U.S. DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

Nonlinear least-squares inversion of transient induced polarization data (Program NLSTIP)

by

Walter L. Anderson
and
Bruce D. Smith

Open-File Report 84-514

1984

DISCLAIMER

This program was written in FORTRAN-77 for a VAX-11/780 system*. Although program tests have been made, no guarantee (expressed or implied) is made by the authors or the Geological Survey regarding program correctness, accuracy, or proper execution on all computer systems.

* Any use of trade names in this report is for descriptive purposes only and does not imply endorsement by the U.S. Geological Survey. This report is preliminary and has not been reviewed for conformity with U.S. Geological Survey editorial standards.

CONTENTS

ABSTRACT

A computer program is presented that inverts transient induced polarization (IP) data using an adaptive nonlinear least-squares (NLS) method. Multiple Cole-Cole relaxation models may be used to define the transient IP forward function, where the system waveform is either a single on-off step in time, or an arbitrary periodic bipolar time response. A digital filtering algorithm is the default method used for evaluating the transient function in the NLS solution; optionally, a direct numerical quadrature method is also available. An inversion example using an IP field data set is given in numerical and graphical form. Program parameters are defined, and the VAX-11 operating instructions are summarized. The FORTRAN program source is listed in an appendix.

INTRODUCTION

Inversion of transient (time-domain) induced polarization (IP) data is provided by program NLSTIP, which is described in this report. The numerical technique uses a general adaptive nonlinear least-squares algorithm originally developed by Dennis and others (1979, 1981), and extended externally for constrained nonlinear regression by Anderson (1982). The transient IP forward problem, as required in the inverse solution, uses a combination of methods proposed by Lee (1981) and Guptasarma (1982). The latter reference uses a fast digital filter approach, and is the standard default method used in NLSTIP. (Note that a stand-alone forward transient IP program can be easily obtained from the NLSTIP subprograms, as described by Anderson, 1984.)

This report summarizes the general nonlinear least-squares (NLS) inversion method discussed by Anderson (1982), but as it applies to observed transient IP data. As an example, the Scintrex IPR-11 receiver (Johnson, 1983) can be used to obtain digital IP transients with a variety of control options, as well as other available digital-acquisition IP systems. A Cole-Cole relaxation model response (Pelton and others, 1978), defined in the frequency-domain, is Fourier transformed to obtain the transient IP response. Up to four arbitrary relaxation models can be combined by multiplication as described by Major and Silic (1981, p. 917) in one execution of NLSTIP. The combination of multiple IP relaxations or dispersions has been the subject of some controversy among the experts (Major and Silic, 1981; Washburne, 1982). We have chosen to use multiplicative combinations rather than additive models because the frequency-domain response has a more intuitive behavior. Washburne (1982, p. 79) gives an extensive discussion of the relative merits of the two ways to combine multiple relaxations, which include allowances for EM coupling effects. This topic, however, has not been rigorously studied nor reported in the literature for time-domain measurements.

Additionally, multiple (or concatenated) IP data sets from different sites and (or) different switching times can be processed simultaneously by NLSTIP to obtain a joint inversion. The transient system waveform can be a single on-off step in time, or it can be specified as an arbitrary on-off bipolar pulse waveform, in which case the total duration time containing a number of periods of the waveform can be further specified. For more than one step, the final transient response at the end of several periods is computed by superpositon of unit step-function responses shifted in time (Wait, 1982, p. 72-74).

The remainder of this report contains 1) a summary of the general computations, 2) a description of the required program parameters, and 3) the VAX operating instructions. Appendix 1 offers some suggestions in converting the VAX

program to other computer systems; Appendix 2 lists a simple input/output test example; and Appendix 3 gives a FORTRAN-77 source listing.

SUMMARY OF CALCULATIONS

The NLS inversion method (Anderson, 1982; Dennis and others, 1979, 1981) requires a twice-continuously differentiable nonlinear objective function describing the model equation as a function of the unknown parameters. For a single Cole-Cole relaxation model, and for a single step-function system waveform, the transient IP objective function can be written (Lee, 1981) as

$$V(t) = m R_0 I_0 \left[\sum_{n=0}^{\infty} \frac{(-1)^n}{\Gamma(n c + 1)} \left(\frac{t}{\tau}\right)^{nc} \right] \qquad (1)$$

where R_0 = amplitude voltage at zero frequency ($R_0>0$),
m = chargeability factor ($0 \le m \le 1$),
τ = time constant ($\tau>0$ seconds),
c = frequency dependency factor ($0<c\le 1$),
I_0 = system waveform current (amps),
Γ = Gamma function of a real argument,
and $V(t)$ = transient voltage at time $t>0$ seconds.

The four model parameters (R_0, m, τ, c) are to be determined by the NLS inversion program, given an observed data set for $V(t)$ defined over a range of t, and an initial starting estimate. The form of (1) is not practical for direct implementation, due to the slow convergence of the series, and hence a direct quadrature of the integral form given by Lee (1981; eq. 8) can be used,

$$V(t) = \frac{I_0}{\pi} \int_0^{\infty} \frac{m R_0 e^{-xt/\tau} x^c \sin \pi c}{x[(1 + x^c \cos \pi c)^2 + x^{2c} \sin^2 \pi c]} dx. \qquad (2)$$

The infinite integration limit in (2) may be replaced by a finite bound, x_0, using the minimum floating-point range of $\exp(-x_0 t/\tau)$ on the computer being used; for the VAX system, we define $x_0 = 87\tau/t$ for all t. Using an adaptive finite Gaussian quadrature (Patterson, 1973), $V(t)$ can be computed to a user-specified relative error or tolerance (usually .001 is sufficient in practice). It should be noted that even though Lee (1981) provided an asymptotic formula for large t, the quadrature method used to evaluate (2) applies over all t of practical interest.

The numerical method used to calculate (2) works well for a single forward solution. However, for an inverse solution, many forward evaluations are necessary. Therefore the faster digital filter procedure outlined by Guptasarma (1982) was chosen as the default method in NLSTIP, and $V(t)$ is given by the summation

$$V(t) = \sum_{r=1}^{21} \phi_r \, \mathrm{Re}[H(\omega_r)], \tag{3}$$

where

$$H(\omega) = R_0\left\{1 - m\left[1 - \frac{1}{1 + (j\omega\tau)^c}\right]\right\}, \tag{4}$$

and

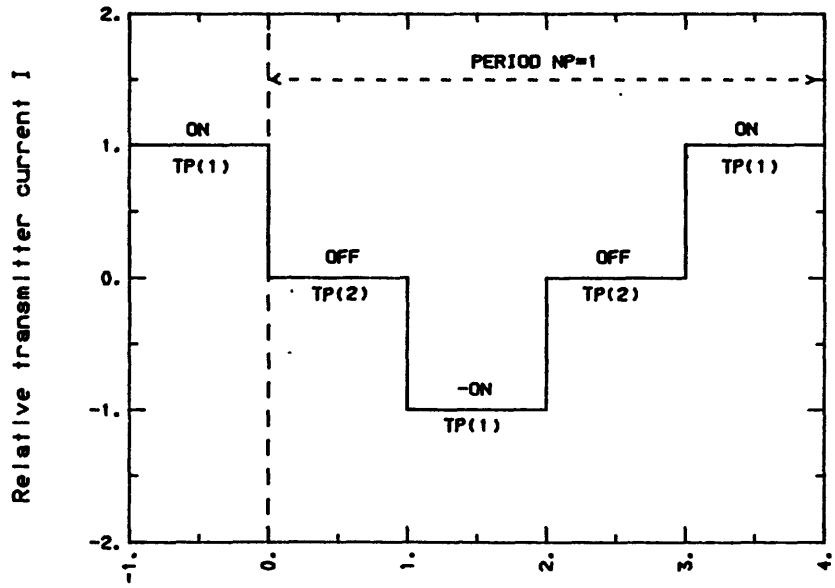
$$\omega_r = 10^{(a_r - \log_{10} t)}, \tag{5}$$

with the filter abscissas a_r and coefficients ϕ_r as given in Guptasarma (1982, Table 1).

Unfortunately, the digital filter method using (3)-(5) is not always sufficiently accurate at late times for some models. The program has a hybrid option (parameter IOPT=-1) that uses numerical integration of (2) in place of detectable "noisy" filtered perturbations on the transient tail (if any). Generally, the default digital filter method (IOPT=0) is good to about 3-5 places in the early and mid-times, and for many models, the late-time "ripple" (if any) does not seriously effect the final least-squares solution. The direct numerical integration option (IOPT=1) is not recommended for routine work, because the total computer time would be much larger than for any of the other IOPT cases.

The time-domain IP system bipolar waveform is illustrated in Figure 1.

Fig.1--IP system waveform for one period



An option to vary on-off times (parameter array TP) and the total number of periods (parameter NP) is provided for special cases. For instance, the Scintrex IPR-11 broadband time-domain receiver has equal on-off times of 0.2, 1.0, 2.0, or 4.0 seconds (via a selectable switch); here, TP(1)=TP(2) and NP>0 would be specified.

The transient responses given by formulas (2) or (3) are derived for a single on-off step waveform. In order to compute the final transient response for an arbitrary waveform as in Figure 1, the superposition procedure described by Wait (1982, p. 72-74) can be followed. For example, for NP=1 and TP(1)=TP(2)=T, the transient becomes

$$V(t) = A(t) - A(t+T) - A(t+2T) + A(t+3T) \qquad (6)$$

where A(t), A(t+T), etc., are the step responses at time t and t+T, etc., respectively. Note that the signs in (6) account for the reverse polarity as shown in Figure 1. For more than one period (e.g., NP=2), the following superposition is added to (6):

$$A(t+4T) - A(t+5T) - A(t+6T) + A(t+7T).$$

This process is continued for all subsequent periods until a user-selected value of NP is reached. In general, we assume that the transient response has decayed sufficiently to be negligible at the longest time shift, (4NP-1)T; i.e., a sufficiently large value of NP should be specified such that a steady-state transient response is achieved. Note that a steady-state test can be easily performed incrementally for a given data set by observing the changes in the least-squares model parameters for increasing values of NP. However, in practice, extremely large values of NP should be avoided in the interest of conserving computer time.

The unknown Cole-Cole model parameters used in NLSTIP are denoted by the vector B(J), J=1,2,...,K, which has the required order as defined in Table 1. (See parameter MODEL in the section: $INIT PARAMETER DEFINITIONS below.)

Table 1.--Cole-Cole parameter names and order assumed

B(J)	Parameter name	MODEL	K
B(1)	R0		
B(2)	m	1	4
B(3)	Tau		
B(4)	c		
B(5)	m2		
B(6)	Tau2	2	7
B(7)	c2		
B(8)	m3		
B(9)	Tau3	3	10
B(10)	c3		
B(11)	m4		
B(12)	Tau4	4	13
B(13)	c4		

Many NLS options are available in the interface subprogram NLSOL (Anderson, 1982), which the reader should become familiar with before running NLSTIP. Following the NLS notation in Anderson (1982, p.11-12), we let X(I,1)=t and Y(I)=V(t) be the observed transient data arrays. Program NLSTIP then reads the observed data matrix

$$(Y(I),X(I,1),I=1,2,...,N)$$

using an arbitrary object- or run-time input format (see any FORTRAN manual). Since Y(I) can range several decades in magnitude for all I, it is advised that a weighted least-squares option be used (see IWT=1 or 2, Anderson, 1982, p.14-15), which requires the augumented data matrix

$$(Y(I),X(I,1),X(I,2),I=1,2,...,N),$$

where X(I,2) is the standard deviation (IWT=1) of observation Y(I), or X(I,2) is the variance (IWT=2). Note that if X(I,2) is unknown, one may use a statistical weighting factor of 1/Y(I) (Bevington, 1969, p.108) by setting X(I,2)=Y(I) and IWT=2; this procedure is preferable to using unity weights (IWT=0).

An analytical partial derivative subprogram (PCODE) was not included in this version of NLSTIP, so that the estimated finite-difference derivative option (IDER=1) must be used. This option requires only the forward-problem solution subprogram FCODE. Appendix 3 contains a listing of FCODE, which includes the methods described above in (2)-(6), as selected by parameter IOPT.

Because realizable Cole-Cole models are sought to fit the given data, a constrained minimization type solution (SP=3 or 4) is advised, along with an initial guess array B(J) and reasonable lower and higher parameter bound arrays, BL(J) and BH(J) respectively, where $BL(J) \leq B(J) \leq BH(J)$, J=1,2,...,K (see Anderson, 1982, p.17). This approach limits parameter space searching, and in some cases may avoid false starts or catastrophic overflow conditions from poor B(J) estimates and data. In addition, individual parameters can be fixed in the least-squares program by specifying parameters IP and IB (Anderson, 1982, p.13). In particular, for the IPR-11 receiver, one should always fix B(1)=1000.

PARAMETERS, FILES AND DATA REQUIRED

Two general classes of NAMELIST parameters are required: $PARMS and $INIT. All $PARMS parameters (excluding the ISTOP=0 option), program files (FOR005-FOR016), and data ordering requirements used by NLSTIP are identical to those described in detail for subprogram NLSOL (Anderson, 1982, p.9-21). (Familiarity with the $PARMS defined in the latter reference is assumed; definitions of these parameters are not repeated here in the interest of brevity.) Note, however, that the ordering of the $PARMS estimated parameter vector B(J) used by NLSTIP must be given exactly as described above in Table 1. The $INIT model parameters required by NLSTIP must be given after the object-time format statement on FOR005 (see Anderson, 1982, p.10, item 5). For a typical data input, refer to the EXAMPLE below and to Appendix 2.

$INIT PARAMETER DEFINITIONS

$INIT parameters (nondefault parameters must be given):

MODEL=1 (default) uses a single Cole-Cole model defined in Table 1 and requires that $PARMS K=4 be given.

MODEL>1 but <5 uses the combined Cole-Cole models as defined in Table 1; note that the corresponding $PARMS K value must be supplied correctly as indicated in Table 1, otherwise unpredictable results could occur.

ISTEP=0 (default) defines a normal transient decay response.

ISTEP=1 may be used to compute the final transient as $R_0 - V(t)$ as in Wait (1982, p. 71, Figure 1).

MAXPTS= the maximum number of points allowed in the adaptive Gaussian quadrature (Patterson, 1973); this parameter is used only if |IOPT|=1 below. (default MAXPTS=5000.)

EPS= is the requested Gaussian integration tolerance used to compute V(t) by (2) for all t; this parameter is used only if |IOPT|=1 below. (default EPS=0.1E-2.)

TP()= is the array defining the selected waveform on-off

time intervals (in seconds) as follows:

TP(1)=0 (default) if a single on-off step is implied.

TP(1)>0 specifies the "on" time interval between each off interval (see Figure 1).

TP(2)=0 (default) assumes TP(2)=TP(1) whenever TP(1)>0; e.g., for an IPR-11 receiver, this option should always be used.

TP(2)>0 specifies the "off" time that is different from the on interval whenever TP(1)>0 (see Figure 1).

NP=0 (default) if a single on-off step waveform is to be used; in this case, TP(1)=0 also implies this option.

NP>0 indicates the number of periods of the bipolar waveform in Figure 1 to use.

REC= is the receiver time (seconds) used to record the transient IP response. This value is used only for documentation purposes in the output file listings. If omitted, REC=0.0 is assumed without an error flag.

IM=1 (default) specifies that a single transient data set is input in the data matrix (see DATA MATRIX NOTES below) along with TP() and NP as given. Make sure that $PARMS M=1 whenever $INIT IM=1 (a dual requirement that is not possible to cross-check by the general-purpose NLSOL subprogram).

IM=2 specifies that a joint inversion of more than one transient data set is input using TP(1)=X(I,2) and NP as given. See DATA MATRIX NOTES below when IM=2 is used. Again, make sure that $PARMS M=2 whenever $INIT IM=2 is selected.

IM=3 specifies that a joint inversion of more than one transient data set is input using TP(1)=X(I,2) and NP=X(I,3). See DATA MATRIX NOTES below when IM=3 is used. Again, make sure that $PARMS M=3 whenever $INIT IM=3 is selected.

IM=4 specifies that a joint inversion of more than one transient data set is input using TP(1)=X(I,2), NP=X(I,3), and TP(2)=X(I,4). See DATA MATRIX NOTES below when IM=4 is used. Again, make sure that $PARMS M=4 whenever $INIT IM=4 is selected.

IOPT=0 (default) selects only the digital filtering method to compute V(t) as defined in Guptasarma (1982). Currently, IOPT=0 must be used whenever MODEL>1 or IM>1.

IOPT=1 selects only the Gaussian quadrature method to compute V(t) as described above in eq. (2). Currently, IOPT=1 can only be used when MODEL=1 and IM=1. This option in not recommended for routine work because the total computer time would be much larger than with IOPT=0; however, for difficult theoretical cases, IOPT=1 might prove to be useful.

IOPT=-1 selects the IOPT=0 method for V(t) until possibly |V(t)|<1E-6, after which V(t) is replaced by the IOPT=1 method for larger t. Currently, IOPT=-1 can only be used when MODEL=1 and IM=1.

$END [end of $INIT parameters; the "END" is optional.]

EXAMPLE OF INPUT PARAMETERS AND DATA ORDERING

```
EXAMPLE WITH OBJECT DATA MATRIX ON FOR005 (IALT=5)
$PARMS N=10,M=1,K=4,SP=3, IALT=5,
NITER=50,IPRT=-2,IDER=1, IWT=1, IP=1,IB=1,
BL=1000,0,2*.1E-7,
BH=1000,1,.1E5,1,
B= 1000,.5,1,.5$
(3G12.5)
  40.200      0.90000E-01   .011
  35.800      0.15000       .095
  29.300      0.21000       .1
  30.700      0.27000       .13
  21.500      0.48000       .21
  17.900      0.84000       .25
  13.600       1.2000       .4
  11.400       1.7400       .42
  8.3000       2.4600       .65
  7.7000       3.1800       .985
$INIT MODEL=1,TP=4.0,NP=   2,REC=4.0,IM=1$
```

(See Appendix 2 for a complete input/output example, and ANOTHER EXAMPLE below.)

DATA MATRIX NOTES

The data matrix (defined following Table 1) is read under the object-time format statement, and is defined as the sequence of ordered rows:

```
(Y(I),(X(I,L),L=1,M*),I=1,N),
```

where M*=M if IWT=0 (default), or M*=M+1 if IWT=1 or 2. In the above example, IWT=1, M=1, and therefore three columns are required in the data matrix row, where in this case, the last column represents the standard deviation of observation Y(I). The number of items read per record depends on $PARMS M,IWT and the $INIT IM option selected. The various data matrix options are summarized as follows:

(a) Single transient input defined by $INIT IM=1 (requires $PARMS M=1 and $INIT IM=1; max. 3 items per record):
- Y(I)= I-th observed transient value V(t) at t=X(I,1).
- X(I,1)= I-th observed time (sec.), where X(I+1,1)>X(I,1) is required.
- X(I,2)= weight factor of I-th observation (include only if $PARMS IWT>0).

(b) Joint transient input defined by $INIT IM=2 (requires $PARMS M=2 and $INIT IM=2; max. 4 items per record):
- Y(I)= I-th observed transient value V(t) at t=X(I,1).
- X(I,1)= I-th observed time (sec.), where X(I+1,1)>X(I,1)

is required until another data set is encountered as indicated by a change in X(I,2) below.

X(I,2)= TP(1) to use for this data set (should be constant for all I within this data set); note that when X(I,2) changes, this condition indicates that another data set follows, and therefore, the time column X(I,1) may be reset, etc.

X(I,3)= weight factor of I-th observation (include only if $PARMS IWT>0).

(c) Joint transient input defined by $INIT IM=3 (requires $PARMS M=3 and $INIT IM=3; max. 5 items per record):

Y(I)= I-th observed transient value V(t) at t=X(I,1).

X(I,1)= I-th observed time (sec.), where X(I+1,1)>X(I,1) is required until another data set is encountered as indicated by a change in X(I,2) below.

X(I,2)= TP(1) to use for this data set (should be constant for all I within this data set); note that when X(I,2) changes, this condition indicates that another data set follows, and therefore, the time column X(I,1) may be reset, etc.

X(I,3)= NP to use for this data set (should be constant for all I within this data set); this value should only be changed whenever X(I,2)=TP(1) also changes.

X(I,4)= weight factor of I-th observation (include only if $PARMS IWT>0).

(d) Joint transient input defined by $INIT IM=4 (requires $PARMS M=4 and $INIT IM=4; max. 6 items per record):

Y(I)= I-th observed transient value V(t) at t=X(I,1).

X(I,1)= I-th observed time (sec.), where X(I+1,1)>X(I,1) is required until another data set is encountered as indicated by a change in X(I,2) below.

X(I,2)= TP(1) to use for this data set (should be constant for all I within this data set); note that when X(I,2) changes, this condition indicates that another data set follows, and therefore, the time column X(I,1) may be reset, etc.

X(I,3)= NP to use for this data set (should be constant for all I within this data set); this value should only be changed whenever X(I,2)=TP(1) also changes.

X(I,4)= TP(2) to use for this data set (should be constant for all I within this data set); this value should only be changed whenever X(I,2)=TP(1) also changes.

X(I,5)= weight factor of I-th observation (include only if $PARMS IWT>0).

