=1.0/(66.0-U*R-Q)                                          00006650
      R=26.0-U                                                       00006660
      C(I)=R                                                         00006670
      D(I)=Y(I+3)-3.0*(Y(I+2)-Y(I+1))-Y(I)-U*S-Q*T                   00006680
      Q=P                                                            00006690
      P=B(I)                                                         00006700
      T=S                                                            00006710
      S=D(I)                                                         00006720
1     CONTINUE                                                       00006730
      D(N+2)=0.0                                                     00006740
      N1=N+1                                                         00006750
      D(N1)=0.0                                                      00006760
      DO 2 J=1,N                                                     00006770
      I=N1-J                                                         00006780
      D(I)=(D(I)-C(I)*D(I+1)-D(I+2))*B(I)                            00006790
2     CONTINUE                                                       00006800
      N=NY-1                                                         00006810
      Q=0.0                                                          00006820
      V=D(1)                                                         00006830
      T=V                                                            00006840
      R=V                                                            00006850
      DO 3 I=2,N                                                     00006860
      P=Q                                                            00006870
      Q=R                                                            00006880
      R=D(I)                                                         00006890
      S=T                                                            00006900
      T=P-Q-Q+R                                                      00006910
      F(I)=T                                                         00006920
      U=5.0*(-P+Q)                                                   00006930
      E(I)=U                                                         00006940
      D(I)=10.0*(P+Q)                                                00006950
      C(I)=0.5*(Y(I+1)+Y(I-1)+S-T)-Y(I)-U                            00006960
      B(I)=0.5*(Y(I+1)-Y(I-1)-S-T)-D(I)                              00006970
3     CONTINUE                                                       00006980
      F(1)=V                                                         00006990
      E(1)=0.0                                                       00007000
      E(NY)=0.0                                                      00007010
      D(1)=0.0                                                       00007020
      D(NY)=0.0                                                      00007030
      C(1)=C(2)-10.0*V                                               00007040
      C(NY)=C(NY-1)+10.0*T                                           00007050
      B(1)=Y(2)-Y(1)-C(1)-V                                          00007060
      B(NY)=Y(NY)-Y(NY-1)+C(NY)-T                                    00007070
4     RETURN                                                         00007080
      END                                                            00007090

      SUBROUTINE RECUR2(G,V1,F1,L1)                                  00007100
```

```
C  RECUR2 RECURSIVELY COMPUTES THE ELEMENTS WHICH ARE USED                  00007110
C  IN THE LAYERED-EARTH HANKEL TRANSFORM KERNELS                           00007120
C--ORIGINAL ALGORITHM FROM ANDERSON(1974) HAS BEEN                         00007130
C  MODIFIED TO USE COMPLEX CONDUCTIVITIES IF ICMPLX=1                      00007140
C                                                                          00007150
      COMMON/MODEL/K,D,M                                                   00007160
      COMMON/CFLAG/CK,ICMPLX                                               00007170
      REAL K(10),D(9)                                                      00007180
      COMPLEX C,VM,V1,F1,L1,E,ONE,CK(10)                                   00007190
      DATA ONE/(1.0,0.0)/                                                  00007200
      F1=ONE                                                               00007210
      L1=ONE                                                               00007220
      G2=G*G                                                               00007230
      IF(ICMPLX.EQ.0) VM=CSQRT(CMPLX(G2,2.*K(M)))                          00007240
      IF(ICMPLX.EQ.1) VM=CSQRT(CMPLX(G2-2.*AIMAG(CK(M)),2.*REAL(CK(M))))00007250
      IF(M.EQ.1) GO TO 2                                                   00007260
      J=M-1                                                                00007270
    1 IF(ICMPLX.EQ.0) V1=CSQRT(CMPLX(G2,2.*K(J)))                          00007280
      IF(ICMPLX.EQ.1) V1=CSQRT(CMPLX(G2-2.*AIMAG(CK(J)),2.*REAL(CK(J))))00007290
      E=CEXP(-V1*D(J))                                                     00007300
      C=(ONE-E)/(ONE+E)                                                    00007310
      F1=(VM*F1+V1*C)/(V1+VM*F1*C)                                         00007320
      IF(ICMPLX.EQ.0) E=K(J+1)*V1+K(J)*VM*L1*C                             00007330
      IF(ICMPLX.EQ.1) E=CK(J+1)*V1+CK(J)*VM*L1*C                           00007340
      IF(REAL(E).EQ.0.0.AND.AIMAG(E).EQ.0.0) E=CMPLX(1.E-30,1.E-30)        00007350
      IF(ICMPLX.EQ.0) L1=(K(J)*VM*L1+K(J+1)*V1*C)/E                        00007360
      IF(ICMPLX.EQ.1) L1=(CK(J)*VM*L1+CK(J+1)*V1*C)/E                      00007370
      IF(J.EQ.1) GO TO 3                                                   00007380
      J=J-1                                                                00007390
      VM=V1                                                                00007400
      GO TO 1                                                              00007410
    2 V1=VM                                                                00007420
    3 RETURN                                                               00007430
      END                                                                  00007440

      SUBROUTINE RECURS(G,V1,F1)                                           00007450
C  RECURS RECURSIVELY COMPUTES THE ELEMENTS WHICH ARE USED                 00007460
C  IN THE LAYERED-EARTH HANKEL TRANSFORM KERNELS                           00007470
C--ORIGINAL ALGORITHM FROM ANDERSON(1974) HAS BEEN                         00007480
C--MODIFIED TO USE COMPLEX CONDUCTIVITIES IF ICMPLX=1                      00007490
C                                                                          00007500
      COMPLEX C,VM,V1,F1,EVD,ONE,T,CK(10)                                  00007510
      REAL K(10),D(9)                                                      00007520
      COMMON/MODEL/K,D,M                                                   00007530
      COMMON/CFLAG/CK,ICMPLX                                               00007540
      DATA ONE/(1.0,0.0)/                                                  00007550
      F1=ONE                                                               00007560
      G2=G*G                                                               00007570
      IF(ICMPLX.EQ.0) VM=CSQRT(CMPLX(G2,2.0*K(M)))                         00007580
      IF(ICMPLX.EQ.1) VM=CSQRT(CMPLX(G2-2.*AIMAG(CK(M)),2.*REAL(CK(M))))00007590
      IF(M.EQ.1) GO TO 30                                                  00007600
      J=M-1                                                                00007610
```

```
   10 IF(ICMPLX.EQ.0) V1=CSQRT(CMPLX(G2,2.0*K(J)))                          00007620
      IF(ICMPLX.EQ.1) V1=CSQRT(CMPLX(G2-2.*AIMAG(CK(J)),2.*REAL(CK(J))))00007630
      EVD=-V1*D(J)                                                         00007640
      EVD=CEXP(EVD)                                                        00007650
   20 C=(ONE-EVD)/(ONE+EVD)                                                00007660
      T=VM*F1                                                              00007670
      F1=(T+V1*C)/(V1+T*C)                                                 00007680
      IF(J.EQ.1) GO TO 40                                                  00007690
      J=J-1                                                                00007700
      VM=V1                                                                00007710
      GO TO 10                                                             00007720
   30 V1=VM                                                                00007730
   40 RETURN                                                               00007740
      END                                                                  00007750

      SUBROUTINE SETSPL(FUNC,DEL,RMAX,RMIN)                                00007760
C--COMPUTE THE QUINTIC SPLINE COEFFICIENTS TO                              00007770
C  REPRESENT FUNC(R) FOR THE RANGE RMAX TO RMIN.                          00007780
C  'SETSPL' CALLS 'FUNC' (WHICH CALLS 'ZLAGH1 OR ZLAGH0') AND 'QUINT'.    00007790
C                                                                         00007800
C  PARAMETERS:                                                            00007810
C                                                                         00007820
C  FUNC =  EXTERNAL DECLARED COMPLEX FUNCTION DEFINING THE DIPOLE FIELD   00007830
C          FUNCTION WITH CALLING SEQ: FUNC(B,NEW,R), WHERE                00007840
C           B = ANY IND. NO.                                              00007850
C           NEW = 1 FIRST TIME, 0 OTHERWISE (REF: ZLAGH1 OR ZLAGH0)       00007860
C           R = B*DEL FOR ANY B OR DEL (SKIN DEPTH).                      00007870
C  DEL  =  SKIN DEPTH.                                                    00007880
C  RMAX = MAXIMUM R AT WHICH FUNC WILL NEED TO BE EVALUATED               00007890
C  RMIN = MINIMUM R AT WHICH FUNC WILL NEED TO BE EVALUATED               00007900
C                                                                         00007910
      COMMON/FINERR/HAKTOL,FINTOL,INTYPE,NFIN,NEV,MEV,ESUM,LW             00007920
      COMMON/SPLN80/FDR(80),AR(80),BR(80),CR(80),DR(80),ER(80),           00007930
     & FDI(80),AI(80),BI(80),CI(80),DI(80),EI(80),RLM1,DELRLM,NB          00007940
      COMMON/CONST/DELL,DEL2,Z2DEL3                                        00007950
      COMPLEX FUNC,ESUM,FD,Z2DEL3                                          00007960
C--ISIZE IS THE MAXIMUM POSSIBLE NUMBER OF NODES IN QUINTIC SPLINE        00007970
C  AND ALSO IS THE DIMENSION OF ALL ARRAYS IN COMMON AREA SPLN80.          00007980
      DATA ISIZE/80/                                                       00007990
      DELL=DEL                                                             00008000
      DEL2=DEL*DEL                                                         00008010
      Z2DEL3=CMPLX(0.0,2./(DEL2*DEL))                                      00008020
      BMAX=RMAX/DEL                                                        00008030
      BMIN=RMIN/DEL                                                        00008040
      NB=AINT(5.*ALOG(BMAX/BMIN))+2                                        00008050
      NB=MAX(NB,3)                                                         00008060
      XO=ALOG(BMIN)+NB*0.2                                                 00008070
      NB=NB+3                                                              00008080
C--RANGE OF RMIN,RMAX EXTENDED BY AT LEAST 2 ON EACH                      00008090
C  END IN ORDER TO MAKE THE END CONDITIONS CHOSEN FOR                      00008100
C  THE SPLINE IRRELEVANT TO THE REAL RANGE OF INTEREST.                   00008110
      NRMAX=ISIZE/NB                                                       00008120
```

```
      IF(NFIN.LE.NRMAX) GO TO 3                                          00008130
      IF(NRMAX.GT.0.0) GO TO 2                                           00008140
      WRITE(6,100)                                                       00008150
  100 FORMAT(43H ERROR IN SETSPL: INSUFFICIENT SPLINE NODES)             00008160
      STOP                                                               00008170
    2 NFIN=NRMAX                                                         00008180
      WRITE(6,110) NFIN                                                  00008190
  110 FORMAT(43H ERROR IN SETSPL: NFIN TOO LARGE, RESET TO ,I2)          00008200
3     DELRLM=.2/FLOAT(NFIN)                                              00008210
      X0=X0-DELRLM                                                       00008220
      DO 5 ITIME=1,NFIN                                                  00008230
          NEW=1                                                          00008240
          X0=X0+DELRLM                                                   00008250
          DO 5 J=1,NB                                                    00008260
              I=(NB+1)-J                                                 00008270
              I=NFIN*(I-1)+ITIME                                         00008280
              XX=X0-0.2*J                                                00008290
              BM=EXP(XX)                                                 00008300
              RM=BM*DEL                                                  00008310
              IF(I.EQ.1) RLM1=ALOG(RM)                                   00008320
              FD=FUNC(BM,NEW,RM)                                         00008330
              FDR(I)=REAL(FD)                                            00008340
              FDI(I)=AIMAG(FD)                                           00008350
    5         NEW=0                                                      00008360
      NB=NFIN*NB                                                         00008370
      CALL QUINT(NB,FDR,AR,BR,CR,DR,ER)                                  00008380
      CALL QUINT(NB,FDI,AI,BI,CI,DI,EI)                                  00008390
   10 RETURN                                                             00008400
      END                                                                00008410

C--ZQUAD PACKAGE (ZBLOCK,ZQUAD1,ZSUB1,ZSUBA1,ZQUAD2,ZSUB2,ZSUBA2)        00008420
C  FOR AUTOMATIC COMPLEX GAUSSIAN DOUBLE INTEGRATION OVER A              00008430
C  FINITE INTERVAL.                                                      00008440
C                                                                        00008450
C--MODIFIED BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO     00008460
C  12/30/75.                                                             00008470
C                                                                        00008480
C--USAGE:                                                               00008490
C                                                                        00008500
C  USE 'ZSUB1' OR 'ZSUBA1' FOR 1ST COMPLEX INTEGRATION (CALLS ZQUAD1)    00008510
C  AND 'ZSUB2' OR 'ZSUBA2' FOR 2ND COMPLEX INTEGRATION (CALLS ZQUAD2)    00008520
C                                                                        00008530
C--REFERENCES:                                                          00008540
C                                                                        00008550
C     (1)       PATTERSON,T.N.L, 1973, ALGORITHM FOR AUTOMATIC           00008560
C               NUMERICAL INTEGRATION OVER A FINITE INTERVAL "D1:        00008570
C               ACM COMM. V.16, NO.11, P.694-699.                        00008580
C     (2)       ANDERSON,W.L., 1974, ELECTROMAGNETIC FIELDS ABOUT A      00008590
C               FINITE ELECTRIC WIRE SOURCE:                             00008600
C               N.T.I.S REPORT PB-238199, 209P.                          00008610
C                                                                        00008620
C--NOTES:                                                               00008630
```

```
C                                                                       00008640
C     (A).  SEE REF(1) FOR A COMPLETE DISCUSSION OF THE BASIC           00008650
C           ALGORITHM(S) AS ORIGINALLY DEVELOPED FOR                    00008660
C           SINGLE REAL FUNCTION AUTOMATIC GAUSSIAN INTEGRATION.        00008670
C     (B).  SEE REF(2) FOR A MODIFIED VERSION FOR SINGLE COMPLEX        00008680
C           FUNCTION AUTOMATIC GAUSSIAN INTEGRATION.                    00008690
C     (C).  ALL CALLING PARMS USED BELOW IN THE ZQUAD PACKAGE ARE       00008700
C           IDENTICAL TO THOSE USED IN REF(2).  THEREFORE, SEE          00008710
C           REF(2) FOR COMMENTS ON THESE ANALOGOUS ROUTINES.            00008720
C           REF(1) MAY ALSO BE USED FOR DEFINITIONS OF MOST OF          00008730
C           THE PARMS...                                                00008740
C                                                                       00008750
C-- MULTICS VERSION USES CALL ZBLOCK TO INITILIZE COMMON/ZQUADP         00008760
C   FOR OTHER SYSTEMS, CHANGE SUBROUINTE ZBLOCK TO A                    00008770
C   BLOCK DATA SUBPROGRAM -- AND REMOVE THE ASSIGNMENTS STATEMENTS.     00008780
C                                                                       00008790
      SUBROUTINE ZBLOCK                                                 00008800
      DIMENSION P(381)                                                  00008810
      COMMON/ZQUADP/Q(381)                                              00008820
      DATA MULTICS/0/                                                   00008830
      DATA                                                              00008840
     * P( 1),P( 2),P( 3),P( 4),P( 5),P( 6),P( 7),                       00008850
     * P( 8),P( 9),P(10),P(11),P(12),P(13),P(14),                       00008860
     * P(15),P(16),P(17),P(18),P(19),P(20),P(21),                       00008870
     * P(22),P(23),P(24),P(25),P(26),P(27),P(28)/                       00008880
     *  0.77459666924148337704E 00,0.55555555555555555556E 00,          00008890
     *  0.88888888888888888889E 00,0.26848808986833344073E 00,          00008900
     *  0.96049126870802028342E 00,0.10465622602646726519E 00,          00008910
     *  0.43424374934680255800E 00,0.40139741477596222291E 00,          00008920
     *  0.45091653865847414235E 00,0.13441525524378422036E 00,          00008930
     *  0.51603282997079739697E-01,0.20062852937698902103E 00,          00008940
     *  0.99383196321275502221E 00,0.17001719629940260339E-01,          00008950
     *  0.88845923287225699889E 00,0.92927195315124537686E-01,          00008960
     *  0.62110294673722640294E 00,0.17151190913639138079E 00,          00008970
     *  0.22338668642896688163E 00,0.21915685840158749640E 00,          00008980
     *  0.22551049979820668739E 00,0.67207754295990703540E-01,          00008990
     *  0.25807598096176653565E-01,0.10031427861179557877E 00,          00009000
     *  0.84345657393211062463E-02,0.46462893261757986541E-01,          00009010
     *  0.85755920049990351154E-01,0.10957842105592463824E 00/          00009020
      DATA                                                              00009030
     * P(29),P(30),P(31),P(32),P(33),P(34),P(35),                       00009040
     * P(36),P(37),P(38),P(39),P(40),P(41),P(42),                       00009050
     * P(43),P(44),P(45),P(46),P(47),P(48),P(49),                       00009060
     * P(50),P(51),P(52),P(53),P(54),P(55),P(56)/                       00009070
     *  0.99909812496766759766E 00,0.25447807915618744154E-02,          00009080
     *  0.98153114955374010687E 00,0.16446049854387810934E-01,          00009090
     *  0.92965485742974005667E 00,0.35957103307129322097E-01,          00009100
     *  0.83672593816886873550E 00,0.56979509494123357412E-01,          00009110
     *  0.70249620649152707861E 00,0.76879620499003531043E-01,          00009120
     *  0.53131974364437562397E 00,0.93627109981264473617E-01,          00009130
     *  0.33113539325797683309E 00,0.10566989358023480974E 00,          00009140
     *  0.11248894313318662575E 00,0.11195687302095345688E 00,          00009150
```

```
*  0.11275525672076869161E 00,0.33603877148207730542E-01,                00009160
*  0.12903800100351265626E-01,0.50157139305899537414E-01,                00009170
*  0.42176304415588548391E-02,0.23231446639910269443E-01,                00009180
*  0.42877960025007734493E-01,0.54789210527962865032E-01,                00009190
*  0.12651565562300680114E-02,0.82230079572359296693E-02,                00009200
*  0.17978551568128270333E-01,0.28489754745833548613E-01/                00009210
 DATA                                                                   00009220
* P(57),P(58),P(59),P(60),P(61),P(62),P(63),                            00009230
* P(64),P(65),P(66),P(67),P(68),P(69),P(70),                            00009240
* P(71),P(72),P(73),P(74),P(75),P(76),P(77),                            00009250
* P(78),P(79),P(80),P(81),P(82),P(83),P(84)/                            00009260
*  0.38439810249455532039E-01,0.46813554990628012403E-01,                00009270
*  0.52834946790116519862E-01,0.55978436510476319408E-01,                00009280
*  0.99987288812035761194E 00,0.36322148184553065969E-03,                00009290
*  0.99720625937222195908E 00,0.25790497946856882724E-02,                00009300
*  0.98868475754742947994E 00,0.61155068221172463397E-02,                00009310
*  0.97218287474858179658E 00,0.10498246909621321898E-01,                00009320
*  0.94634285837340290515E 00,0.15406750466559497802E-01,                00009330
*  0.91037115695700429250E 00,0.20594233915912711149E-01,                00009340
*  0.86390793819369047715E 00,0.25869679327214746911E-01,                00009350
*  0.80694053195021761186E 00,0.31073551111687964880E-01,                00009360
*  0.73975604435269475868E 00,0.36064432780782572640E-01,                00009370
*  0.66290966002478059546E 00,0.40715510116944318934E-01,                00009380
*  0.57719571005204581484E 00,0.44914531653632197414E-01,                00009390
*  0.48361802694584102756E 00,0.48564330406673198716E-01/                00009400
 DATA                                                                   00009410
* P( 85),P( 86),P( 87),P( 88),P( 89),P( 90),P( 91),                     00009420
* P( 92),P( 93),P( 94),P( 95),P( 96),P( 97),P( 98),                     00009430
* P( 99),P(100),P(101),P(102),P(103),P(104),P(105),                     00009440
* P(106),P(107),P(108),P(109),P(110),P(111),P(112)/                     00009450
*  0.38335932419873034692E 00,0.51583253952048458777E-01,                00009460
*  0.27774982202182431507E 00,0.53905499335266063927E-01,                00009470
*  0.16823525155220746498E 00,0.55481404356559363988E-01,                00009480
*  0.56344313046592789972E-01,0.56277699831254301273E-01,                00009490
*  0.56377628360384717388E-01,0.16801938574103865271E-01,                00009500
*  0.64519000501757369228E-02,0.25078569652949768707E-01,                00009510
*  0.21088152457266328793E-02,0.11615723319955134727E-01,                00009520
*  0.21438980012503867246E-01,0.27394605263981432516E-01,                00009530
*  0.63260731936263354422E-03,0.41115039786546930472E-02,                00009540
*  0.89892757840641357233E-02,0.14244877372916774306E-01,                00009550
*  0.19219905124727766019E-01,0.23406777495314006201E-01,                00009560
*  0.26417473395058259931E-01,0.27989218255238159704E-01,                00009570
*  0.18073956444538835782E-03,0.12895240826104173921E-02,                00009580
*  0.30577534101755311361E-02,0.52491234548088591251E-02/                00009590
 DATA                                                                   00009600
* P(113),P(114),P(115),P(116),P(117),P(118),P(119),                     00009610
* P(120),P(121),P(122),P(123),P(124),P(125),P(126),                     00009620
* P(127),P(128),P(129),P(130),P(131),P(132),P(133),                     00009630
* P(134),P(135),P(136),P(137),P(138),P(139),P(140)/                     00009640
*  0.77033752332797418482E-02,0.10297116957956355524E-01,                00009650
*  0.12934839663607373455E-01,0.15536775555843982440E-01,                00009660
*  0.18032216390391286320E-01,0.20357755058472159467E-01,                00009670
```