ANOTHER EXAMPLE

The following example of $INIT IM=4 case illustrates the most complicated data matrix, where $PARMS IWT=2 reads Y(I) again for the weight factor X(I,5):

```
EXAMPLE WITH $INIT IM=4 AND IWT=2 (NOTE T1 IN FORMAT)
$PARMS N=20,M=4,K=4,SP=3,NITER=10,IDER=1,
IP=1,IB=1, IWT=2, IALT=5,
BL=1000,0,2*.1E-7, BH=1000,1,.1E5,1,
B=1000,.5,1,.5$
(5G12.5,T1,G12.5)
  102.40      0.45000E-02  1.0000      2.0000      1.5
  94.200      0.75000E-02  1.0000      2.0000      1.5
  80.200      0.10500E-01  1.0000      2.0000      1.5
  59.200      0.13500E-01  1.0000      2.0000      1.5
  40.700      0.24000E-01  1.0000      2.0000      1.5
  41.700      0.42000E-01  1.0000      2.0000      1.5
  37.000      0.60000E-01  1.0000      2.0000      1.5
  33.200      0.87000E-01  1.0000      2.0000      1.5
  30.800      0.12300      1.0000      2.0000      1.5
  28.500      0.15900      1.0000      2.0000      1.5
  43.900      0.45000E-01  2.0000      2.0000      2.
  39.600      0.75000E-01  2.0000      2.0000      2.
  37.300      0.10500      2.0000      2.0000      2.
  24.700      0.13500      2.0000      2.0000      2.
  23.200      0.24000      2.0000      2.0000      2.
  17.000      0.42000      2.0000      2.0000      2.
  13.500      0.60000      2.0000      2.0000      2.
  11.200      0.87000      2.0000      2.0000      2.
  8.2000       1.2300      2.0000      2.0000      2.
  6.9000       1.5900      2.0000      2.0000      2.
$INIT IM=4,MODEL=1,NP=4,REC=2$
```

SPECIAL OBJECT FORMAT PHRASES

If an existing data matrix file does not have the proper defined column ordering in the form (Y(I),X(I,J),J=1,M), then the FORTRAN "Tn" format phrase (as used in the above example) may be used to begin at any column n in the data record. For example, the format (T41,F10.0,T1,2F10.0) will select Y(I) using column 41-50 and X(I,1) beginning at column 1. See any FORTRAN-77 coding manual for other allowable object (run) time format phrases (e.g., the F-format, use of "/" to skip records, etc.). Note that "tab"-characters must <u>not</u> be used when creating the data matrix file FOR010.

VAX OPERATING INSTRUCTIONS

In general, the basic steps described to run NLSOL (Anderson, 1982, p.22-24) can be followed to run NLSTIP in either on-line or batch modes. That is, the parameter and data matrix files may be associated with the logical names

FOR005 and FOR010, respectively, using the VAX-DCL statements:

```
$ASSIGN parameter_file_name FOR005
$ASSIGN data_matrix_file_name FOR010
```

The program can be executed by using the DCL command:

```
$RUN NLSTIP !(use $RUN [WANDERSON]NLSTIP on USGS VAX)
```

If the data matrix is included on FOR005 (i.e., using IALT=5), then the FOR010 assignment is not necessary.

In addition, program NLSTIP has a useful "restart file" (called FOR005.TMP) that is automatically provided each time the program is executed. File FOR005.TMP contains a copy of all parameters on FOR005, plus the last solution B-vector obtained; note that $PARMS ISTOP=0 (Anderson, 1982, p.14) cannot be used because FOR005 is positioned at EOF in creating FOR005.TMP. If desired, one can easily continue (or restart) more iterations simply by using the DCL commands:

```
$ASSIGN FOR005.TMP FOR005
$RUN NLSTIP !(use $RUN [WANDERSON]NLSTIP on USGS VAX)
```

Note that FOR005.TMP may also be edited (using any VAX editor) for other parameter changes, if desired. Also, the reassignment of FOR005 using FOR005.TMP only needs to be done once for multiple continuation runs.

By default, the master print (disk) file is called FOR016.DAT, unless otherwise assigned. This file can be TYPEd or PRINTed on a line printer. Also, file FOR016 may be used as an input file to a plot routine; e.g., to plot the observed (OBS), calculated (CAL), and residual (RES) curves. If program NLSTIP is run on-line, then a shorter terminal print file on FOR006 contains some of the information as on FOR016, but as controlled by parameter IPRT (Anderson, 1982, p.15).

ERROR MESSAGES

Almost all $PARMS syntactical errors are flagged and printed on files FOR006 and FOR016 and the job is aborted (see Anderson, 1982, p.24). However, some cross references (or dual inputs) are not checked; for example, the relationships between $PARMS M and $INIT IM, and $PARMS K and $INIT MODEL in Table 1, respectively, are not double checked by program NLSTIP. This is because a general-purpose nonlinear least-squares algorithm (NLSOL) is being used as a control program, but the model input is external to the particular nonlinear problem requirements (NLSTIP) read by subprogram SUBZ (see Anderson, 1982, p.38). Therefore, the user is responsible for providing exactly K

parameter estimates in B(I),I=1,2,...,K (see Table 1), and that $INIT IM is equal to $PARMS M (otherwise, unpredictable results could occur that are unchecked).

PRINTED OUTPUT

All input parameters are output on files FOR006 and FOR016, with the $INIT parameters given first, followed by all $PARMS parameters given or assumed by default. (Refer to Appendix 2 for a complete sample output listing.)

Specific names (e.g., IT, NF, ...) used by NLSOL in the output listings are tabulated in Anderson (1982, p.25-26). Program NLSTIP also provides a summary listing of the final solution vector B and names at the end of the output file.

REFERENCES

Anderson, W.L., 1982, Adaptive nonlinear least-squares solution for constrained or unconstrained minimization problems (Subprogram NLSOL): USGS Open-File Rept. 82-68, 65 p.

---------, 1984, A general interface for producing forward solution programs (Subprogram FWDSOL): USGS Open-File Rept. 84-348, 43 p.

Bevington, P.R., 1969, Data reduction and error analysis for the physical sciences: McGraw-Hill, N.Y., 336 p.

Dennis, J.E., Gay, D.M., and Welsch R.E., 1979, An adaptive nonlinear least-squares algorithm: Univ. of Wisconsin MRC Tech. Sum. Rept. 2010 (also available as NTIS Rept. AD-A079-716),

---------, 1981, An adaptive nonlinear least-squares algorithm: ACM Trans. on Math. Software, v. 7, no. 3, p. 348-368.

Guptasarma, D., 1982, Computation of the time-domain response of a polarizable ground: Geophysics, v. 47, no. 11, p. 1574-1576.

Johnson, I.M., 1983, Spectral IP parameters as determined through time domain measurements: Presented at the SEG Annual International Meeting, Las Vegas, Nevada, Sept. 11-15, 16 p.

Lee, T., 1981, Short Note--The Cole-Cole model in time domain induced polarization: Geophysics, v. 46, no. 6, p. 932-933.

Major, J., and Silic, J., 1981, Restrictions on the use of Cole-Cole dispersion models in complex resistivity

interpretation: Geophysics, v. 46, no. 6, p. 916-931.

Patterson, T.N.L., 1973, Algorithm 468--Algorithm for automatic numerical integration over a finite interval: Comm. of ACM, v. 16, no. 11, p. 694-699.

Pelton, W.H., Ward, S.H., Hallof, P.G., Sill, W.R., and Nelson, P.H., 1978, Mineral discrimination and removal of inductive coupling with multifrequency IP: Geophysics, v. 34, no. 3, p. 588-609.

Wait, J., 1982, Geo-Electromagnetism: Academic Press, New York, N.Y., 268 p.

Washburne, J.C., 1982, Parameterization of spectral induced polarization data and laboratory and in situ spectral induced polarization measurements: West Shasta copper-zinc district, Shasta, Calif.: MS thesis T-2591 (unpublished), Colo. School of Mines, 268 p.

Appendix 1.-- Conversion to other systems

This program (and associated subprograms) was written in extended ANSI-standard FORTRAN-77 for the VAX-11/780 system. Conversion to systems without an ANSI-FORTRAN-77 compiler would necessitate extensive changes, particularly for all CHARACTER-type variables, IF-THEN-ELSE phrases, etc.

Changes for non-VAX systems might include some (or all) of the following FORTRAN-77 constructs and VAX concepts:

(1) Variable names with more than 6-characters.
(2) Character strings delimited by single-quote characters (e.g., 'STRING'); also, character string concatination (e.g., 'STRING1'//'STRING2').
(3) Passing variable-length character strings in subroutine calls; e.g., CHARACTER*(*) passed length character arguments.
(4) Suppression of arithmetic or exponential underflow messages; note that a VAX-11 result is automatically set to 0.0 after any underflow--which is assumed for this program package. If the target system does not set underflows to 0.0, and suppress warning messages, then a suitable conversion procedure must be used for proper operation of this program package.
(5) Replacement of any special VAX-dependent CALLS or statements (e.g., CALL LIB$INDEX, etc.)
(6) VAX non-ANSI NAMELIST input and output statements.

Appendix 2.-- Test problem input/output listing

The following input files (FOR005 and FOR010) were used to run a test problem using a field data set from an IPR-11 receiver for program NLSTIP on a VAX system. The corresponding output file (FOR016) is listed following FOR010. In addition, file FOR016.DAT was used to plot the final observed (OBS) and calculated (CAL) curves using an external plotter. The symbol "0" represents Y(I) in the plot, and the solid line represents a curve drawn through the calculated (CAL) points.

FOR005

```
SAUDI STA.35/B3: MODEL=1 [B]
$PARMS N=10,M=1,K=4,SP=3,
NITER=50,IPRT=-2,IDER=1,IWT=2,IP=1,IB=1,
BL=.1E-7,0,2*.1E-7,
BH=1000,1,.1E5,1,
B=  1000.0,.5,1,.5$
(2G12.5,T1,G12.5)
$INIT MODEL=1,TP=2.0,NP=   2,REC=2.0$
```

FOR010

```
  43.900      0.45000E-01
  39.600      0.75000E-01
  37.300      0.10500
  24.700      0.13500
  23.200      0.24000
  17.000      0.42000
  13.500      0.60000
  11.200      0.87000
  8.2000       1.2300
  6.9000       1.5900
```

FOR016

```
  <NLSTIP>:      SAUDI STA.35/B3: MODEL=1 [B]

$INIT
MODEL    =            1,
ISTEP    =            0,
MAXPTS   =         5000,
EPS      =  1.0000000E-03,
TP       =   2.000000     ,  0.0000000E+00,
NP       =            2,
REC      =   2.000000     ,
IM       =            1,
IOPT     =            0
$END
```

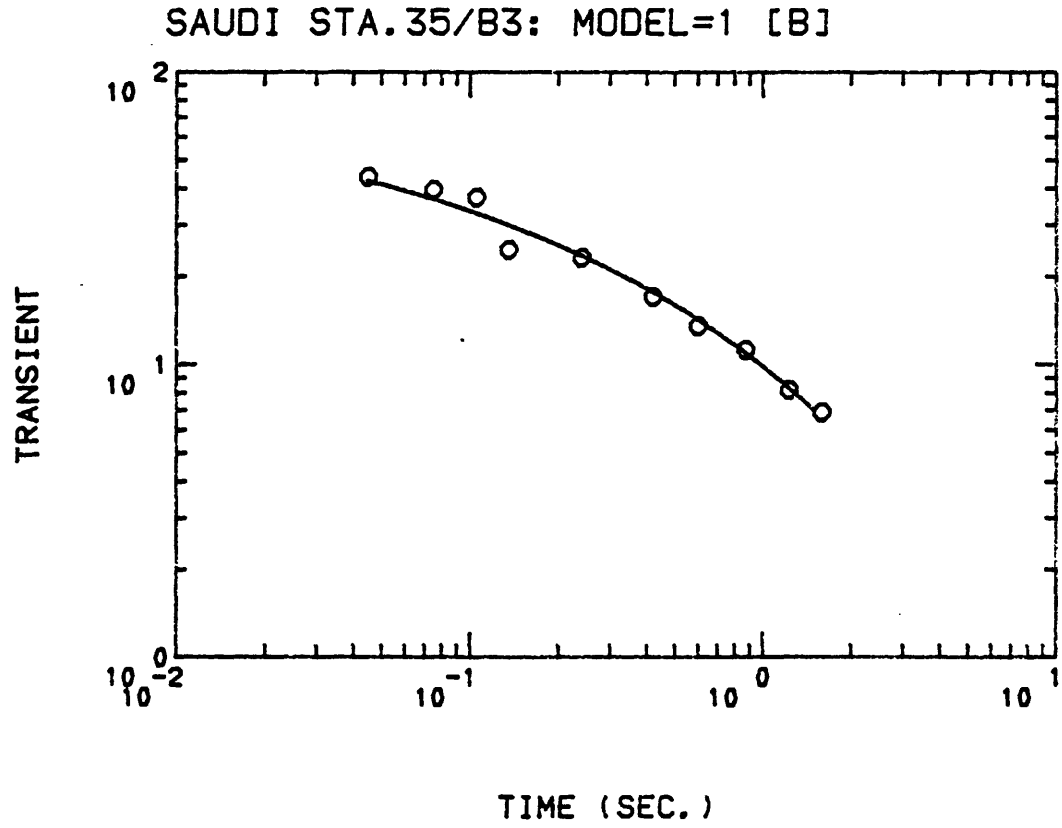
PARAMETER ORDER & NAMES--

1	B(1)=	R0
2	B(2)=	m
3	B(3)=	Tau
4	B(4)=	c

```
{NLSOL}:        SAUDI STA.35/B3: MODEL=1 [B]

N=        10    K=         4    IP=        1    M=         1    IALT=     10
ISTOP=     1    IWT=       2    IDER=      1    IPRT=     -2    NITER=    50
IOUT=      1    SP=        3

PARAMETERS HELD FIXED: IB=  1

FMT=(2G12.5,T1,G12.5)


PARAMETER LOWER BOUNDS: BL=

 0.99999999E-08  0.00000000E+00  0.99999999E-08  0.99999999E-08

INITIAL PARAMETERS: B=

 0.10000000E+04  0.50000000E+00  0.10000000E+01  0.50000000E+00

PARAMETER HIGHER BOUNDS: BH=

 0.10000000E+04  0.10000000E+01  0.10000000E+05  0.10000000E+01

PARAMETER INDEX:  1  2  3  4
REORDERED AS...:  2  3  4

REORDERED PARAMETERS:

 0.50000000E+00  0.10000000E+01  0.50000000E+00

** NLITR (IDER=0) OR NL2SNO (IDER=1) CALLED:  1 **


    I      INITIAL X(I)        D(I)

    1     0.785398E+00      0.164E+03
    2     0.100002E-01      0.140E+04
    3     0.785398E+00      0.184E+03

   IT    NF       F            DF          COSMAX       VAR

    0     1  0.225E+04                  0.100E+01
    1     2  0.773E+02  0.217E+04  0.994E+00  0.700E+01
    2     3  0.193E+01  0.753E+02  0.680E+00  0.693E+01
    3     5  0.106E+01  0.870E+00  0.189E+00  0.209E+01
    4     6  0.104E+01  0.200E-01  0.282E+00  0.305E+00
    5     7  0.100E+01  0.409E-01  0.129E+00  0.119E+00
    6    10  0.100E+01 -0.229E-02  0.129E+00  0.963E-04

***** VARIABILITY CONVERGENCE *****

FUNCTION     0.999781D+00  VARIABILITY   0.962956E-04
FUNC. EVALS       10       GRAD. EVALS        6
GRAD. NORM   0.245633E+02  COSMAX        0.128697E+00

    I       FINAL X(I)         D(I)           G(I)

    1     0.435899E+00      0.637E+02    -0.441E+00
    2     0.697591E-02      0.313E+03    -0.246E+02
    3     0.545296E+00      0.509E+02    -0.387E+00

COVARIANCE = SCALE * (J**T * J)**-1
```

```
ROW  1    0.9775E-01
ROW  2   -0.1172E-02  0.3099E-04
ROW  3   -0.1214E+00  0.1443E-02  0.1509E+00

  I     OBS.Y(I)       CAL            RES         %RES.ERR      X(I,1)         X(I,2)         X(I,3)         X(I,4)         WT(I)
  1  0.439000E+02  0.425394E+02  0.136E+01  0.319853E+01  0.450000E-01  0.439000E+02  0.000000E+00  0.000000E+00  0.227790E-01
  2  0.396000E+02  0.367316E+02  0.287E+01  0.780899E+01  0.750000E-01  0.396000E+02  0.000000E+00  0.000000E+00  0.252525E-01
  3  0.373000E+02  0.328760E+02  0.442E+01  0.134567E+02  0.105000E+00  0.373000E+02  0.000000E+00  0.000000E+00  0.268096E-01
  4  0.247000E+02  0.300053E+02 -0.531E+01 -0.176812E+02  0.135000E+00  0.247000E+02  0.000000E+00  0.000000E+00  0.404858E-01
  5  0.232000E+02  0.235612E+02 -0.361E+00 -0.153285E+01  0.240000E+00  0.232000E+02  0.000000E+00  0.000000E+00  0.431034E-01
  6  0.170000E+02  0.176599E+02 -0.660E+00 -0.373645E+01  0.420000E+00  0.170000E+02  0.000000E+00  0.000000E+00  0.588235E-01
  7  0.135000E+02  0.142097E+02 -0.710E+00 -0.499459E+01  0.600000E+00  0.135000E+02  0.000000E+00  0.000000E+00  0.740741E-01
  8  0.112000E+02  0.109662E+02  0.234E+00  0.213213E+01  0.870000E+00  0.112000E+02  0.000000E+00  0.000000E+00  0.892857E-01
  9  0.820000E+01  0.832813E+01 -0.128E+00 -0.153846E+01  0.123000E+01  0.820000E+01  0.000000E+00  0.000000E+00  0.121951E+00
 10  0.690000E+01  0.663794E+01  0.262E+00  0.394792E+01  0.159000E+01  0.690000E+01  0.000000E+00  0.000000E+00  0.144928E+00

** RMSERR=  0.29033587E+01

CORRELATION MATRIX
 2  0.1000E+01
 3 -0.6732E+00  0.1000E+01
 4 -0.9996E+00  0.6673E+00  0.1000E+01

  **PARM_SOL.   STD_ERROR   REL_ERROR    % ERROR **

 2  0.1783E+00  0.3126E+00  0.1754E+01  0.1754E+03
 3  0.4866E+00  0.5567E-02  0.1144E-01  0.1144E+01
 4  0.2690E+00  0.3884E+00  0.1444E+01  0.1444E+03


********  E N D  ********     SAUDI STA.35/B3: MODEL=1 [B]

PARAMETER NAME   FINAL SOLUTION

B( 1)  *  RO  =  0.10000000E+04
B( 2)     m   =  0.17827475E+00
B( 3)     Tau =  0.48662505E+00
B( 4)     c   =  0.26902008E+00

       * FIXED
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TOTAL "ELAPSED" TIME=            78.24 SEC. (    1 MIN. 18.24 SEC.)
CPU_TIME=          25.62 SEC. (    0 M. 25.62 S.)    CPU % = 32.75%
BUF.I/O_COUNT=       126
DIR.I/O_COUNT=         7
PAGE_FAULTS=         272
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```


SAUDI STA.35/B3: MODEL=1 [B]
TRANSIENT
10 2
10 1
10 0
10 -2
10 -1
10 0
10 1
TIME (SEC.)

Appendix 3.-- Source code availability and listing

Source Code Availability

The current version of the source code may be obtained by writing directly to the first author*, and enclosing a magnetic tape to be copied and returned. This method of releasing the source code was selected in order to satisfy requests for the latest (e.g., possibly updated) version. [The attached listing does not include the adaptive nonlinear least-squares algorithm (Dennis and others, 1979) due to its length; however, the complete algorithm is available on the distributed tape.]

Unless otherwise requested, the magnetic tape will be recorded in the following mode:

> Industry compatible: 9-track, standard ANSI-labeled, ASCII-mode, odd-parity, 800-bpi density, 80-character card-image records (blocked 50-card images, or 4000-characters, per physical block), and contained on a file named "NLSTIP.VAX".

---

* present address is:

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

Source Listing

The attached subprograms are listed in the following order:

```
00000010  [MAIN PROGRAM]
00000170  SUBROUTINE TIP_FCODE
00002010  REAL*8 FUNCTION FUN
00002160  SUBROUTINE FILTIP
00002670  SUBROUTINE TIP_SUBZ
00003770  SUBROUTINE TIP_SUBEND
00004590  SUBROUTINE DUMYPCODE
00004630  SUBROUTINE CPUTIME
00005270  REAL*8 FUNCTION DQSUBA
00006660  SUBROUTINE ERRMSG
00007000  SUBROUTINE NLSOL2
00013240  SUBROUTINE NLITR
00014300  SUBROUTINE INTRAN
00014890  SUBROUTINE CALCR
00015380  SUBROUTINE NONBLANK
00015510  SUBROUTINE PRENAM
00016090  SUBROUTINE PROCINFO
00016460  SUBROUTINE WARN
00016800  REAL FUNCTION ASINH
00016880  SUBROUTINE DQUAD
00020370  FUNCTION ERF
00020700  FUNCTION ERFINV
00021500  INTEGER FUNCTION LOC
00021610  SUBROUTINE NL2SOL
00026180  SUBROUTINE NL2SNO
00027730  SUBROUTINE NL2ITR
00034810  SUBROUTINE ASSESS
00038810  SUBROUTINE COVCLC
00042970  SUBROUTINE DFAULT
00043860  REAL FUNCTION DOTPRD
00044230  SUBROUTINE DUPDAT
00044810  SUBROUTINE GQTSTP
00050730  SUBROUTINE ITSMRY
00053030  SUBROUTINE LINVRT
00053460  SUBROUTINE LITVMU
00053780  SUBROUTINE LIVMUL
00054090  SUBROUTINE LMSTEP
00059200  SUBROUTINE LSQRT
00059850  REAL FUNCTION LSVMIN
00061640  SUBROUTINE LTSQAR
00062000  SUBROUTINE PARCHK
00063920  SUBROUTINE QAPPLY
00064820  SUBROUTINE QRFACT
00067210  SUBROUTINE RPTMUL
00067960  SUBROUTINE SLUPDT
00068580  SUBROUTINE SLVMUL
00069040  LOGICAL FUNCTION STOPX
```

```
00069270  SUBROUTINE VAXPY
00069400  SUBROUTINE VCOPY
00069530  SUBROUTINE VSCOPY
00069660  REAL FUNCTION V2NORM
00070210  INTEGER FUNCTION IMDCON
00070380  REAL FUNCTION RMDCON
00071420  FUNCTION TCHEB
```

```
C <NLSTIP>: COLE-COLE INVERSE TRANSIENT IP SOUNDINGS     <1/24/84>      00000010
C  USING $INIT MODEL>=1 & <5 (IOPT=0; MODEL>1 & IOPT=+-1 NOT AVAIL.);    00000020
C               NP=0 FOR SINGLE STEP RESPONSE (ON-OFF)                  00000030
C                 >1 FOR BIPOLAR WAVEFORM WITH NP-PERIODS AND           00000040
C                    TP(1)>0 IS 4*NP-1 STEPS PER NP, ETC.               00000050
C  (SEE DOCUMENTATION OF NLSTIP FOR DETAILED $INIT DEFINITIONS.)         00000060
C                                                                        00000070
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO, USA.        00000080
C                                                                        00000090
C                                                                        00000100
      EXTERNAL TIP_FCODE,TIP_SUBZ,DUMYPCODE,TIP_SUBEND                   00000110
      CALL SETTIME                                                       00000120
      CALL NLSOL2(TIP_FCODE,DUMYPCODE,TIP_SUBZ,TIP_SUBEND)               00000130
      CALL CPUTIME(6,16)                                                 00000140
      CALL EXIT                                                          00000150
      END                                                                00000160
      SUBROUTINE TIP_FCODE(Y,X,B,PRNT,F,IN,IDER)                         00000170
C--FUNCTION EVALUATION FOR 'FWDTIP' OR 'NLSTIP' (SEE NOTES)              00000180
C                                                                        00000190
C--PARAMETERS--                                                          00000200
C                                                                        00000210
C       Y=       OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)              00000220
C       X=       OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)          00000230
C       B=       CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)              00000240
C       PRNT=    WORK AND PRINT ARRAY (DIM. 5)                           00000250
C       F=       OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. IN  00000260
C       IN=      OBSERVATION NO. TO EVAL. F (1<=IN<=N)                   00000270
C       IDER=    0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED) 00000280
C                1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED) 00000290
C  (NOTE THAT IDER=1 REQUIRED FOR THIS VERSION.)                         00000300
C                                                                        00000310
      CHARACTER*3 ICHRS                                                  00000320
      REAL Y(1),X(500,5),B(1),PRNT(5),                                   00000330
     1 BSAVE(20),W1(200),W2(200)                                         00000340
      REAL*8 DQSUBA,RAT,RELERR,SINA,COSA,A,PI,DEPS,X0                    00000350
      COMMON/PASS/RAT,SINA,COSA                                          00000360
      EXTERNAL FUN                                                       00000370
      COMMON/TIP/DEPS,BB(13),TMIN,TMAX,TP(2),                            00000380
     1 MPARM,MODEL,ISTEP,IFIRST,NN,MAXPTS,NP,IOPT,IM                     00000390
      COMMON/TCOM/T(500),VSAVE(500)                                      00000400
      DATA PI/3.141592653589793D0/                                       00000410
20    DO 30 J=1,5                                                        00000420
30    PRNT(J)=X(IN,J)                                                    00000430
      IF(IM.GT.1) GO TO 900   !!SPECIAL CASE IM>1 AT 900                 00000440
      IF(IN.GT.1) GO TO 800                                              00000450
      IF(IDER.EQ.1) GO TO 8001                                           00000460
```

```
35      DO J=1,MPARM                                                    00000470
          BB(J)=B(J)                                                    00000480
        ENDDO                                                           00000490
        A=PI*B(4)                                                       00000500
        COSA=DCOS(A)                                                    00000510
        SINA=DSIN(A)                                                    00000520
C--GET TRANSIENT V(T) FOR ALL T IN [TMIN,TMAX] BACKWARD FROM TMAX       00000530
        IWARN=0                                                         00000540
        IEND=1                                                          00000550
36        DO 70 I=NN,IEND,-1                                            00000560
          T(I)=X(I,1)                                                   00000570
        IF(IOPT.LE.0.AND.IEND.EQ.1) CALL FILTIP(T(I),TRANS,*71)         00000580
        IF(IWARN.EQ.1) GO TO 71                                         00000590
        RAT=T(I)/B(3)                                                   00000600
        X0=87.D0/RAT                                                    00000610
C--NOW GET TRANSIENT INTEGRAL FROM 0 TO X0                              00000620
          TRANS=.31830989*DQSUBA(0.0D0,X0,DEPS,NPTS,ITEM,               00000630
     1    RELERR,FUN,MAXPTS)                                            00000640
          IF(NPTS.GT.MAXPTS) THEN                                       00000650
            ENCODE(3,69,ICHRS) I                                        00000660
69          FORMAT(I3)                                                  00000670
            IWARN=1                                                     00000680
            CALL WARN('NPTS>MAXPTS AT I=1 TO'//ICHRS,0,6,16,*71)        00000690
          ENDIF                                                         00000700
71        VSAVE(I)=TRANS                                                00000710
70      CONTINUE                                                        00000720
C--CHECK IF NP>0 PERIODS SELECTED                                       00000730
        IF(NP.GT.0) THEN  !! SEE ENDIF BELOW                            00000740
          TSUM=0.0                                                      00000750
          DO INP=1,NP     !! SEE ENDDO BELOW                            00000760
            DO J=2,1,-1                                                 00000770
              TSUM=TSUM+TP(J)                                           00000780
              IWARN=0                                                   00000790
              DO I=NN,IEND,-1                                           00000800
               IF(IWARN.EQ.0) THEN                                      00000810
                TI=X(I,1)+TSUM                                          00000820
                IF(IOPT.LE.0.AND.IEND.EQ.1) CALL FILTIP(TI,TRANS,*72)   00000830
                RAT=TI/B(3)                                             00000840
                X0=87.D0/RAT                                            00000850
                TRANS=.31830989*DQSUBA(0.0D0,X0,DEPS,NPTS,ITEM,         00000860
     1           RELERR,FUN,MAXPTS)                                     00000870
                IF(NPTS.GT.MAXPTS) IWARN=1                              00000880
               ENDIF                                                    00000890
72              VSAVE(I)=VSAVE(I)-TRANS                                 00000900
              ENDDO                                                     00000910
            ENDDO                                                       00000920
              TSUM=TSUM+TP(2)                                           00000930
              IWARN=0                                                   00000940
              DO I=NN,IEND,-1                                           00000950
               IF(IWARN.EQ.0) THEN                                      00000960
                TI=X(I,1)+TSUM                                          00000970
                IF(IOPT.LE.0.AND.IEND.EQ.1) CALL FILTIP(TI,TRANS,*73)   00000980
                RAT=TI/B(3)                                             00000990
                X0=87.D0/RAT                                            00001000
                TRANS=.31830989*DQSUBA(0.0D0,X0,DEPS,NPTS,ITEM,         00001010
```

```
     1          RELERR,FUN,MAXPTS)                                          00001020
               IF(NPTS.GT.MAXPTS) IWARN=1                                   00001030
              ENDIF                                                         00001040
73             VSAVE(I)=VSAVE(I)+TRANS                                      00001050
            ENDDO                                                           00001060
          IF(INP.LT.NP) THEN                                                00001070
            TSUM=TSUM+TP(1)                                                 00001080
            IWARN=0                                                         00001090
            DO I=NN,IEND,-1                                                 00001100
             IF(IWARN.EQ.0) THEN                                            00001110
              TI=X(I,1)+TSUM                                                00001120
              IF(IOPT.LE.0.AND.IEND.EQ.1) CALL FILTIP(TI,TRANS,*74)         00001130
              RAT=TI/B(3)                                                   00001140
              X0=98.D0/RAT                                                  00001150
              TRANS=.31830989*DQSUBA(0.0D0,X0,DEPS,NPTS,ITEM,               00001160
     1          RELERR,FUN,MAXPTS)                                          00001170
              IF(NPTS.GT.MAXPTS) IWARN=1                                    00001180
             ENDIF                                                          00001190
74           VSAVE(I)=VSAVE(I)+TRANS                                        00001200
            ENDDO                                                           00001210
          ENDIF                                                             00001220
        ENDDO !! GET NEXT INP                                               00001230
       ENDIF   !! END OF "IF(NP.GT.0) THEN" ABOVE                           00001240
       IF(IOPT.EQ.-1.AND.IEND.EQ.1) THEN                                    00001250
         DO I=1,NN                                                          00001260
           IF(VSAVE(I).LE.1.E-6) THEN                                       00001270
             IEND=I                                                         00001280
             IF(IEND.GT.1) GO TO 36                                         00001290
           ENDIF                                                            00001300
         ENDDO                                                              00001310
       ENDIF                                                                00001320
         DO I=2,NN                                                          00001330
           IF(VSAVE(I).EQ.0.0) THEN                                         00001340
             VSAVE(I)=VSAVE(I-1)                                            00001350
           ENDIF                                                            00001360
         ENDDO                                                              00001370
       IF(ISTEP.EQ.1) THEN                                                  00001380
C--GET REVERSE STEP ONLY IF ISTEP=1                                         00001390
         DO I=1,NN                                                          00001400
           VSAVE(I)=ABS(B(1)-VSAVE(I))                                      00001410
         ENDDO                                                              00001420
       ENDIF                                                                00001430
       IF(IDER.EQ.0) GO TO 600                                              00001440
       IFIRST=0                                                             00001450
       DO 80 J=1,MPARM                                                      00001460
80     BSAVE(J)=B(J)                                                        00001470
C--GET PRE-COMPUTED TRANSIENT                                               00001480
600    F=VSAVE(IN)                                                          00001490
       RETURN                                                               00001500
800    IF(IDER.EQ.0) GO TO 600                                              00001510
C--IDER=1 EST.DER.OPTION                                                    00001520
8001   IF(IFIRST.EQ.1) GO TO 35                                             00001530
       DO 802 J=1,MPARM                                                     00001540
         IF(B(J).NE.BSAVE(J)) GO TO 35                                      00001550
802    CONTINUE                                                             00001560
```

```
      GO TO 600                                                          00001570
C                                                                        00001580
C--SPEICAL CASE IM>1; DO ONLY INPUT INDEX "IN" AND IOPT=0 FOR NOW.       00001590
C                                                                        00001600
900   DO J=1,MPARM                                                       00001610
        BB(J)=B(J)                                                       00001620
      ENDDO                                                              00001630
      A=PI*B(4)                                                          00001640
      COSA=DCOS(A)                                                       00001650
      SINA=DSIN(A)                                                       00001660
      TP(1)=PRNT(2)                                                      00001670
      IF(IM.EQ.3) NP=IFIX(PRNT(3))                                       00001680
      IF(IM.EQ.4) THEN                                                   00001690
        TP(2)=PRNT(4)                                                    00001700
        IF(TP(2).LE.0.0) TP(2)=TP(1)                                     00001710
      ELSE                                                               00001720
        TP(2)=TP(1)                                                      00001730
      ENDIF                                                              00001740
      CALL FILTIP(X(IN,1),F,*910)                                        00001750
910   IF(NP.GT.0) THEN  !!! SEE ENDIF BELOW                              00001760
        TSUM=0.0                                                         00001770
        DO INP=1,NP      !!! SEE ENDDO BELOW                             00001780
          DO J=2,1,-1                                                    00001790
            TSUM=TSUM+TP(J)                                              00001800
            TI=X(IN,1)+TSUM                                              00001810
            CALL FILTIP(TI,TRANS,*920)                                   00001820
920         F=F-TRANS                                                    00001830
          ENDDO                                                          00001840
          TSUM=TSUM+TP(2)                                                00001850
          TI=X(IN,1)+TSUM                                                00001860
          CALL FILTIP(TI,TRANS,*930)                                     00001870
930       F=F+TRANS                                                      00001880
         IF(INP.LT.NP) THEN                                              00001890
           TSUM=TSUM+TP(1)                                               00001900
           TI=X(IN,1)+TSUM                                               00001910
           CALL FILTIP(TI,TRANS,*940)                                    00001920
940        F=F+TRANS                                                     00001930
         ENDIF                                                           00001940
        ENDDO   !!! GET NEXT INP                                         00001950
      ENDIF     !!! END OF "IF(NP.GT.0) THEN" ABOVE                      00001960
      IF(ISTEP.EQ.1) F=ABS(B(1)-F)                                       00001970
      VSAVE(IN)=F                                                        00001980
      RETURN                                                             00001990
      END                                                                00002000
      REAL*8 FUNCTION FUN(X)                                             00002010
C--INTEGRAND FOR TRANSIENT INTEGRAL FROM 0 TO X0, WHERE                  00002020
C  X0 IS UPPER BOUND DETERMINED AS X0=87.D0/RAT, RAT=T(I)/TAU            00002030
C  (USED ONLY IF IOPT=1)                                                 00002040
C                                                                        00002050
      IMPLICIT REAL*8 (A-H,O-Z)                                          00002060
      REAL*4 B,TMIN,TMAX,TP                                              00002070
      COMMON/PASS/RAT,SINA,COSA                                          00002080
      COMMON/TIP/DEPS,B(13),TMIN,TMAX,TP(2),                             00002090
     1 MPARM,MODEL,ISTEP,IFIRST,NN,MAXPTS,NP,IOPT,IM                     00002100
      XC=X**B(4)                                                         00002110
```

```
      FUN=B(1)*B(2)*XC*SINA*DEXP(-X*RAT)/                                00002120
     1 ( X*((1.D0+XC*COSA)**2+(XC*SINA)**2) )                            00002130
      RETURN                                                             00002140
      END                                                                00002150
      SUBROUTINE FILTIP(T,V,*)                                           00002160
C--FILTER SOLUTION FOR V(T) FOR COLE-COLE IP USING REF 1                 00002170
C  (FILTIP USED ONLY IF IOPT=0 OR -1).                                   00002180
C                                                                        00002190
C  REF 1: GUPTASARMA, D., 1982, GEOPHYSICS, V.47, N.11, P.1574-1576      00002200
C                                                                        00002210
C--PARMS:                                                                00002220
C     T=TIME IN SECONDS >0.0                                             00002230
C     V=OUTPUT V(T) VIA FILTER                                           00002240
C     *=OUTPUT LABEL RETURN (MUST BE GIVEN AS USED BELOW)                00002250
C                                                                        00002260
C--E.G., IF(IOPT.EQ.0) CALL FILTIP(T(I),TRANS,*71)                       00002270
C                                                                        00002280
      COMPLEX Z,Z1,ZF,ONE                                                00002290
      DIMENSION A(21),W(21)                                              00002300
      REAL*8 DEPS                                                        00002310
      COMMON/TIP/DEPS,B(13),TMIN,TMAX,TP(2),                             00002320
     1 MPARM,MODEL,ISTEP,IFIRST,NN,MAXPTS,NP,IOPT,IM                     00002330
      DATA ONE/(1.0,0.0)/                                                00002340
      DATA PI2/1.570796327/                                              00002350
      DATA A/-3.82704,-3.56608,-3.30512,-3.04416,-2.7832,-2.52224,       00002360
     1 -2.26128,-2.00032,-1.73936,-1.4784,-1.21744,-.95648,              00002370
     2 -.69552,-.43456,-.1736,.08736,.34832,.60928,.87024,               00002380
     3 1.1312,1.39216/                                                   00002390
      DATA W/.349998E-3,-.418371E-3,.772828E-3,-.171356E-3,              00002400
     1 .1022172E-2,.897638E-3,.2208974E-2,.3844944E-2,                   00002410
     2 .680904E-2,.013029162,.022661391,.042972904,.075423603,           00002420
     3 .139346367,.234486236,.366178323,.284615486,-.235691746,          00002430
     4 .046994188,-.5901946E-2,.570165E-3/                               00002440
C--FILTER SOLUTION FOR ANY T (REF 1)                                     00002450
      V=0.0                                                              00002460
      DO I=1,21   !! SEE ENDDO BELOW                                     00002470
        OMEGA=10.**(A(I)-ALOG10(T))                                      00002480
        Z=(OMEGA*B(3))**B(4)                                             00002490
        Z1=Z*(CEXP(CMPLX(0.0,B(4)*PI2)))                                 00002500
        ZF=ONE-CMPLX(B(2),0.0)*(ONE-ONE/(ONE+Z1))                        00002510
        IF(MODEL.EQ.1) GO TO 30                                          00002520
        DO IMODEL=2,MODEL                                                00002530
          I1=3*IMODEL-1                                                  00002540
          I2=I1+1                                                        00002550
          I3=I1+2                                                        00002560
          Z=(OMEGA*B(I2))**B(I3)                                         00002570
          Z1=Z*(CEXP(CMPLX(0.0,B(I3)*PI2)))                              00002580
          ZF=ZF*(ONE-CMPLX(B(I1),0.0)*(ONE-ONE/(ONE+Z1)))                00002590
        ENDDO                                                            00002600
30      ZF=CMPLX(B(1),0.0)*ZF                                            00002610
        V=V+REAL(ZF)*W(I)                                                00002620
      ENDDO    !! GET NEXT I                                           00002630
      V=B(1)-V                                                           00002640
      RETURN 1                                                           00002650
      END                                                                00002660
```

```
      SUBROUTINE TIP_SUBZ(Y,X,B,PRNT,NPRINT,N,TITLE,IOUT)                00002670
C-- INITIALIZATION ROUTINE (CALLED ONCE) FOR FWDTIP                      00002680
C                                                                       00002690
C  SUBZ IS CALLED AFTER THE DATA Y(I),X(I,5) ARE READ.                  00002700
C  SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL $INIT                   00002710
C  PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE...            00002720
C                                                                       00002730
C--PARAMETERS--                                                         00002740
C       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.                         00002750
C       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY             00002760
C               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES              00002770
C       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)                   00002780
C       TITLE=  ALPHA TITLE ARRAY READ IN BY PGM IMSLMQ.                00002790
C       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED                     00002800
C               0 IF ONLY UNIT 6 PRINT FILE USED.                       00002810
C                                                                       00002820
      CHARACTER*80 TITLE                                                00002830
      CHARACTER*4 PARNAM(13)                                            00002840
      REAL*8 DEPS                                                       00002850
      REAL   Y(1),X(500,5),B(1),PRNT(1), TPSAVE(2)                      00002860
      NAMELIST/INIT/MODEL,ISTEP,MAXPTS,EPS,TP,NP,REC,IM,IOPT            00002870
      COMMON/TIP/DEPS,BB(13),TMIN,TMAX,TP(2),                           00002880
     1 MPARM,MODEL,ISTEP,IFIRST,NN,MAXPTS,NP,IOPT,IM                    00002890
      DATA ISUBZ/0/,PARNAM/'R0','M','TAU','C','M2','TAU2','C2',         00002900
     1 'M3','TAU3','C3','M4','TAU4','C4'/                               00002910
      IF(ISUBZ.NE.0) THEN                                               00002920
        TP(1)=TPSAVE(1)                                                 00002930
        TP(2)=TPSAVE(2)                                                 00002940
        NP=NPSAVE                                                       00002950
        GO TO 10                                                        00002960
      ENDIF                                                             00002970
C--PRESET                                                               00002980
      ISUBZ=1                                                           00002990
      MODEL=1                                                           00003000
      MAXPTS=5000                                                       00003010
      ISTEP=0                                                           00003020
      EPS=.1E-2                                                         00003030
      NP=0                                                              00003040
      TP(1)=0.0                                                         00003050
      TP(2)=0.0                                                         00003060
      REC=0.0                                                           00003070
      IM=1                                                              00003080
      IOPT=0                                                            00003090
   10 READ(99,INIT)                                                     00003100
      CALL NONBLANK(TITLE,NONBLK)                                       00003110
      WRITE(6,20) TITLE                                                 00003120
   20 FORMAT('1<NLSTIP>:',5X,A<NONBLK>/)                                00003130
      IF(IOUT.EQ.1) WRITE(16,20) TITLE                                  00003140
      WRITE(6,INIT)                                                     00003150
      IF(IOUT.EQ.1) WRITE(16,INIT)                                      00003160
C--TEST $INIT PARMS                                                     00003170
      IF(MODEL.LT.1.OR.MODEL.GT.4)                                      00003180
     1 CALL ERRMSG('MODEL<1 OR >4',0,6,16)                              00003190
      IF(IOPT.LT.-1.OR.IOPT.GT.1)CALL ERRMSG(                           00003200
     1 'IOPT<-1 OR >1',0,6,16)                                          00003210
```

```
      IF(MODEL.GT.1.AND.IOPT.NE.0)CALL ERRMSG(                          00003220
     1 'MODEL>1 & IOPT=1 OR -1 NOT AVAILABLE.',0,6,16)                  00003230
      IF(IM.EQ.1.AND.NP.GT.0.AND.TP(1).LE.0.0)CALL ERRMSG(              00003240
     1 'IM=1 AND NP>0 BUT TP(1)<=0?',0,6,16)                            00003250
      IF(IM.EQ.2.AND.NP.LE.0)CALL ERRMSG(                               00003260
     1 'IM=2 AND NP<=0?',0,6,16)                                        00003270
      IF(ISTEP.LT.0.OR.ISTEP.GT.1)CALL ERRMSG(                          00003280
     1 'ISTEP<0 OR >1',0,6,16)                                          00003290
      IF(IM.LT.1.OR.IM.GT.4)CALL ERRMSG(                                00003300
     1 'IM<1 OR >4',0,6,16)                                             00003310
      IF(IM.GT.1.AND.IOPT.NE.0)CALL ERRMSG(                             00003320
     1 'IM>1 AND IOPT NOT 0',0,6,16)                                    00003330
      MPARM=3*MODEL+1                                                   00003340
      DO I=1,MPARM                                                      00003350
        IF(B(I).EQ.0.0)CALL ERRMSG(                                     00003360
     1 'SOME B(I)=0 FOR I=1,3*MODEL+1',0,6,16)                          00003370
      ENDDO                                                             00003380
C--TEST X(I, ) DATA FOR GIVEN IM BEFORE PROCEEDING--                    00003390
      IF(IM.EQ.1) THEN                                                  00003400
      DO 40 I=2,N                                                       00003410
      IF(X(I,1).LE.X(I-1,1).OR.X(I,1).LE.0.0)CALL ERRMSG(               00003420
     & 'IM=1 AND SOME X(I,1)<=0.0 OR NOT INCREASING.',7,6,16)           00003430
40    CONTINUE                                                          00003440
      ENDIF                                                             00003450
      IF(IM.GT.1) THEN                                                  00003460
       DO I=1,N                                                         00003470
        IF(X(I,1).LE.0.0)CALL ERRMSG(                                   00003480
     1  'IM>1 AND SOME X(I,1)<=0.0?',0,6,16)                           00003490
        IF(X(I,2).LE.0.0)CALL ERRMSG(                                   00003500
     1  'IM>1 AND SOME X(I,2)=TP(1)<=0',0,6,16)                        00003510
        IF(IM.EQ.3.AND.X(I,3).LE.0.0)CALL ERRMSG(                       00003520
     1  'IM=3 AND SOME X(I,3)=NP<=0',0,6,16)                           00003530
       ENDDO                                                            00003540
      ENDIF                                                             00003550
C--PRESET SOME GLOBAL CONSTANTS                                         00003560
      IFIRST=1                                                          00003570
      DEPS=EPS                                                          00003580
      IF(TP(1).GT.0.0.AND.TP(2).LE.0.0) TP(2)=TP(1)                     00003590
      TPSAVE(1)=TP(1)                                                   00003600
      TPSAVE(2)=TP(2)                                                   00003610
      NPSAVE=NP                                                         00003620
      NN=N                                                              00003630
      TMIN=X(1,1)                                                       00003640
      TMAX=X(N,1)                                                       00003650
C--PRINT PARAMETER NAMES AND ORDER                                      00003660
      WRITE(6,50)                                                       00003670
50    FORMAT(////' PARAMETER ORDER & NAMES--'/)                         00003680
      IF(IOUT.EQ.1) WRITE(16,50)                                        00003690
      DO I=1,MPARM                                                      00003700
        WRITE(6,60) I,I,PARNAM(I)                                       00003710
60      FORMAT(5X,I3,5X,'B(',I2,')=',4X,A4)                             00003720
        IF(IOUT.EQ.1) WRITE(16,60) I,I,PARNAM(I)                        00003730
      ENDDO                                                             00003740
      RETURN                                                            00003750
      END                                                               00003760
```