```
*  0.22457265826816098707E-01,0.24282165203336599358E-01,                00009680
*  0.25791626976024229388E-01,0.26952749667633031963E-01,                00009690
*  0.27740702178279681994E-01,0.28138849915627150636E-01,                00009700
*  0.99998243035489159858E 00,0.50536095207862517625E-04,                00009710
*  0.99959879967191068325E 00,0.37774664632698466027E-03,                00009720
*  0.99831663531840739253E 00,0.93836984854238150079E-03,                00009730
*  0.99572410469840718851E 00,0.16811428654214699063E-02,                00009740
*  0.99149572117810613240E 00,0.25687649437940203731E-02,                00009750
*  0.98537149959852037111E 00,0.35728927835172996494E-02,                00009760
*  0.97714151463970571416E 00,0.46710503721143217474E-02,                00009770
*  0.96663785155841656709E 00,0.58434498758356395076E-02/                00009780
 DATA                                                                   00009790
* P(141),P(142),P(143),P(144),P(145),P(146),P(147),                      00009800
* P(148),P(149),P(150),P(151),P(152),P(153),P(154),                      00009810
* P(155),P(156),P(157),P(158),P(159),P(160),P(161),                      00009820
* P(162),P(163),P(164),P(165),P(166),P(167),P(168)/                      00009830
*  0.95373000642576113641E 00,0.70724899954335554680E-02,                00009840
*  0.93832039777959288365E 00,0.83428387539681577056E-02,                00009850
*  0.92034002547001242073E 00,0.96411772970253669653E-02,                00009860
*  0.89974489977694003664E 00,0.10955733387837901648E-01,                00009870
*  0.87651341448470526974E 00,0.12275830560082770087E-01,                00009880
*  0.85064449476835027976E 00,0.13591571009765546790E-01,                00009890
*  0.82215625436498040737E 00,0.14893641664815182035E-01,                00009900
*  0.79108493379984836143E 00,0.16173218729577719942E-01,                00009910
*  0.75748396638051363793E 00,0.17421930159464173747E-01,                00009920
*  0.72142308537009891548E 00,0.18631848256138790186E-01,                00009930
*  0.68298743109107922809E 00,0.19795495048097499488E-01,                00009940
*  0.64227664250975951377E 00,0.20905851445812023852E-01,                00009950
*  0.59940393024224289297E 00,0.21956366305317824939E-01,                00009960
*  0.55449513263193254887E 00,0.22940964229387748761E-01/                00009970
 DATA                                                                   00009980
* P(169),P(170),P(171),P(172),P(173),P(174),P(175),                      00009990
* P(176),P(177),P(178),P(179),P(180),P(181),P(182),                      00010000
* P(183),P(184),P(185),P(186),P(187),P(188),P(189),                      00010010
* P(190),P(191),P(192),P(193),P(194),P(195),P(196)/                      00010020
*  0.50768775753371660215E 00,0.23854052106038540080E-01,                00010030
*  0.45913001198983233287E 00,0.24690524744487676909E-01,                00010040
*  0.40897982122988867241E 00,0.25445769965464765813E-01,                00010050
*  0.35740383783153215238E 00,0.26115673376706097680E-01,                00010060
*  0.30457644155671404334E 00,0.26696622927450359906E-01,                00010070
*  0.25067873030348317661E 00,0.27185513229624791819E-01,                00010080
*  0.19589750271110015392E 00,0.27579749566481873035E-01,                00010090
*  0.14042423315256017459E 00,0.27877251476613701609E-01,                00010100
*  0.84454040083710883710E-01,0.28076455793817246607E-01,                00010110
*  0.28184648949745694339E-01,0.28176319033016602131E-01,                00010120
*  0.28188814180192358694E-01,0.84009692870519326354E-02,                00010130
*  0.32259500250878684614E-02,0.12539284826474884353E-01,                00010140
*  0.10544076228633167722E-02,0.58078616599775673635E-02,                00010150
*  0.10719490006251933623E-01,0.13697302631990716258E-01/                00010160
 DATA                                                                   00010170
* P(197),P(198),P(199),P(200),P(201),P(202),P(203),                      00010180
* P(204),P(205),P(206),P(207),P(208),P(209),P(210),                      00010190
```

```
* P(211),P(212),P(213),P(214),P(215),P(216),P(217),                    00010200
* P(218),P(219),P(220),P(221),P(222),P(223),P(224)/                    00010210
*  0.31630366082226447689E-03,0.20557519893273465236E-02,             00010220
*  0.44946378920320678616E-02,0.71224386864583871532E-02,             00010230
*  0.96099525623638830097E-02,0.11703388747657003101E-01,             00010240
*  0.13208736697529129966E-01,0.13994609127619079852E-01,             00010250
*  0.90372734658751149261E-04,0.64476204130572477933E-03,             00010260
*  0.15288767050877655684E-02,0.26245617274044295626E-02,             00010270
*  0.38516876166398709241E-02,0.51485584789781777618E-02,             00010280
*  0.64674198318036867274E-02,0.77683877779219912200E-02,             00010290
*  0.90161081951956431600E-02,0.10178877529236079733E-01,             00010300
*  0.11228632913408049354E-01,0.12141082601668299679E-01,             00010310
*  0.12895813488012114694E-01,0.13476374833816515982E-01,             00010320
*  0.13870351089139840997E-01,0.14069424957813575318E-01,             00010330
*  0.25157870384280661489E-04,0.18887326450650491366E-03,             00010340
*  0.46918492424785040975E-03,0.84057143271072246365E-03/             00010350
 DATA                                                                 00010360
* P(225),P(226),P(227),P(228),P(229),P(230),P(231),                    00010370
* P(232),P(233),P(234),P(235),P(236),P(237),P(238),                    00010380
* P(239),P(240),P(241),P(242),P(243),P(244),P(245),                    00010390
* P(246),P(247),P(248),P(249),P(250),P(251),P(252)/                    00010400
*  0.12843824718970101768E-02,0.17864463917586498247E-02,             00010410
*  0.23355251860571608737E-02,0.29217249379178197538E-02,             00010420
*  0.35362449977167777340E-02,0.41714193769840788528E-02,             00010430
*  0.48205888648512683476E-02,0.54778666939189508240E-02,             00010440
*  0.61379152800413850435E-02,0.67957855048827733948E-02,             00010450
*  0.74468208324075910174E-02,0.80866093647088599710E-02,             00010460
*  0.87109650797320868736E-02,0.93159241280693950932E-02,             00010470
*  0.98977475240487497440E-02,0.10452925722906011926E-01,             00010480
*  0.10978183152658912470E-01,0.11470482114693874380E-01,             00010490
*  0.11927026053019270040E-01,0.12345262372243838455E-01,             00010500
*  0.12722884982732382906E-01,0.13057836688353048840E-01,             00010510
*  0.13348311463725179953E-01,0.13592756614812395910E-01,             00010520
*  0.13789874783240936517E-01,0.13938625738306850804E-01,             00010530
*  0.14038227896908623303E-01,0.14088159516508301065E-01/             00010540
 DATA                                                                 00010550
* P(253),P(254),P(255),P(256),P(257),P(258),P(259),                    00010560
* P(260),P(261),P(262),P(263),P(264),P(265),P(266),                    00010570
* P(267),P(268),P(269),P(270),P(271),P(272),P(273),                    00010580
* P(274),P(275),P(276),P(277),P(278),P(279),P(280)/                    00010590
*  0.99999759637974846462E 00,0.69379364324108267170E-05,             00010600
*  0.99994399620705437576E 00,0.53275293669780613125E-04,             00010610
*  0.99976049092443204733E 00,0.13575491094922871973E-03,             00010620
*  0.99938033802502358193E 00,0.24921240048299729402E-03,             00010630
*  0.99874561446809511470E 00,0.38974528447328229322E-03,             00010640
*  0.99780535449595727456E 00,0.55429531493037471492E-03,             00010650
*  0.99651414591489027385E 00,0.74028280424450333046E-03,             00010660
*  0.99483150280062100052E 00,0.94536151685852538246E-03,             00010670
*  0.99272134428278861533E 00,0.11674841174299594077E-02,             00010680
*  0.99015137040077015918E 00,0.14049079956551446427E-02,             00010690
*  0.98709252795403406719E 00,0.16561127281544526052E-02,             00010700
*  0.98351865757863272876E 00,0.19197129710138724125E-02,             00010710
```

```
*  0.979406281670862683B1E 00,0.219440692536383883B8E-02,                00010720
*  0.974734459752402667766E 00,0.247895822665756793O7E-02/               00010730
 DATA                                                                    00010740
* P(281),P(282),P(283),P(284),P(285),P(286),P(287),                       00010750
* P(288),P(289),P(290),P(291),P(292),P(293),P(294),                       00010760
* P(295),P(296),P(297),P(298),P(299),P(300),P(301),                       00010770
* P(302),P(303),P(304),P(305),P(306),P(307),P(308)/                       00010780
*  0.969484659502459231776E 00,0.277219576459345099406E-02,              00010790
*  0.963640621569812132526E 00,0.307301843470257832346E-02,              00010800
*  0.957188216109860962746E 00,0.338039799108692038236E-02,              00010810
*  0.950115297521294876566E 00,0.369337791702565081836E-02,              00010820
*  0.942411565191083059816E 00,0.401106872407502339896E-02,              00010830
*  0.934068436157725788006E 00,0.433264096809298285456E-02,              00010840
*  0.925078932907075652366E 00,0.465731729975685477736E-02,              00010850
*  0.915437587155765040646E 00,0.498436456476553860126E-02,              00010860
*  0.905140358813261595196E 00,0.531308660518705656636E-02,              00010870
*  0.894184568335555022866E 00,0.564281810138444415856E-02,              00010880
*  0.882568840247341906846E 00,0.597291956550816580496E-02,              00010890
*  0.870293055548113905856E 00,0.630277344908575871726E-02,              00010900
*  0.857358310886232156536E 00,0.663178124290188789416E-02,              00010910
*  0.843766882672708601046E 00,0.695936140939042293946E-02/              00010920
 DATA                                                                    00010930
* P(309),P(310),P(311),P(312),P(313),P(314),P(315),                       00010940
* P(316),P(317),P(318),P(319),P(320),P(321),P(322),                       00010950
* P(323),P(324),P(325),P(326),P(327),P(328),P(329),                       00010960
* P(330),P(331),P(332),P(333),P(334),P(335),P(336)/                       00010970
*  0.829522194637401400186E 00,0.728494798055380706396E-02,              00010980
*  0.814628787655137413446E 00,0.760798966571905658326E-02,              00010990
*  0.799092290960841401806E 00,0.792794933429484911036E-02,              00011000
*  0.782919394118283016396E 00,0.824430376303286803066E-02,              00011010
*  0.766117819303760090726E 00,0.855654356130768961926E-02,              00011020
*  0.748696293616936602826E 00,0.886417320948249426416E-02,              00011030
*  0.730664521242181261336E 00,0.916671116356078840676E-02,              00011040
*  0.712033155362252034596E 00,0.946368999383006529436E-02,              00011050
*  0.692813769779114702896E 00,0.975465653631741146116E-02,              00011060
*  0.673018830230418479206E 00,0.100391720440568407986E-01,              00011070
*  0.652661665410017496106E 00,0.103168123309476216826E-01,              00011080
*  0.631756437711194230416E 00,0.105871679048851979316E-01,              00011090
*  0.610318113715186400166E 00,0.108498440893373140996E-01,              00011100
*  0.588362434447662541436E 00,0.111044611340069265376E-01/              00011110
 DATA                                                                    00011120
* P(337),P(338),P(339),P(340),P(341),P(342),P(343),                       00011130
* P(344),P(345),P(346),P(347),P(348),P(349),P(350),                       00011140
* P(351),P(352),P(353),P(354),P(355),P(356),P(357),                       00011150
* P(358),P(359),P(360),P(361),P(362),P(363),P(364)/                       00011160
*  0.565905885423654422626E 00,0.113506543159805966026E-01,              00011170
*  0.542965666498311490496E 00,0.115880740330439525686E-01,              00011180
*  0.519559661537457021996E 00,0.118163858908302357636E-01,              00011190
*  0.495706407918761460176E 00,0.120352707852795626306E-01,              00011200
*  0.471425065871658876936E 00,0.122444249816119858996E-01,              00011210
*  0.446735387662028473746E 00,0.124435601907140352636E-01,              00011220
*  0.421657686626163300066E 00,0.126324036435420787656E-01,              00011230
```

```
     *  0.39621280605761593918E 00,0.12810698163877361967E-01,          00011240
     *  0.37042208795007823014E 00,0.12978202239537399286E-01,          00011250
     *  0.34430734159943802278E 00,0.13134690091960152836E-01,          00011260
     *  0.31789081206847668318E 00,0.13279951743930530650E-01,          00011270
     *  0.29119514851824668196E 00,0.13413793085110098513E-01,          00011280
     *  0.26424337241092676194E 00,0.13536035934956213614E-01,          00011290
     *  0.23705884558982972721E 00,0.13646518102571291428E-01/          00011300
      DATA                                                             00011310
     * P(365),P(366),P(367),P(368),P(369),P(370),P(371),                00011320
     * P(372),P(373),P(374),P(375),P(376),P(377),P(378),                00011330
     * P(379),P(380),P(381)/                                           00011340
     *  0.20966523824318119477E 00,0.13745093443001896632E-01,          00011350
     *  0.18208649675925219825E 00,0.13831631909506428676E-01,          00011360
     *  0.15434681148137810869E 00,0.13906019601325461264E-01,          00011370
     *  0.12647058437230196685E 00,0.13968158806516938516E-01,          00011380
     *  0.98482396598119202090E-01,0.14017968039456608810E-01,          00011390
     *  0.70406976042855179063E-01,0.14055382072649964277E-01,          00011400
     *  0.42269164765363603212E-01,0.14080351962553661325E-01,          00011410
     *  0.14093886410782462614E-01,0.14092845069160408355E-01,          00011420
     *  0.14094407090096179347E-01/                                     00011430
      IF(MULTICS.EQ.1) RETURN                                          00011440
      DO 1 I=1,381                                                     00011450
1     Q(I)=P(I)                                                        00011460
      MULTICS=1                                                        00011470
      RETURN                                                           00011480
      END                                                              00011490

      COMPLEX FUNCTION ZEX(B,NEW,R)                                    00011500
C--ZEX COMPUTES THE P(R) TERM WHICH IS                                 00011510
C  DOUBLE INTEGRATED OVER FINITE LIMITS.                               00011520
C  IT IS PART OF THE EQUATION FOR THE                                  00011530
C  ELECTRIC FIELD OF AN ELECTRIC DIPOLE.                               00011540
C                                                                      00011550
C       B    INDUCTION NUMBER                                          00011560
C       R    DISTANCE                                                  00011570
C       NEW CONTROLS ZLAGHO INTEGRATION                                00011580
C                                                                      00011590
      COMPLEX ZLAGHO,TWODEL3,ONE                                       00011600
      EXTERNAL F3                                                      00011610
      COMMON/PARM/ISTEP,A1,A2,A3,A4,A5,M,TOL                           00011620
      COMMON/CONST/DEL,DEL2,TWODEL3                                    00011630
      DATA ONE/(1.0,0.0)/                                              00011640
      ZEX=CMPLX(0.0,0.0)                                               00011650
      IF(M.EQ.1) GO TO 2                                               00011660
      ZEX=ZLAGHO(ALOG(B),F3,TOL,LW,NEW)/B                              00011670
    2 ZEX=TWODEL3*ZEX+(ONE-(ONE+CMPLX(B,B))*CEXP(-CMPLX(B,B)))/R**3    00011680
      RETURN                                                           00011690
      END                                                              00011700

        COMPLEX FUNCTION ZHANKO(X,FUN,TOL,L)                           00011710
C--REVISED VERSION: 12-13-76 -- SEE NOTE(2) BELOW.                     00011720
C--INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*JO(G*B)*DG' DEFINED AS THE   00011730
```

```
C  COMPLEX HANKEL TRANSFORM OF ORDER 0 AND ARGUMENT X(=ALOG(B))          00011740
C  BY CONVOLUTION FILTERING WITH COMPLEX FUNCTION 'FUN'--AND            00011750
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....       00011760
C                                                                       00011770
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.            00011780
C                                                                       00011790
C--PARAMETERS:                                                          00011800
C                                                                       00011810
C     X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE HANKEL TRANSFORM  00011820
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)   00011830
C               OF A REAL ARGUMENT G.                                   00011840
C              NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN  00011850
C                  CALLING PROGRAM AND IN SUBPROGRAM FUN.               00011860
C                  THE COMPLEX FUNCTION FUN SHOULD BE A MONOTONE        00011870
C                  DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00011880
C               FOR REAL-ONLY FUNCTIONS, SUBPROGRAM 'RHANKO' IS ADVISED;00011890
C               HOWEVER, TWO REAL-FUNCTIONS F1(G),F2(G) MAY BE          00011900
C               INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G))00011910
C       TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,     00011920
C                 IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00011930
C                 THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,      00011940
C                 TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON       00011950
C                 THE FUNCTION FUN AND PARAMETER X...IN GENERAL,        00011960
C               A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'  00011970
C               BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00011980
C               RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00011990
C               APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,   00012000
C               ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...  00012010
C       L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE        00012020
C                 CONVOLUTION (L DEPENDS ON TOL AND FUN).               00012030
C               MIN.L=20 AND MAX.L=193--WHICH COULD                     00012040
C                 OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING  00012050
C                 VERY FAST...                                          00012060
C                                                                       00012070
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZHANKO; THE HANKEL 00012080
C  TRANSFORM IS THEN ZHANKO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM   00012090
C  THIS ROUTINE....                                                     00012100
C                                                                       00012110
C--USAGE-- 'ZHANKO' IS CALLED AS FOLLOWS:                               00012120
C     ...                                                               00012130
C     COMPLEX Z,ZHANKO,ZF                                               00012140
C     EXTERNAL ZF                                                       00012150
C     ...                                                               00012160
C     Z=ZHANKO(ALOG(B),ZF,TOL,L)/B                                      00012170
C     ...                                                               00012180
C     END                                                               00012190
C     COMPLEX FUNCTION ZF(G)                                            00012200
C     ...USER SUPPLIED CODE...                                          00012210
C     END                                                               00012220
C                                                                       00012230
C--NOTES:                                                               00012240
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THIS SUBPROGRAM;   00012250
```

```
C               THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS 00012260
C               EXP-UNDERFLOW'S TO 0.0.....                               00012270
C         (2).  THIS SUBPROGRAM IS AN ANSI REVISION OF THE ORIGINAL       00012280
C               PUBLISHED VERSION IN NTIS REPT. PB-242-800, P.45-48;      00012290
C               IMPROVEMENTS HAVE BEEN MADE IN OVERALL EXECUTION TIME,    00012300
C               HOWEVER, THE CALLING SEQUENCE AND FILTER WEIGHTS          00012310
C               WERE NOT CHANGED.                                         00012320
C         (3).  ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED             00012330
C               TO SAVE STORAGE.                                          00012340
        COMPLEX FUN,C,CMAX                                                00012350
        DOUBLE PRECISION A,E,ER,Y1,Y                                      00012360
        DIMENSION T(2),TMAX(2)                                            00012370
        DIMENSION WT(193),W1(76),W2(76),W3(41)                            00012380
        EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1))          00012390
        EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                               00012400
        DATA E/1.2214027581601698340 D0/,ER/.81873075307798185900 D0/     00012410
C--JO-EXTENDED FILTER WEIGHT ARRAYS:                                      00012420
       DATA W1/                                                           00012430
      1 5.8565723E-08, 7.1143477E-11,-7.8395565E-11, 8.7489547E-11,       00012440
      2-8.9007811E-11, 9.8790055E-11,-9.8675347E-11, 1.1118797E-10,       00012450
      3-1.0893474E-10, 1.2543400E-10,-1.1979399E-10, 1.4200767E-10,       00012460
      4-1.3106341E-10, 1.6153229E-10,-1.4238602E-10, 1.8486236E-10,       00012470
      5-1.5315381E-10, 2.1319755E-10,-1.6238115E-10, 2.4824144E-10,       00012480
      6-1.6850378E-10, 2.9243813E-10,-1.6909302E-10, 3.4934366E-10,       00012490
      7-1.6043759E-10, 4.2417082E-10,-1.3690001E-10, 5.2458440E-10,       00012500
      8-8.9946096E-11, 6.6188220E-10,-6.6964033E-12, 8.5276151E-10,       00012510
      9 1.3222770E-10, 1.1219600E-09, 3.5591442E-10, 1.5061956E-09,       00012520
      1 7.0795382E-10, 2.0600379E-09, 1.2535947E-09, 2.8646623E-09,       00012530
      2 2.0904225E-09, 4.0409101E-09, 3.3642886E-09, 5.7687700E-09,       00012540
      3 5.2930786E-09, 8.3164338E-09, 8.2021809E-09, 1.2083635E-08,       00012550
      4 1.2577400E-08, 1.7666303E-08, 1.9143895E-08, 2.5953011E-08,       00012560
      5 2.8983953E-08, 3.8268851E-08, 4.3712685E-08, 5.6590075E-08,       00012570
      6 6.5740136E-08, 8.3864288E-08, 9.8662323E-08, 1.2448811E-07,       00012580
      7 1.4784461E-07, 1.8501974E-07, 2.2129198E-07, 2.7524203E-07,       00012590
      8 3.3094739E-07, 4.0974828E-07, 4.9462868E-07, 6.1030809E-07,       00012600
      9 7.3891802E-07, 9.0939667E-07, 1.1034727E-06, 1.3554600E-06,       00012610
      1 1.6474556E-06, 2.0207696E-06, 2.4591294E-06, 3.0131400E-06/       00012620
       DATA W2/                                                           00012630
      1 3.6701680E-06, 4.4934101E-06, 5.4770076E-06, 6.7015208E-06,       00012640
      2 8.1726989E-06, 9.9954201E-06, 1.2194425E-05, 1.4909101E-05,       00012650
      3 1.8194388E-05, 2.2239184E-05, 2.7145562E-05, 3.3174088E-05,       00012660
      4 4.0499452E-05, 4.9486730E-05, 6.0421440E-05, 7.3822001E-05,       00012670
      5 9.0141902E-05, 1.1012552E-04, 1.3448017E-04, 1.6428337E-04,       00012680
      6 2.0062570E-04, 2.4507680E-04, 2.9930366E-04, 3.6560582E-04,       00012690
      7 4.4651421E-04, 5.4541300E-04, 6.6612648E-04, 8.1365181E-04,       00012700
      8 9.9374786E-04, 1.2138120E-03, 1.4824945E-03, 1.8107657E-03,       00012710
      9 2.2115938E-03, 2.7012675E-03, 3.2991969E-03, 4.0295817E-03,       00012720
      1 4.9214244E-03, 6.0106700E-03, 7.3405529E-03, 8.9643708E-03,       00012730
      2 1.0946310E-02, 1.3365017E-02, 1.6314985E-02, 1.9910907E-02,       00012740
      3 2.4289325E-02, 2.9612896E-02, 3.6070402E-02, 4.3876936E-02,       00012750
      4 5.3264829E-02, 6.4465091E-02, 7.7664144E-02, 9.2918324E-02,       00012760
      5 1.1000121E-01, 1.2811102E-01, 1.4543025E-01, 1.5832248E-01,       00012770
```

```
     6 1.6049224E-01, 1.4170064E-01, 8.8788108E-02,-1.1330934E-02,      00012780
     7-1.5331864E-01,-2.9094670E-01,-2.9084655E-01,-2.9708834E-02,      00012790
     8 3.9009601E-01, 1.7999785E-01,-4.1858139E-01, 1.5317216E-01,      00012800
     9 6.5184953E-02,-1.0751806E-01, 7.8429567E-02,-4.6019124E-02,      00012810
     1 2.5309571E-02,-1.3904823E-02, 7.8187120E-03,-4.5190369E-03/      00012820
      DATA W3/                                                          00012830
     1 2.6729062E-03,-1.6073718E-03, 9.7715622E-04,-5.9804407E-04,      00012840
     2 3.6749320E-04,-2.2635296E-04, 1.3960805E-04,-8.6172618E-05,      00012850
     3 5.3212947E-05,-3.2867888E-05, 2.0304203E-05,-1.2543926E-05,      00012860
     4 7.7499633E-06,-4.7882430E-06, 2.9584108E-06,-1.8278645E-06,      00012870
     5 1.1293571E-06,-6.9778174E-07, 4.3113019E-07,-2.6637753E-07,      00012880
     6 1.6458373E-07,-1.0168954E-07, 6.2829807E-08,-3.8819969E-08,      00012890
     7 2.3985272E-08,-1.4819520E-08, 9.1563774E-09,-5.6573541E-09,      00012900
     8 3.4954514E-09,-2.1597005E-09, 1.3343946E-09,-8.2447148E-10,      00012910
     9 5.0941033E-10,-3.1474631E-10, 1.9447072E-10,-1.2015685E-10,      00012920
     1 7.4241055E-11,-4.5871468E-11, 2.8343095E-11,-1.7513137E-11,      00012930
     2 6.9049613E-12/                                                   00012940
C--$$ENDATA                                                             00012950
C                                                                       00012960
      A=DBLE(EXP(-X-26.3045570))                                        00012970
      ZHANKO=(0.0,0.0)                                                  00012980
      CMAX=(0.0,0.0)                                                    00012990
      L=18                                                              00013000
      Y1=A*0.13120148080287689880 D+12                                  00013010
      Y=Y1                                                              00013020
      DO 110 I=129,146                                                  00013030
      Y=Y*E                                                             00013040
      C=FUN(SNGL(Y))*WT(I)                                              00013050
      ZHANKO=ZHANKO+C                                                   00013060
      TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                                  00013070
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                                  00013080
  110 CONTINUE                                                          00013090
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150                   00013100
      CMAX=TOL*CMAX                                                     00013110
      DO 120 I=147,193                                                  00013120
      Y=Y*E                                                             00013130
      C=FUN(SNGL(Y))*WT(I)                                              00013140
      ZHANKO=ZHANKO+C                                                   00013150
      L=L+1                                                             00013160
      IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130       00013170
  120 CONTINUE                                                          00013180
  130 Y=Y1*E                                                            00013190
      DO 140 I=1,128                                                    00013200
      Y=Y*ER                                                            00013210
      C=FUN(SNGL(Y))*WT(129-I)                                          00013220
      ZHANKO=ZHANKO+C                                                   00013230
      L=L+1                                                             00013240
      IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190       00013250
  140 CONTINUE                                                          00013260
      GO TO 190                                                         00013270
  150 Y=A                                                               00013280
      DO 160 I=1,128                                                    00013290
```

```
      Y=Y*E                                                          00013300
      C=FUN(SNGL(Y))*WT(I)                                           00013310
      ZHANKO=ZHANKO+C                                                00013320
      L=L+1                                                          00013330
      IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170                      00013340
  160 CONTINUE                                                       00013350
  170 Y=A*0.708966799407196320l D+17                                 00013360
      DO 180 I=1,47                                                  00013370
      Y=Y*ER                                                         00013380
      C=FUN(SNGL(Y))*WT(194-I)                                       00013390
      ZHANKO=ZHANKO+C                                                00013400
      L=L+1                                                          00013410
      IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190                      00013420
  180 CONTINUE                                                       00013430
  190 RETURN                                                         00013440
      END                                                            00013450

      COMPLEX FUNCTION ZLAGHO(X,FUN,TOL,L,NEW)                       00013460
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE            00013470
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*JO(G*B)*DG' DEFINED AS THE  00013480
C  COMPLEX HANKEL TRANSFORM OF ORDER 0 AND ARGUMENT X(=ALOG(B))       00013490
C  BY CONVOLUTION FILTERING WITH COMPLEX FUNCTION 'FUN'--AND          00013500
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....     00013510
C                                                                    00013520
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.         00013530
C                                                                    00013540
C--PARAMETERS:                                                       00013550
C                                                                    00013560
C     * X      = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE HANKEL TRANSFORM 00013570
C                'ZLAGHO' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,  00013580
C                SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT, 00013590
C                THEN SUBPROGRAM 'ZHANKO' IS ADVISED FOR GENERAL USE.   00013600
C                (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).           00013610
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)   00013620
C               OF A REAL ARGUMENT G.                                  00013630
C              NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN  00013640
C                  CALLING PROGRAM AND IN SUBPROGRAM FUN.              00013650
C                  THE COMPLEX FUNCTION FUN SHOULD BE A MONOTONE       00013660
C                  DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...00013670
C                FOR REAL-ONLY FUNCTIONS, SUBPROGRAM 'RLAGHO' IS ADVISED;00013680
C                HOWEVER, TWO REAL-FUNCTIONS F1(G),F2(G) MAY BE       00013690
C                INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G))00013700
C       TOL=       REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,   00013710
C                  IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.00013720
C                  THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,    00013730
C                  TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON     00013740
C                  THE FUNCTION FUN AND PARAMETER X...IN GENERAL,      00013750
C                A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY' 00013760
C                BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY00013770
C                RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN 00013780
C                APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,  00013790
C                ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...  00013800
```

```
C        L=          RESULTING NO. FILTER WTS. USED IN THE VARIABLE          00013810
C                    CONVOLUTION (L DEPENDS ON TOL AND FUN).                 00013820
C                  MIN.L=20 AND MAX.L=193--WHICH COULD                       00013830
C                    OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING    00013840
C                    VERY FAST...                                            00013850
C      * NEW=      1 IS NECESSARY 1ST TIME OR BRAND NEW X.                   00013860
C                  0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20          00013870
C                    IS ASSUMED INTERNALLY BY THIS ROUTINE.                  00013880
C                    NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL                00013890
C                    STILL ASSUME X=(LAST X)-0.20 ANYWAY...                  00013900
C                    IT IS THE USERS RESPONSIBILITY TO NORMALIZE              00013910
C                    BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).00013920
C                  THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT        00013930
C                    TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A              00013940
C                    SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING 00013950
C                    ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...           00013960
C                    THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED              00013970
C                    KERNELS WILL BE USED IN THE LAGGED CONVOLUTION          00013980
C                    WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS      00013990
C                    WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)              00014000
C                                                                            00014010
C--THE RESULTING COMPLEX CONVOLUTION SUM IS GIVEN IN ZLAGHO; THE HANKEL      00014020
C  TRANSFORM IS THEN ZLAGHO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM        00014030
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...               00014040
C                                                                            00014050
C--USAGE-- 'ZLAGHO' IS CALLED AS FOLLOWS:                                    00014060
C      ...                                                                   00014070
C      COMPLEX Z,ZLAGHO,ZF                                                   00014080
C      EXTERNAL ZF                                                           00014090
C      ...                                                                   00014100
C      Z=ZLAGHO(ALOG(B),ZF,TOL,L,NEW)/B                                      00014110
C      ...                                                                   00014120
C      END                                                                   00014130
C      COMPLEX FUNCTION ZF(G)                                                00014140
C      ...USER SUPPLIED CODE...                                              00014150
C      END                                                                   00014160
C                                                                            00014170
C--NOTES:                                                                    00014180
C        (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM         00014190
C      BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS           00014200
C      ANY & ALL EXP-UNDERFLOW'S TO 0.0....                                  00014210
C      (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION       00014220
C      METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN               00014230
C      THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,       00014240
C      PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'         00014250
C      BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING      00014260
C      ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).         00014270
C      FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:                      00014280
C                  ...                                                       00014290
C                  NB=AINT(5.*ALOG(BMAX/BMIN))+1                             00014300
C                  NB1=NB+1                                                  00014310
C                  X0=ALOG(BMAX)+.2                                          00014320
```

```
C                NEW=1                                                          00014330
C                DO 1 J=1,NB                                                    00014340
C                I=NB1-J                                                        00014350
C                X=XO-.2*J                                                      00014360
C                ARG(I)=EXP(X)                                                  00014370
C                Z(I)=ZLAGHO(X,ZF,TOL,L,NEW)/ARG(I)                             00014380
C      1         NEW=0                                                          00014390
C                ...                                                            00014400
C      (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),Z(I),I=1,NB FOR             00014410
C      ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,         00014420
C      TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)                 00014430
C      SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.                      00014440
C      (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY                 00014450
C      ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW                     00014460
C      BMAX,BMIN AND BY SETTING NEW=1....                                       00014470
C      (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE00014480
C                                                                               00014490
C                                                                               00014500
        COMPLEX FUN,C,CMAX,SAVE                                                 00014510
        DIMENSION KEY(193),SAVE(193),T(2),TMAX(2)                               00014520
        DIMENSION YT(193),Y1(76),Y2(76),Y3(41)                                  00014530
        EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                                     00014540
        EQUIVALENCE (YT(1),Y1(1)),(YT(77),Y2(1)),(YT(153),Y3(1))                00014550
C--JO-EXTENDED FILTER WEIGHT ARRAYS:                                            00014560
       DATA Y1/                                                                 00014570
      1 5.8565723E-08, 7.1143477E-11,-7.8395565E-11, 8.7489547E-11,             00014580
      2-8.9007811E-11, 9.8790055E-11,-9.8675347E-11, 1.1118797E-10,             00014590
      3-1.0893474E-10, 1.2543400E-10,-1.1979399E-10, 1.4200767E-10,             00014600
      4-1.3106341E-10, 1.6153229E-10,-1.4238602E-10, 1.8486236E-10,             00014610
      5-1.5315381E-10, 2.1319755E-10,-1.6238115E-10, 2.4824144E-10,             00014620
      6-1.6850378E-10, 2.9243813E-10,-1.6909302E-10, 3.4934366E-10,             00014630
      7-1.6043759E-10, 4.2417082E-10,-1.3690001E-10, 5.2458440E-10,             00014640
      8-8.9946096E-11, 6.6188220E-10,-6.6964033E-12, 8.5276151E-10,             00014650
      9 1.3222770E-10, 1.1219600E-09, 3.5591442E-10, 1.5061956E-09,             00014660
      1 7.0795382E-10, 2.0600379E-09, 1.2535947E-09, 2.8646623E-09,             00014670
      2 2.0904225E-09, 4.0409101E-09, 3.3642886E-09, 5.7687700E-09,             00014680
      3 5.2930786E-09, 8.3164338E-09, 8.2021809E-09, 1.2083635E-08,             00014690
      4 1.2577400E-08, 1.7666303E-08, 1.9143895E-08, 2.5953011E-08,             00014700
      5 2.8983953E-08, 3.8268851E-08, 4.3712685E-08, 5.6590075E-08,             00014710
      6 6.5740136E-08, 8.3864288E-08, 9.8662323E-08, 1.2448811E-07,             00014720
      7 1.4784461E-07, 1.8501974E-07, 2.2129198E-07, 2.7524203E-07,             00014730
      8 3.3094739E-07, 4.0974828E-07, 4.9462868E-07, 6.1030809E-07,             00014740
      9 7.3891802E-07, 9.0939667E-07, 1.1034727E-06, 1.3554600E-06,             00014750
      1 1.6474556E-06, 2.0207696E-06, 2.4591294E-06, 3.0131400E-06/             00014760
       DATA Y2/                                                                 00014770
      1 3.6701680E-06, 4.4934101E-06, 5.4770076E-06, 6.7015208E-06,             00014780
      2 8.1726989E-06, 9.9954201E-06, 1.2194425E-05, 1.4909101E-05,             00014790
      3 1.8194388E-05, 2.2239184E-05, 2.7145562E-05, 3.3174088E-05,             00014800
      4 4.0499452E-05, 4.9486730E-05, 6.0421440E-05, 7.3822001E-05,             00014810
      5 9.0141902E-05, 1.1012552E-04, 1.3448017E-04, 1.6428337E-04,             00014820
      6 2.0062570E-04, 2.4507680E-04, 2.9930366E-04, 3.6560582E-04,             00014830
      7 4.4651421E-04, 5.4541300E-04, 6.6612648E-04, 8.1365181E-04,             00014840
```

```
     8 9.9374786E-04, 1.2138120E-03, 1.4824945E-03, 1.8107657E-03,      00014850
     9 2.2115938E-03, 2.7012675E-03, 3.2991969E-03, 4.0295817E-03,      00014860
     1 4.9214244E-03, 6.0106700E-03, 7.3405529E-03, 8.9643708E-03,      00014870
     2 1.0946310E-02, 1.3365017E-02, 1.6314985E-02, 1.9910907E-02,      00014880
     3 2.4289325E-02, 2.9612896E-02, 3.6070402E-02, 4.3876936E-02,      00014890
     4 5.3264829E-02, 6.4465091E-02, 7.7664144E-02, 9.2918324E-02,      00014900
     5 1.1000121E-01, 1.2811102E-01, 1.4543025E-01, 1.5832248E-01,      00014910
     6 1.6049224E-01, 1.4170064E-01, 8.8788108E-02,-1.1330934E-02,      00014920
     7-1.5331864E-01,-2.9094670E-01,-2.9084655E-01,-2.9708834E-02,      00014930
     8 3.9009601E-01, 1.7999785E-01,-4.1858139E-01, 1.5317216E-01,      00014940
     9 6.5184953E-02,-1.0751806E-01, 7.8429567E-02,-4.6019124E-02,      00014950
     1 2.5309571E-02,-1.3904823E-02, 7.8187120E-03,-4.5190369E-03/      00014960
      DATA Y3/                                                         00014970
     1 2.6729062E-03,-1.6073718E-03, 9.7715622E-04,-5.9804407E-04,      00014980
     2 3.6749320E-04,-2.2635296E-04, 1.3960805E-04,-8.6172618E-05,      00014990
     3 5.3212947E-05,-3.2867888E-05, 2.0304203E-05,-1.2543926E-05,      00015000
     4 7.7499633E-06,-4.7882430E-06, 2.9584108E-06,-1.8278645E-06,      00015010
     5 1.1293571E-06,-6.9778174E-07, 4.3113019E-07,-2.6637753E-07,      00015020
     6 1.6458373E-07,-1.0168954E-07, 6.2829807E-08,-3.8819969E-08,      00015030
     7 2.3985272E-08,-1.4819520E-08, 9.1563774E-09,-5.6573541E-09,      00015040
     8 3.4954514E-09,-2.1597005E-09, 1.3343946E-09,-8.2447148E-10,      00015050
     9 5.0941033E-10,-3.1474631E-10, 1.9447072E-10,-1.2015685E-10,      00015060
     1 7.4241055E-11,-4.5871468E-11, 2.8343095E-11,-1.7513137E-11,      00015070
     2 6.9049613E-12/                                                 00015080
C--$$ENDATA                                                           00015090
C                                                                     00015100
      IF(NEW) 10,30,10                                                00015110
10    LAG=-1                                                          00015120
      X0=-X-26.30455704                                               00015130
      DO 20 IR=1,193                                                  00015140
20    KEY(IR)=0                                                       00015150
30    LAG=LAG+1                                                       00015160
      ZLAGH0=(0.0,0.0)                                                00015170
      CMAX=(0.0,0.0)                                                  00015180
      L=0                                                             00015190
      ASSIGN 110 TO M                                                 00015200
      I=129                                                           00015210
      GO TO 200                                                       00015220
110   TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                                00015230
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                                00015240
      I=I+1                                                           00015250
      IF(I.LE.146) GO TO 200                                          00015260
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) GO TO 150                 00015270
      CMAX=TOL*CMAX                                                   00015280
      ASSIGN 120 TO M                                                 00015290
      I=128                                                           00015300
      GO TO 200                                                       00015310
120   IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130     00015320
      I=I-1                                                           00015330
      IF(I.GT.0) GO TO 200                                            00015340
130   ASSIGN 140 TO M                                                 00015350
      I=147                                                           00015360
```

```
      GO TO 200                                                          00015370
140   IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 190        00015380
      I=I+1                                                              00015390
      IF(I.LE.193) GO TO 200                                             00015400
      GO TO 190                                                          00015410
150   ASSIGN 160 TO M                                                    00015420
      I=1                                                                00015430
      GO TO 200                                                          00015440
160   IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 170                          00015450
      I=I+1                                                              00015460
      IF(I.LE.128) GO TO 200                                             00015470
170   ASSIGN 180 TO M                                                    00015480
      I=193                                                              00015490
      GO TO 200                                                          00015500
180   IF(T(1).EQ.0.0.AND.T(2).EQ.0.0) GO TO 190                          00015510
      I=I-1                                                              00015520
      IF(I.GE.147) GO TO 200                                             00015530
190   RETURN                                                             00015540
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)               00015550
200   LOOK=I+LAG                                                         00015560
      IQ=LOOK/194                                                        00015570
      IR=MOD(LOOK,194)                                                   00015580
      IF(IR.EQ.0) IR=1                                                   00015590
      IROLL=IQ*193                                                       00015600
      IF(KEY(IR).LE.IROLL) GO TO 220                                     00015610
210   C=SAVE(IR)*YT(I)                                                   00015620
      ZLAGH0=ZLAGH0+C                                                    00015630
      L=L+1                                                              00015640
      GO TO M,(110,120,140,160,180)                                      00015650
220   KEY(IR)=IROLL+IR                                                   00015660
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))                              00015670
      GO TO 210                                                          00015680
      END                                                                00015690