```
      SUBROUTINE TIP_SUBEND(Y,X,B,K,N,TITLE,IOUT)                    00003770
C** SUBEND TERMINATION ROUTINE                                       00003780
C  ALSO GIVES RESTART $PARMS ON UNIT=4 AS 'FOR005.TMP'               00003790
C                                                                    00003800
      CHARACTER*132 LINE                                             00003810
      CHARACTER*80 TITLE                                             00003820
      CHARACTER*4 PARNAM(13)                                         00003830
      DATA PARNAM/'R0','M','TAU','C','M2','TAU2','C2',               00003840
     1 'M3','TAU3','C3','M4','TAU4','C4'/                           00003850
      CHARACTER*1 FLAG(19)                                           00003860
      COMMON/FIXDAT/DUM(3020),IB(19),IP,IDUM(3)                      00003870
      REAL Y(1),X(500,5),B(1)                                        00003880
      DO I=1,K                                                       00003890
        FLAG(I)=' '                                                  00003900
        IF(IP.GT.0) THEN                                             00003910
          DO J=1,IP                                                  00003920
            IF(IB(J).NE.0) FLAG(IB(J))='*'                           00003930
          ENDDO                                                      00003940
        ENDIF                                                        00003950
      ENDDO                                                          00003960
      CALL NONBLANK(TITLE,NB)                                        00003970
      WRITE(6,10) TITLE                                              00003980
10    FORMAT(//' ********  E N D  ********',5X,A<NB>//               00003990
     1 ' PARAMETER NAME',3X,'FINAL SOLUTION'/)                       00004000
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                               00004010
      DO 30 I=1,K                                                    00004020
      WRITE(6,20) I,FLAG(I),PARNAM(I),B(I)                           00004030
20    FORMAT(1X,'B(',I2,')',2X,A1,2X,A,'=',E16.8)                    00004040
      IF(IOUT.EQ.1) WRITE(16,20) I,FLAG(I),PARNAM(I),B(I)            00004050
30    CONTINUE                                                       00004060
C** GENERATE RESTART $PARMS ON FOR005.TMP                            00004070
60    REWIND 5                                                       00004080
      OPEN(UNIT=4,FILE='FOR005.TMP',STATUS='NEW',                    00004090
     1 CARRIAGECONTROL='LIST')                                       00004100
      READ(5,65,END=999) LINE                                        00004110
65    FORMAT(A)                                                      00004120
      CALL NONBLANK(LINE,NB)                                         00004130
      WRITE(4,66) LINE                                               00004140
66    FORMAT(A<NB>)                                                  00004150
      IDOL=0                                                         00004160
70    READ(5,65,END=999) LINE                                        00004170
      I=INDEX(LINE,'$')                                              00004180
      IF(I.NE.0) THEN                                                00004190
        IF(IDOL.EQ.0) THEN                                           00004200
          IDOL=1                                                     00004210
          J=INDEX(LINE(I+1:),'$')                                    00004220
          IF(J.NE.0) THEN                                            00004230
            IDOL=2                                                   00004240
            LINE(J:J)=','                                            00004250
          ENDIF                                                      00004260
        ELSE                                                         00004270
          IDOL=2                                                     00004280
          LINE(I:I)=','                                              00004290
        ENDIF                                                        00004300
      ENDIF                                                          00004310
```

```
      CALL NONBLANK(LINE,NB)                                          00004320
      WRITE(4,66) LINE                                                00004330
      IF(IDOL.LT.2) GO TO 70                                          00004340
      LINE(1:)='B='                                                   00004350
      DO 80 I=1,K                                                     00004360
      ENCODE(16,90,LINE(3:18)) B(I)                                   00004370
90    FORMAT(G16.8)                                                   00004380
      IF(I.LT.K) THEN                                                 00004390
        LINE(19:19)=','                                               00004400
      ELSE                                                            00004410
        LINE(19:19)='$'                                               00004420
      ENDIF                                                           00004430
      CALL NONBLANK(LINE,NB)                                          00004440
      WRITE(4,66) LINE                                                00004450
      LINE(1:2)='  '                                                  00004460
80    CONTINUE                                                        00004470
100   READ(5,65,END=999) LINE                                         00004480
      CALL NONBLANK(LINE,NB)                                          00004490
      WRITE(4,66) LINE                                                00004500
      GO TO 100                                                       00004510
999   IF(IP.GT.0) THEN                                                00004520
        WRITE(6,110)                                                  00004530
110     FORMAT(/8X,'* FIXED')                                         00004540
        IF(IOUT.EQ.1) WRITE(16,110)                                   00004550
      ENDIF                                                           00004560
      RETURN                                                          00004570
      END                                                             00004580
      SUBROUTINE DUMYPCODE()                                          00004590
C--DUMMY PCODE FOR USE IN 'MARQRT' OR 'NLSOL'                         00004600
      CALL ERRMSG('IDER=0 NOT AVAILABLE IN THIS VERSION.',4,6,16)     00004610
      END                                                             00004620
      SUBROUTINE CPUTIME(I1,I2)                                       00004630
C                                                                     00004640
C  CPUTIME WRITES "ELAPSED & CPU" TIME FROM PREVIOUS "CALL SETTIME" ON 00004650
C  FORTRAN UNITS I1 (IF NOT 0) AND I2 (IF NOT 0).                     00004660
C                                                                     00004670
C  WILL EJECT FIRST IF I1>0 (OR I2>0).                                00004680
C  DOUBLE SPACE FIRST IF I1<0 (OR I2<0).                              00004690
C                                                                     00004700
C  E.G., USE TO TIME ELAPSED & CPU TIME FOR PROGRAM OR CODE SEGMENTS AS:00004710
C                                                                     00004720
C     CALL SETTIME    ! DON'T FORGET TO DO THIS!                      00004730
C     >>>>> THE CODE TO TIME IS HERE <<<<<  ! USUALLY A COMPLETE PROGRAM00004740
C     CALL CPUTIME(-6,16) ! OR USE I1 OR I2=0 TO OMIT WRITE.          00004750
C     >>>>> ALSO CAN USE CALL GETTIME(CPU) TO GET JUST THE CPU (SEC)  00004760
C           SINCE THE LAST CALL SETTIME WAS DONE.                     00004770
C                                                                     00004780
      SAVE                                                            00004790
      INTEGER*4 ABSVAL(4),INCRVAL(4)                                  00004800
      CALL PROCINFO(ABSVAL,INCRVAL)                                   00004810
      TIMES=SECNDS(TIME0)                                             00004820
      MIN=TIMES/60.0                                                  00004830
      SEC=AMOD(TIMES,60.0)                                            00004840
      CPUSEC=INCRVAL(1)*.01                                           00004850
      IMIN=CPUSEC/60.0                                                00004860
```

```
      CSEC=AMOD(CPUSEC,60.0)                                            00004870
      PCPU=100.*(CPUSEC/TIMES)                                          00004880
      IF(I1.NE.0) THEN                                                  00004890
        IF(I1.GT.0) THEN                                                00004900
          J=1                                                           00004910
        ELSE                                                            00004920
          J=0                                                           00004930
        ENDIF                                                           00004940
      WRITE(IABS(I1),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,         00004950
     1 (INCRVAL(I),I=2,4)                                               00004960
60    FORMAT(I1,65('$')/' TOTAL "ELAPSED" TIME=',F16.2,' SEC. (',       00004970
     1 I4,' MIN.',F6.2,' SEC.)'/                                        00004980
     2 ' CPU_TIME=',F15.2,' SEC. (',I4,' M. ',F5.2,                     00004990
     1 ' S.)   CPU % =',F6.2,'%'/                                       00005000
     3 ' BUF.I/O_COUNT=',I10/                                           00005010
     4 ' DIR.I/O_COUNT=',I10/                                           00005020
     5 ' PAGE_FAULTS=',2X,I10/                                          00005030
     6 ' ',65('$')//)                                                   00005040
      ENDIF                                                             00005050
      IF(I2.NE.0) THEN                                                  00005060
        IF(I2.GT.0) THEN                                                00005070
          J=1                                                           00005080
        ELSE                                                            00005090
          J=0                                                           00005100
        ENDIF                                                           00005110
      WRITE(IABS(I2),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,         00005120
     1 (INCRVAL(I),I=2,4)                                               00005130
      ENDIF                                                             00005140
      RETURN                                                            00005150
C** ENTRY 'CALL SETTIME'--MUST BE DONE BEFORE 'CALL CPUTIME(I1,I2)'     00005160
      ENTRY SETTIME()                                                   00005170
      TIME0=SECNDS(0.0)                                                 00005180
      CALL PROCINFO(ABSVAL,INCRVAL)                                     00005190
      RETURN                                                            00005200
C** ENTRY 'CALL GETTIME(CPU)'--TO GET CPU(SEC) SINCE LAST CALL SETTIME  00005210
      ENTRY GETTIME(CPU)                                                00005220
      CALL PROCINFO(ABSVAL,INCRVAL)                                     00005230
      CPU=INCRVAL(1)*.01                                                00005240
      RETURN                                                            00005250
      END                                                               00005260
      REAL*8 FUNCTION DQSUBA(A, B, EPSIL, NPTS, ICHECK, RELERR, F,MEV)  00005270
C--DOUBLE-PRECISION VERSION BY W.L.ANDERSON-- 9/21/81.                  00005280
      IMPLICIT REAL*8 (A-H,O-Z)                                         00005290
C   THIS FUNCTION ROUTINE PERFORMS AUTOMATIC INTEGRATION                00005300
C OVER A FINITE INTERVAL USING THE BASIC INTEGRATION                    00005310
C ALGORITHM QUAD TOGETHER WITH,  IF NECESSARY AN ADAPTIVE               00005320
C SUBDIVISION PROCESS.  IT IS GENERALLY MORE EFFICIENT THAN             00005330
C THE NON-ADAPTIVE ALGORITHM QSUB BUT IS LIKELY TO BE LESS              00005340
C RELIABLE(SEE COMP.J.,14,189,1971).                                    00005350
C   THE CALL TAKES THE FORM                                             00005360
C           DQSUBA(A,B,EPSIL,NPTS,ICHECK,RELERR,F,MEV)                  00005370
C AND CAUSES F(X) TO BE INTEGRATED OVER (A,B) WITH RELATIVE             00005380
C ERROR HOPEFULLY NOT EXCEEDING EPSIL.  SHOULD QUAD CONVERGE            00005390
C (ICHECK=0) THEN DQSUBA WILL RETURN THE VALUE OBTAINED BY IT           00005400
C OTHERWISE SUBDIVISION WILL BE INVOKED AS A RESCUE                     00005410
```

```
C OPERATION IN AN ADAPTIVE MANNER. THE ARGUMENT RELERR GIVES          00005420
C A CRUDE ESTIMATE OF THE ACTUAL RELATIVE ERROR OBTAINED.             00005430
C   THE SUBDIVISION STRATEGY IS AS FOLLOWS                            00005440
C AT EACH STAGE OF THE PROCESS AN INTERVAL IS PRESENTED FOR           00005450
C SUBDIVISION (INITIALLY THIS WILL BE THE WHOLE INTERVAL              00005460
C (A,B)). THE INTERVAL IS HALVED AND QUAD APPLIED TO EACH             00005470
C SUBINTERVAL. SHOULD QUAD FAIL ON THE FIRST SUBINTERVAL              00005480
C THE SUBINTERVAL IS STACKED FOR FUTURE SUBDIVISION AND THE           00005490
C SECOND SUBINTERVAL IMMEDIATELY EXAMINED. SHOULD QUAD FAIL           00005500
C ON THE SECOND SUBINTERVAL THE SUBINTERVAL IS                        00005510
C IMMEDIATELY SUBDIVIDED AND THE WHOLE PROCESS REPEATED.              00005520
C EACH TIME A CONVERGED RESULT IS OBTAINED IT IS                      00005530
C ACCUMULATED AS THE PARTIAL VALUE OF THE INTEGRAL. WHEN              00005540
C QUAD CONVERGES ON BOTH SUBINTERVALS THE INTERVAL LAST               00005550
C STACKED IS CHOSEN NEXT FOR SUBDIVISION AND THE PROCESS              00005560
C REPEATED. A SUBINTERVAL IS NOT EXAMINED AGAIN ONCE A                00005570
C CONVERGED RESULT IS OBTAINED FOR IT SO THAT A SPURIOUS              00005580
C CONVERGENCE IS MORE LIKELY TO SLIP THROUGH THAN FOR THE             00005590
C NON-ADAPTIVE ALGORITHM QSUB.                                        00005600
C   THE CONVERGENCE CRITERION OF QUAD IS SLIGHTLY RELAXED             00005610
C IN THAT A PANEL IS DEEMED TO HAVE BEEN SUCCESSFULLY                 00005620
C INTEGRATED IF EITHER QUAD CONVERGES OR THE ESTIMATED                00005630
C ABSOLUTE ERROR COMMITTED ON THIS PANEL DOES NOT EXCEED              00005640
C EPSIL TIMES THE ESTIMATED ABSOLUTE VALUE OF THE INTEGRAL            00005650
C OVER (A,B). THIS RELAXATION IS TO TRY TO TAKE ACCOUNT OF            00005660
C A COMMON SITUATION WHERE ONE PARTICULAR PANEL CAUSES                00005670
C SPECIAL DIFFICULTY, PERHAPS DUE TO A SINGULARITY OF SOME            00005680
C TYPE. IN THIS CASE QUAD COULD OBTAIN NEARLY EXACT                   00005690
C ANSWERS ON ALL OTHER PANELS AND SO THE RELATIVE ERROR FOR           00005700
C THE TOTAL INTEGRATION WOULD BE ALMOST ENTIRELY DUE TO THE           00005710
C DELINQUENT PANEL. WITHOUT THIS CONDITION THE COMPUTATION            00005720
C MIGHT CONTINUE DESPITE THE REQUESTED RELATIVE ERROR BEING           00005730
C ACHIEVED.                                                           00005740
C   THE OUTCOME OF THE INTEGRATION IS INDICATED BY ICHECK.            00005750
C   ICHECK=0  -  CONVERGENCE OBTAINED WITHOUT INVOKING SUB-           00005760
C                DIVISION. THIS WOULD CORRESPOND TO THE               00005770
C                DIRECT USE OF QUAD.                                  00005780
C   ICHECK=1  -  RESULT OBTAINED AFTER INVOKING SUBDIVISION.          00005790
C   ICHECK=2  -  AS FOR ICHECK=1 BUT AT SOME POINT THE                00005800
C                RELAXED CONVERGENCE CRITERION WAS USED.              00005810
C                THE RISK OF UNDERESTIMATING THE RELATIVE             00005820
C                ERROR WILL BE INCREASED. IF NECESSARY,               00005830
C                CONFIDENCE MAY BE RESTORED BY CHECKING               00005840
C                EPSIL AND RELERR FOR A SERIOUS DISCREPANCY.          00005850
C   ICHECK NEGATIVE                                                   00005860
C                IF DURING THE SUBDIVISION PROCESS THE STACK          00005870
C                OF DELINQUENT INTERVALS BECOMES FULL (IT IS          00005880
C                PRESENTLY SET TO HOLD AT MOST 100 NUMBERS)           00005890
C                A RESULT IS OBTAINED BY CONTINUING THE               00005900
C                INTEGRATION IGNORING CONVERGENCE FAILURES            00005910
C                WHICH CANNOT BE ACCOMMODATED ON THE STACK.           00005920
C                THIS OCCURRENCE IS FLAGGED BY RETURNING              00005930
C                ICHECK WITH NEGATIVE SIGN.                           00005940
C  THE RELIABILITY OF THE ALGORITHM WILL DECREASE FOR LARGE           00005950
C  VALUES OF EPSIL. IT IS RECOMMENDED THAT EPSIL SHOULD               00005960
```

```
C  GENERALLY BE LESS THAN ABOUT 0.001.                                    00005970
      DIMENSION RESULT(8), STACK(100)                                     00005980
      EXTERNAL F                                                          00005990
      DATA ISMAX/100/                                                     00006000
      CALL DQUAD(A, B, RESULT, K, EPSIL, NPTS, ICHECK, F,MEV)             00006010
      DQSUBA = RESULT(K)                                                  00006020
      RELERR =0.0D0                                                       00006030
      IF(DQSUBA.NE.0.0)RELERR=                                            00006040
     1 DABS((RESULT(K)-RESULT(K-1))/DQSUBA)                               00006050
C CHECK IF SUBDIVISION IS NEEDED                                          00006060
      IF (ICHECK.EQ.0) RETURN                                             00006070
C SUBDIVIDE                                                               00006080
      ESTIM=DABS(DQSUBA*EPSIL)                                            00006090
      RELERR =0.0D0                                                       00006100
      DQSUBA =0.0D0                                                       00006110
      IS = 1                                                              00006120
      IC = 1                                                              00006130
      SUB1 = A                                                            00006140
      SUB3 = B                                                            00006150
   10 SUB2 = (SUB1+SUB3)*0.5D0                                            00006160
      CALL  DQUAD(SUB1, SUB2, RESULT, K, EPSIL, NF, ICHECK, F,MEV)        00006170
      NPTS = NPTS + NF                                                    00006180
      IF(NPTS.GE.MEV) GO TO 50                                            00006190
      COMP=DABS(RESULT(K)-RESULT(K-1))                                    00006200
      IF (ICHECK.EQ.0) GO TO 30                                           00006210
      IF(COMP.LE.ESTIM) GO TO 70                                          00006220
      IF (IS.GE.ISMAX) GO TO 20                                           00006230
C STACK SUBINTERVAL (SUB1,SUB2) FOR FUTURE EXAMINATION                    00006240
      STACK(IS) = SUB1                                                    00006250
      IS = IS + 1                                                         00006260
      STACK(IS) = SUB2                                                    00006270
      IS = IS + 1                                                         00006280
      GO TO 40                                                            00006290
   20 IC = -IABS(IC)                                                      00006300
   30 DQSUBA = DQSUBA + RESULT(K)                                         00006310
      RELERR = RELERR + COMP                                              00006320
   40 CALL DQUAD(SUB2, SUB3, RESULT, K, EPSIL, NF, ICHECK, F,MEV)         00006330
      NPTS = NPTS + NF                                                    00006340
      IF(NPTS.GE.MEV) GO TO 50                                            00006350
      COMP=DABS(RESULT(K)-RESULT(K-1))                                    00006360
      IF (ICHECK.EQ.0) GO TO 50                                           00006370
      IF(COMP.LE.ESTIM) GO TO 80                                          00006380
C SUBDIVIDE INTERVAL (SUB2,SUB3)                                          00006390
      SUB1 = SUB2                                                         00006400
      GO TO 10                                                            00006410
   50 DQSUBA = DQSUBA + RESULT(K)                                         00006420
      RELERR = RELERR + COMP                                              00006430
      IF(NPTS.GE.MEV) RETURN                                              00006440
      IF (IS.EQ.1) GO TO 60                                               00006450
C SUBDIVIDE THE DELINQUENT INTERVAL LAST STACKED                          00006460
      IS = IS - 1                                                         00006470
      SUB3 = STACK(IS)                                                    00006480
      IS = IS - 1                                                         00006490
      SUB1 = STACK(IS)                                                    00006500
      GO TO 10                                                            00006510
```

```
C SUBDIVISION RESULT                                                    00006520
   60 ICHECK = IC                                                       00006530
      IF(DQSUBA.NE.0.0) THEN                                            00006540
      RELERR=RELERR/DABS(DQSUBA)                                        00006550
      ELSE                                                              00006560
          RELERR=0.0D0                                                  00006570
      ENDIF                                                             00006580
      RETURN                                                            00006590
C RELAXED CONVERGENCE                                                   00006600
   70 IC = ISIGN(2,IC)                                                  00006610
      GO TO 30                                                          00006620
   80 IC = ISIGN(2,IC)                                                  00006630
      GO TO 50                                                          00006640
      END                                                               00006650
      SUBROUTINE ERRMSG(MSG,ISKIP,IUNIT1,IUNIT2)                        00006660
C                                                                       00006670
C  GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780                  00006680
C                                                                       00006690
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                                  00006700
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2         00006710
C        > 0 FOR ONE BLANK LINE BEFORE.                                 00006720
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).     00006730
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).     00006740
C                                                                       00006750
C  MESSAGES ARE WRITTEN IN THE FORM:                                    00006760
C                                                                       00006770
C  {ERRMSG}: _MSG_HERE_                                                 00006780
C                                                                       00006790
      CHARACTER*(*) MSG                                                 00006800
      I=LEN(MSG)                                                        00006810
      DO 1 J=1,2                                                        00006820
        IF(J.EQ.1) THEN                                                 00006830
       JUNIT=IUNIT1                                                     00006840
          ELSE                                                          00006850
       JUNIT=IUNIT2                                                     00006860
        ENDIF                                                           00006870
        IF(JUNIT.GT.0) THEN                                             00006880
          IF(ISKIP.EQ.0) THEN                                           00006890
            WRITE(JUNIT,2) MSG                                          00006900
            ELSE                                                        00006910
       WRITE(JUNIT,3) MSG                                               00006920
          ENDIF                                                         00006930
      ENDIF                                                             00006940
1     CONTINUE                                                          00006950
      CALL EXIT                                                         00006960
2     FORMAT(1X,'{ERRMSG}: ',A<I>)                                      00006970
3     FORMAT(/1X,'{ERRMSG}: ',A<I>)                                     00006980
      END                                                               00006990
      SUBROUTINE NLSOL2(FCODE,PCODE,SUBZ,SUBEND)                        00007000
C                                                                       00007010
C>> NLSOL2 IS A REVISED VERSION OF NLSOL WITHOUT CALL NAMELIST;         00007020
C   I.E., NLSOL2 USES THE CURRENT VAX-11/780 VER 3.4 NAMELIST, AND      00007030
C   SUBZ MUST BE CHANGED TO READ(99,INIT) FROM CALL NAMELIST(5,'$INIT'),00007040
C   WHERE CALL PRENAM(5,99) IS USED TO ENSURE UNIT=5 NAMELIST IS IN     00007050
C   PROPER FORMAT ON SCRATCH UNIT=99 (FOR099 DELETED ON RETURN TO VMS). 00007060
```

```
C                                                                       00007070
C {NLSOL2}: GENERAL NONLINEAR LEAST-SQUARES SOLUTION   {11/2/83}        00007080
C           USING DENNIS ET AL (1979; SEE REF1 BELOW)    -------        00007090
C           ADAPTIVE NONLINEAR LEAST-SQUARES ALGORITHM.                 00007100
C                                                                       00007110
C** THIS IS AN INTERFACE ROUTINE WRITTEN FOR THE VAX-11/780 BY          00007120
C   W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.              00007130
C                                                                       00007140
C** THIS INTERFACE (NLSOL) HAS ADDITIONAL OPTIONS (BESIDE REF1) TO:     00007150
C   (1) PERFORM EITHER UNCONSTRAINED OR UP TO 4-TYPES OF CONSTRAINED    00007160
C       ADAPTIVE NONLINEAR REGRESSION FOR ARBITRARY NONLINEAR PROBLEMS. 00007170
C       (I.E., PARTIAL OR FULL LOWER/HIGHER PARAMETER BOUNDS, ETC.)     00007180
C   (2) HOLDING CERTAIN PARAMETERS FIXED (I.E., AS CONSTANTS) IN THE    00007190
C       LEAST-SQUARES (THIS IS ANOTHER FORM OF CONSTRAINING SOLUTION    00007200
C       SPACE).                                                         00007210
C   (3) PROVIDE FOR WEIGHTED OBSERVATIONS (I.E., WEIGHTED LEAST-SQUARES)00007220
C   (4) OBJECT (RUN)-TIME CONTROL OF READING THE DATA MATRIX, PLUS      00007230
C       MANY OTHER I/O OPTIONS, ETC.                                    00007240
C   (5) OPTIONALLY, ONE CAN USE EITHER ESTIMATED PARTIAL DERIVATIVES, OR00007250
C       ANALYTICAL PARTIAL DERIVATIVES (IF SUBROUTINE PCODE AVAILABLE). 00007260
C                                                                       00007270
C** THE USER ONLY NEEDS TO WRITE SUBROUTINES FCODE, PCODE, SUBZ, AND    00007280
C   SUBEND (SEE DETAILS BELOW) EXACTLY AS USED IN SUBROUTINE 'MARQRT'   00007290
C   (SEE REF2) OR 'IMSLMQ' (SEE REF3).  ALSO, THE SAME PARAMETER FILE   00007300
C   FOR005 AND OBJECT (RUN)-TIME DATA MATRIX FILE FOR010 AS USED BY     00007310
C   EITHER MARQRT OR IMSLMQ MAY BE USED IN 'NLSOL'.                     00007320
C                                                                       00007330
C** NLSOL CALLS NLITR WHICH CALLS 'NL2ITR' AS PUBLISHED BY DENNIS ET AL,00007340
C   (SEE REF1, P. 38), OR 'NL2SNO' (SEE REF1, P. 35).                   00007350
C                                                                       00007360
C** REF1:  DENNIS, J.E., ET AL, 1979, AN ADAPTIVE NONLINEAR LEAST-      00007370
C          SQUARES ALGORITHM, NTIS REPORT AD-A079-716.                  00007380
C                                                                       00007390
C   REF2:  ANDERSON, W.L., 1980, PROGRAM MARQHXY: INVERSION OF HX AND HY00007400
C          FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN-  00007410
C          FILE REPT. 80-901.                                           00007420
C                                                                       00007430
C   REF3:  ANDERSON, W.L., 1980, PROGRAM IMSLEXY: INVERSION OF EX AND EY00007440
C          FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN-  00007450
C          FILE REPT. 80-1073.                                          00007460
C                                                                       00007470
C***********************************************************************00007480
C                                                                       00007490
C****  THE USER MUST DECLARE THE CALLING PARAMETERS AS EXTERNAL IN THE  00007500
C      CALLING PROGRAM (ANY DESIRED NAMES MAY BE USED).                 00007510
C E.G.,                                                                 00007520
C                                                                       00007530
C [MAIN]:                                                               00007540
C     EXTERNAL MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND                      00007550
C     CALL NLSOL2(MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND)                  00007560
C     STOP   !<OR USE>: CALL EXIT                                       00007570
C     END                                                               00007580
C [FCODE]:                                                              00007590
C     SUBROUTINE MY_FCODE(Y,X,B,W,F,IN,IDER)                            00007600
C     USER WRITTEN TO EVALUATE THE NONLINEAR OBJECTIVE FUNCTION (F)     00007610
```

```
C      USED IN NLSOL AS THE WEIGHTED SUM OF (Y(IN)-F)**2, WHERE          00007620
C      Y= OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N, WHERE N IS          00007630
C       GIVEN IN $PARMS NAMELIST INPUT--SEE BELOW).                      00007640
C      X= OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,M, WHERE           00007650
C       M IS IN $PARMS INPUT).                                           00007660
C      B= CURRENT PARAMETER ESTIMATES (DIM. K, WHERE                     00007670
C       K IS IN $PARMS INPUT).                                           00007680
C      W= WORK ARRAY (DIM. 5)--MAY BE USED TO PASS DATA TO PCODE.        00007690
C      F= (OUTPUT) THE FUNCTION VALUE EVALUATED FOR THE GIVEN            00007700
C       Y,X, AND B ARRAYS AT THE OBSERVATION NO. 'IN'.                   00007710
C      IN= (INPUT) OBSERVATION NO. TO EVALUATE F (1.LE.IN.LE.N),         00007720
C       WHICH IS CONTROLLED EXTERNALLY BY 'NLSOL'. USUALLY,              00007730
C       IN=1,2,...,N--BUT NOT ALWAYS.                                    00007740
C      IDER= 0 IF ANALYTICAL DERIVATIVES ARE USED (PCODE CALLED          00007750
C         AFTER FCODE).                                                 00007760
C          = 1 IF ESTIMATED DERIVATIVES ARE USED (PCODE NOT CALLED       00007770
C         AFTER FCODE).                                                 00007780
C      DIMENSION Y(1),X(500,5),B(1),W(5)                                 00007790
C>>>>> INSERT USER CODE HERE TO EVALUATE F  <<<<<                        00007800
C      END                                                               00007810
C [PCODE]: >> PCODE MAY BE A DUMMY NAME IF ONLY IDER=1 IS TO BE USED. <<00007820
C      SUBROUTINE MY_PCODE(P,X,B,W,F,IN,IP,IB)                           00007830
C      USER WRITTEN TO EVALUATE THE ANALYTICAL PARTIAL DERIVATIVES OF    00007840
C      F WITH RESPECT TO B(J),J=1,2,...,K, AT OBSERVATION 'IN', WHERE    00007850
C      P= (OUTPUT) PARTIAL DERIVATIVE ARRAY (DIM. K, WHERE               00007860
C       K IS IN $PARMS INPUT).                                           00007870
C      X,B,W ARE THE SAME AS USED IN FCODE (SEE ABOVE).                  00007880
C      F= LAST FUNCTION VALUE FROM FCODE AT OBSERVATION IN.              00007890
C       (NOTE THAT F MAY NOT BE NEEDED, BUT IS AVAILABLE ANYWAY)         00007900
C      IN= (INPUT) OBSERVATION NO. TO EVALUATE P ARRAY, WHICH IS         00007910
C       CONTROLLED EXTERNALLY BY 'NLSOL' (1.LE.IN.LE.N).                 00007920
C      IP= (INPUT) THE NO. OF B-PARAMETERS HELD FIXED IN THE LEAST-      00007930
C       SQUARES (0.LE.IP.LE.K-1; USE IP=0 IF NONE).                      00007940
C      IB= ARRAY OF B-PARAMETER INDICES HELD FIXED IF IP.GT.0.           00007950
C       NOTE THAT THE INDICES IN IB ARRAY MAY BE IN ANY ORDER,           00007960
C       BUT MUST BE BETWEEN 1 AND K (K IS IN $PARMS INPUT).              00007970
C      DIMENSION P(1),X(500,5),B(1),W(5),IB(1)                           00007980
C>>>>>  INSERT USER CODE HERE TO EVALUATE P  <<<<<                       00007990
C      END                                                               00008000
C [SUBZ]:                                                                00008010
C      SUBROUTINE MY_SUBZ(Y,X,B,W,NW,N,TITLE,IOUT)                       00008020
C      USER WRITTEN INITIALIZATION ROUTINE (CALLED ONCE BY 'NLSOL').     00008030
C      SUBZ MAY BE USED TO CHECK Y(IN),X(IN,M) AFTER INPUT VIA           00008040
C      OBJECT (RUN)-TIME INPUT (SEE BELOW) ON UNIT IALT. ALSO, SUBZ      00008050
C      MAY BE USED TO READ ADDITIONAL $INIT PARAMETERS, AND TO LOAD      00008060
C       ANY COMMON BLOCKS IF NEEDED IN THE USERS FCODE,PCODE.            00008070
C      Y,X,B,W  ARE THE SAME AS USED IN FCODE (SEE ABOVE).               00008080
C      NW= USE ANY DUMMY INTEGER VARIABLE (THIS IS                       00008090
C       TO MAINTAIN COMPATIBILITY WITH 'MARQRT' OR 'IMSLMQ').            00008100
C      N= NO. OF OBSERVATIONS IN Y(N),X(N,M) ARRAYS, WHERE               00008110
C       K.GE.N.LE.500 (N,M,K  ARE IN $PARMS INPUT).                      00008120
C      TITLE= (INPUT) 80-CHARACTER HEADING (SEE INPUT FOR005 BELOW).     00008130
C      IOUT= 1 IF TO WRITE OUTPUT ON BOTH FOR006 AND FOR016.             00008140
C          = 0 IF TO WRITE OUTPUT ONLY ON FOR006.                        00008150
C      DIMENSION Y(1),X(500,5),B(1),W(5)                                 00008160
```

```
C     CHARACTER*80 TITLE                                                    00008170
C>>>>> INSERT USER CODE HERE FOR ANY INITIALIZATION DESIRED  <<<<<          00008180
C     END                                                                   00008190
C [SUBEND]:                                                                 00008200
C     SUBROUTINE MY_SUBEND(Y,X,B,K,N,TITLE,IOUT)                            00008210
C     USER WRITTEN TERMINATION ROUTINE (CALLED ONCE BY 'NLSOL').            00008220
C     SUBEND MAY BE USED TO OUTPUT THE FINAL SOLUTION VECTOR B(I),          00008230
C     I=1,2,...,K, IN OTHER FORMS, ETC., AS DESIRED. [OR IT MAY BE A        00008240
C     DUMMY ROUTINE; I.E., JUST RETURNS.]                                   00008250
C     Y,X,K,N,TITLE,IOUT  ARE THE SAME AS IN SUBZ AND FCODE.                00008260
C     B= (INPUT) IS THE FINAL SOLUTION VECTOR AS DETERMINED BY              00008270
C      'NLSOL' (SEE REF1 FOR DETAILS).                                      00008280
C     DIMENSION Y(1),X(500,5),B(1)                                          00008290
C     CHARACTER*80 TITLE                                                    00008300
C>>>>>  INSERT USER CODE HERE FOR ANY TERMINATION SUMMARY DESIRED  <<<<<00008310
C     END                                                                   00008320
C                                                                           00008330
C*************************************************************************00008340
C                                                                           00008350
C** INPUT ORDER ON FOR005 (PARAMETER FILE LOGICAL NAME):                    00008360
C                                                                           00008370
C 1.  TITLE (MAX. 80-CHARACTERS--ALWAYS READ BEFORE $PARMS INPUT).          00008380
C 2.  $PARMS  -- POSSIBLE NAMES ARE: N,M,K,B(),IP,IB(),IALT,IWT,IDER,       00008390
C       BL(),BH(),IPRT,IOUT,NITER,SP, -- PLUS FOLLOWING PARAMETERS FROM     00008400
C     REF1 (NL2SOL), P.31-35: IV(),V().                                     00008410
C 3.  (OBJECT-RUN-TIME FORMAT STATEMENT) TO DESCRIBE THE FORMAT OF THE      00008420
C     DATA MATRIX ROW Y(I),(X(I,J),J=1,M*) READ ON FILE IALT, WHERE         00008430
C     M*=M (IF IWT=0) OR M*=M+1 (IF IWT>0), M.LE.4, AND I=1,2,...,N.        00008440
C (3A). INSERT DATA MATRIX HERE ONLY IF IALT=5.                             00008450
C 4.  $INIT OPTIONAL NAMELIST USED FOR READING PROBLEM-DEPENDENT            00008460
C     PARAMETERS USED IN SUBROUTINE SUBZ (SEE ABOVE).                       00008470
C 5.  OPTIONALLY, REPEAT STEPS 1-4, IF PARAMETER ISTOP=0 WAS USED           00008480
C     IN THE LAST STEP 2.                                                   00008490
C                                                                           00008500
C** OUTPUT IS GIVEN ON FOR006 (ON-LINE USUALLY) AND ON FOR016(IF IOUT=1)00008510
C   FOR016 CONTAINS ALL PRINTABLE OUTPUT SELECTED VIA $PARMS IPRT,IOUT.   00008520
C   NOTE: IPRT=0 GIVES ABBREVIATED OUTPUT ON FOR006 (BUT MORE ON FOR016)00008530
C         IPRT=1 OR -2 GIVES DETAILED OUTPUT ON BOTH 6 AND 16.            00008540
C         IPRT=-1 GIVES MODERATE OUTPUT ON 6 (DETAILED ON 16).            00008550
C                                                                           00008560
C** TO RUN ON VAX (ELIMINATE <> DELIMITERS IN SUBSTITUTIONS):              00008570
C                                                                           00008580
C   $ASSIGN <PARAMETER FILE NAME> FOR005                                    00008590
C   $ASSIGN <DATA MATRIX FILE NAME> FOR010                                  00008600
C   $RUN <MAIN NAME>                                                        00008610
C                                                                           00008620
C [NOTE: NLSOL2 USES SCRATCH UNIT FOR099 VIA CALL PRENAM(5,99)]             00008630
C                                                                           00008640
C*************************************************************************00008650
C                                                                           00008660
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00008670
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF             00008680
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                            00008690
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                                   00008700
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS.     00008710
```

```
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:            00008720
      PARAMETER (K1DIM=KDIM-1,K2DIM=KDIM+KDIM,M1DIM=MDIM-1,             00008730
     1 IVDIM=KDIM+60,NKVDIM=96+(KDIM+3)*NDIM+(KDIM*(7*KDIM+43))/2)      00008740
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00008750
C                                                                       00008760
      REAL*4 L                                                          00008770
      DIMENSION B(KDIM),SQWT(NDIM),IB(K1DIM),C(KDIM),INDEX(KDIM),       00008780
     1 IV(IVDIM),V(NKVDIM),CBOUND(K2DIM),                               00008790
     2 BL(KDIM),BH(KDIM),CL(KDIM),CH(KDIM),SE(KDIM),                    00008800
     3 W(KDIM),PARM(4),IRATIO(2),PRNT(5)                                00008810
      INTEGER SP,SCALEP,SY,SCALEY                                       00008820
      CHARACTER*3 CHAR3                                                 00008830
      CHARACTER*6 CALLED                                                00008840
      CHARACTER*80 TITLE                                                00008850
      CHARACTER*132 LINE132                                             00008860
      CHARACTER*72 FMT                                                  00008870
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,     00008880
     1 IDER,K,ISP                                                       00008890
      COMMON/BOUNDS/BL,BH                                               00008900
      COMMON/REVCOM/R(NDIM)                                             00008910
      EQUIVALENCE (SQWT(1),X(1,MDIM)),(N,NOBS),(K,KPARMS),(M,MVARS),    00008920
     1 (CL(1),CBOUND(1)),(CH(1),CBOUND(KDIM+1))                         00008930
      EXTERNAL FCODE,PCODE,CALCR                                        00008940
      NAMELIST/PARMS/N,M,K,B,IP,IB,IALT,IWT,IDER,IPRT,IOUT,NITER,SP,    00008950
     1 BL,BH,IV,V,ISTOP                                                 00008960
C**                                                                     00008970
C PRENAM WILL ENSURE UNIT=5 NAMELIST IS IN PROPER FORMAT                00008980
C ON SCRATCH UNIT=99 (DELETED ON RETURN TO VMS). MAKE SURE SUBZ HAS     00008990
C BEEN MODIFIED TO READ(99,INIT) INSTEAD OF READ(5,INIT), ETC.          00009000
C                                                                       00009010
      CALL PRENAM(5,99)                                                 00009020
C                                                                       00009030
C** READ NLSOL TITLE LINE                                              00009040
C                                                                       00009050
      READ(99,10,ERR=9000,END=9010) TITLE                               00009060
10    FORMAT(A80)                                                       00009070
C                                                                       00009080
C**PRESET DEFAULT PARMS (SOME MUST BE GIVEN IN $PARMS ELSE AN ERROR)    00009090
C                                                                       00009100
      N=0                                                               00009110
      K=0                                                               00009120
      IP=0                                                              00009130
      M=0                                                               00009140
      IALT=10      !NOTE IALT=99 CANNOT BE USED                         00009150
      ISTOP=1                                                           00009160
      ICALL=1                                                           00009170
      IWT=0                                                             00009180
      IDER=0                                                            00009190
      IPRT=0                                                            00009200
      NITER=10                                                          00009210
      IOUT=1                                                            00009220
      SP=0                                                              00009230
      DO 20 I=1,KDIM                                                    00009240
      IF(I.LT.KDIM) IB(I)=0                                             00009250
      BL(I)=0.0                                                         00009260
```

```
      B(I)=0.0                                                          00009270
      BH(I)=0.0                                                         00009280
20    CONTINUE                                                          00009290
22    IV(1)=10                                                          00009300
C**                                                                     00009310
C  PRESET NLITR                                                         00009320
C**                                                                     00009330
      CALL DFAULT(IV,V)                                                 00009340
C**                                                                     00009350
C** OVERRIDE FOR IV(15)=3 DEFAULT (MAY BE CHANGED VIA $PARMS INPUT)     00009360
C**                                                                     00009370
      IV(15)=3                                                          00009380
C**                                                                     00009390
C  READ $PARMS ON FOR005 (VIA FOR099)                                   00009400
C**                                                                     00009410
C//30 CALL NAMELIST(5,'$PARMS',*9020) !! OLD NLSOL METHOD               00009420
30    READ(99,PARMS,END=9020)                                           00009430
C**                                                                     00009440
C  SET EQUIVALENT PARAMETERS IN DIFFERENT COMMON'S                      00009450
C**                                                                     00009460
      ISP=SP                                                            00009470
      DO 32 I=1,KDIM                                                    00009480
      BFIX(I)=B(I)                                                      00009490
      IF(I.LT.KDIM) IIB(I)=IB(I)                                        00009500
32    CONTINUE                                                          00009510
      IIP=IP                                                            00009520
C**                                                                     00009530
C  TEST $PARMS BEFORE PROCEEDING                                        00009540
C**                                                                     00009550
      IF(IP.LT.0.OR.IP.GT.K1DIM)CALL ERRMSG('IP<0 OR IP>19',0,6,16)     00009560
      KIP=K-IP                                                          00009570
      IF(N.LT.1.OR.N.GT.NDIM.OR.N.LT.KIP)                               00009580
     1 CALL ERRMSG('N<1,N>500,OR N<K-IP',0,6,16)                        00009590
      IF(K.LT.1.OR.K.GT.KDIM.OR.KIP.LT.1)                               00009600
     1 CALL ERRMSG('K<1,K>20,OR K-IP<1',0,6,16)                         00009610
      IF(M.LT.1.OR.M.GT.M1DIM)CALL ERRMSG('M<1 OR M>4',0,6,16)          00009620
      IF(IALT.EQ.6.OR.IALT.EQ.13.OR.IALT.EQ.16.OR.IALT.EQ.4.OR.         00009630
     1 IALT.EQ.99)                                                      00009640
     1 CALL ERRMSG('IALT=4,6,13,16 OR 99',0,6,16)                       00009650
      IF(ISTOP.EQ.0.AND.IALT.EQ.5)                                      00009660
     1   CALL ERRMSG('ISTOP=0 BUT IALT=5',0,6,16)                       00009670
      IF(IWT.LT.0.OR.IWT.GT.2)CALL ERRMSG('IWT<0 OR IWT>2',0,6,16)      00009680
      IF(IDER.LT.0.OR.IDER.GT.1)CALL ERRMSG('IDER<0 OR IDER>1',0,6,16)  00009690
      IF(SP.LT.0.OR.SP.GT.4)CALL ERRMSG('SP<0 OR SP>4',0,6,16)          00009700
      IF(IP.GT.0) THEN                                                  00009710
        DO J=1,IP                                                       00009720
          IF(IB(J).LT.1.OR.IB(J).GT.K) THEN                             00009730
            ENCODE(3,43,CHAR3) J                                        00009740
            CALL ERRMSG('IP>0 AND IB(J)<1 OR IB(J)>K FOR J='//          00009750
     1      CHAR3,0,6,16)                                               00009760
          ENDIF                                                         00009770
        ENDDO                                                           00009780
      ENDIF                                                             00009790
      IF(SP.EQ.0.OR.SP.EQ.2) GO TO 41                                   00009800
      DO 40 I=1,KPARMS                                                  00009810
```