      SUBROUTINE ZQUAD1(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)            00015700
      COMPLEX F,RESULT,FUNCT,FZERO,ACUM                                  00015710
      DIMENSION FUNCT(127),P(381),RESULT(8)                              00015720
      COMMON/ZQUADP/P                                                    00015730
C--FOLLOWING CALL ONLY FOR MULTICS SYSTEM:                               00015740
      CALL ZBLOCK                                                        00015750
      ICHECK = 0                                                         00015760
C CHECK FOR TRIVIAL CASE.                                                00015770
      IF (A.EQ.B) GO TO 70                                               00015780
C SCALE FACTORS.                                                         00015790
      SUM = (B+A)/2.0                                                    00015800
      DIFF = (B-A)/2.0                                                   00015810
C 1-POINT GAUSS                                                          00015820
      FZERO = F(SUM)                                                     00015830
      RESULT(1) = 2.0*FZERO*DIFF                                         00015840
      I = 0                                                              00015850
      IOLD = 0                                                           00015860
      INEW = 1                                                           00015870
```

```
      K = 2                                                          00015880
      ACUM = (0.0,0.0)                                               00015890
      GO TO 30                                                       00015900
   10 IF (K.EQ.8) GO TO 50                                           00015910
      IF(INEW+IOLD.GE.MEV) GO TO 60                                  00015920
      K = K + 1                                                      00015930
      ACUM = (0.0,0.0)                                               00015940
C CONTRIBUTION FROM FUNCTION VALUES ALREADY COMPUTED.                00015950
      DO 20 J=1,IOLD                                                 00015960
        I = I + 1                                                    00015970
        ACUM = ACUM + P(I)*FUNCT(J)                                  00015980
   20 CONTINUE                                                       00015990
C CONTRIBUTION FROM NEW FUNCTION VALUES.                             00016000
   30 IOLD = IOLD + INEW                                             00016010
      DO 40 J=INEW,IOLD                                              00016020
        I = I + 1                                                    00016030
        X = P(I)*DIFF                                                00016040
        FUNCT(J) = F(SUM+X) + F(SUM-X)                               00016050
        I = I + 1                                                    00016060
        ACUM = ACUM + P(I)*FUNCT(J)                                  00016070
   40 CONTINUE                                                       00016080
      INEW = IOLD + 1                                                00016090
      I = I + 1                                                      00016100
      RESULT(K) = (ACUM+P(I)*FZERO)*DIFF                             00016110
C CHECK FOR CONVERGENCE.                                             00016120
      IF(ABS(REAL(RESULT(K))-REAL(RESULT(K-1))).LE.EPSIL*            00016130
     $ABS(REAL(RESULT(K))).AND.                                      00016140
     $ ABS(AIMAG(RESULT(K))-AIMAG(RESULT(K-1))).LE.EPSIL*            00016150
     $ABS(AIMAG(RESULT(K)))) GO TO 60                                00016160
      GO TO 10                                                       00016170
C CONVERGENCE NOT ACHIEVED.                                          00016180
   50 ICHECK = 1                                                     00016190
C NORMAL TERMINATION.                                                00016200
   60 NPTS = INEW + IOLD                                             00016210
      RETURN                                                         00016220
C TRIVIAL CASE                                                       00016230
   70 K = 2                                                          00016240
      RESULT(1) = (0.0,0.0)                                          00016250
      RESULT(2) = (0.0,0.0)                                          00016260
      NPTS = 0                                                       00016270
      RETURN                                                         00016280
      END                                                            00016290

      SUBROUTINE ZQUAD2(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)        00016300
      COMPLEX F,RESULT,FUNCT,FZERO,ACUM                              00016310
      DIMENSION FUNCT(127), P(381), RESULT(8)                        00016320
      COMMON/ZQUADP/P                                                00016330
C--FOLLOWING CALL ONLY FOR MULTICS SYSTEM;                           00016340
      CALL ZBLOCK                                                    00016350
      ICHECK = 0                                                     00016360
C CHECK FOR TRIVIAL CASE.                                            00016370
      IF (A.EQ.B) GO TO 70                                           00016380
```

```
C SCALE FACTORS.                                                        00016390
      SUM = (B+A)/2.0                                                   00016400
      DIFF = (B-A)/2.0                                                  00016410
C 1-POINT GAUSS                                                         00016420
      FZERO = F(SUM)                                                    00016430
      RESULT(1) = 2.0*FZERO*DIFF                                        00016440
      I = 0                                                             00016450
      IOLD = 0                                                          00016460
      INEW = 1                                                          00016470
      K = 2                                                             00016480
      ACUM = (0.0,0.0)                                                  00016490
      GO TO 30                                                          00016500
   10 IF (K.EQ.8) GO TO 50                                              00016510
      IF(INEW+IOLD.GE.MEV) GO TO 60                                     00016520
      K = K + 1                                                         00016530
      ACUM = (0.0,0.0)                                                  00016540
C CONTRIBUTION FROM FUNCTION VALUES ALREADY COMPUTED.                   00016550
      DO 20 J=1,IOLD                                                    00016560
        I = I + 1                                                       00016570
        ACUM = ACUM + P(I)*FUNCT(J)                                     00016580
   20 CONTINUE                                                          00016590
C CONTRIBUTION FROM NEW FUNCTION VALUES.                                00016600
   30 IOLD = IOLD + INEW                                                00016610
      DO 40 J=INEW,IOLD                                                 00016620
        I = I + 1                                                       00016630
        X = P(I)*DIFF                                                   00016640
        FUNCT(J) = F(SUM+X) + F(SUM-X)                                  00016650
        I = I + 1                                                       00016660
        ACUM = ACUM + P(I)*FUNCT(J)                                     00016670
   40 CONTINUE                                                          00016680
      INEW = IOLD + 1                                                   00016690
      I = I + 1                                                         00016700
      RESULT(K) = (ACUM+P(I)*FZERO)*DIFF                                00016710
C CHECK FOR CONVERGENCE.                                                00016720
      IF(ABS(REAL(RESULT(K))-REAL(RESULT(K-1))).LE.EPSIL*               00016730
     $ABS(REAL(RESULT(K))).AND.                                         00016740
     $ ABS(AIMAG(RESULT(K))-AIMAG(RESULT(K-1))).LE.EPSIL*               00016750
     $ABS(AIMAG(RESULT(K)))) GO TO 60                                   00016760
      GO TO 10                                                          00016770
C CONVERGENCE NOT ACHIEVED.                                             00016780
   50 ICHECK = 1                                                        00016790
C NORMAL TERMINATION.                                                   00016800
   60 NPTS = INEW + IOLD                                                00016810
      RETURN                                                            00016820
C TRIVIAL CASE                                                          00016830
   70 K = 2                                                             00016840
      RESULT(1) = (0.0,0.0)                                             00016850
      RESULT(2) = (0.0,0.0)                                             00016860
      NPTS = 0                                                          00016870
      RETURN                                                            00016880
      END                                                               00016890
```

```
      COMPLEX FUNCTION ZSUB1(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)  00016900
      COMPLEX RELERR,F,RESULT,ESTIM,COMP                                00016910
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION                00016920
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION                    00016930
C ALGORITHM ZQUAD1, TOGETHER WITH, IF NECESSARY, A NON-                 00016940
C ADAPTIVE SUBDIVISION PROCESS.                                         00016950
      DIMENSION RESULT(8)                                               00016960
      INTEGER BAD, OUT                                                  00016970
      LOGICAL RHS                                                       00016980
      EXTERNAL F                                                        00016990
      DATA NMAX/4096/                                                   00017000
      CALL ZQUAD1(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)          00017010
      ZSUB1 = RESULT(K)                                                 00017020
      RELERR = (0.0,0.0)                                                00017030
      IF(REAL(ZSUB1).NE.0.0.AND.AIMAG(ZSUB1).NE.0.0) RELERR=            00017040
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUB1),              00017050
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUB1))                  00017060
C CHECK IF SUBDIVISION IS NEEDED.                                       00017070
      IF (ICHECK.EQ.0) RETURN                                           00017080
C SUBDIVIDE                                                             00017090
      ESTIM=ZSUB1*EPSIL                                                 00017100
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))                   00017110
      IC = 1                                                            00017120
      RHS = .FALSE.                                                     00017130
      N = 1                                                             00017140
      H = B - A                                                         00017150
      BAD = 1                                                           00017160
   10 ZSUB1 = (0.0,0.0)                                                 00017170
      RELERR = (0.0,0.0)                                                00017180
      H = H*0.5                                                         00017190
      N = N + N                                                         00017200
C INTERVAL (A,B) DIVIDED INTO N EQUAL SUBINTERVALS.                     00017210
C INTEGRATE OVER SUBINTERVALS BAD TO (BAD+1) WHERE TROUBLE              00017220
C HAS OCCURRED.                                                         00017230
      M1 = BAD                                                          00017240
      M2 = BAD + 1                                                      00017250
      OUT = 1                                                           00017260
      GO TO 50                                                          00017270
C INTEGRATE OVER SUBINTERVALS 1 TO (BAD-1)                              00017280
   20 M1 = 1                                                            00017290
      M2 = BAD - 1                                                      00017300
      RHS = .FALSE.                                                     00017310
      OUT = 2                                                           00017320
      GO TO 50                                                          00017330
C INTEGRATE OVER SUBINTERVALS (BAD+2) TO N.                             00017340
   30 M1 = BAD + 2                                                      00017350
      M2 = N                                                            00017360
      OUT = 3                                                           00017370
      GO TO 50                                                          00017380
C SUBDIVISION RESULT                                                    00017390
   40 ICHECK = IC                                                       00017400
      IF(REAL(ZSUB1).EQ.0.0) GO TO 42                                   00017410
```

```
      IF(AIMAG(ZSUB1).EQ.0.0) GO TO 44                                  00017420
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUB1)),                       00017430
     $ AIMAG(RELERR)/ABS(AIMAG(ZSUB1)))                                 00017440
      RETURN                                                            00017450
   42 IF(AIMAG(ZSUB1).EQ.0.0) GO TO 46                                  00017460
      RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUB1)))                 00017470
      RETURN                                                            00017480
   44 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUB1)),0.0)                   00017490
      RETURN                                                            00017500
   46 RELERR=(0.0,0.0)                                                  00017510
      RETURN                                                            00017520
C INTEGRATE OVER SUBINTERVALS M1 TO M2.                                 00017530
   50 IF (M1.GT.M2) GO TO 90                                            00017540
      DO 80 JJ=M1,M2                                                    00017550
        J = JJ                                                          00017560
C EXAMINE FIRST THE LEFT OR RIGHT HALF OF THE SUBDIVIDED                00017570
C TROUBLESOME INTERVAL DEPENDING ON THE OBSERVED TREND.                 00017580
        IF (RHS) J = M2 + M1 - JJ                                       00017590
        ALPHA = A + H*(J-1)                                             00017600
        BETA = ALPHA + H                                                00017610
        CALL ZQUAD1(ALPHA, BETA, RESULT, M, EPSIL, NF, ICHECK, F,MEV)   00017620
        COMP = (RESULT(M)-RESULT(M-1))                                  00017630
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                      00017640
        NPTS = NPTS + NF                                                00017650
      IF(NPTS.GE.MEV) GO TO 70                                          00017660
        IF (ICHECK.NE.1) GO TO 70                                       00017670
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                                 00017680
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 100                           00017690
C SUBINTERVAL J HAS CAUSED TROUBLE.                                     00017700
C CHECK IF FURTHER SUBDIVISION SHOULD BE CARRIED OUT.                   00017710
        IF (N.EQ.NMAX) GO TO 60                                         00017720
        BAD = 2*J - 1                                                   00017730
        RHS = .FALSE.                                                   00017740
        IF ((J-2*(J/2)).EQ.0) RHS = .TRUE.                              00017750
        GO TO 10                                                        00017760
   60   IC = -IABS(IC)                                                  00017770
   70   ZSUB1 = ZSUB1 + RESULT(M)                                       00017780
   80 CONTINUE                                                          00017790
      RELERR = RELERR + COMP                                            00017800
90    IF(OUT-2) 20,30,40                                                00017810
C RELAXED CONVERGENCE                                                   00017820
  100 IC = ISIGN(2,IC)                                                  00017830
      GO TO 70                                                          00017840
      END                                                               00017850

      COMPLEX FUNCTION ZSUB2(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)  00017860
      COMPLEX RELERR,F,RESULT,ESTIM,COMP                                00017870
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION                00017880
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION                    00017890
C ALGORITHM ZQUAD2, TOGETHER WITH, IF NECESSARY, A NON-                 00017900
C ADAPTIVE SUBDIVISION PROCESS.                                         00017910
      DIMENSION RESULT(8)                                               00017920
```

```
      INTEGER BAD, OUT                                                  00017930
      LOGICAL RHS                                                       00017940
      EXTERNAL F                                                        00017950
      DATA NMAX/4096/                                                   00017960
      CALL ZQUAD2(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)          00017970
      ZSUB2 = RESULT(K)                                                 00017980
      RELERR = (0.0,0.0)                                                00017990
      IF(REAL(ZSUB2).NE.0.0.AND.AIMAG(ZSUB2).NE.0.0) RELERR=            00018000
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUB2),              00018010
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUB2))                  00018020
C CHECK IF SUBDIVISION IS NEEDED.                                       00018030
      IF (ICHECK.EQ.0) RETURN                                           00018040
C SUBDIVIDE                                                             00018050
      ESTIM=ZSUB2*EPSIL                                                 00018060
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))                   00018070
      IC = 1                                                            00018080
      RHS = .FALSE.                                                     00018090
      N = 1                                                             00018100
      H = B - A                                                         00018110
      BAD = 1                                                           00018120
   10 ZSUB2 = (0.0,0.0)                                                 00018130
      RELERR = (0.0,0.0)                                                00018140
      H = H*0.5                                                         00018150
      N = N + N                                                         00018160
C INTERVAL (A,B) DIVIDED INTO N EQUAL SUBINTERVALS.                     00018170
C INTEGRATE OVER SUBINTERVALS BAD TO (BAD+1) WHERE TROUBLE              00018180
C HAS OCCURRED.                                                         00018190
      M1 = BAD                                                          00018200
      M2 = BAD + 1                                                      00018210
      OUT = 1                                                           00018220
      GO TO 50                                                          00018230
C INTEGRATE OVER SUBINTERVALS 1 TO (BAD-1)                              00018240
   20 M1 = 1                                                            00018250
      M2 = BAD - 1                                                      00018260
      RHS = .FALSE.                                                     00018270
      OUT = 2                                                           00018280
      GO TO 50                                                          00018290
C INTEGRATE OVER SUBINTERVALS (BAD+2) TO N.                             00018300
   30 M1 = BAD + 2                                                      00018310
      M2 = N                                                            00018320
      OUT = 3                                                           00018330
      GO TO 50                                                          00018340
C SUBDIVISION RESULT                                                    00018350
   40 ICHECK = IC                                                       00018360
      IF(REAL(ZSUB2).EQ.0.0) GO TO 42                                   00018370
      IF(AIMAG(ZSUB2).EQ.0.0) GO TO 44                                  00018380
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUB2)),                       00018390
     $ AIMAG(RELERR)/ABS(AIMAG(ZSUB2)))                                 00018400
      RETURN                                                            00018410
   42 IF(AIMAG(ZSUB2).EQ.0.0) GO TO 46                                  00018420
      RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUB2)))                 00018430
      RETURN                                                            00018440
```

```
   44 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUB2)),0.0)                    00018450
      RETURN                                                             00018460
   46 RELERR=(0.0,0.0)                                                   00018470
      RETURN                                                             00018480
C INTEGRATE OVER SUBINTERVALS M1 TO M2.                                  00018490
   50 IF (M1.GT.M2) GO TO 90                                             00018500
      DO 80 JJ=M1,M2                                                     00018510
        J = JJ                                                           00018520
C EXAMINE FIRST THE LEFT OR RIGHT HALF OF THE SUBDIVIDED                 00018530
C TROUBLESOME INTERVAL DEPENDING ON THE OBSERVED TREND.                  00018540
        IF (RHS) J = M2 + M1 - JJ                                        00018550
        ALPHA = A + H*(J-1)                                              00018560
        BETA = ALPHA + H                                                 00018570
        CALL ZQUAD2(ALPHA, BETA, RESULT, M, EPSIL, NF, ICHECK, F,MEV)    00018580
        COMP = (RESULT(M)-RESULT(M-1))                                   00018590
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                       00018600
        NPTS = NPTS + NF                                                 00018610
      IF(NPTS.GE.MEV) GO TO 70                                           00018620
        IF (ICHECK.NE.1) GO TO 70                                        00018630
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                                  00018640
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 100                            00018650
C SUBINTERVAL J HAS CAUSED TROUBLE.                                      00018660
C CHECK IF FURTHER SUBDIVISION SHOULD BE CARRIED OUT.                    00018670
        IF (N.EQ.NMAX) GO TO 60                                          00018680
        BAD = 2*J - 1                                                    00018690
        RHS = .FALSE.                                                    00018700
        IF ((J-2*(J/2)).EQ.0) RHS = .TRUE.                               00018710
        GO TO 10                                                         00018720
   60   IC = -IABS(IC)                                                   00018730
   70   ZSUB2 = ZSUB2 + RESULT(M)                                        00018740
   80 CONTINUE                                                           00018750
      RELERR = RELERR + COMP                                             00018760
90    IF(OUT-2) 20,30,40                                                 00018770
C RELAXED CONVERGENCE                                                    00018780
  100 IC = ISIGN(2,IC)                                                   00018790
      GO TO 70                                                           00018800
      END                                                                00018810

      COMPLEX FUNCTION ZSUBA1(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)  00018820
      COMPLEX RELERR,F,RESULT,ESTIM,COMP                                 00018830
C    THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION                00018840
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION                     00018850
C ALGORITHM ZQUAD1 TOGETHER WITH,  IF NECESSARY AN ADAPTIVE              00018860
C SUBDIVISION PROCESS.  IT IS GENERALLY MORE EFFICIENT THAN              00018870
C THE NON-ADAPTIVE ALGORITHM ZSUB1 BUT IS LIKELY TO BE LESS              00018880
C RELIABLE(SEE COMP.J.,14,189,1971).                                     00018890
      DIMENSION RESULT(8), STACK(100)                                    00018900
      EXTERNAL F                                                         00018910
      DATA ISMAX/100/                                                    00018920
      CALL ZQUAD1(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)           00018930
      ZSUBA1 = RESULT(K)                                                 00018940
      RELERR = (0.0,0.0)                                                 00018950
```

```
      IF(REAL(ZSUBA1).NE.0.0.AND.AIMAG(ZSUBA1).NE.0.0) RELERR=          00018960
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUBA1),             00018970
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUBA1))                 00018980
C CHECK IF SUBDIVISION IS NEEDED                                        00018990
      IF (ICHECK.EQ.0) RETURN                                           00019000
C SUBDIVIDE                                                             00019010
      ESTIM=ZSUBA1*EPSIL                                                00019020
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))                   00019030
      RELERR = (0.0,0.0)                                                00019040
      ZSUBA1 = (0.0,0.0)                                                00019050
      IS = 1                                                            00019060
      IC = 1                                                            00019070
      SUB1 = A                                                          00019080
      SUB3 = B                                                          00019090
   10 SUB2 = (SUB1+SUB3)*0.5                                            00019100
      CALL  ZQUAD1(SUB1, SUB2, RESULT, K, EPSIL, NF, ICHECK, F,MEV)     00019110
      NPTS = NPTS + NF                                                  00019120
      IF(NPTS.GE.MEV) GO TO 50                                          00019130
      COMP = (RESULT(K)-RESULT(K-1))                                    00019140
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                      00019150
      IF (ICHECK.EQ.0) GO TO 30                                         00019160
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                                 00019170
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 70                            00019180
      IF (IS.GE.ISMAX) GO TO 20                                         00019190
C STACK SUBINTERVAL (SUB1,SUB2) FOR FUTURE EXAMINATION                  00019200
      STACK(IS) = SUB1                                                  00019210
      IS = IS + 1                                                       00019220
      STACK(IS) = SUB2                                                  00019230
      IS = IS + 1                                                       00019240
      GO TO 40                                                          00019250
   20 IC = -IABS(IC)                                                    00019260
   30 ZSUBA1 = ZSUBA1 + RESULT(K)                                       00019270
      RELERR = RELERR + COMP                                            00019280
   40 CALL ZQUAD1(SUB2, SUB3, RESULT, K, EPSIL, NF, ICHECK, F,MEV)      00019290
      NPTS = NPTS + NF                                                  00019300
      IF(NPTS.GE.MEV) GO TO 50                                          00019310
      COMP = (RESULT(K)-RESULT(K-1))                                    00019320
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                      00019330
      IF (ICHECK.EQ.0) GO TO 50                                         00019340
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                                 00019350
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 80                            00019360
C SUBDIVIDE INTERVAL (SUB2,SUB3)                                        00019370
      SUB1 = SUB2                                                       00019380
      GO TO 10                                                          00019390
   50 ZSUBA1 = ZSUBA1 + RESULT(K)                                       00019400
      RELERR = RELERR + COMP                                            00019410
      IF(NPTS.GE.MEV) RETURN                                            00019420
      IF (IS.EQ.1) GO TO 60                                             00019430
C SUBDIVIDE THE DELINQUENT INTERVAL LAST STACKED                        00019440
      IS = IS - 1                                                       00019450
      SUB3 = STACK(IS)                                                  00019460
      IS = IS - 1                                                       00019470
```