```
        IF(SP.EQ.1) THEN                                                    00009820
            IF(IP.GT.0) THEN                                                00009830
            DO 42 J=1,IP                                                    00009840
              IF(I.EQ.IB(J)) GO TO 40                                       00009850
42          CONTINUE                                                        00009860
            ENDIF                                                           00009870
          IF(B(I).LE.0.) THEN                                               00009880
            ENCODE(3,43,CHAR3) I                                            00009890
43           FORMAT(I2,'.')                                                 00009900
               CALL ERRMSG('SP=1 AND B(I)<=0 FOR I='//CHAR3,0,6,16)         00009910
             ENDIF                                                          00009920
        ELSE IF(SP.GT.2) THEN                                               00009930
          IF(B(I).LT.BL(I).OR.B(I).GT.BH(I).OR.BL(I).GT.BH(I)) THEN         00009940
           ENCODE(3,43,CHAR3) I                                             00009950
            CALL ERRMSG('SP>2 AND B(I)<BL(I), '//                           00009960
     1       'B(I)>BH(I), OR BL(I)>BH(I)'//                                 00009970
     2       ' FOR I='//CHAR3,0,6,16)                                       00009980
          ENDIF                                                             00009990
         IF(BL(I).EQ.BH(I)) THEN                                            00010000
            IF(IP.GT.0) THEN                                                00010010
              DO 45 J=1,IP                                                  00010020
                IF(I.EQ.IB(J)) GO TO 40                                     00010030
45          CONTINUE                                                        00010040
             ENDIF                                                          00010050
            ENCODE(3,43,CHAR3) I                                            00010060
            CALL ERRMSG('SP>2 AND BL(I)=BH(I) BUT B(I) NOT HELD '//         00010070
     1       'FIXED FOR I='//CHAR3,0,6,16)                                  00010080
          ENDIF                                                             00010090
        ENDIF                                                               00010100
40     CONTINUE                                                             00010110
41     IF(IV(1).EQ.10) THEN                                                 00010120
C**                                                                         00010130
C  NOTE CALL DFAULT(IV,V) WAS PRESET BEFORE $PARMS READ                     00010140
C**                                                                         00010150
        IV(18)=NITER                                                        00010160
        IF(IPRT.GT.-3.AND.IPRT.LT.1) THEN                                   00010170
          IV(19)=-1                                                         00010180
        ELSE                                                                00010190
          IV(19)=IPRT                                                       00010200
        ENDIF                                                               00010210
        IF(IOUT.EQ.0) THEN                                                  00010220
          IV(21)=6                                                          00010230
        ELSE                                                                00010240
          IV(21)=16                                                         00010250
        ENDIF                                                               00010260
      ENDIF                                                                 00010270
      IF(IP.GT.0) THEN                                                      00010280
        DO 50 I=1,IP                                                        00010290
          IF(IB(I).LE.0)CALL ERRMSG('IP>0 BUT SOME IB(I)<=0',0,6,16)        00010300
50      CONTINUE                                                            00010310
      ENDIF                                                                 00010320
C                                                                           00010330
C  READ OBJECT(RUN)-TIME FORMAT FOR DATA MATRIX FROM FILE IALT.             00010340
C                                                                           00010350
      READ(99,60,ERR=9000,END=9010) FMT                                     00010360
```

```
60     FORMAT(A72)                                                          00010370
       IF(IWT.EQ.0) THEN                                                    00010380
         M1=MVARS                                                           00010390
       ELSE                                                                 00010400
         M1=MVARS+1                                                         00010410
       ENDIF                                                                00010420
       DO 70 I=1,NOBS                                                       00010430
        IF(IALT.EQ.5) THEN                                                  00010440
         READ(99,FMT,ERR=9030,END=9040) Y(I),(X(I,J),J=1,M1)                00010450
        ELSE                                                                00010460
         READ(IALT,FMT,ERR=9030,END=9040) Y(I),(X(I,J),J=1,M1)              00010470
        ENDIF                                                               00010480
         IF(IWT.EQ.0.OR.X(I,M1).EQ.0.0) THEN                                00010490
           SQWT(I)=1.0                                                      00010500
           GO TO 70                                                         00010510
         ELSE IF(IWT.EQ.1) THEN                                             00010520
           SQWT(I)=1.0/X(I,M1)                                              00010530
         ELSE                                                               00010540
           SQWT(I)=1.0/SQRT(ABS(X(I,M1)))                                   00010550
         ENDIF                                                              00010560
70     CONTINUE                                                             00010570
C                                                                           00010580
C  INITIALIZE VIA CALL SUBZ (READ $INIT AND TEST, LOAD COMMON, ETC.)        00010590
C  (AGAIN, MAKE SURE SUBZ USES A READ(99,INIT), ETC.)                       00010600
C                                                                           00010610
       CALL SUBZ(Y,X,BFIX,PRNT,NPRNT,N,TITLE,IOUT)                          00010620
C      ******************************************                           00010630
C                                                                           00010640
C  WRITE $PARMS ON FOR006 AND FOR016 (THE LATTER IF IOUT=1)                 00010650
C                                                                           00010660
       CALL NONBLANK(TITLE,NB)                                              00010670
       WRITE(6,80) TITLE,N,K,IP,M,IALT,ISTOP,IWT,IDER,IPRT,NITER,IOUT,SP    00010680
80     FORMAT('1{NLSOL}:',8X,A<NB>//' N=',4X,I6,T18,'K=',4X,I6,T34,'IP=',   00010690
      1 3X,I6,T50,'M=',4X,I6,T66,'IALT=',1X,I6/' ISTOP=',I6,T18,'IWT=',     00010700
      2 2X,I6,T34,'IDER=',I7,T50,'IPRT=',I7,T66,'NITER=',I6/' IOUT=',       00010710
      3 5X,I2,T18,'SP=',3X,I6)                                              00010720
       IF(IOUT.NE.0)                                                        00010730
      1WRITE(16,80)TITLE,N,K,IP,M,IALT,ISTOP,IWT,IDER,IPRT,NITER,IOUT,SP    00010740
       IF(IP.GT.0) THEN                                                     00010750
         WRITE(6,90) (IB(I),I=1,IP)                                         00010760
90       FORMAT(/' PARAMETERS HELD FIXED: IB=',20I3)                        00010770
         IF(IOUT.NE.0) WRITE(16,90) (IB(I),I=1,IP)                          00010780
       ENDIF                                                                00010790
       CALL NONBLANK(FMT,NB)                                                00010800
       WRITE(6,100) FMT                                                     00010810
100    FORMAT(/' FMT=',A<NB>/)                                              00010820
       IF(IOUT.NE.0) WRITE(16,100) FMT                                      00010830
       IF(SP.GT.2) THEN                                                     00010840
         WRITE(6,111) (BL(I),I=1,KPARMS)                                    00010850
111      FORMAT(/' PARAMETER LOWER BOUNDS: BL='//(5E16.8))                  00010860
         IF(IOUT.NE.0) WRITE(16,111) (BL(I),I=1,KPARMS)                     00010870
       ENDIF                                                                00010880
       WRITE(6,110) (B(I),I=1,KPARMS)                                       00010890
110    FORMAT(/' INITIAL PARAMETERS: B='//(5E16.8))                         00010900
       IF(IOUT.NE.0) WRITE(16,110) (B(I),I=1,KPARMS)                        00010910
```

```
      IF(SP.GT.2) THEN                                                  00010920
        WRITE(6,112) (BH(I),I=1,KPARMS)                                 00010930
112     FORMAT(/' PARAMETER HIGHER BOUNDS: BH='//(5E16.8))              00010940
        IF(IOUT.NE.0) WRITE(16,112) (BH(I),I=1,KPARMS)                  00010950
      ENDIF                                                             00010960
      DO 120 I=1,KDIM                                                   00010970
120     INDEX(I)=I                                                      00010980
      IF(IP.EQ.0) THEN                                                  00010990
        DO 130 I=1,KPARMS                                               00011000
        IF(SP.GT.2) THEN                                                00011010
          CL(I)=BL(I)                                                   00011020
          CH(I)=BH(I)                                                   00011030
        ENDIF                                                           00011040
130     C(I)=B(I)                                                       00011050
      ELSE                                                              00011060
C                                                                       00011070
C  REORDER B TO C WHEN IP>0 (AND BL,BH TO CL,CH, RESPECTIVELY)          00011080
C                                                                       00011090
        IM=0                                                            00011100
        DO 150 I=1,KPARMS                                               00011110
        DO 140 J=1,IP                                                   00011120
          IF(I.EQ.IB(J)) GO TO 150                                      00011130
140     CONTINUE                                                        00011140
        IM=IM+1                                                         00011150
        C(IM)=B(I)                                                      00011160
        IF(SP.GT.2) THEN                                                00011170
          CL(IM)=BL(I)                                                  00011180
          CH(IM)=BH(I)                                                  00011190
        ENDIF                                                           00011200
        INDEX(IM)=I                                                     00011210
150     CONTINUE                                                        00011220
        WRITE(6,160) (I,I=1,KPARMS)                                     00011230
160     FORMAT(/' PARAMETER INDEX:',20I3)                               00011240
        IF(IOUT.NE.0) WRITE(16,160) (I,I=1,KPARMS)                      00011250
        WRITE(6,170) (INDEX(I),I=1,KIP)                                 00011260
170     FORMAT(' REORDERED AS...:',20I3)                                00011270
        IF(IOUT.NE.0) WRITE(16,170) (INDEX(I),I=1,KIP)                  00011280
        WRITE(6,180) (C(I),I=1,KIP)                                     00011290
180     FORMAT(/' REORDERED PARAMETERS:'//(5E16.8))                     00011300
        IF(IOUT.NE.0) WRITE(16,180) (C(I),I=1,KIP)                      00011310
      ENDIF                                                             00011320
C                                                                       00011330
C  PERFORM INITIAL PARAMETER TRANSFORMS VIA SP (SCALEP)                 00011340
C                                                                       00011350
      IF(SP.EQ.0) GO TO 220                                             00011360
      DO 210 I=1,KIP                                                    00011370
        GO TO (201,202,203,203),SP                                      00011380
201     C(I)=ALOG(C(I))                                                 00011390
        GO TO 210                                                       00011400
202     C(I)=ASINH(C(I))                                                00011410
        GO TO 210                                                       00011420
203     TEM=(C(I)-CL(I))/(CH(I)-CL(I))                                  00011430
        IF(SP.EQ.3) THEN                                                00011440
          C(I)=ASIN(SQRT(TEM))                                          00011450
        ELSE                                                            00011460
```

```
          C(I)=ERFINV(2.0*TEM-1.0)                                      00011470
        ENDIF                                                           00011480
210   CONTINUE                                                          00011490
C                                                                       00011500
C  INTERFACE WITH NL2ITR USING MARQRT FCODE AND PCODE (IF IDER=0)       00011510
C                                                                       00011520
220   ENCODE(6,222,CALLED) ICALL                                        00011530
222   FORMAT(I3,' **')                                                  00011540
      WRITE(6,221) CALLED                                               00011550
221   FORMAT('0** NLITR (IDER=0) OR NL2SNO (IDER=1) CALLED:',A6/)       00011560
      IF(IOUT.NE.0) WRITE(16,221) CALLED                                00011570
      IF(IDER.EQ.0) THEN                                                00011580
        CALL NLITR(NOBS,KIP,C,IV,V,CBOUND,FCODE,PCODE)                  00011590
C       ********************************************                    00011600
      ELSE                                                              00011610
        CALL NL2SNO(NOBS,KIP,C,CALCR,IV,V,IDUMMY,CBOUND,FCODE)          00011620
C       ******************************************************          00011630
      ENDIF                                                             00011640
C                                                                       00011650
C  GET INVERSE PARAMETER TRANSFORMATION OF SOLUTION VECTOR C            00011660
C                                                                       00011670
      IF(SP.EQ.0) GO TO 229                                             00011680
      DO 228 I=1,KIP                                                    00011690
        GO TO (224,225,226,226),SP                                      00011700
224     C(I)=EXP(C(I))                                                  00011710
        GO TO 228                                                       00011720
225     C(I)=SINH(C(I))                                                 00011730
        GO TO 228                                                       00011740
226     TEM=CH(I)-CL(I)                                                 00011750
        IF(SP.EQ.3) THEN                                                00011760
          C(I)=CL(I)+TEM*SIN(C(I))**2                                   00011770
        ELSE                                                            00011780
          C(I)=CL(I)+0.5*TEM*(1.0+ERF(C(I)))                            00011790
        ENDIF                                                           00011800
228   CONTINUE                                                          00011810
C                                                                       00011820
C  OUTPUT SELECTED RESULTS ON FOR006 (ALL RESULTS ON FOR016 IF IOUT=1)  00011830
C                                                                       00011840
229   IF(IOUT.NE.0.AND.IPRT.NE.0) THEN                                  00011850
        I=1                                                             00011860
        REWIND 16                                                       00011870
230     READ(16,232,END=240) LINE132                                    00011880
232     FORMAT(A)                                                       00011890
        IF(I.EQ.1) THEN                                                 00011900
C                                                                       00011910
C  VAX FUNCTION 'LIB$INDEX' USED TO DISTINGUISH FROM ARRAY 'INDEX'      00011920
C                                                                       00011930
          IF(LIB$INDEX(LINE132,'CALLED:'//CALLED).EQ.0) GO TO 230       00011940
          I=0                                                           00011950
          GO TO 230                                                     00011960
        ENDIF                                                           00011970
        IF(LIB$INDEX(LINE132,'OBS.Y(I)').NE.0) GO TO 236                00011980
        IF(LIB$INDEX(LINE132,'COVARIANCE - SCALE').NE.0) GO TO 236      00011990
        CALL NONBLANK(LINE132,J)                                        00012000
        IF(J.LE.0) GO TO 230                                            00012010
```

```
        WRITE(6,234) LINE132                                              00012020
234     FORMAT(A<J>)                                                      00012030
        GO TO 230                                                         00012040
236     READ(16,232,END=240) LINE132                                      00012050
        GO TO 236                                                         00012060
      ENDIF                                                               00012070
240   IF(IOUT.NE.0) WRITE(16,250)                                         00012080
250   FORMAT(/3X,'I',4X,'OBS.Y(I)',6X,'CAL',11X,'RES',8X,                 00012090
     1 '%RES.ERR',6X,'X(I,1)',8X,                                         00012100
     2 'X(I,2)',8X,'X(I,3)',8X,'X(I,4)',8X,'WT(I)')                       00012110
      IF(IPRT.EQ.-2) WRITE(6,250)                                         00012120
      SUMF2=0.0                                                           00012130
      IF(IDER.NE.0) IADR=IV(50)-1                                         00012140
      DO 270 I=1,NOBS                                                     00012150
        IF(IDER.EQ.0) THEN                                                00012160
          F2=R(I)                                                         00012170
        ELSE                                                              00012180
          F2=V(IADR+I)                                                    00012190
        ENDIF                                                             00012200
        RES=F2/SQWT(I)                                                    00012210
        CAL=Y(I)-RES                                                      00012220
        IF(CAL.NE.0.0) THEN                                               00012230
          PERR=100.0*RES/ABS(CAL)                                         00012240
        ELSE                                                              00012250
          PERR=0.0                                                        00012260
        ENDIF                                                             00012270
        WT=SQWT(I)**2                                                     00012280
        SUMF2=SUMF2+RES**2                                                00012290
        IF(IPRT.EQ.-2)WRITE(6,260) I,Y(I),CAL,RES,PERR,                   00012300
     1 (X(I,J),J=1,4),WT                                                  00012310
260     FORMAT(1X,I3,2E14.6,E11.3,6E14.6)                                 00012320
        IF(IOUT.NE.0) WRITE(16,260) I,Y(I),CAL,RES,PERR,                  00012330
     1 (X(I,J),J=1,4),WT                                                  00012340
270   CONTINUE                                                            00012350
      IF(NOBS.EQ.KIP) THEN                                                00012360
        RMSERR=0.0                                                        00012370
      ELSE                                                                00012380
        RMSERR=SQRT(SUMF2/(NOBS-KIP))                                     00012390
      ENDIF                                                               00012400
      WRITE(6,280) RMSERR                                                 00012410
280   FORMAT(/' ** RMSERR=',E16.8)                                        00012420
      IF(IOUT.NE.0) WRITE(16,280) RMSERR                                  00012430
      IF(IV(26).LE.0) GO TO 380                                           00012440
C                                                                         00012450
C  A COVARIANCE MATRIX WAS COMPUTED (GET ADDITIONAL STATISTICS)           00012460
C                                                                         00012470
      IADR=IV(26)-1                                                       00012480
      IF(IPRT.LT.-1) WRITE(6,290)                                         00012490
290   FORMAT(/' COVARIANCE MATRIX')                                       00012500
      DO 320 I=1,KIP                                                      00012510
      DO 300 J=1,I                                                        00012520
300     W(J)=V(IADR+LOC(J,I))                                             00012530
        SE(I)=SQRT(ABS(W(I)))                                             00012540
        IF(IPRT.LT.-1) WRITE(6,310) INDEX(I),(W(J),J=1,I)                 00012550
310     FORMAT(1X,I2,10E12.4/(3X,10E12.4))                                00012560
```

```
320   CONTINUE                                                          00012570
C                                                                       00012580
C  GET CORRELATION COEFFICIENT MATRIX                                   00012590
C                                                                       00012600
      IF(IOUT.NE.0) WRITE(16,330)                                       00012610
330   FORMAT(/' CORRELATION MATRIX')                                    00012620
      IF(IPRT.LT.0) WRITE(6,330)                                        00012630
      DO 350 I=1,KIP                                                    00012640
        IF(SE(I).EQ.0.0) THEN                                           00012650
          W(I)=1.0                                                      00012660
        ENDIF                                                           00012670
      DO 340 J=1,I                                                      00012680
        IF(SE(J).NE.0.0) W(J)=V(IADR+LOC(J,I))/(SE(I)*SE(J))            00012690
340   CONTINUE                                                          00012700
      IF(IOUT.NE.0) WRITE(16,310) INDEX(I),(W(J),J=1,I)                 00012710
      IF(IPRT.LT.0) WRITE(6,310) INDEX(I),(W(J),J=1,I)                  00012720
350   CONTINUE                                                          00012730
C                                                                       00012740
C  PRINT PARAMETER STANDARD ERRORS (SE) AND RELATIVE ERRORS             00012750
C                                                                       00012760
      WRITE(6,360)                                                      00012770
360   FORMAT(/'   **PARM_SOL.   STD_ERROR   REL_ERROR    % ERROR **'/)   00012780
      IF(IOUT.NE.0) WRITE(16,360)                                       00012790
      DO 370 I=1,KIP                                                    00012800
        RELERR=0.0                                                      00012810
        IF(C(I).NE.0.0) RELERR=SE(I)/C(I)                               00012820
        PERR=100.*RELERR                                                00012830
        WRITE(6,310) INDEX(I),C(I),SE(I),RELERR,PERR                    00012840
        IF(IOUT.NE.0) WRITE(16,310) INDEX(I),C(I),SE(I),RELERR,PERR     00012850
370   CONTINUE                                                          00012860
C                                                                       00012870
C  PUT SOLUTION C AND BFIX TOGETHER (IF IP>0)                           00012880
C                                                                       00012890
380   DO 390 I=1,KIP                                                   00012900
390   W(I)=C(I)                                                         00012910
      IF(IP.EQ.0) GO TO 420                                             00012920
      IM=0                                                              00012930
      DO 410 I=1,KPARMS                                                 00012940
        W(I)=BFIX(I)                                                    00012950
      DO 400 J=1,IP                                                     00012960
        IF(I.EQ.IB(J)) GO TO 410                                        00012970
400   CONTINUE                                                          00012980
        IM=IM+1                                                         00012990
        W(I)=C(IM)                                                      00013000
410   CONTINUE                                                          00013010
420   CALL SUBEND(Y,X,W,K,N,TITLE,IOUT)                                 00013020
C       **********************************                              00013030
      IF(ISTOP.NE.1) THEN                                               00013040
        READ(99,10,ERR=9000,END=9010) TITLE                             00013050
        IF(IALT.NE.5) REWIND IALT                                       00013060
        ICALL=ICALL+1                                                   00013070
        GO TO 22                                                        00013080
      ENDIF                                                             00013090
C                                                                       00013100
C** RETURN FROM NLSOL2                                                 00013110
```

```
C                                                                       00013120
      RETURN                                                            00013130
9000  CALL ERRMSG('ERR=9000 READING FOR005',0,6,16)                     00013140
9010  CALL ERRMSG('PREMATURE E.O.F (END=9010) READING FOR005',0,6,16)   00013150
9020  CALL ERRMSG('END=9020 READING FOR005 NAMELIST',0,6,16)            00013160
9030  CALL ERRMSG('END=9030 READING FILE IALT',0,6,16)                  00013170
9040  CALL ERRMSG('PREMATURE E.O.F (END=9040) READING FILE IALT',       00013180
     1 0,6,16)                                                          00013190
C                                                                       00013200
C** END OF SUBROUTINE NLSOL2                                           00013210
C                                                                       00013220
      END                                                               00013230
      SUBROUTINE NLITR(N,KIP,C,IV,V,CBOUND,FCODE,PCODE)                 00013240
C                                                                       00013250
C**CALCULATES BOTH THE RESIDUAL VECTOR R(N) & ANALYTICAL JACOBIAN       00013260
C  JAC(N,KIP) BY 'REVERSE COMMUNICATION VIA INTERNAL CALL NL2ITR'       00013270
C  (SEE REF1, P. 38).                                                   00013280
C                                                                       00013290
C     N = NO. OBSERVATIONS <=500 (SEE NDIM BELOW)                       00013300
C     KIP = NO. ADJUSTABLE PARAMETERS =K-IIP WHERE                      00013310
C        K=TOTAL PARAMETERS, IIP=NO. PARAMETERS HELD FIXED              00013320
C          IN IIB(IIP) VIA COMMON/FIXDAT/                               00013330
C     C() = I/O PARAMETER VECTOR (SUPPLIED BY NL2ITR)                   00013340
C        WHICH ARE THE UNCONSTRAINED PARAMETERS IN NL2ITR.              00013350
C     IV() = SAME CONTROL INFORMATION SET BY NLSOL (OR NL2ITR).         00013360
C     V() = SAME CONTROL INFORMATION SET BY NLSOL (OR NL2ITR).          00013370
C     CBOUND = INPUT ARRAY OF LOW AND HIGH BOUNDS USED ONLY WHEN SP>2.  00013380
C     FCODE = EXTERNAL FUNCTION NAME (SAME AS USED IN 'MARQRT' OR       00013390
C        'IMSLMQ' TO COMPUTE THE NONLINEAR OBJECTIVE FUNCTION).         00013400
C     PCODE = EXTERNAL ANALYTIC DERIVATIVE NAME (SAME AS USED IN        00013410
C        'MARQRT' WHEN IDER=0) CORRESPONDING TO EACH FCODE CALL.        00013420
C                                                                       00013430
C**SEE REF1 (P.38) FOR MORE DETAILS ON CALLING NL2ITR.                  00013440
C                                                                       00013450
C**OTHER DATA IN COMMON/FIXDAT/ MUST BE PRESET.                         00013460
C                                                                       00013470
C                                                                       00013480
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00013490
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF         00013500
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                        00013510
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                               00013520
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS. 00013530
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:            00013540
      PARAMETER (K1DIM=KDIM-1)                                          00013550
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00013560
C                                                                       00013570
      INTEGER SP                                                        00013580
      DIMENSION C(1),IV(1),V(1),CBOUND(1),PRNT(5),SQWT(NDIM),           00013590
     1 BIP(KDIM),D(KDIM),R(NDIM),PART(KDIM),W(KDIM)                     00013600
      REAL*4 JAC(NDIM,KDIM)                                             00013610
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,     00013620
     1 IDER,KPARMS,SP                                                   00013630
      COMMON/BOUNDS/BL(KDIM),BH(KDIM)                                   00013640
      COMMON/REVCOM/R                                                   00013650
      EQUIVALENCE (SQWT(1),X(1,MDIM))                                   00013660
```

```
      DATA NN/NDIM/                                                     00013670
C                                                                       00013680
C  GET INVERSE PARAMETER TRANSFORMATION (C TO BIP)                      00013690
C                                                                       00013700
10    CALL INTRAN(KIP,C,CBOUND,BIP)                                     00013710
C                                                                       00013720
C  DETERMINE FROM IV(1) HOW TO CALL NL2ITR                              00013730
      IV1=IV(1)                                                         00013740
        DO 120 I=1,N                                                    00013750
          CALL FCODE(Y,X,BIP,PRNT,F,I,IDER)                             00013760
C         **********************************                            00013770
          IF(IV1.NE.2) R(I)=SQWT(I)*(Y(I)-F)                            00013780
          IF(IV1.EQ.1) GO TO 120                                        00013790
          CALL PCODE(PART,X,BIP,PRNT,F,I,IIP,IIB)                       00013800
C         **************************************                        00013810
C                                                                       00013820
C  SCALE PART(J) VIA SP AND THE DERIVATIVE CHAIN-RULE.                  00013830
C                                                                       00013840
        IF(SP.EQ.0) GO TO 80                                            00013850
         IF(SP.EQ.1) THEN                                               00013860
           DO 11 K=1,KPARMS                                             00013870
11         PART(K)=BIP(K)*PART(K)                                       00013880
         ELSE IF(SP.EQ.2) THEN                                          00013890
           DO 12 K=1,KPARMS                                             00013900
           IF(PART(K).EQ.0.0) GO TO 12                                  00013910
           TEM=BIP(K)+SQRT(BIP(K)**2+1.0)                               00013920
           PART(K)=0.5*(TEM+1.0/TEM)*PART(K)                            00013930
12         CONTINUE                                                     00013940
         ELSE IF (SP.EQ.3) THEN                                         00013950
           DO 13 K=1,KPARMS                                             00013960
           IF(PART(K).EQ.0.0) GO TO 13                                  00013970
           PART(K)=2.*PART(K)*SQRT((BIP(K)-BL(K))*                      00013980
     1        (BH(K)-BIP(K)))                                           00013990
13         CONTINUE                                                     00014000
         ELSE IF(SP.EQ.4) THEN                                          00014010
           DO 14 K=1,KPARMS                                             00014020
           IF(PART(K).EQ.0.0) GO TO 14                                  00014030
           TEM=BH(K)-BL(K)                                              00014040
           PART(K)=0.56418958*PART(K)*TEM*EXP(-(ERFINV(2.*(BIP(K)-      00014050
     1      BL(K))/TEM-1.))**2)                                         00014060
14         CONTINUE                                                     00014070
         ENDIF                                                          00014080
80       IF(IIP.EQ.0) THEN                                              00014090
           DO 90 J=1,KIP                                                00014100
90         JAC(I,J)=-SQWT(I)*PART(J)                                    00014110
         ELSE                                                           00014120
           IM=0                                                         00014130
           DO 110 K=1,KPARMS                                            00014140
           DO 100 J=1,IIP                                               00014150
             IF(K.EQ.IIB(J)) GO TO 110                                  00014160
100        CONTINUE                                                     00014170
           IM=IM+1                                                      00014180
           JAC(I,IM)=-SQWT(I)*PART(K)                                   00014190
110        CONTINUE                                                     00014200
         ENDIF                                                          00014210
```

```
120      CONTINUE                                                        00014220
C                                                                        00014230
C                                                                        00014240
      CALL NL2ITR(D,IV,JAC,N,NN,KIP,R,V,C)                               00014250
C     ************************************                               00014260
      IF(IV(1).EQ.1.OR.IV(1).EQ.2) GO TO 10                              00014270
      RETURN                                                             00014280
      END                                                                00014290
      SUBROUTINE INTRAN(KIP,C,CBOUND,BIP)                                00014300
C                                                                        00014310
C**INVERSE PARAMETER TRANSFORMATION USED IN 'NLSOL','NLITR'.             00014320
C                                                                        00014330
C  CALCULATES CONSTRAINED PARAMETERS FOR FCODE OR PCODE BACK FROM THE    00014340
C  UNCONSTRAINED PARAMETERS IN 'NL2ITR' OR 'NL2SNO'                      00014350
C                                                                        00014360
C     KIP = NO. ADJUSTABLE PARAMETERS = K-IIP (IIP IN COMMON/FIXDAT)     00014370
C     C() = INPUT UNCONSTRAINED VECTOR (DIM. KIP)                        00014380
C     CBOUND = INPUT CONSTRAINED BOUNDS, IF ANY.                         00014390
C     BIP() = OUTPUT CONSTRAINED VECTOR (DIM. KPARMS--IN COMMON).        00014400
C                                                                        00014410
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00014420
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF          00014430
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                         00014440
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                                00014450
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS.   00014460
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:             00014470
      PARAMETER (K1DIM=KDIM-1)                                           00014480
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00014490
C                                                                        00014500
      INTEGER SP                                                         00014510
      DIMENSION C(1),CBOUND(1),BIP(1),CTEM(KDIM)                         00014520
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,      00014530
     1 IDER,KPARMS,SP                                                    00014540
      IF(SP.EQ.0) THEN                                                   00014550
        DO 10 I=1,KIP                                                    00014560
10      CTEM(I)=C(I)                                                     00014570
      ELSE                                                               00014580
        DO 50 I=1,KIP                                                    00014590
          GO TO (20,30,40,40),SP                                         00014600
20        CTEM(I)=EXP(C(I))                                              00014610
          GO TO 50                                                       00014620
30        CTEM(I)=SINH(C(I))                                             00014630
          GO TO 50                                                       00014640
40        DIF=CBOUND(KDIM+I)-CBOUND(I)                                   00014650
          IF(SP.EQ.3) THEN                                               00014660
            CTEM(I)=CBOUND(I)+DIF*SIN(C(I))**2                           00014670
          ELSE                                                           00014680
            CTEM(I)=CBOUND(I)+0.5*DIF*(1.0+ERF(C(I)))                    00014690
          ENDIF                                                          00014700
50      CONTINUE                                                         00014710
      ENDIF                                                              00014720
      IF(IIP.EQ.0) THEN                                                  00014730
        DO 60 I=1,KIP                                                    00014740
60      BIP(I)=CTEM(I)                                                   00014750
      ELSE                                                               00014760
```

```
        IM=0                                                            00014770
        DO 80 I=1,KPARMS                                                00014780
          BIP(I)=BFIX(I)                                                00014790
        DO 70 J=1,IIP                                                   00014800
          IF(I.EQ.IIB(J)) GO TO 80                                      00014810
70      CONTINUE                                                        00014820
        IM=IM+1                                                         00014830
        BIP(I)=CTEM(IM)                                                 00014840
80      CONTINUE                                                        00014850
      ENDIF                                                             00014860
      RETURN                                                            00014870
      END                                                               00014880
      SUBROUTINE CALCR(N,KIP,C,NF,R,LASTNF,CBOUND,FCODE)                00014890
C                                                                       00014900
C**CALCULATES RESIDUAL VECTOR R(N) FOR 'NL2SNO' WHEN IDER=1.            00014910
C                                                                       00014920
C     N = NO. OBSERVATIONS <=500 (SEE NDIM BELOW)                       00014930
C     KIP = NO. ADJUSTABLE PARAMETERS =K-IIP WHERE                      00014940
C        K=TOTAL PARAMETERS, IIP=NO. PARAMETERS HELD FIXED              00014950
C          IN IIB(IIP) VIA COMMON/FIXDAT/                               00014960
C     C() = INPUT PARAMETER VECTOR (SUPPLIED BY NL2SNO)                 00014970
C        WHICH ARE THE UNCONSTRAINED PARAMETERS IN NL2SNO.              00014980
C     NF = INVOCATION COUNT (INPUT)FOR USE BY NL2SNO OR NL2SOL.         00014990
C     R() = OUTPUT WEIGHTED RESIDUAL VECTOR (DIM. N)                    00015000
C     LASTNF = LAST NF (ON EXIT FOR POSSIBLE USE IN CALCJ OR NL2SOL).   00015010
C     CBOUND = INPUT ARRAY OF LOW AND HIGH BOUNDS USED ONLY WHEN SP>2.  00015020
C     FCODE = EXTERNAL FUNCTION NAME (SAME AS USED IN 'MARQRT' OR       00015030
C        'IMSLMQ' TO COMPUTE THE NONLINEAR OBJECTIVE FUNCTION).         00015040
C                                                                       00015050
C**OTHER DATA IN COMMON/FIXDAT/ MUST BE PRESET.                         00015060
C                                                                       00015070
C                                                                       00015080
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00015090
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF         00015100
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                        00015110
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                               00015120
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS.  00015130
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:            00015140
      PARAMETER (K1DIM=KDIM-1)                                          00015150
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00015160
C                                                                       00015170
      INTEGER SP                                                        00015180
      DIMENSION C(1),R(1),CBOUND(1),PRNT(5),SQWT(NDIM),BIP(KDIM)        00015190
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,     00015200
     1 IDER,KPARMS,SP                                                   00015210
      EQUIVALENCE (SQWT(1),X(1,MDIM))                                   00015220
C                                                                       00015230
C  GET INVERSE PARAMETER TRANSFORMATION (C TO BIP)                      00015240
C                                                                       00015250
      CALL INTRAN(KIP,C,CBOUND,BIP)                                     00015260
C                                                                       00015270
C  COMPUTE RESIDUAL VECTOR R(N) USING BIP IN FCODE                      00015280
C                                                                       00015290
      DO 10 I=1,N                                                       00015300
        CALL FCODE(Y,X,BIP,PRNT,F,I,IDER)                               00015310
```

```
C       ********************************                                00015320
        R(I)=SQWT(I)*(Y(I)-F)                                           00015330
10      CONTINUE                                                        00015340
        LASTNF=NF                                                       00015350
        RETURN                                                          00015360
        END                                                             00015370
        SUBROUTINE NONBLANK(C,NB)                                       00015380
C--DETERMINE NON-BLANK CHAR LENGTH (=NB ON EXIT) OF C*(*)               00015390
C  NOTE THAT NB WILL BE IN [0,LEN(C)].                                  00015400
C                                                                       00015410
        CHARACTER*(*) C                                                 00015420
        L=LEN(C)                                                        00015430
        DO 10 I=L,1,-1                                                  00015440
          NB=I                                                          00015450
          IF(C(I:I).NE.' ') RETURN                                      00015460
10      CONTINUE                                                        00015470
        NB=0                                                            00015480
        RETURN                                                          00015490
        END                                                             00015500
        SUBROUTINE PRENAM(INUNIT,ITMP)                                  00015510
C--PRENAM CAN BE CALLED PRIOR TO READ(ITMP,NAME,...) TO SHIFT ALL       00015520
C  NAMELIST INPUT $NAME ... FROM COL.1 AND BEYOND ON INUNIT TO          00015530
C  NAMELIST INPUT $NAME ... FROM COL.2 AND BEYOND ON ITMP (UNIT=ITMP    00015540
C  IS DELETED AFTER CLOSING OR END OF PROCESS).  NOTE ITMP MAY BE       00015550
C  ANY UNIT NUMBER NOT BEING USED (BUT CANNOT BE INUNIT OR 6).          00015560
C                                                                       00015570
C--USAGE:                                                               00015580
C                                                                       00015590
C       NAMELIST/ANYNAME/...                                            00015600
C       ...                                                             00015610
C       CALL PRENAM(5,1)                                                00015620
C       ...                                                             00015630
C       READ(1,ANYNAME,END=99,ERR=999)                                  00015640
C       ...                                                             00015650
C                                                                       00015660
C--NOTE:  BECAUSE EARLIER VERSIONS (<3.0) OF VAX-11 FORTRAN-77          00015670
C       DID NOT HAVE NAMELIST AVAILABLE, A SIMULATED CALL NAMELIST WAS  00015680
C       USED BY MANY USERS.  IN PARTICULAR, W.L.ANDERSON USED A         00015690
C       SIMULATED CALL NAMELIST(INUNIT,'$ANYNAM',*99) SUBROUTINE WHICH  00015700
C       COULD CONTAIN $ANYNAM LISTS BEGINNING IN COL.1 TO 80;  BUT      00015710
C       SINCE VERSION 3.0 OF VAX-11 FORTRAN-77 REQUIRES THE INPUT       00015720
C       $ANYNAM LIST TO BEGIN IN COL.2 OR BEYOND, SUBROUTINE PRENAM     00015730
C       CAN BE USED ONCE TO MEET THIS REQUIREMENT, AND BECOMES          00015740
C       TRANSPARENT TO THE END USER'S INPUT FILE PREPARATION (COL.1-ON) 00015750
C                                                                       00015760
        CHARACTER*200 C                                                 00015770
        IF(ITMP.EQ.6.OR.ITMP.EQ.INUNIT) CALL ERRMSG(                    00015780
     1  '{PRENAM}: ITMP=6 OR ITMP=INUNIT VIOLATION',0,6,0)              00015790
        OPEN(UNIT=ITMP,STATUS='SCRATCH',FILE='PRENAM.TMP',ERR=999)      00015800
        NONAME=0                                                        00015810
10      READ(INUNIT,20,END=99,ERR=888) C                                00015820
20      FORMAT(A)                                                       00015830
        CALL NONBLANK(C,NC)                                             00015840
        IF(NC.EQ.0) NC=1                                                00015850
        IF(NONAME.EQ.0) THEN                                            00015860
```

```
        I=INDEX(C,'$')                                                  00015870
        IF(I.EQ.0) THEN                                                 00015880
          WRITE(ITMP,30) C                                              00015890
30        FORMAT(A<NC>)                                                 00015900
        ELSE                                                            00015910
          NONAME=1                                                      00015920
          WRITE(ITMP,40) C(I:NC)                                        00015930
40        FORMAT(1X,A)                                                  00015940
          I=INDEX(C(I+1:NC),'$')                                        00015950
          IF(I.NE.0) NONAME=0                                           00015960
        ENDIF                                                           00015970
      ELSE                                                              00015980
        WRITE(ITMP,40) C(1:NC)                                          00015990
        I=INDEX(C,'$')                                                  00016000
        IF(I.NE.0) NONAME=0                                             00016010
      ENDIF                                                             00016020
      GO TO 10                                                          00016030
99    REWIND ITMP                                                       00016040
      RETURN                                                            00016050
888   CALL ERRMSG('{PRENAM}: ERROR IN READING INUNIT',0,6,0)            00016060
999   CALL ERRMSG('{PRENAM}: CANNOT OPEN UNIT=ITMP',0,6,0)              00016070
      END                                                               00016080
      SUBROUTINE PROCINFO(ABS_VALUES,INCR_VALUES)                       00016090
C                                                                       00016100
C** SUBROUTINE TO OBTAIN ABSOLUTE AND INCREMENTAL VALUES OF PROCESS     00016110
C   PARAMETERS: CPU TIME, BUFFERED I/O COUNT, DIRECT I/O COUNT, AND     00016120
C   PAGE FAULTS.                                                        00016130
C                                                                       00016140
      IMPLICIT INTEGER*2(W),INTEGER*4(L)                                00016150
      PARAMETER (JPI$_CPUTIM = '00000407'X,                             00016160
     1 JPI$_BUFIO = '0000040C'X,JPI$_DIRIO = '0000040B'X,               00016170
     2 JPI$_PAGEFLTS= '0000040A'X)                                      00016180
      INTEGER*4 ABS_VALUES(4),INCR_VALUES(4),LCL_VALUES(4)              00016190
      COMMON/ITEMLIST/                                                  00016200
     1 W_LEN1,W_CODE1,L_ADDR1,L_LENADDR1,                               00016210
     2 W_LEN2,W_CODE2,L_ADDR2,L_LENADDR2,                               00016220
     3 W_LEN3,W_CODE3,L_ADDR3,L_LENADDR3,                               00016230
     4 W_LEN4,W_CODE4,L_ADDR4,L_LENADDR4,                               00016240
     5 W_LEN5,W_CODE5                                                   00016250
      DATA W_LEN1,W_LEN2,W_LEN3,W_LEN4,W_LEN5/5*4/                      00016260
      DATA W_CODE1/JPI$_CPUTIM/,                                        00016270
     1 W_CODE2/JPI$_BUFIO/,                                             00016280
     2 W_CODE3/JPI$_DIRIO/,                                             00016290
     3 W_CODE4/JPI$_PAGEFLTS/,                                          00016300
     4 W_CODE5/0/                                                       00016310
      DATA L_LENADDR1,L_LENADDR2,L_LENADDR3,L_LENADDR4/4*0/             00016320
      L_ADDR1=%LOC(LCL_VALUES(1))                                       00016330
      L_ADDR2=%LOC(LCL_VALUES(2))                                       00016340
      L_ADDR3=%LOC(LCL_VALUES(3))                                       00016350
      L_ADDR4=%LOC(LCL_VALUES(4))                                       00016360
C** PERFORM THE SYSTEM SERVICE CALL                                     00016370
      CALL SYS$GETJPI(,,,W_LEN1,,,)                                     00016380
C** ASSIGN THE NEW VALUES TO THE ARGUMENTS                              00016390
      DO I=1,4                                                          00016400
        INCR_VALUES(I)=LCL_VALUES(I)-ABS_VALUES(I)                      00016410
```

```
        ABS_VALUES(I)=LCL_VALUES(I)                                     00016420
      END DO                                                            00016430
      RETURN                                                            00016440
      END                                                               00016450
      SUBROUTINE WARN(MSG,ISKIP,IUNIT1,IUNIT2,*)                        00016460
C                                                                       00016470
C  GENERAL WARNING MESSAGE OUTPUT AND RETURN 1 ON VAX-11/780            00016480
C                                                                       00016490
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                                  00016500
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2         00016510
C          > 0 FOR ONE BLANK LINE BEFORE.                               00016520
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).     00016530
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).     00016540
C                                                                       00016550
C  MESSAGES ARE WRITTEN IN THE FORM:                                    00016560
C                                                                       00016570
C  {WARN}: _MSG_HERE_                                                   00016580
C                                                                       00016590
      CHARACTER*(*) MSG                                                 00016600
      I=LEN(MSG)                                                        00016610
      DO 1 J=1,2                                                        00016620
        IF(J.EQ.1) THEN                                                 00016630
       JUNIT=IUNIT1                                                     00016640
          ELSE                                                          00016650
       JUNIT=IUNIT2                                                     00016660
        ENDIF                                                           00016670
        IF(JUNIT.GT.0) THEN                                             00016680
          IF(ISKIP.EQ.0) THEN                                           00016690
            WRITE(JUNIT,2) MSG                                          00016700
            ELSE                                                        00016710
       WRITE(JUNIT,3) MSG                                               00016720
          ENDIF                                                         00016730
      ENDIF                                                             00016740
1     CONTINUE                                                          00016750
      RETURN 1                                                          00016760
2     FORMAT(1X,'{WARN}: ',A<I>)                                        00016770
3     FORMAT(/1X,'{WARN}: ',A<I>)                                       00016780
      END                                                               00016790
      REAL FUNCTION ASINH(X)                                            00016800
C--INVERSE HYPERBOLIC SIN FUNCTION                                      00016810
C                                                                       00016820
      REAL*8 X2                                                         00016830
      X2=X                                                              00016840
      ASINH=DLOG(X2+DSQRT(X2*X2+1.0D0))                                 00016850
      RETURN                                                            00016860
      END                                                               00016870
      SUBROUTINE DQUAD(A,B,RESULT,K,EPSIL,NPTS,ICHECK,F,MEV)            00016880
C--DOUBLE-PRECISION VERSION BY W.L.ANDERSON-- 9/21/81.                  00016890
      IMPLICIT REAL*8 (A-H,O-Z)                                         00016900
      DIMENSION FUNCT(127),P(381),RESULT(8)                             00016910
C THIS SUBROUTINE ATTEMPTS TO CALCULATE THE INTEGRAL OF F(X)            00016920
C OVER THE INTERVAL *A* TO *B* WITH RELATIVE ERROR NOT                  00016930
C EXCEEDING *EPSIL*.                                                    00016940
C THE RESULT IS OBTAINED USING A SEQUENCE OF 1,3,7,15,31,63,            00016950
C 127, AND 255 POINT INTERLACING FORMULAE(NO INTEGRAND                  00016960
```