```
      SUB1 = STACK(IS)                                                  00019480
      GO TO 10                                                          00019490
C SUBDIVISION RESULT                                                    00019500
   60 ICHECK = IC                                                       00019510
      IF(REAL(ZSUBA1).EQ.0.0) GO TO 62                                  00019520
      IF(AIMAG(ZSUBA1).EQ.0.0) GO TO 64                                 00019530
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA1)),                      00019540
     $ AIMAG(RELERR)/ABS(AIMAG(ZSUBA1)))                                00019550
      RETURN                                                            00019560
   62 IF(AIMAG(ZSUBA1).EQ.0.0) GO TO 66                                 00019570
      RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUBA1)))                00019580
      RETURN                                                            00019590
   64 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA1)),0.0)                  00019600
      RETURN                                                            00019610
   66 RELERR=(0.0,0.0)                                                  00019620
      RETURN                                                            00019630
C RELAXED CONVERGENCE                                                   00019640
   70 IC = ISIGN(2,IC)                                                  00019650
      GO TO 30                                                          00019660
   80 IC = ISIGN(2,IC)                                                  00019670
      GO TO 50                                                          00019680
      END                                                               00019690

      COMPLEX FUNCTION ZSUBA2(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV) 00019700
      COMPLEX RELERR,F,RESULT,ESTIM,COMP                                00019710
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION                00019720
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION                    00019730
C ALGORITHM ZQUAD2 TOGETHER WITH,  IF NECESSARY AN ADAPTIVE             00019740
C SUBDIVISION PROCESS.  IT IS GENERALLY MORE EFFICIENT THAN             00019750
C THE NON-ADAPTIVE ALGORITHM ZSUB2 BUT IS LIKELY TO BE LESS             00019760
C RELIABLE(SEE COMP.J.,14,189,1971).                                    00019770
      DIMENSION RESULT(8), STACK(100)                                   00019780
      EXTERNAL F                                                        00019790
      DATA ISMAX/100/                                                   00019800
      CALL ZQUAD2(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)          00019810
      ZSUBA2 = RESULT(K)                                                00019820
      RELERR = (0.0,0.0)                                                00019830
      IF(REAL(ZSUBA2).NE.0.0.AND.AIMAG(ZSUBA2).NE.0.0) RELERR=          00019840
     $ CMPLX(ABS(REAL(RESULT(K)-RESULT(K-1)))/REAL(ZSUBA2),             00019850
     $ ABS(AIMAG(RESULT(K)-RESULT(K-1)))/AIMAG(ZSUBA2))                 00019860
C CHECK IF SUBDIVISION IS NEEDED                                        00019870
      IF (ICHECK.EQ.0) RETURN                                           00019880
C SUBDIVIDE                                                             00019890
      ESTIM=ZSUBA2*EPSIL                                                00019900
      ESTIM=CMPLX(ABS(REAL(ESTIM)),ABS(AIMAG(ESTIM)))                   00019910
      RELERR = (0.0,0.0)                                                00019920
      ZSUBA2 = (0.0,0.0)                                                00019930
      IS = 1                                                            00019940
      IC = 1                                                            00019950
      SUB1 = A                                                          00019960
      SUB3 = B                                                          00019970
   10 SUB2 = (SUB1+SUB3)*0.5                                            00019980
```

```
      CALL  ZQUAD2(SUB1, SUB2, RESULT, K, EPSIL, NF, ICHECK, F,MEV)     00019990
      NPTS = NPTS + NF                                                  00020000
      IF(NPTS.GE.MEV) GO TO 50                                          00020010
      COMP = (RESULT(K)-RESULT(K-1))                                    00020020
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                      00020030
      IF (ICHECK.EQ.0) GO TO 30                                         00020040
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                                 00020050
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 70                            00020060
      IF (IS.GE.ISMAX) GO TO 20                                         00020070
C STACK SUBINTERVAL (SUB1,SUB2) FOR FUTURE EXAMINATION                  00020080
      STACK(IS) = SUB1                                                  00020090
      IS = IS + 1                                                       00020100
      STACK(IS) = SUB2                                                  00020110
      IS = IS + 1                                                       00020120
      GO TO 40                                                          00020130
   20 IC = -IABS(IC)                                                    00020140
   30 ZSUBA2 = ZSUBA2 + RESULT(K)                                       00020150
      RELERR = RELERR + COMP                                            00020160
   40 CALL ZQUAD2(SUB2, SUB3, RESULT, K, EPSIL, NF, ICHECK, F,MEV)      00020170
      NPTS = NPTS + NF                                                  00020180
      IF(NPTS.GE.MEV) GO TO 50                                          00020190
      COMP = (RESULT(K)-RESULT(K-1))                                    00020200
      COMP=CMPLX(ABS(REAL(COMP)),ABS(AIMAG(COMP)))                      00020210
      IF (ICHECK.EQ.0) GO TO 50                                         00020220
      IF(REAL(COMP).LE.REAL(ESTIM).AND.                                 00020230
     $ AIMAG(COMP).LE.AIMAG(ESTIM)) GO TO 80                            00020240
C SUBDIVIDE INTERVAL (SUB2,SUB3)                                        00020250
      SUB1 = SUB2                                                       00020260
      GO TO 10                                                          00020270
   50 ZSUBA2 = ZSUBA2 + RESULT(K)                                       00020280
      RELERR = RELERR + COMP                                            00020290
      IF(NPTS.GE.MEV) RETURN                                            00020300
      IF (IS.EQ.1) GO TO 60                                             00020310
C SUBDIVIDE THE DELINQUENT INTERVAL LAST STACKED                        00020320
      IS = IS - 1                                                       00020330
      SUB3 = STACK(IS)                                                  00020340
      IS = IS - 1                                                       00020350
      SUB1 = STACK(IS)                                                  00020360
      GO TO 10                                                          00020370
C SUBDIVISION RESULT                                                    00020380
   60 ICHECK = IC                                                       00020390
      IF(REAL(ZSUBA2).EQ.0.0) GO TO 62                                  00020400
      IF(AIMAG(ZSUBA2).EQ.0.0) GO TO 64                                 00020410
      RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA2)),                      00020420
     $ AIMAG(RELERR)/ABS(AIMAG(ZSUBA2)))                                00020430
      RETURN                                                            00020440
   62 IF(AIMAG(ZSUBA2).EQ.0.0) GO TO 66                                 00020450
      RELERR=CMPLX(0.0,AIMAG(RELERR)/ABS(AIMAG(ZSUBA2)))                00020460
      RETURN                                                            00020470
   64 RELERR=CMPLX(REAL(RELERR)/ABS(REAL(ZSUBA2)),0.0)                  00020480
      RETURN                                                            00020490
   66 RELERR=(0.0,0.0)                                                  00020500
```

```
      RETURN                                                            00020510
C RELAXED CONVERGENCE                                                   00020520
   70 IC = ISIGN(2,IC)                                                  00020530
      GO TO 30                                                          00020540
   80 IC = ISIGN(2,IC)                                                  00020550
      GO TO 50                                                          00020560
      END                                                               00020570

      SUBROUTINE POLAR(ZR,ZI,DEG,AMP)                                   00020580
C        PARMS  ZR = GIVEN REAL(Z) OR X-COORD.                          00020590
C               ZI = GIVEN IMAG(Z) OR Y-COORD.                          00020600
C               DEG= COMPUTED PHASE IN DEGREES (0,360.)                 00020610
C               AMP= COMPUTED AMPLITUDE.                                00020620
C                                                                       00020630
      DATA PI,PI2/3.1415927,6.2831853/                                  00020640
      IF(ZR.EQ.0.AND.ZI.EQ.0) GO TO 9                                   00020650
      PV=ATAN2(ABS(ZI),ABS(ZR))                                         00020660
      IF(ZI.GE.0.AND.ZR.GE.0) GO TO 10                                  00020670
      IF(ZI.GE.0.AND.ZR.LT.0) GO TO 20                                  00020680
      IF(ZI.LT.0.AND.ZR.LE.0) GO TO 30                                  00020690
      RAD=PI2-PV                                                        00020700
      GO TO 40                                                          00020710
    9 DEG=0.                                                            00020720
      AMP=0.                                                            00020730
      RETURN                                                            00020740
   10 RAD=PV                                                            00020750
      GO TO 40                                                          00020760
   20 RAD=PI-PV                                                         00020770
      GO TO 40                                                          00020780
   30 RAD=PI+PV                                                         00020790
   40 AMP=SQRT(ZR*ZR+ZI*ZI)                                             00020800
      DEG=57.29577951*RAD                                               00020810
      RETURN                                                            00020820
      END                                                               00020830

      SUBROUTINE ERRMSG(MSG,M5,I6,I9)                                   00020840
C--ERROR MESSAGE WRITE ROUTINE AND STOP, WHERE--                        00020850
C                                                                       00020860
C       MSG=    ANY MULTIPLE OF 5 CHARACTERS--MAX. OF 120               00020870
C               (USE NH---------- FORM FOR ANSI COMPATIBILITY)          00020880
C       M5=     NO.CHARS IN MSG/5 (REMAINDER MUST BE 0) 1.LE.M5.LE.24   00020890
C       I6=     1ST UNIT  FOR WRITE(I6, ) MSG -- USUALLY I6=6 FOR LPT.  00020900
C               IF I6.LE.0 UNIT I6 IGNORED.                             00020910
C       I9=     2ND UNIT  FOR WRITE(I9, ) MSG --                        00020920
C               IF I9.LE.0, UNIT I9 IGNORED.                            00020930
C--MESSAGE WRITTEN IN FORM--                                            00020940
C       /ERROR--MSG HERE                                                00020950
C                                                                       00020960
      DIMENSION MSG(30)                                                 00020970
      J=5*M5                                                            00020980
      K=J/4+MOD(J,4)                                                    00020990
      IF(I6.GT.0) WRITE(I6,10) (MSG(I),I=1,K)                           00021000
```

```
   10 FORMAT(/8H ERROR--,30A4)                                          00021010
      IF(I9.GT.0) WRITE(I9,10) (MSG(I),I=1,K)                           00021020
      CALL CLOSE_FILE('-ALL')                                           00021030
C                                                                       00021040
      STOP                                                              00021050
      END                                                               00021060

      COMPLEX FUNCTION ZHANKS(N,B,FUN,TOL,NF,NEW)                       00021070
C=======================================================================00021080
C  COMPLEX HANKEL TRANSFORMS OF ORDER 0 OR 1 FOR RELATED (SAVED) KERNELS00021090
C  AND FIXED TRANSFORM ARGUMENT B.GT.0.                                 00021100
C                                                                       00021110
C--REF: ANDERSON, W.L., 1979, GEOPHYSICS, VOL. 44, NO. 7, P. 1287-1305. 00021120
C                                                                       00021130
C--SUBPROGRAM ZHANKS EVALUATES THE INTEGRAL FROM 0 TO INFINITY OF       00021140
C  FUN(G)*JN(G*B)*DG, DEFINED AS THE COMPLEX HANKEL TRANSFORM OF        00021150
C  ORDER N (=0 OR 1) AND TRANSFORM ARGUMENT B.GT.0.  THE METHOD IS BY   00021160
C  ADAPTIVE DIGITAL FILTERING OF THE COMPLEX KERNEL FUNCTION FUN,       00021170
C  USING DIRECT AND/OR PREVIOUSLY SAVED KERNEL FUNCTION VALUES.         00021180
C                                                                       00021190
C--PARAMETERS (ALL INPUT, EXCEPT NF)                                    00021200
C                                                                       00021210
C     N      = ORDER (=0 OR 1) OF THE HANKEL TRANSFORM TO BE EVALUATED. 00021220
C     B      = REAL TRANSFORM ARGUMENT B.GT.0.0 OF THE HANKEL TRANSFORM. 00021230
C              IF NEW=0, B IS ASSUMED EQUAL TO THE LAST B USED WHEN NEW=100021240
C              (SEE PARAMETER NEW AND SUBPROGRAM USAGE BELOW).           00021250
C     FUN(G)= EXTERNAL DECLARED COMPLEX FUNCTION NAME (USER SUPPLIED)   00021260
C              OF A REAL ARGUMENT G.GT.0. THIS REFERENCE MUST BE SUPPLIED00021270
C              EVEN WHEN NEW=0, SINCE THE ADAPTIVE CONVOLUTION           00021280
C              MAY NEED SOME DIRECT FUNCTION CALLS (E.G. IF TOL REDUCED).00021290
C              IF PARAMETERS OTHER THAN G ARE REQUIRED IN FUN, USE COMMON00021300
C              IN THE CALLING PROGRAM AND IN SUBPROGRAM FUN.  BOTH       00021310
C              REAL AND IMAGINARY PARTS OF THE COMPLEX FUNCTION FUN(G)   00021320
C              MUST BE CONTINUOUS BOUNDED FUNCTIONS FOR G.GT.0.0. FOR A  00021330
C              REAL FUNCTION F1(G), FUN=CMPLX(F1(G),0.0) MAY BE USED.    00021340
C              TWO INDEPENDENT REAL-FUNCTIONS F1(G),F2(G) MAY BE         00021350
C              INTEGRATED IN PARALLEL BY WRITING FUN=CMPLX(F1(G),F2(G)). 00021360
C     TOL    = REQUESTED REAL TRUNCATION TOLERANCE ACCEPTED AT THE FILTER00021370
C              TAILS FOR ADAPTIVE FILTERING.  A TRUNCATION CRITERION IS  00021380
C              DEFINED DURING CONVOLUTION IN A FIXED ABSCISSA RANGE AS   00021390
C              THE MAX. ABSOLUTE CONVOLVED PRODUCT TIMES TOL.  TYPICALLY,00021400
C              TOL.LE.0.00001 WOULD GIVE ABOUT .01 PER CENT ACCURACY     00021410
C              FOR WELL-BEHAVED KERNELS AND MODERATE VALUES OF B.  FOR   00021420
C              VERY LARGE OR SMALL B, A VERY SMALL TOL SHOULD BE USED.   00021430
C              IN GENERAL, DECREASING THE TOLERANCE WOULD PRODUCE HIGHER 00021440
C              ACCURACY IN THE CONVOLUTION SINCE MORE FILTER WEIGHTS ARE 00021450
C              USED (UNLESS EXPONENT UNDERFLOWS OCCUR IN THE KERNEL      00021460
C              EVALUATION -- SEE NOTE (1) BELOW).                        00021470
C              FOR MAXIMUM ACCURACY POSSIBLE, TOL=0.0 MAY BE USED.       00021480
C     NF     = TOTAL NUMBER OF DIRECT FUN CALLS USED DURING CONVOLUTION  00021490
C              FOR ANY VALUE OF NEW (NF IS AN OUTPUT PARAMETER).         00021500
C              NF IS IN THE RANGE 21.LE.NF.LE.283 WHEN NEW=1.  USUALLY,  00021510
```

```
C                NF IS MUCH LESS THAN 283 (OR 0) WHEN NEW=0.                    00021520
C      NEW  =1 IS REQUIRED FOR THE VERY FIRST CALL TO ZHANKS, OR IF             00021530
C                FORCING DIRECT FUNCTION FUN(G) CALLS, E.G., IF USING           00021540
C                ZHANKS FOR UNRELATED KERNELS.                                  00021550
C                NEW=1 INITIALIZES COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE      00021560
C                FOR NSAVE COMPLEX KERNEL VALUES IN FSAVE AND CORRESPONDING     00021570
C                REAL ARGUMENTS IN GSAVE FOR THE GIVEN PARAMETER B.             00021580
C      NEW  =0 TO USE RELATED KERNELS (MODIFIED BY USER) CURRENTLY STORED       00021590
C                IN COMMON/SAVE/. FUN IS CALLED ONLY IF REQUIRED                00021600
C                DURING THE CONVOLUTION.  ADDITIONAL FUNCTION VALUES WHEN       00021610
C                NEEDED ARE AUTOMATICALLY ADDED TO THE COMMON/SAVE/ BLOCK.      00021620
C                                                                               00021630
C      ******* NOTE THAT IT IS THE USERS RESPONSIBILITY TO MODIFY THE           00021640
C                COMMON FSAVE() VALUES FOR NEW=0 CALLS, EXTERNALLY IN           00021650
C                THE USERS CALLING PROGRAM (SEE SUBPROGRAM USAGE BELOW).        00021660
C                                                                               00021670
C===============================================================================00021680
C--SUBPROGRAM USAGE-- ZHANKS IS CALLED AS FOLLOWS                               00021690
C      ...                                                                      00021700
C      COMPLEX Z1,Z2,ZHANKS,FSAVE                                               00021710
C      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                                  00021720
C      EXTERNAL ZF1,ZF2                                                         00021730
C      ...                                                                      00021740
C      Z1=ZHANKS(N1,B,ZF1,TOL,NF1,1)                                            00021750
C      DO 1 I=1,NSAVE                                                           00021760
C   C--MODIFY FSAVE IN COMMON/SAVE/ TO OBTAIN RELATED ZF2 FROM ZF1.             00021770
C   C--E.G. FSAVE(I)=GSAVE(I)*FSAVE(I) -- FOR RELATION ZF2(G)=G*ZF1(G)          00021780
C    1 CONTINUE                                                                 00021790
C      Z2=ZHANKS(N2,B,ZF2,TOL,NF2,0)                                            00021800
C      ...                                                                      00021810
C      END                                                                      00021820
C      COMPLEX FUNCTION ZF1(G)                                                  00021830
C      ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF1(G), G.GT.0.           00021840
C      END                                                                      00021850
C      COMPLEX FUNCTION ZF2(G)                                                  00021860
C      ...USER SUPPLIED CODE FOR DIRECT EVALUATION OF ZF2(G), G.GT.0.           00021870
C      END                                                                      00021880
C===============================================================================00021890
C--NOTES                                                                        00021900
C        (1).  EXP-UNDERFLOW MAY OCCUR IN EXECUTING THIS SUBPROGRAM.            00021910
C              THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS        00021920
C              EXP-UNDERFLOW TO 0.0.                                            00021930
C        (2).  ANSI FORTRAN (AMERICAN STANDARD X3.9-1966) IS USED, EXCEPT       00021940
C              DATA STATEMENTS MAY NEED TO BE CHANGED FOR SOME COMPILERS.       00021950
C              TO CONVERT ZHANKS TO THE NEW AMERICAN STANDARD FORTRAN           00021960
C              (X3.9-1978), ADD THE FOLLOWING DECLARATION TO THIS ROUTINE       00021970
C              SAVE Y1,ISAVE                                                    00021980
C        (3).  THE FILTER ABSCISSA CORRESPONDING TO EACH FILTER WEIGHT          00021990
C              IS GENERATED IN DOUBLE-PRECISION (TO REDUCE ROUND-OFF),          00022000
C              BUT IS USED IN SINGLE-PRECISION IN FUNCTION FUN.                 00022010
C        (4).  NO CHECKS ARE MADE ON CALLING PARAMETERS (TO SAVE TIME),         00022020
C              HENCE UNPREDICTABLE RESULTS COULD OCCUR IF ZHANKS                00022030
```

```
C              IS CALLED INCORRECTLY (OR IF FUN OR COMMON IS IN ERROR).  00022040
C=======================================================================00022050
C                                                                        00022060
      COMPLEX FUN,C,CMAX,FSAVE                                           00022070
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                            00022080
      DOUBLE PRECISION E,ER,Y1,Y                                         00022090
      DIMENSION T(2),TMAX(2)                                             00022100
      DIMENSION WT0(283),WA0(76),WB0(76),WC0(76),WD0(55),                00022110
     * WT1(283),WA1(76),WB1(76),WC1(76),WD1(55)                         00022120
      EQUIVALENCE (WT0(1),WA0(1)),(WT0(77),WB0(1)),(WT0(153),WC0(1)),    00022130
     * (WT0(229),WD0(1)),(WT1(1),WA1(1)),(WT1(77),WB1(1)),              00022140
     * (WT1(153),WC1(1)),(WT1(229),WD1(1))                              00022150
      EQUIVALENCE (C,T(1)),(CMAX,TMAX(1))                                00022160
C-----E=DEXP(.2D0), ER=1.0D0/E                                           00022170
      DATA E/1.2214027581601698340 D0/,ER/.81873075307798185 9 D0/       00022180
C--JO-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT0 ARRAY)           00022190
      DATA WA0/                                                          00022200
     * 2.1969101E-11, 4.1201161E-09,-6.1322980E-09, 7.2479291E-09,       00022210
     *-7.9821627E-09, 8.5778983E-09,-9.1157294E-09, 9.6615250E-09,       00022220
     *-1.0207546E-08, 1.0796633E-08,-1.1393033E-08, 1.2049873E-08,       00022230
     *-1.2708789E-08, 1.3446466E-08,-1.4174300E-08, 1.5005577E-08,       00022240
     *-1.5807160E-08, 1.6747136E-08,-1.7625961E-08, 1.8693427E-08,       00022250
     *-1.9650840E-08, 2.0869789E-08,-2.1903555E-08, 2.3305308E-08,       00022260
     *-2.4407377E-08, 2.6033678E-08,-2.7186773E-08, 2.9094334E-08,       00022270
     *-3.0266804E-08, 3.2534013E-08,-3.3672072E-08, 3.6408936E-08,       00022280
     *-3.7425022E-08, 4.0787921E-08,-4.1543242E-08, 4.5756842E-08,       00022290
     *-4.6035233E-08, 5.1425075E-08,-5.0893896E-08, 5.7934897E-08,       00022300
     *-5.6086570E-08, 6.5475248E-08,-6.1539913E-08, 7.4301996E-08,       00022310
     *-6.7117043E-08, 8.4767837E-08,-7.2583120E-08, 9.7366568E-08,       00022320
     *-7.7553611E-08, 1.1279873E-07,-8.1416723E-08, 1.3206914E-07,       00022330
     *-8.3217217E-08, 1.5663185E-07,-8.1482581E-08, 1.8860593E-07,       00022340
     *-7.3963141E-08, 2.3109673E-07,-5.7243707E-08, 2.8867452E-07,       00022350
     *-2.6163525E-08, 3.6808773E-07, 2.7049871E-08, 4.7932617E-07,       00022360
     * 1.1407365E-07, 6.3720626E-07, 2.5241961E-07, 8.6373487E-07,       00022370
     * 4.6831433E-07, 1.1916346E-06, 8.0099716E-07, 1.6696015E-06,       00022380
     * 1.3091334E-06, 2.3701475E-06, 2.0803829E-06, 3.4012978E-06/       00022390
      DATA WB0/                                                          00022400
     * 3.2456774E-06, 4.9240402E-06, 5.0005198E-06, 7.1783540E-06,       00022410
     * 7.6367633E-06, 1.0522038E-05, 1.1590021E-05, 1.5488635E-05,       00022420
     * 1.7510398E-05, 2.2873836E-05, 2.6368006E-05, 3.3864387E-05,       00022430
     * 3.9610390E-05, 5.0230379E-05, 5.9397373E-05, 7.4612122E-05,       00022440
     * 8.8951409E-05, 1.1094809E-04, 1.3308026E-04, 1.6511335E-04,       00022450
     * 1.9895671E-04, 2.4587195E-04, 2.9728181E-04, 3.6629770E-04,       00022460
     * 4.4402013E-04, 5.4589361E-04, 6.6298832E-04, 8.1375348E-04,       00022470
     * 9.8971624E-04, 1.2132772E-03, 1.4772052E-03, 1.8092022E-03,       00022480
     * 2.2045122E-03, 2.6980811E-03, 3.2895354E-03, 4.0238764E-03,       00022490
     * 4.9080203E-03, 6.0010999E-03, 7.3216878E-03, 8.9489225E-03,       00022500
     * 1.0919448E-02, 1.3340696E-02, 1.6276399E-02, 1.9873311E-02,       00022510
     * 2.4233627E-02, 2.9555699E-02, 3.5990069E-02, 4.3791529E-02,       00022520
     * 5.3150319E-02, 6.4341372E-02, 7.7506720E-02, 9.2749987E-02,       00022530
     * 1.0980561E-01, 1.2791555E-01, 1.4525830E-01, 1.5820085E-01,       00022540
     * 1.6058576E-01, 1.4196085E-01, 8.9781222E-02,-1.0238278E-02,       00022550
```

```
     *-1.5083434E-01,-2.9059573E-01,-2.9105437E-01,-3.7973244E-02,      00022560
     * 3.8273717E-01, 2.2014118E-01,-4.7342635E-01, 1.9331133E-01,      00022570
     * 5.3839527E-02,-1.1909845E-01, 9.9317051E-02,-6.6152628E-02,      00022580
     * 4.0703241E-02,-2.4358316E-02, 1.4476533E-02,-8.6198067E-03/      00022590
      DATA WC0/                                                        00022600
     * 5.1597053E-03,-3.1074602E-03, 1.8822342E-03,-1.1456545E-03,      00022610
     * 7.0004347E-04,-4.2904226E-04, 2.6354444E-04,-1.6215439E-04,      00022620
     * 9.9891279E-05,-6.1589037E-05, 3.7996921E-05,-2.3452250E-05,      00022630
     * 1.4479572E-05,-8.9417427E-06, 5.5227518E-06,-3.4114252E-06,      00022640
     * 2.1074101E-06,-1.3019229E-06, 8.0433617E-07,-4.9693681E-07,      00022650
     * 3.0702417E-07,-1.8969219E-07, 1.1720069E-07,-7.2412496E-08,      00022660
     * 4.4740283E-08,-2.7643004E-08, 1.7079403E-08,-1.0552634E-08,      00022670
     * 6.5200311E-09,-4.0284597E-09, 2.4890232E-09,-1.5378695E-09,      00022680
     * 9.5019040E-10,-5.8708696E-10, 3.6273937E-10,-2.2412348E-10,      00022690
     * 1.3847792E-10,-8.5560821E-11, 5.2865474E-11,-3.2664392E-11,      00022700
     * 2.0182948E-11,-1.2470979E-11, 7.7057678E-12,-4.7611713E-12,      00022710
     * 2.9415274E-12,-1.8170081E-12, 1.1221034E-12,-6.9271067E-13,      00022720
     * 4.2739744E-13,-2.6344388E-13, 1.6197105E-13,-9.9147443E-14,      00022730
     * 6.0487998E-14,-3.6973097E-14, 2.2817964E-14,-1.4315547E-14,      00022740
     * 9.1574735E-15,-5.9567236E-15, 3.9209969E-15,-2.5911739E-15,      00022750
     * 1.6406939E-15,-8.8248590E-16, 3.0195409E-16, 2.2622634E-17,      00022760
     *-8.0942556E-17,-3.7172363E-17, 1.9299542E-16,-3.3388160E-16,      00022770
     * 4.6174116E-16,-5.8627358E-16, 7.2227767E-16,-8.7972941E-16,      00022780
     * 1.0211793E-15,-1.0940039E-15, 1.0789555E-15,-9.7089714E-16/      00022790
      DATA WD0/                                                        00022800
     * 7.4110927E-16,-4.1700094E-16, 8.5977184E-17, 1.3396469E-16,      00022810
     *-1.7838410E-16, 4.8975421E-17, 1.9398153E-16,-5.0046989E-16,      00022820
     * 8.3280985E-16,-1.1544640E-15, 1.4401527E-15,-1.6637066E-15,      00022830
     * 1.7777129E-15,-1.7322187E-15, 1.5247247E-15,-1.1771155E-15,      00022840
     * 6.9747910E-16,-1.2088956E-16,-4.8382957E-16, 1.0408292E-15,      00022850
     *-1.5220450E-15, 1.9541597E-15,-2.4107448E-15, 2.9241438E-15,      00022860
     *-3.5176475E-15, 4.2276125E-15,-5.0977851E-15, 6.1428456E-15,      00022870
     *-7.3949962E-15, 8.8597601E-15,-1.0515959E-14, 1.2264584E-14,      00022880
     *-1.3949870E-14, 1.5332490E-14,-1.6146782E-14, 1.6084121E-14,      00022890
     *-1.4962523E-14, 1.2794804E-14,-9.9286701E-15, 6.8825809E-15,      00022900
     *-4.0056107E-15, 1.5965079E-15,-7.2732961E-18,-4.0433218E-16,      00022910
     *-6.5679655E-16, 3.3011866E-15,-7.3545910E-15, 1.2394851E-14,      00022920
     *-1.7947697E-14, 2.3774303E-14,-3.0279168E-14, 3.9252831E-14,      00022930
     *-5.5510504E-14, 9.0505371E-14,-1.7064873E-13/                    00022940
C--END OF JO FILTER WEIGHTS                                           00022950
C                                                                     00022960
C--J1-TRANSFORM FILTER WEIGHT ARRAYS (EQUIVALENT TO WT1 ARRAY)        00022970
      DATA WA1/                                                       00022980
     *-4.2129715E-16, 5.3667031E-15,-7.1183962E-15, 8.9478500E-15,      00022990
     *-1.0767891E-14, 1.2362265E-14,-1.3371129E-14, 1.3284178E-14,      00023000
     *-1.1714302E-14, 8.4134738E-15,-3.7726725E-15,-1.4263879E-15,      00023010
     * 6.1279163E-15,-9.1102765E-15, 9.9696405E-15,-9.3649955E-15,      00023020
     * 8.6009018E-15,-8.9749846E-15, 1.1153987E-14,-1.4914821E-14,      00023030
     * 1.9314024E-14,-2.3172388E-14, 2.5605477E-14,-2.6217555E-14,      00023040
     * 2.5057768E-14,-2.2485539E-14, 1.9022752E-14,-1.5198084E-14,      00023050
     * 1.1422464E-14,-7.9323958E-15, 4.8421406E-15,-2.1875032E-15,      00023060
     *-3.2177842E-17, 1.8637565E-15,-3.3683643E-15, 4.6132219E-15,      00023070
```

```
*-5.6209538E-15, 6.4192841E-15,-6.8959928E-15, 6.9895792E-15,      00023080
*-6.5355935E-15, 5.6125163E-15,-4.1453931E-15, 2.6358827E-15,      00023090
*-9.5104370E-16, 1.4600474E-16, 5.6166519E-16, 8.2899246E-17,      00023100
* 5.0032100E-16, 4.3752205E-16, 2.1052293E-15,-9.5451973E-16,      00023110
* 6.4004437E-15,-2.1926177E-15, 1.1651003E-14, 5.8415433E-16,      00023120
* 1.8044664E-14, 1.0755745E-14, 3.0159022E-14, 3.3506138E-14,      00023130
* 5.8709354E-14, 8.1475200E-14, 1.2530006E-13, 1.8519112E-13,      00023140
* 2.7641786E-13, 4.1330823E-13, 6.1506209E-13, 9.1921659E-13,      00023150
* 1.3698462E-12, 2.0447427E-12, 3.0494477E-12, 4.5501001E-12,      00023160
* 6.7870250E-12, 1.0126237E-11, 1.5104976E-11, 2.2536053E-11/      00023170
 DATA WB1/                                                        00023180
* 3.3617368E-11, 5.0153839E-11, 7.4818173E-11, 1.1161804E-10,      00023190
* 1.6651222E-10, 2.4840923E-10, 3.7058109E-10, 5.5284353E-10,      00023200
* 8.2474468E-10, 1.2303750E-09, 1.8355034E-09, 2.7382502E-09,      00023210
* 4.0849867E-09, 6.0940898E-09, 9.0913020E-09, 1.3562651E-08,      00023220
* 2.0233058E-08, 3.0184244E-08, 4.5029477E-08, 6.7176304E-08,      00023230
* 1.0021488E-07, 1.4950371E-07, 2.2303208E-07, 3.3272689E-07,      00023240
* 4.9636623E-07, 7.4049804E-07, 1.1046805E-06, 1.6480103E-06,      00023250
* 2.4585014E-06, 3.6677163E-06, 5.4714550E-06, 8.1626422E-06,      00023260
* 1.2176782E-05, 1.8166179E-05, 2.7099223E-05, 4.0428804E-05,      00023270
* 6.0307294E-05, 8.9971508E-05, 1.3420195E-04, 2.0021123E-04,      00023280
* 2.9860417E-04, 4.4545291E-04, 6.6423156E-04, 9.9073275E-04,      00023290
* 1.4767050E-03, 2.2016806E-03, 3.2788147E-03, 4.8837292E-03,      00023300
* 7.2596811E-03, 1.0788355E-02, 1.5973323E-02, 2.3612041E-02,      00023310
* 3.4655327E-02, 5.0608141E-02, 7.2827752E-02, 1.0337889E-01,      00023320
* 1.4207357E-01, 1.8821315E-01, 2.2996815E-01, 2.5088500E-01,      00023330
* 2.0334626E-01, 6.0665451E-02,-2.0275683E-01,-3.5772336E-01,      00023340
*-1.8280529E-01, 4.7014634E-01, 7.2991233E-03,-3.0614594E-01,      00023350
* 2.4781735E-01,-1.1149185E-01, 2.5985386E-02, 1.0850279E-02,      00023360
*-2.2830217E-02, 2.4644647E-02,-2.2895284E-02, 2.0197032E-02/      00023370
 DATA WC1/                                                        00023380
*-1.7488968E-02, 1.5057670E-02,-1.2953923E-02, 1.1153254E-02,      00023390
*-9.6138436E-03, 8.2952090E-03,-7.1628361E-03, 6.1882910E-03,      00023400
*-5.3482055E-03, 4.6232056E-03,-3.9970542E-03, 3.4560118E-03,      00023410
*-2.9883670E-03, 2.5840861E-03,-2.2345428E-03, 1.9323046E-03,      00023420
*-1.6709583E-03, 1.4449655E-03,-1.2495408E-03, 1.0805480E-03,      00023430
*-9.3441130E-04, 8.0803899E-04,-6.9875784E-04, 6.0425624E-04,      00023440
*-5.2253532E-04, 4.5186652E-04,-3.9075515E-04, 3.3790861E-04,      00023450
*-2.9220916E-04, 2.5269019E-04,-2.1851585E-04, 1.8896332E-04,      00023460
*-1.6340753E-04, 1.4130796E-04,-1.2219719E-04, 1.0567099E-04,      00023470
*-9.1379828E-05, 7.9021432E-05,-6.8334412E-05, 5.9092726E-05,      00023480
*-5.1100905E-05, 4.4189914E-05,-3.8213580E-05, 3.3045496E-05,      00023490
*-2.8576356E-05, 2.4711631E-05,-2.1369580E-05, 1.8479514E-05,      00023500
*-1.5980307E-05, 1.3819097E-05,-1.1950174E-05, 1.0334008E-05,      00023510
*-8.9364160E-06, 7.7278366E-06,-6.6827083E-06, 5.7789251E-06,      00023520
*-4.9973715E-06, 4.3215167E-06,-3.7370660E-06, 3.2316575E-06,      00023530
*-2.7946015E-06, 2.4166539E-06,-2.0898207E-06, 1.8071890E-06,      00023540
*-1.5627811E-06, 1.3514274E-06,-1.1686576E-06, 1.0106059E-06,      00023550
*-8.7392952E-07, 7.5573750E-07,-6.5353002E-07, 5.6514528E-07,      00023560
*-4.8871388E-07, 4.2261921E-07,-3.6546333E-07, 3.1603732E-07/      00023570
 DATA WD1/                                                        00023580
*-2.7329579E-07, 2.3633470E-07,-2.0437231E-07, 1.7673258E-07,      00023590
```

```
     *-1.5283091E-07, 1.3216174E-07,-1.1428792E-07, 9.8831386E-08,      00023600
     *-8.5465227E-08, 7.3906734E-08,-6.3911437E-08, 5.5267923E-08,      00023610
     *-4.7793376E-08, 4.1329702E-08,-3.5740189E-08, 3.0906612E-08,      00023620
     *-2.6726739E-08, 2.3112160E-08,-1.9986424E-08, 1.7283419E-08,      00023630
     *-1.4945974E-08, 1.2924650E-08,-1.1176694E-08, 9.6651347E-09,      00023640
     *-8.3580023E-09, 7.2276490E-09,-6.2501673E-09, 5.4048822E-09,      00023650
     *-4.6739154E-09, 4.0418061E-09,-3.4951847E-09, 3.0224895E-09,      00023660
     *-2.6137226E-09, 2.2602382E-09,-1.9545596E-09, 1.6902214E-09,      00023670
     *-1.4616324E-09, 1.2639577E-09,-1.0930164E-09, 9.4519327E-10,      00023680
     *-8.1736202E-10, 7.0681930E-10,-6.1122713E-10, 5.2856342E-10,      00023690
     *-4.5707937E-10, 3.9526267E-10,-3.4180569E-10, 2.9557785E-10,      00023700
     *-2.5560176E-10, 2.2103233E-10,-1.9113891E-10, 1.6528994E-10,      00023710
     *-1.4294012E-10, 1.2361991E-10,-8.2740936E-11/                     00023720
C--END OF J1 FILTER WEIGHTS                                             00023730
C                                                                       00023740
      NONE=0                                                            00023750
      IF(NEW.EQ.0) GO TO 100                                            00023760
      NSAVE=0                                                           00023770
C-----INITIALIZE KERNEL ABSCISSA GENERATION FOR GIVEN B                 00023780
      Y1=0.73588526614797944600D0/DBLE(B)                               00023790
  100 ZHANKS=(0.0,0.0)                                                  00023800
      CMAX=(0.0,0.0)                                                    00023810
      NF=0                                                              00023820
      Y=Y1                                                              00023830
C-----BEGIN RIGHT-SIDE CONVOLUTION AT WEIGHT 131 (EITHER NEW=1 OR 0)    00023840
      ASSIGN 110 TO M                                                   00023850
      I=131                                                             00023860
      Y=Y*E                                                             00023870
      GO TO 200                                                         00023880
  110 TMAX(1)=AMAX1(ABS(T(1)),TMAX(1))                                  00023890
      TMAX(2)=AMAX1(ABS(T(2)),TMAX(2))                                  00023900
      I=I+1                                                             00023910
      Y=Y*E                                                             00023920
      IF(I.LE.149) GO TO 200                                            00023930
      IF(TMAX(1).EQ.0.0.AND.TMAX(2).EQ.0.0) NONE=1                      00023940
C-----ESTABLISH TRUNCATION CRITERION (CMAX=CMPLX(TMAX(1),TMAX(2))       00023950
      CMAX=TOL*CMAX                                                     00023960
      ASSIGN 120 TO M                                                   00023970
      GO TO 200                                                         00023980
C-----CHECK FOR FILTER TRUNCATION AT RIGHT END                          00023990
  120 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2)) GO TO 130       00024000
      I=I+1                                                             00024010
      Y=Y*E                                                             00024020
      IF(I.LE.283) GO TO 200                                            00024030
  130 Y=Y1                                                              00024040
C-----CONTINUE WITH LEFT-SIDE CONVOLUTION AT WEIGHT 130                 00024050
      ASSIGN 140 TO M                                                   00024060
      I=130                                                             00024070
      GO TO 200                                                         00024080
C-----CHECK FOR FILTER TRUNCATION AT LEFT END                           00024090
  140 IF(ABS(T(1)).LE.TMAX(1).AND.ABS(T(2)).LE.TMAX(2).AND.             00024100
     * NONE.EQ.0) GO TO 190                                            00024110
```

```
      I=I-1                                                              00024120
      Y=Y*ER                                                             00024130
      IF(I.GT.0) GO TO 200                                               00024140
C-----RETURN WITH ISAVE=1 PRESET FOR POSSIBLE NEW=0 USE.                 00024150
  190 ISAVE=1                                                            00024160
C-----NORMALIZE BY B TO ACCOUNT FOR INTEGRATION RANGE CHANGE             00024170
      ZHANKS=ZHANKS/B                                                    00024180
      RETURN                                                             00024190
C-----SAVE/RETRIEVE PSEUDO-SUBROUTINE (CALL FUN ONLY WHEN NECESSARY)     00024200
  200 G=SNGL(Y)                                                          00024210
      IF(NEW) 300,210,300                                                00024220
  210 IF(ISAVE.GT.NSAVE) GO TO 300                                       00024230
      ISAVE0=ISAVE                                                       00024240
  220 IF(G.EQ.GSAVE(ISAVE)) GO TO 240                                    00024250
      ISAVE=ISAVE+1                                                      00024260
      IF(ISAVE.LE.NSAVE) GO TO 220                                       00024270
      ISAVE=ISAVE0                                                       00024280
C-----G NOT IN COMMON/SAVE/----- EVALUATE FUN.                           00024290
      GO TO 300                                                          00024300
C-----G FOUND IN COMMON/SAVE/----- USE FSAVE AS GIVEN.                   00024310
  240 C=FSAVE(ISAVE)                                                     00024320
      ISAVE=ISAVE+1                                                      00024330
C-----SWITCH ON ORDER N                                                  00024340
  250 IF(N) 270,260,270                                                  00024350
  260 C=C*WT0(I)                                                         00024360
      GO TO 280                                                          00024370
  270 C=C*WT1(I)                                                         00024380
  280 ZHANKS=ZHANKS+C                                                    00024390
      GO TO M,(110,120,140)                                              00024400
C-----DIRECT FUN EVALUATION (AND ADD TO END OF COMMON/SAVE/)             00024410
  300 NSAVE=NSAVE+1                                                      00024420
      C=FUN(G)                                                           00024430
      NF=NF+1                                                            00024440
      FSAVE(NSAVE)=C                                                     00024450
      GSAVE(NSAVE)=G                                                     00024460
      GO TO 250                                                          00024470
      END                                                                00024480

      SUBROUTINE SAVER(I,J)                                              00024490
C====================================================================== 00024500
C--GENERAL UTILITY TO MODIFY COMMON/SAVE/ AS FOLLOWS                     00024510
C  CALL SAVER(I,J) WILL REPLACE FSAVE() ARRAY WITH                       00024520
C  FSAVE(K)=GSAVE(K)**I * FSAVE(K)**J                                    00024530
C  FOR K=1,NSAVE.                                                        00024540
C                                                                        00024550
C  INPUT PARAMETERS (I,J) MAY BE NEGATIVE,ZERO, OR POSITIVE INTEGERS.    00024560
C                                                                        00024570
C--SUBROUTINE SAVER MAY BE USED IN CONJUNCTION WITH SUBPROGRAM ZHANKS    00024580
C  TO MODIFY SAVED KERNELS WHEN USING NEW=0 OPTION (SEE ZHANKS).         00024590
C                                                                        00024600
      COMPLEX FSAVE                                                      00024610
      COMMON/SAVE/FSAVE(283),GSAVE(283),NSAVE                            00024620
```

```
      DO 1 K=1,NSAVE                                                    00024630
      FSAVE(K)=CMPLX(GSAVE(K)**I,0.0)*(FSAVE(K)**J)                     00024640
1     CONTINUE                                                          00024650
      RETURN                                                            00024660
      END                                                               00024670

      SUBROUTINE IMSLMQ(SUBZ,SUBEND)                                    00024680
C--IMSLMQ-- DERIVATIVE-FREE 'IMSL' MARQUARDT INVERSION--5/4/79          00024690
C  FOR SOLVING GENERAL NONLINEAR LEAST SQUARES PROBLEMS. USER NEED ONLY 00024700
C  WRITE SUBROUTINES 'FCODE, SUBZ, AND SUBEND' EXACTLY AS USED          00024710
C  IN PROGRAM 'MARQRT'.  ALSO, THE SAME PARAMETER FILE05 AND DATA       00024720
C  FILE10 MAY BE USED BY 'IMSLMQ' AS IN 'MARQRT'.                       00024730
C                                                                       00024740
C--NOTE: 'FCODE' CANNOT BE PASSED AS EXTERNAL DUE TO THE                00024750
C  'BLACK-BOX' NATURE OF IMSL ROUTINE 'ZXSSQ' (SEE IMSL DOC.).          00024760
C  THUS, ONE SHOULD RENAME ACTUAL NAME TO 'FCODE' FOR USE HERE.         00024770
C  (I.E., SEE CALL FCODE IN 'FPXSSQ'--EXTERNAL FUNCTION FOR ZXSSQ).     00024780
C                                                                       00024790
C--THE USER MUST DECLARE THE CALLING PARAMETERS                         00024800
C  SUBZ,SUBEND (ANY DESIRED NAMES MAY BE USED) AS EXTERNAL IN           00024810
C  MAIN CALLING PROGRAM; E.G.,                                          00024820
C                                                                       00024830
C       EXTERNAL SUBZ,SUBEND                                            00024840
C       CALL IMSLMQ(SUBZ,SUBEND)                                        00024850
C       STOP                                                            00024860
C       END                                                             00024870
C                                                                       00024880
C--THIS INTERFACE BETWEEN 'MARQRT' AND 'IMSLMQ' WAS WRITTEN BY          00024890
C  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.              00024900
C                                                                       00024910
C--SEE DOCUMENTATION OF 'MARQLOOPS', USGS OPEN-FILE REPT 79-240 (1979), 00024920
C  FOR DETAILS ON CODING THE REQUIRED SUBROUTINES FCODE,SUBZ, AND       00024930
C  SUBEND. ALSO SEE IMSL DOCUMENTATION FOR 'ZXSSQ'.                     00024940
C                                                                       00024950
C--THE INPUT ORDER ON FILE05 (PARAMETER FILE) IS:                       00024960
C                                                                       00024970
C  1.  TITLE  (MAX. 80-CHARACTER TITLE--ALWAYS REQUIRED).               00024980
C  2.  $PARMS (SAME PARMS DEFINITIONS FOR PGM MARQRT FOR PARAMETERS:    00024990
C         N,K,IP,M,IALT,ISTOP,IWT,NITER,E,SCALEP,B(),IPRT, AND IB()).   00025000
C         PLUS, ADDITIONAL PARMS FOR 'ZXSSQ' (SEE IMSL DOC):            00025010
C         IOPT,NSIG,MAXFN,EPS,DELTA,PARM(4).                            00025020
C  3.  (OBJECT-TIME FORMAT STATEMENT) FOR READING THE DATA MATRIX       00025030
C       (Y(I),X(I,J),J=1,M*) ON FILE IALT (DEFAULT 10), WHERE M*=M+IWT. 00025040
C  (3A.  INSERT DATA MATRIX HERE ONLY IF IALT=5)                        00025050
C  4.  $INIT OPTIONAL NAMELIST FOR READING ADDITIONAL PARMS IN          00025060
C           SUBROUTINE SUBZ  (WHICH MAY BE A DUMMY SUBROUTINE).         00025070
C  5.  OPTIONALLY, REPEAT STEPS 1-4, IF ISTOP=0 WAS USED IN STEP 2.     00025080
C                                                                       00025090
C--OUTPUT IS GIVEN ON FILE06 (ON-LINE USUALLY), AND                     00025100
C  ON FILE16 (CONTAINS ALL PRINTABLE OUTPUT); FILE06 CONTAINS ONLY      00025110
C  OUTPUT VIA PARM IPRT (0--ABBREVIATED, 1 OR -2 --DETAIL).             00025120
C                                                                       00025130
```

```
C-------------------------------------------------------------------------- 00025140
C--THE USER SHOULD ADD >IML>IMSL TO THE SEARCH RULES ON MULTICS.           00025150
C  (IMSL ROUTINES USED: ZXSSQ,LEQT1P,LUDECP,LUELMP,UERTST,LINV1P)          00025160
C-------------------------------------------------------------------------- 00025170
C                                                                          00025180
      DIMENSION B(20),GRAD(20),TITLE(16),H(270),COV(20,20),SE(20)          00025190
      DIMENSION SQWT(200),IB(20),PRNT(5),C(20),INDEX(20),                  00025200
     & XJAC(200,20),XJTJ(210),WORK(710),PARM(4),F(200)                     00025210
      INTEGER SP,SCALEP,SY,SCALEY                                          00025220
C--THE FOLLOWING CHARACTER STATEMENTS ONLY FOR HONEYWELL MULTICS:           00025230
      CHARACTER*4 FMT(18)                                                  00025240
      CHARACTER*5 TITLE                                                    00025250
      COMMON/FIXDAT/Y(200),X(200,5),BFIX(20),YMAX,IIB(20),IIP,NOBS,K       00025260
      COMMON/PRT/IPRT                                                      00025270
      EXTERNAL FPXSSQ,LNXSSQ                                               00025280
      EQUIVALENCE (SQWT(1),X(1,5)),(SP,SCALEP),(N,NOBS),                   00025290
     & (M,NVARS),(NITER,LIMIT),(SY,SCALEY),(E,EPS),(SS,SSQ),               00025300
     & (IB(1),IIB(1)),(IP,IIP),(B(1),BFIX(1))                              00025310
      NAMELIST/PARMS/N,K,IP,M,IALT,NITER,IB,E,B,IWT,ISTOP,SP,SY,           00025320
     & SCALEP,SCALEY, IDER,IPRT,INON,FF,T,TAU,XL,MODLAM,GAMCR,             00025330
     & DEL,ZETA,IOUT,IOPT,NSIG,MAXFN,EPS,DELTA,PARM                        00025340
C-- NOTE NAMELIST PARMS INCLUDED (BUT IGNORED) FOR COMPATIBILITY           00025350
C  ARE: IDER,INON,FF,T,TAU,XL,MODLAM,GAMCR,DEL,ZETA,IOUT.                  00025360
C  ALSO, SP=SCALEP WILL BE CONSIDERED MODE=1 (ALOG OPTION) ONLY            00025370
C  WHEN SP=2 OR 1 (AND MODE=0 LINEAR WHEN SP=0).  SY=SCALEY IS             00025380
C  IGNORED FOR THIS VERSION OF 'IMSLMQ'.                                   00025390
C                                                                          00025400
C--READ IMSLMQ TITLE LINE                                                  00025410
      READ(5,4) TITLE                                                      00025420
4     FORMAT(16A5)                                                         00025430
C--PRESET DEFAULTS                                                         00025440
      N=0                                                                  00025450
      K=0                                                                  00025460
      M=0                                                                  00025470
      IP=0                                                                 00025480
      IPRT=0                                                               00025490
      ISTOP=1                                                              00025500
      IWT=0                                                                00025510
      IALT=10                                                              00025520
      NITER=10                                                             00025530
      SP=0                                                                 00025540
      MODE=0                                                               00025550
      FMIN=0.0                                                             00025560
      IOPT=1                                                               00025570
      NSIG=3                                                               00025580
      MAXFN=0                                                              00025590
      EPS=0.0                                                              00025600
      DELTA=0.0                                                            00025610
      PARM(1)=.01                                                          00025620
      PARM(2)=2.                                                           00025630
      PARM(3)=120.                                                         00025640
      PARM(4)=.1                                                           00025650
```

```
      DO 5 I=1,20                                                          00025660
      IB(I)=0                                                              00025670
      B(I)=0.0                                                             00025680
5     GRAD(I)=0.0                                                          00025690
C--READ $PARMS                                                             00025700
6     READ(5,PARMS)                                                        00025710
C--TEST $PARMS BEFORE PROCEEDING                                           00025720
      IF(N.GT.200.OR.K.GT.20.OR.M.GT.4.OR.IWT.GT.1.OR.IP.GT.19.OR.         00025730
     & N.LT.1.OR.K.LT.1.OR.M.LT.1.OR.IWT.LT.0.OR.IP.LT.0.OR.               00025740
     & N.LT.K-IP.OR.IALT.EQ.6.OR.IALT.EQ.16)                               00025750
     &CALL ERRMSG('SOME $PARMS OUT OF RANGE.',5,6,16)                      00025760
      DO 7 I=1,K                                                           00025770
      IF(B(I).EQ.0.0)CALL ERRMSG('SOME B(I)=0.0  ',3,6,16)                 00025780
7     CONTINUE                                                             00025790
      IF(MAXFN.EQ.0) MAXFN=2*K*NITER                                       00025800
      IF(IP.LE.0) GO TO 9                                                  00025810
      DO 8 I=1,IP                                                          00025820
      IF(IB(I).GT.0) GO TO 8                                               00025830
      CALL ERRMSG('IP.GT.1 BUT SOME IB(I).LE.0   ',6,6,16)                 00025840
8     CONTINUE                                                             00025850
9     IF(SP.NE.0) MODE=1                                                   00025860
      IF(IPRT.EQ.1) IPRT=-2                                                00025870
C--READ OBJECT FORMAT FOR DATA MATRIX ON FILE IALT.                        00025880
      READ(5,10) (FMT(I),I=1,18)                                           00025890
10    FORMAT(18A4)                                                         00025900
      M1=M+IWT                                                             00025910
      YMAX=0.0                                                             00025920
      DO 11 I=1,N                                                          00025930
      READ(IALT,FMT) Y(I),(X(I,J),J=1,M1)                                  00025940
      SQWT(I)=1.0                                                          00025950
      IF(IWT.EQ.1.AND.X(I,M1).NE.0.0) SQWT(I)=1.0/X(I,M1)                  00025960
      IF(ABS(Y(I)).GT.YMAX) YMAX=ABS(Y(I))                                 00025970
11    CONTINUE                                                             00025980
      IF(IALT.NE.5) REWIND IALT                                            00025990
C--INITIALIZATION VIA CALL SUBZ (READ $INIT, TEST B,X,Y, ETC)              00026000
      CALL SUBZ(Y,X,B,PRNT,NDUM,N,TITLE,1)                                 00026010
C--WRITE $PARMS ON UNIT 6 AND 16                                           00026020
      WRITE(6,60) TITLE,N,K,IP,M,E,IALT,ISTOP,IWT,NITER,SP,IPRT,           00026030
     & IOPT,NSIG,MAXFN,EPS,DELTA,PARM                                      00026040
60    FORMAT('1I M S L M Q  --      ',16A5//' N=',I5,9X,'K=',I4,10X,'IP='  00026050
     & ,I4,9X,'M=',I3,11X,'E=',E10.3/' IALT=',I3,8X,'ISTOP=',I2,8X,       00026060
     & 'IWT=',I2,10X,'NITER=',I6,4X,'SCALEP=',I2/' IPRT=',I3,             00026070
     & 8X,'IOPT=',I2,9X,'NSIG=',I3,8X,'MAXFN=',I6,4X,'EPS=',E10.3/        00026080
     & ' DELTA=',E10.3/' PARM=',4E10.3)                                    00026090
      WRITE(16,60) TITLE,N,K,IP,M,E,IALT,ISTOP,IWT,NITER,SP,IPRT,          00026100
     & IOPT,NSIG,MAXFN,EPS,DELTA,PARM                                      00026110
      IF(IP.EQ.0) GO TO 661                                                00026120
      WRITE(6,660) (IB(I),I=1,IP)                                          00026130
660   FORMAT(' IB=',19I3)                                                  00026140
      WRITE(16,660) (IB(I),I=1,IP)                                         00026150
661   WRITE(6,662) FMT                                                     00026160
662   FORMAT(' FMT=',18A4/)                                                00026170
```

```
      WRITE(16,662) FMT                                                 00026180
      WRITE(6,6000) (B(I),I=1,K)                                        00026190
6000  FORMAT(/' INITIAL PARAMETERS'//(5E16.8))                          00026200
      WRITE(16,6000) (B(I),I=1,K)                                       00026210
C--INTERFACE WITH ZXSSQ USING MARQRT FCODE                              00026220
      DO 1 I=1,20                                                       00026230
1     INDEX(I)=I                                                        00026240
      KIP=K-IP                                                          00026250
      IF(IP.EQ.0) GO TO 400                                             00026260
C--REORDER B TO C WHEN IP>0                                             00026270
      IM=0                                                              00026280
      DO 202 I=1,K                                                      00026290
      DO 201 J=1,IP                                                     00026300
      IF(I.EQ.IB(J)) GO TO 202                                          00026310
201   CONTINUE                                                          00026320
      IM=IM+1                                                           00026330
      C(IM)=B(I)                                                        00026340
      INDEX(IM)=I                                                       00026350
202   CONTINUE                                                          00026360
      WRITE(6,203) (I,I=1,K)                                            00026370
203   FORMAT(/' PARAMETER INDEX:',20I3)                                 00026380
      WRITE(16,203) (I,I=1,K)                                           00026390
      WRITE(6,204) (INDEX(I),I=1,KIP)                                   00026400
204   FORMAT(' REORDERED AS...:',20I3)                                  00026410
      WRITE(16,204) (INDEX(I),I=1,KIP)                                  00026420
      WRITE(6,206) (C(I),I=1,KIP)                                       00026430
206   FORMAT(/' REORDERED PARAMETERS'//(5E16.8))                        00026440
      WRITE(16,206) (C(I),I=1,KIP)                                      00026450
      GO TO 500                                                         00026460
400   DO 401 I=1,K                                                      00026470
401   C(I)=B(I)                                                         00026480
500   CONTINUE                                                          00026490
      IF(MODE.EQ.0) GO TO 12                                            00026500
C--LOG PARAMETERS CHOSEN (MODE=1 OR SP.NE.0)                            00026510
      DO 111 I=1,KIP                                                    00026520
      IF(C(I).LE.0.0)CALL ERRMSG('SP.NE.0 & SOME B(I).LE.0.',5,6,16)    00026530
111   C(I)=ALOG(C(I))                                                   00026540
      CALL ZXSSQ(LNXSSQ,N,KIP,NSIG,EFS,DELTA,MAXFN,IOPT,PARM,           00026550
     & C,SSQ,F,XJAC,200,XJTJ,WORK,INFER,IER)                            00026560
      DO 1111 I=1,KIP                                                  00026570
1111  C(I)=EXP(C(I))                                                    00026580
      GO TO 21                                                          00026590
C--LINEAR PARAMETERS CHOSEN (MODE=0 OR SP=0)                            00026600
12    CALL ZXSSQ(FPXSSQ,N,KIP,NSIG,EFS,DELTA,MAXFN,IOPT,PARM,           00026610
     & C,SSQ,F,XJAC,200,XJTJ,WORK,INFER,IER)                            00026620
C--ZXSSQ ERRMSG CODE HANDLERS                                           00026630
21    IF(IER.EQ.0) GO TO 100                                            00026640
      IF(IER.EQ.129)                                                    00026650
     &CALL ERRMSG('SINGULARITY DETECTED IN JACOBIAN & RECOVERY FAILED',  00026660
     & 10,6,16)                                                         00026670
      IF(IER.EQ.130)                                                    00026680
     &CALL ERRMSG('N,K-IP,IOPT,PARM(1) OR PARM(2) INCORRECT',8,6,16)    00026690
```

```
      IF(IER.EQ.131)                                                    00026700
     &CALL WARN('MARQUARDT PARAMETER EXCEEDED PARM(3)    ',8,6,16,$100) 00026710
      IF(IER.EQ.132) CALL ERRMSG(                                       00026720
     &'AFTER RECOVERY FROM SINGULAR JACOBIAN, B CYCLED BACK AGAIN..',   00026730
     & 12,6,16)                                                         00026740
      IF(IER.EQ.133)CALL WARN('MAXFN EXCEEDED.',3,6,16,$100)            00026750
      IF(IER.EQ.38)CALL WARN('JACOBIAN=0. SOLUTION IS STATIONARY POINT',00026760
     & 8,6,16,$100)                                                     00026770
100   WRITE(6,603) (WORK(I),I=1,5),INFER,IER,SSQ,(C(I),I=1,KIP)         00026780
      WRITE(16,603) (WORK(I),I=1,5),INFER,IER,SSQ,(C(I),I=1,KIP)        00026790
603   FORMAT(//' $$$$  IMSLMQ CONVERGENCE INFORMATION:'//               00026800
     & ' NORM OF GRADIENT',T32,E16.8/' FUNCTION EVALUATIONS',T32,E16.8/ 00026810
     & ' EST. SIGN. DIGITS',T32,E16.8/' MARQUARDT PARAMETER',T32,       00026820
     & E16.8/' NO. ITERATIONS',T32,E16.8/' TYPE CONVERGENCE (INFER)',   00026830
     & T32,I3/' ERROR CODE (IER)',T32,I5/                               00026840
     & ' RESIDUAL SUM-OF-SQUARES (SSQ)=',E16.8//                        00026850
     &' **** FINAL UNSCALED PARAMETERS'//                               00026860
     & (5E16.8))                                                        00026870
99    KK=MAXO((KIP+1)*KIP/2,5)                                          00026880
      DO 80 I=1,KIP                                                     00026890
80    GRAD(I)=2.*WORK(KK+I)                                             00026900
      WRITE(6,82) (GRAD(I),I=1,KIP)                                     00026910
82    FORMAT(/' SCALED GRADIENT'//(5E16.8))                             00026920
      WRITE(16,82) (GRAD(I),I=1,KIP)                                    00026930
      IF(IPRT.EQ.-2) WRITE(6,699)                                       00026940
699   FORMAT(/3X,'I',4X,'OBS.Y(I)',6X,'CAL',11X,'RES',8X,'X(I,1)',8X,   00026950
     & 'WT(I)')                                                         00026960
      WRITE(16,1699)                                                    00026970
1699  FORMAT(/3X,'I',4X,'OBS.Y(I)',6X,'CAL',11X,'RES',8X,'X(I,1)',8X,   00026980
     & 'X(I,2)',8X,'X(I,3)',8X,'X(I,4)',8X,'WT(I)')                     00026990
      SUMF2=0.0                                                         00027000
      DO 110 I=1,NOBS                                                   00027010
      RES=F(I)/SQWT(I)                                                  00027020
      YCAL=Y(I)-RES                                                     00027030
      WT=SQWT(I)*SQWT(I)                                                00027040
      SUMF2=SUMF2+F(I)*F(I)                                             00027050
      IF(IPRT.EQ.-2) WRITE(6,210) I,Y(I),YCAL,RES,X(I,1),WT             00027060
210   FORMAT(1X,I3,2E14.6,E11.3,2E14.6)                                 00027070
      WRITE(16,211) I,Y(I),YCAL,RES,(X(I,J),J=1,4),WT                   00027080
211   FORMAT(1X,I3,2E14.6,E11.3,5E14.6)                                 00027090
110   CONTINUE                                                          00027100
      IF(N.EQ.KIP) RMSERR=0.0                                           00027110
      IF(N.GT.KIP) RMSERR=SQRT(SUMF2/(N-KIP))                           00027120
      WRITE(6,604) RMSERR                                               00027130
604   FORMAT(/' **** RMSERR=',E16.8)                                    00027140
      WRITE(16,604) RMSERR                                              00027150
C--PRINT ON FILE16 (ONLY) THE FINAL SCALED PARTIALS (JACOBIAN)           00027160
      WRITE(16,605)                                                     00027170
605   FORMAT(/' FINAL SCALED PARTIALS (JACOBIAN)')                      00027180
      DO 112 I=1,NOBS                                                   00027190
      WRITE(16,606) I,(XJAC(I,J),J=1,KIP)                               00027200
606   FORMAT(1X,I3,5E16.8/(4X,5E16.8))                                  00027210
```

```
112   CONTINUE                                                          00027220
C--GET INVERSE JACOBIAN TRANSPOSE*JACOBIAN (FROM XJTJ SYMMETRIC MATRIX) 00027230
      CALL LINV1P(XJTJ,KIP,H,5,D1,D2,IER)                               00027240
      IF(IER.GT.128) CALL ERRMSG('IN LINV1P CALL.',3,6,16)              00027250
C--FINAL STATISTICS                                                     00027260
      DO 301 I=1,KIP                                                    00027270
      DO 301 J=1,KIP                                                    00027280
301   COV(I,J)=H(LOC(I,J))                                              00027290
      IF(IPRT.EQ.-2) WRITE(6,120)                                       00027300
120   FORMAT(/' SCALED COVARIANCE MATRIX (INVERSE OF XJTJ)')            00027310
      WRITE(16,120)                                                     00027320
      DO 122 I=1,KIP                                                    00027330
      SE(I)=SQRT(ABS(COV(I,I)))                                         00027340
      IF(IPRT.EQ.-2) WRITE(6,300) INDEX(I),(COV(I,J),J=1,KIP)           00027350
300   FORMAT(1X,I2,10E12.4/(3X,10E12.4))                                00027360
      WRITE(16,300) INDEX(I),(COV(I,J),J=1,KIP)                         00027370
122   CONTINUE                                                          00027380
      IF(IPRT.EQ.-2) WRITE(6,304)                                       00027390
304   FORMAT(/' CORRELATION MATRIX')                                    00027400
      WRITE(16,304)                                                     00027410
      DO 131 I=1,KIP                                                    00027420
      IF(SE(I).EQ.0.0) GO TO 132                                        00027430
      DO 129 J=1,KIP                                                    00027440
      IF(SE(J).EQ.0.0) GO TO 129                                        00027450
      COV(I,J)=COV(I,J)/(SE(I)*SE(J))                                   00027460
129   CONTINUE                                                          00027470
133   IF(IPRT.EQ.-2) WRITE(6,300) INDEX(I),(COV(I,J),J=1,KIP)           00027480
      WRITE(16,300) INDEX(I),(COV(I,J),J=1,KIP)                         00027490
      GO TO 131                                                         00027500
132   COV(I,I)=1.0                                                      00027510
      GO TO 133                                                         00027520
131   CONTINUE                                                          00027530
  125 WRITE(6,303)                                                      00027540
  303 FORMAT(/15H   ** PARAMETER,3X,9HSTD ERROR,3X,                     00027550
     & 31HSTD ERROR/PARAMETER  (UNSCALED))                              00027560
      WRITE(16,303)                                                     00027570
      DO 126 I=1,KIP                                                    00027580
      SE(I)=RMSERR*SE(I)                                                00027590
      IF(SP.GT.0) SE(I)=C(I)*SE(I)                                      00027600
      SEC=SE(I)/C(I)                                                    00027610
      WRITE(6,300) INDEX(I),C(I),SE(I),SEC                              00027620
      WRITE(16,300) INDEX(I),C(I),SE(I),SEC                             00027630
126   CONTINUE                                                          00027640
      DO 600 I=1,KIP                                                    00027650
600   H(I)=C(I)                                                         00027660
      IF(IP.EQ.0) GO TO 601                                             00027670
C--PUT SOL C AND BFIX TOGETHER FOR SUBEND USE.                          00027680
      IM=0                                                              00027690
      DO 127 I=1,K                                                      00027700
      H(I)=B(I)                                                         00027710
      DO 128 J=1,IP                                                     00027720
      IF(I.EQ.IB(J)) GO TO 127                                          00027730
```

```
128   CONTINUE                                                          00027740
      IM=IM+1                                                           00027750
      H(I)=C(IM)                                                        00027760
127   CONTINUE                                                          00027770
601   CALL SUBEND(Y,X,H,K,N,TITLE,1)                                    00027780
      IF(ISTOP.EQ.1) GO TO 999                                          00027790
      READ(5,4) TITLE                                                   00027800
      DO 1000 I=1,20                                                    00027810
1000  B(I)=0.0                                                          00027820
      GO TO 6                                                           00027830
C--FOLLOWING CALL ONLY FOR HONEYWELL MULTICS SYSTEM:                    00027840
999   CALL CLOSE_FILE('-ALL')                                           00027850
C     STOP                                                              00027860
      RETURN                                                            00027870
      END                                                               00027880

      SUBROUTINE FPXSSQ(C,N,KIP,F)                                      00027890
C--CALCULATES RESIDUAL VECTOR F(N) FOR 'ZXSSQ' (EXTERNAL FPXSSQ) FOR    00027900
C  UNSCALED C(KIP) PARAMETER VECTOR AND DATA X(200,5),Y(200) IN FIXDAT. 00027910
C                                                                       00027920
C       C= INPUT VECTOR OF PARAMETERS (LENGTH KIP)                      00027930
C       N= NO. OBS. <= 200                                              00027940
C     KIP= NO. PARAMETERS =K-IP (IP>=0)                                 00027950
C       F= OUTPUT VECTOR OF (WEIGHTED) FUNCTION RESIDUALS (LENGTH N)    00027960
C                                                                       00027970
C--CALLS 'FCODE' AS CODED FOR 'MARQRT' WITH FIXED DATA IN COMMON/FIXDAT/00027980
C                                                                       00027990
      DIMENSION C(1),F(1),PRNT(5),SQWT(200),BIP(20)                     00028000
      COMMON/FIXDAT/Y(200),X(200,5),BFIX(20),YMAX,IIB(20),IIP,NOBS,K    00028010
      EQUIVALENCE (SQWT(1),X(1,5))                                      00028020
      IF(IIP.GT.0) GO TO 2                                              00028030
      DO 1 I=1,N                                                        00028040
      CALL FCODE(Y,X,C,PRNT,FF,I,1)                                     00028050
1     F(I)=SQWT(I)*(Y(I)-FF)                                            00028060
      RETURN                                                            00028070
2     IM=0                                                              00028080
      DO 4 I=1,K                                                        00028090
      BIP(I)=BFIX(I)                                                    00028100
      DO 3 J=1,IIP                                                      00028110
      IF(I.EQ.IIB(J)) GO TO 4                                           00028120
3     CONTINUE                                                          00028130
      IM=IM+1                                                           00028140
      BIP(I)=C(IM)                                                      00028150
4     CONTINUE                                                          00028160
      DO 5 I=1,N                                                        00028170
      CALL FCODE(Y,X,BIP,PRNT,FF,I,1)                                   00028180
5     F(I)=SQWT(I)*(Y(I)-FF)                                            00028190
      RETURN                                                            00028200
      END                                                               00028210

      SUBROUTINE LNXSSQ(C,N,KIP,F)                                      00028220
C--INTERFACES TO 'FPXSSQ' TO ALLOW PARMS TO BE IN LOG OR LINEAR         00028230
```

```
C  SPACE OUTSIDE, BUT ALWAYS LINEAR WITHIN FPXSSQ.                    00028240
C                                                                    00028250
C--CALLS 'FPSXXQ' (AND IN TURN 'FCODE')                              00028260
C                                                                    00028270
      DIMENSION C(1),F(1),CTEM(20)                                   00028280
      COMMON/PRT/IPRT                                                00028290
      DO 1 I=1,KIP                                                   00028300
1     CTEM(I)=EXP_(C(I))                                             00028310
      CALL FPXSSQ(CTEM,N,KIP,F)                                      00028320
      RETURN                                                         00028330
      END                                                            00028340
```

Appendix 2.-- Conversion to other systems

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

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

```
character*n          (replace with logical*n or delete)
call open...         (delete)
call close...        (delete)
exp...               (replace with exp)
dexp...              (replace with dexp)
cexp...              (replace with cexp)
```

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

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

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

6. To replace the IMSL interface routines (IMSLMQ, FPXSSQ, LNXSSQ, and all calls to the IMSL Library) with the nonlinear least squares subprogram MARQRT (available in Anderson, 1979c), replace the main program (lines 00005980-00006090) with the following code:

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

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

to be available (even if not called), the following subprogram could be used:

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

If converting to subprogram MARQRT, all parameters prefixed by an "*" apply; however, parameters iopt,parm(),nsig, eps,delta, and maxfn cannot be used in this case. In addition to subprogram MARQRT, the following subprograms from Anderson (1979b) are required: GJR, UNSCAL, and ASINH.

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

```
mqlvemcpl test problem, model parms b0=.1,.1,.3,.25,.8,100
$parms n=22,k=6,m=2,sp=1,ider=0,iprt=-2,iwt=1,
b=.09,.15,.4,.2,1,110$
(4f10.0)
$init  mm=2,iob=5,rl=100,sl=100,nn=2,kase=2,icmplx=1,lcmplx=2$
```

file10

```
0.001295  .01       1.        .00001
-7.4      .01       2.        .1
0.00129   .025      1.        .00001
-8.49     .025      2.        .1
0.001283  .063      1.        .00001
-9.84     .063      2.        .1
0.001276  .16       1.        .00001
-11.92    .16       2.        .1
0.001268  .4        1.        .00001
-15.9     .4        2.        .1
0.001259  1.        1.        .00001
-24.52    1.        2.        .1
0.001246  2.5       1.        .00001
-43.53    2.5       2.        .1
0.00122   6.3       1.        .00001
-83.23    6.3       2.        .1
0.001161  16.       1.        .00001
-156.84   16.       2.        .1
0.00103   40.       1.        .00001
-265.64   40.       2.        .1
0.0008052 100.      1.        .00001
-346.8    100.      2.        .1
```

```
1E M C P L--     malvemcpl test problem, model parms b0=.1,.1,.3,.25,.8,100


 m=    2         iob=    5       icmplx=    1    kase=    2
 sl=   0.100E+03 rl=   0.100E+03 tol=  0.100E-07

 lcmplx=    2
 fintl1=.100E-03 in1=    1       mev1=  300      nfin=    1
 fintl2=.100E-05 in2=    1       mev2=  300


1i m s l m a  --     malvemcpl test problem, model parms b0=.1,.1,.3,.25,.8,100

 n=   22         k=   6          ip=   0         m=  2           e= 0.000E+00
 ialt= 10        istop= 1        iwt= 1          niter=    10    scalep= 1
 iprt= -2        iopt= 1         nsig=  3        maxfn=   120    eps= 0.000E+00
 delta= 0.000E+00                                .
 parm= 0.100E-01 0.200E+01 0.120E+03 0.100E+00

 fmt=(4f10.0)


 initial parameters

  0.90000000E-01  0.15000000E+00  0.40000000E+00  0.20000000E+00  0.10000000E+01
  0.11000000E+03
0warning--maxfn exceeded.


 $$$$  imslmq convergence information:

 norm of gradient                0.36320558E+04
 function evaluations            0.12100000E+03
 est. sign. digits               0.21958871E+01
 marquardt parameter             0.21209739E-01
 no. iterations                  0.29000000E+02
 type convergence (infer)        0
 error code (ier)                133
 residual sum-of-squares (ssq)=  0.55422972E+02

 **** final unscaled parameters

  0.99836191E-01  0.97348635E-01  0.35017996E+00  0.23472579E+00  0.81935589E+00
  0.10534573E+03

 scaled gradient

  0.27038359E+04 -0.22346078E+04 -0.33793453E+03 -0.69155015E+03  0.38872346E+03
  0.37958121E+03
```

i	obs.y(i)	cal	res	x(i,1)	x(i,2)	x(i,3)	x(i,4)	wt(i)
1	0.129500E-02	0.130226E-02	-0.726E-05	0.100000E-01	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
2	-0.740000E+01	-0.753648E+01	0.136E+00	0.100000E-01	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
3	0.129000E-02	0.129631E-02	-0.631E-05	0.250000E-01	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
4	-0.849000E+01	-0.856146E+01	0.715E-01	0.250000E-01	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
5	0.128300E-02	0.128972E-02	-0.672E-05	0.630000E-01	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
6	-0.984000E+01	-0.986402E+01	0.240E-01	0.630000E-01	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
7	0.127600E-02	0.128249E-02	-0.649E-05	0.160000E+00	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
8	-0.119200E+02	-0.119324E+02	0.124E-01	0.160000E+00	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
9	0.126800E-02	0.127474E-02	-0.674E-05	0.400000E+00	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
10	-0.159000E+02	-0.158902E+02	-0.985E-02	0.400000E+00	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
11	0.125900E-02	0.126566E-02	-0.666E-05	0.100000E+01	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
12	-0.245200E+02	-0.244754E+02	-0.446E-01	0.100000E+01	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
13	0.124600E-02	0.125252E-02	-0.652E-05	0.250000E+01	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
14	-0.435300E+02	-0.433425E+02	-0.187E+00	0.250000E+01	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
15	0.122000E-02	0.122667E-02	-0.667E-05	0.630000E+01	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
16	-0.832300E+02	-0.830257E+02	-0.204E+00	0.630000E+01	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
17	0.116100E-02	0.116612E-02	-0.512E-05	0.160000E+02	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
18	-0.156840E+03	-0.157419E+03	0.579E+00	0.160000E+02	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
19	0.103000E-02	0.103536E-02	-0.536E-05	0.400000E+02	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
20	-0.265640E+03	-0.265391E+03	-0.249E+00	0.400000E+02	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03
21	0.805200E-03	0.811560E-03	-0.636E-05	0.100000E+03	0.100000E+01	0.100000E-04	0.000000E+00	0.100000E+11
22	-0.346800E+03	-0.346893E+03	0.931E-01	0.100000E+03	0.200000E+01	0.100000E+00	0.000000E+00	0.100000E+03

**** rmserr= 0.18611651E+01

final scaled partials (Jacobian)

1	0.98642883E+02	0.31646329E+02	0.35165714E+01	-0.18362811E+01	0.33600423E+01
	-0.64437265E+01				
2	-0.97259989E+01	0.19241101E+02	0.92549404E+02	0.51542303E+02	0.74988357E+02
	-0.18125327E+03				
3	0.98068936E+02	0.31555212E+02	0.43102299E+01	-0.12279366E+01	0.31959843E+01
	-0.77899473E+01				
4	-0.97288013E+01	0.16031380E+02	0.99668590E+02	0.72159225E+02	0.37460612E+02
	-0.20050412E+03				
5	0.97502064E+02	0.31639331E+02	0.52438661E+01	-0.55199177E+00	0.46712477E+01
	-0.93184847E+01				
6	-0.97288013E+01	0.16031380E+02	0.10678140E+03	0.82467686E+02	0.37460612E+02
	-0.21974923E+03				
7	0.96609254E+02	0.31464095E+02	0.56795736E+01	-0.15769545E+00	0.28682879E+01
	-0.11155544E+02				
8	0.32448687E+01	0.19246843E+02	0.11391971E+03	0.92785375E+02	0.37527745E+02
	-0.23578922E+03				
9	0.96084901E+02	0.31478127E+02	0.67220370E+01	0.77734514E+00	0.33600423E+01
	-0.12831324E+02				
10	0.32419663E+02	0.25654802E+02	0.12103252E+03	0.10308461E+03	0.00000000E+00
	-0.24541321E+03				
11	0.95376320E+02	0.31415038E+02	0.75312309E+01	0.14532899E+01	0.28682879E+01
	-0.14643846E+02				
12	0.10374292E+03	0.35266739E+02	0.12813258E+03	0.10308461E+03	0.37460612E+02
	-0.25504294E+03				
13	0.94703147E+02	0.31520151E+02	0.85114732E+01	0.22643775E+01	0.28682879E+01
	-0.16403775E+02				
14	0.25287918E+03	0.41686182E+02	0.13527089E+03	0.92766917E+02	0.37594879E+02
	-0.26786345E+03				
15	0.94277984E+02	0.31730448E+02	0.95695517E+01	0.29403223E+01	0.49989440E+01
	-0.18090073E+02				
16	0.58029165E+03	0.19246843E+02	0.14235821E+03	0.87635758E+02	0.00000000E+00
	-0.28230374E+03				
17	0.93902452E+02	0.31639331E+02	0.10596559E+02	0.36049619E+01	0.43435512E+01
	-0.19832483E+02				
18	0.11573326E+04	-0.14111289E+03	0.92390067E+02	0.20524634E+02	0.00000000E+00
	-0.28715163E+03				
19	0.93753632E+02	0.28555285E+02	0.10736617E+02	0.37176098E+01	0.40993523E+00
	-0.21164685E+02				
20	0.19777097E+04	-0.72467350E+03	-0.12808159E+03	-0.31008440E+02	-0.36520740E+02
	-0.10587252E+03				
21	0.92208908E+02	0.16555615E+02	0.67998732E+01	0.21911824E+01	0.12705055E+01
	-0.18821045E+02				
22	0.21201821E+04	-0.19336876E+04	-0.64081587E+03	-0.29886230E+03	0.75189758E+02
	0.80203947E+03				

scaled covariance matrix (inverse of xJtJ)

1	0.4763E-05	0.7141E-05	0.2942E-05	0.3994E-04	0.1320E-04	0.2080E-04
2	0.7141E-05	0.2963E-04	-0.1026E-03	0.8143E-04	0.6041E-04	-0.5765E-05
3	0.2942E-05	-0.1026E-03	0.6482E-03	-0.1723E-03	-0.1950E-03	0.2218E-03
4	0.3994E-04	0.8143E-04	-0.1723E-03	0.5796E-03	0.1029E-03	0.1604E-03
5	0.1320E-04	0.6041E-04	-0.1950E-03	0.1029E-03	0.1922E-03	-0.2365E-04
6	0.2080E-04	-0.5765E-05	0.2218E-03	0.1604E-03	-0.2365E-04	0.1734E-03

correlation matrix

1	0.1000E+01	0.6011E+00	0.5295E-01	0.7602E+00	0.4364E+00	0.7238E+00
2	0.6011E+00	0.1000E+01	-0.7407E+00	0.6214E+00	0.8006E+00	-0.8042E-01
3	0.5295E-01	-0.7407E+00	0.1000E+01	-0.2811E+00	-0.5526E+00	0.6617E+00
4	0.7602E+00	0.6214E+00	-0.2811E+00	0.1000E+01	0.3083E+00	0.5059E+00
5	0.4364E+00	0.8006E+00	-0.5526E+00	0.3083E+00	0.1000E+01	-0.1296E+00
6	0.7238E+00	-0.8042E-01	0.6617E+00	0.5059E+00	-0.1296E+00	0.1000E+01

** parameter	std error	std error/parameter (unscaled)
1	0.9984E-01	0.4055E-03	0.4062E-02
2	0.9735E-01	0.9862E-03	0.1013E-01
3	0.3502E+00	0.1659E-01	0.4738E-01
4	0.2347E+00	0.1052E-01	0.4481E-01
5	0.8194E+00	0.2114E-01	0.2580E-01
6	0.1053E+03	0.2582E+01	0.2451E-01

1E M C P L-- malvemcpl test problem, model parms b0=.1,.1,.3,.25,.8,100

final unscaled parameters--

	rho(0)	chargeability	c	tau
1	0.99836191E-01	0.10016408E+02		
2	0.97348635E-01	0.10272358E+02	0.35017996E+00	0.23472579E+00	0.81935589E+00

		lyr	depth
6	0.10534573E+03	1	0.10534573E+03