```
C EVALUATIONS ARE WASTED) OF RESPECTIVE DEGREE 1,5,11,23,                00016970
C 47,95,191 AND 383. THE FORMULAE ARE BASED ON THE OPTIMAL               00016980
C EXTENSION OF THE 3-POINT GAUSS FORMULA.  DETAILS OF                    00016990
C THE FORMULAE ARE GIVEN IN *THE OPTIMUM ADDITION OF POINTS              00017000
C TO QUADRATURE FORMULAE* BY T.N.L. PATTERSON,MATHS.COMP.                00017010
C VOL 22,847-856,1968.                                                   00017020
C                      *** INPUT ***                                     00017030
C A        LOWER LIMIT OF INTEGRATION.                                   00017040
C B        UPPER LIMIT OF INTEGRATION.                                   00017050
C EPSIL    RELATIVE ACCURACY REQUIRED. WHEN THE RELATIVE                 00017060
C          DIFFERENCE OF TWO SUCCESSIVE FORMULAE DOES NOT                00017070
C          EXCEED *EPSIL* THE LAST FORMULA COMPUTED IS TAKEN             00017080
C          AS THE RESULT.                                                00017090
C F        F(X) IS THE INTEGRAND.                                        00017100
C                      *** OUTPUT ***                                    00017110
C RESULT   THIS ARRAY,WHICH SHOULD BE DECLARED TO HAVE AT                00017120
C          LEAST 8 ELEMENTS, HOLDS THE RESULTS OBTAINED BY               00017130
C          THE 1,3,7, ETC., POINT FORMULAE. THE NUMBER OF                00017140
C          FORMULAE COMPUTED DEPENTS ON *EPSIL*.                         00017150
C K        RESULT(K) HOLDS THE VALUE OF THE INTEGRAL TO THE              00017160
C          SPECIFIED RELATIVE ACCURACY.                                  00017170
C NPTS     NUMBER INTEGRAND EVALUATIONS.                                 00017180
C ICHECK   ON EXIT NORMALLY ICHECK=0. HOWEVER IF CONVERGENCE             00017190
C          TO THE ACCURACY REQUESTED IS NOT ACHIEVED ICHECK=1            00017200
C          ON EXIT.                                                      00017210
C MEV      MAX.ALLOWABLE EVALUATIONS BEFORE ACCEPTING RESULT             00017220
C            WITH NPTS>=MEV...                                           00017230
C ABSCISSAE AND WEIGHTS OF QUADRATURE RULES ARE STACKED IN               00017240
C ARRAY *P* IN THE ORDER IN WHICH THEY ARE NEEDED.                       00017250
      DATA                                                               00017260
     * P( 1),P( 2),P( 3),P( 4),P( 5),P( 6),P( 7),                       00017270
     * P( 8),P( 9),P(10),P(11),P(12),P(13),P(14),                       00017280
     * P(15),P(16),P(17),P(18),P(19),P(20),P(21),                       00017290
     * P(22),P(23),P(24),P(25),P(26),P(27),P(28)/                       00017300
     *  0.77459666924148337704D 00,0.55555555555555555556D 00,          00017310
     *  0.88888888888888888889D 00,0.26848808986833344073D 00,          00017320
     *  0.96049126870802028342D 00,0.10465622602646726519D 00,          00017330
     *  0.43424374934680255800D 00,0.40139741477596222291D 00,          00017340
     *  0.45091653865847414235D 00,0.13441525524378422036D 00,          00017350
     *  0.51603282997079739697D-01,0.20062852937698902103D 00,          00017360
     *  0.99383196321275502221D 00,0.17001719629940260339D-01,          00017370
     *  0.88845923287225699889D 00,0.92927195315124537686D-01,          00017380
     *  0.62110294673722640294D 00,0.17151190913639138079D 00,          00017390
     *  0.22338668642896688163D 00,0.21915685840158749640D 00,          00017400
     *  0.22551049979820668739D 00,0.67207754295990703540D-01,          00017410
     *  0.25807598096176653565D-01,0.10031427861179557877D 00,          00017420
     *  0.84345657393211062463D-02,0.46462893261757986541D-01,          00017430
     *  0.85755920049990351154D-01,0.10957842105592463824D 00/          00017440
      DATA                                                               00017450
     * P(29),P(30),P(31),P(32),P(33),P(34),P(35),                       00017460
     * P(36),P(37),P(38),P(39),P(40),P(41),P(42),                       00017470
     * P(43),P(44),P(45),P(46),P(47),P(48),P(49),                       00017480
     * P(50),P(51),P(52),P(53),P(54),P(55),P(56)/                       00017490
     *  0.99900981249676675976 6D 00,0.25447807915618744154D-02,        00017500
     *  0.98153114955374010687D 00,0.16446049854387810934D-01,          00017510
```

```
*  0.92965485742974005667D 00,0.35957103307129322097D-01,          00017520
*  0.83672593816886873550D 00,0.56979509494123357412D-01,          00017530
*  0.70249620649152707861D 00,0.76879620499003531043D-01,          00017540
*  0.53131974364437562397D 00,0.93627109981264473617D-01,          00017550
*  0.33113539325797683309D 00,0.10566989358023480974D 00,          00017560
*  0.11248894313318662575D 00,0.11195687302095345688D 00,          00017570
*  0.11275525672076869161D 00,0.33603877148207730542D-01,          00017580
*  0.12903800100351265626D-01,0.50157139305899537414D-01,          00017590
*  0.42176304415588548391D-02,0.23231446639910269443D-01,          00017600
*  0.42877960025007734493D-01,0.54789210527962865032D-01,          00017610
*  0.12651565562300680114D-02,0.82230079572359296693D-02,          00017620
*  0.17978551568128270333D-01,0.28489754745833548613D-01/          00017630
 DATA                                                             00017640
* P(57),P(58),P(59),P(60),P(61),P(62),P(63),                      00017650
* P(64),P(65),P(66),P(67),P(68),P(69),P(70),                      00017660
* P(71),P(72),P(73),P(74),P(75),P(76),P(77),                      00017670
* P(78),P(79),P(80),P(81),P(82),P(83),P(84)/                      00017680
*  0.38439810249455532039D-01,0.46813554990628012403D-01,          00017690
*  0.52834946790116519862D-01,0.55978436510476319408D-01,          00017700
*  0.99987288812035761194D 00,0.36322148184553065969D-03,          00017710
*  0.99720625937222195908D 00,0.25790497946856882724D-02,          00017720
*  0.98868475754742947994D 00,0.61155068221172463397D-02,          00017730
*  0.97218287474858179658D 00,0.10498246909621321898D-01,          00017740
*  0.94634285837340290515D 00,0.15406750466559497802D-01,          00017750
*  0.91037115695700429250D 00,0.20594233915912711149D-01,          00017760
*  0.86390793819369047715D 00,0.25869679327214746911D-01,          00017770
*  0.80694053195021761186D 00,0.31073551111687964880D-01,          00017780
*  0.73975604435269475868D 00,0.36064432780782572640D-01,          00017790
*  0.66290966002478059546D 00,0.40715510116944318934D-01,          00017800
*  0.57719571005204581484D 00,0.44914531653632197414D-01,          00017810
*  0.48361802694584102756D 00,0.48564330406673198716D-01/          00017820
 DATA                                                             00017830
* P( 85),P( 86),P( 87),P( 88),P( 89),P( 90),P( 91),               00017840
* P( 92),P( 93),P( 94),P( 95),P( 96),P( 97),P( 98),               00017850
* P( 99),P(100),P(101),P(102),P(103),P(104),P(105),               00017860
* P(106),P(107),P(108),P(109),P(110),P(111),P(112)/               00017870
*  0.38335932419873034692D 00,0.51583253952048458777D-01,          00017880
*  0.27774982202182431507D 00,0.53905499335266063927D-01,          00017890
*  0.16823525155220746498D 00,0.55481404356559363988D-01,          00017900
*  0.56344313046592789972D-01,0.56277699831254301273D-01,          00017910
*  0.56377628360384717388D-01,0.16801938574103865271D-01,          00017920
*  0.64519000501757369228D-02,0.25078569652949768707D-01,          00017930
*  0.21088152457266328793D-02,0.11615723319955134727D-01,          00017940
*  0.21438980012503867246D-01,0.27394605263981432516D-01,          00017950
*  0.63260731936263354422D-03,0.41115039786546930472D-02,          00017960
*  0.89892757840641357233D-02,0.14244877372916774306D-01,          00017970
*  0.19219905124727766019D-01,0.23406777495314006201D-01,          00017980
*  0.26417473395058259931D-01,0.27989218255238159704D-01,          00017990
*  0.18073956444538835782D-03,0.12895240826104173921D-02,          00018000
*  0.30577534101755311361D-02,0.52491234548088591251D-02/          00018010
 DATA                                                             00018020
* P(113),P(114),P(115),P(116),P(117),P(118),P(119),               00018030
* P(120),P(121),P(122),P(123),P(124),P(125),P(126),               00018040
* P(127),P(128),P(129),P(130),P(131),P(132),P(133),               00018050
* P(134),P(135),P(136),P(137),P(138),P(139),P(140)/               00018060
```

```
     *  0.77033752332797418482D-02,0.10297116957956355524D-01,          00018070
     *  0.12934839663607373455D-01,0.15536775555843982440D-01,          00018080
     *  0.18032216390391286320D-01,0.20357755058472159467D-01,          00018090
     *  0.22457265826816098707D-01,0.24282165203336599358D-01,          00018100
     *  0.25791626976024229388D-01,0.26952749667633031963D-01,          00018110
     *  0.27740702178279681994D-01,0.28138849915627150636D-01,          00018120
     *  0.99998243035489159858D 00,0.50536095207862517625D-04,          00018130
     *  0.99959879967191068325D 00,0.37774664632698466027D-03,          00018140
     *  0.99831663531840739253D 00,0.93836984854238150079D-03,          00018150
     *  0.99572410469840718851D 00,0.16811428654214699063D-02,          00018160
     *  0.99149572117810613240D 00,0.25687649437940203731D-02,          00018170
     *  0.98537149959852037111D 00,0.35728927835172996494D-02,          00018180
     *  0.97714151463970571416D 00,0.46710503721143217474D-02,          00018190
     *  0.96663785155841656709D 00,0.58434498758356395076D-02/          00018200
      DATA                                                            00018210
     * P(141),P(142),P(143),P(144),P(145),P(146),P(147),                00018220
     * P(148),P(149),P(150),P(151),P(152),P(153),P(154),                00018230
     * P(155),P(156),P(157),P(158),P(159),P(160),P(161),                00018240
     * P(162),P(163),P(164),P(165),P(166),P(167),P(168)/                00018250
     *  0.95373000642576113641D 00,0.70724899954335554680D-02,          00018260
     *  0.93832039777959288365D 00,0.83428387539681577056D-02,          00018270
     *  0.92034002547001242073D 00,0.96411777297025366953D-02,          00018280
     *  0.89974489977694003664D 00,0.10955733387837901648D-01,          00018290
     *  0.87651341448470526974D 00,0.12275830560082770087D-01,          00018300
     *  0.85064449476835027976D 00,0.13591571009765546790D-01,          00018310
     *  0.82215625436498040737D 00,0.14893641664815182035D-01,          00018320
     *  0.79108493379984836143D 00,0.16173218729577719942D-01,          00018330
     *  0.75748396638051363793D 00,0.17421930159464173747D-01,          00018340
     *  0.72142308537009891548D 00,0.18631848256138790186D-01,          00018350
     *  0.68298743109107922809D 00,0.19795495048097499488D-01,          00018360
     *  0.64227664250975951377D 00,0.20905851445812023852D-01,          00018370
     *  0.59940393024224289297D 00,0.21956366305317824939D-01,          00018380
     *  0.55449513263193254887D 00,0.22940964229387748761D-01/          00018390
      DATA                                                            00018400
     * P(169),P(170),P(171),P(172),P(173),P(174),P(175),                00018410
     * P(176),P(177),P(178),P(179),P(180),P(181),P(182),                00018420
     * P(183),P(184),P(185),P(186),P(187),P(188),P(189),                00018430
     * P(190),P(191),P(192),P(193),P(194),P(195),P(196)/                00018440
     *  0.50768775753371660215D 00,0.23854052106038540080D-01,          00018450
     *  0.45913001198983233287D 00,0.24690524744487676909D-01,          00018460
     *  0.40897982122988867241D 00,0.25445769965464765813D-01,          00018470
     *  0.35740383783153215238D 00,0.26115673376706097680D-01,          00018480
     *  0.30457644155671404334D 00,0.26696622927450359906D-01,          00018490
     *  0.25067873030348317661D 00,0.27185513229624791819D-01,          00018500
     *  0.19589750271110015392D 00,0.27579749566481873035D-01,          00018510
     *  0.14042423315256017459D 00,0.27877251476613701609D-01,          00018520
     *  0.84454040083710883710D-01,0.28076455793817246607D-01,          00018530
     *  0.28184648949745694339D-01,0.28176319033016602131D-01,          00018540
     *  0.28188814180192358694D-01,0.84009692870519326354D-02,          00018550
     *  0.32259500250878684614D-02,0.12539284826474884353D-01,          00018560
     *  0.10544076228633167722D-02,0.58078616599775673635D-02,          00018570
     *  0.10719490006251933623D-01,0.13697302631990716258D-01/          00018580
      DATA                                                            00018590
     * P(197),P(198),P(199),P(200),P(201),P(202),P(203),                00018600
     * P(204),P(205),P(206),P(207),P(208),P(209),P(210),                00018610
```

```
     * P(211),P(212),P(213),P(214),P(215),P(216),P(217),                  00018620
     * P(218),P(219),P(220),P(221),P(222),P(223),P(224)/                  00018630
     *  0.31630366082226447689D-03,0.20557519893273465236D-02,            00018640
     *  0.44946378920320678616D-02,0.71224386864583871532D-02,            00018650
     *  0.96099525623638830097D-02,0.11703388747657003101D-01,            00018660
     *  0.13208736697529129966D-01,0.13994609127619079852D-01,            00018670
     *  0.90372734658751149261D-04,0.64476204130572477933D-03,            00018680
     *  0.15288767050877655684D-02,0.26245617274044295626D-02,            00018690
     *  0.38516876166398709241D-02,0.51485584789781777618D-02,            00018700
     *  0.64674198318036867274D-02,0.77683877779219912200D-02,            00018710
     *  0.90161081951956431600D-02,0.10178877529236079733D-01,            00018720
     *  0.11228632913408049354D-01,0.12141082601668299679D-01,            00018730
     *  0.12895813488012114694D-01,0.13476374833816515982D-01,            00018740
     *  0.13870351089139840997D-01,0.14069424957813575318D-01,            00018750
     *  0.25157870384280661489D-04,0.18887326450650491366D-03,            00018760
     *  0.46918492424785040975D-03,0.84057143271072246365D-03/            00018770
      DATA                                                              00018780
     * P(225),P(226),P(227),P(228),P(229),P(230),P(231),                  00018790
     * P(232),P(233),P(234),P(235),P(236),P(237),P(238),                  00018800
     * P(239),P(240),P(241),P(242),P(243),P(244),P(245),                  00018810
     * P(246),P(247),P(248),P(249),P(250),P(251),P(252)/                  00018820
     *  0.12843824718970101768D-02,0.17864463917586498247D-02,            00018830
     *  0.23355251860571608737D-02,0.29217249379178197538D-02,            00018840
     *  0.35362449977167777340D-02,0.41714193769840788528D-02,            00018850
     *  0.48205888648512683476D-02,0.54778666939189508240D-02,            00018860
     *  0.61379152800413850435D-02,0.67957855048827733948D-02,            00018870
     *  0.74468208324075910174D-02,0.80866093647888599710D-02,            00018880
     *  0.87109650797320868736D-02,0.93159241280693950932D-02,            00018890
     *  0.98977475240487497440D-02,0.10452925722906011926D-01,            00018900
     *  0.10978183152658912470D-01,0.11470482114693874380D-01,            00018910
     *  0.11927026053019270040D-01,0.12345262372243838455D-01,            00018920
     *  0.12722884982732382906D-01,0.13057836688353048840D-01,            00018930
     *  0.13348311463725179953D-01,0.13592756614812395910D-01,            00018940
     *  0.13789874783240936517D-01,0.13938625738306850804D-01,            00018950
     *  0.14038227896908623303D-01,0.14088159516508301065D-01/            00018960
      DATA                                                              00018970
     * P(253),P(254),P(255),P(256),P(257),P(258),P(259),                  00018980
     * P(260),P(261),P(262),P(263),P(264),P(265),P(266),                  00018990
     * P(267),P(268),P(269),P(270),P(271),P(272),P(273),                  00019000
     * P(274),P(275),P(276),P(277),P(278),P(279),P(280)/                  00019010
     *  0.99999759637974846462D 00,0.69379364324108267170D-05,            00019020
     *  0.99994399620705437576D 00,0.53275293669780613125D-04,            00019030
     *  0.99976049092443204733D 00,0.13575491094922871973D-03,            00019040
     *  0.99938033802502358193D 00,0.24921240048299729402D-03,            00019050
     *  0.99874561446809511470D 00,0.38974528447328229322D-03,            00019060
     *  0.99780535449595727456D 00,0.55429531493037471492D-03,            00019070
     *  0.99651414591489027385D 00,0.74028280424450333046D-03,            00019080
     *  0.99483150280062100052D 00,0.94536151685852538246D-03,            00019090
     *  0.99272134428278861533D 00,0.11674841174299594077D-02,            00019100
     *  0.99015137040077015918D 00,0.14049079956551446427D-02,            00019110
     *  0.98709252795403406719D 00,0.16561127281544526052D-02,            00019120
     *  0.98351865757863272876D 00,0.19197129710138724125D-02,            00019130
     *  0.97940628167086268381D 00,0.21944069253638388388D-02,            00019140
     *  0.97473445975240266776D 00,0.24789582266575679307D-02/            00019150
      DATA                                                              00019160
```

```
     * P(281),P(282),P(283),P(284),P(285),P(286),P(287),                00019170
     * P(288),P(289),P(290),P(291),P(292),P(293),P(294),                00019180
     * P(295),P(296),P(297),P(298),P(299),P(300),P(301),                00019190
     * P(302),P(303),P(304),P(305),P(306),P(307),P(308)/                00019200
     *  0.96948465950245923177D 00,0.27721957645934509940D-02,          00019210
     *  0.96364062156981213252D 00,0.30730184347025783234D-02,          00019220
     *  0.95718821610986096274D 00,0.33803979910869203823D-02,          00019230
     *  0.95011529752129487656D 00,0.36933779170256508183D-02,          00019240
     *  0.94241156519108305981D 00,0.40110687240750233989D-02,          00019250
     *  0.93406843615772578800D 00,0.43326409680929828545D-02,          00019260
     *  0.92507893290707565236D 00,0.46573172997568547773D-02,          00019270
     *  0.91543758715576504064D 00,0.49843645647655386012D-02,          00019280
     *  0.90514035881326159519D 00,0.53130866051870565663D-02,          00019290
     *  0.89418456833555902286D 00,0.56428181013844441585D-02,          00019300
     *  0.88256884024734190684D 00,0.59729195655081658049D-02,          00019310
     *  0.87029305554811390585D 00,0.63027734490857587172D-02,          00019320
     *  0.85735831088623215653D 00,0.66317812429018878941D-02,          00019330
     *  0.84376688267270860104D 00,0.69593614093904229394D-02/          00019340
      DATA                                                            00019350
     * P(309),P(310),P(311),P(312),P(313),P(314),P(315),                00019360
     * P(316),P(317),P(318),P(319),P(320),P(321),P(322),                00019370
     * P(323),P(324),P(325),P(326),P(327),P(328),P(329),                00019380
     * P(330),P(331),P(332),P(333),P(334),P(335),P(336)/                00019390
     *  0.82952219463740140018D 00,0.72849479805538070639D-02,          00019400
     *  0.81462878765513741344D 00,0.76079896657190565832D-02,          00019410
     *  0.79909229096084140180D 00,0.79279493342948491103D-02,          00019420
     *  0.78291939411828301639D 00,0.82443037630328680306D-02,          00019430
     *  0.76611781930376009072D 00,0.85565435613076896192D-02,          00019440
     *  0.74869629361693660282D 00,0.88641732094824942641D-02,          00019450
     *  0.73066452124218126133D 00,0.91667111635607884067D-02,          00019460
     *  0.71203315536225203459D 00,0.94636899938300652943D-02,          00019470
     *  0.69281376977911470289D 00,0.97546565363174114611D-02,          00019480
     *  0.67301883023041847920D 00,0.10039172044056840798D-01,          00019490
     *  0.65266166541001749610D 00,0.10316812330947621682D-01,          00019500
     *  0.63175643711119423041D 00,0.10587167904885197931D-01,          00019510
     *  0.61031811371518640016D 00,0.10849844089337314099D-01,          00019520
     *  0.58836243444766254143D 00,0.11104461134006926537D-01/          00019530
      DATA                                                            00019540
     * P(337),P(338),P(339),P(340),P(341),P(342),P(343),                00019550
     * P(344),P(345),P(346),P(347),P(348),P(349),P(350),                00019560
     * P(351),P(352),P(353),P(354),P(355),P(356),P(357),                00019570
     * P(358),P(359),P(360),P(361),P(362),P(363),P(364)/                00019580
     *  0.56590588542365442262D 00,0.11350654315980596602D-01,          00019590
     *  0.54296566649831149049D 00,0.11588074033043952568D-01,          00019600
     *  0.51955966153745702199D 00,0.11816385890830235763D-01,          00019610
     *  0.49570640791876146017D 00,0.12035270785279562630D-01,          00019620
     *  0.47142506587165887693D 00,0.12244424981611985899D-01,          00019630
     *  0.44673538766202847374D 00,0.12443560190714035263D-01,          00019640
     *  0.42165768662616330006D 00,0.12632403643542078765D-01,          00019650
     *  0.39621280605761593918D 00,0.12810698163877361967D-01,          00019660
     *  0.37042208795007823014D 00,0.12978202239537399286D-01,          00019670
     *  0.34430734159943802278D 00,0.13134690091960152836D-01,          00019680
     *  0.31789081206847668318D 00,0.13279951743930530650D-01,          00019690
     *  0.29119514851824668196D 00,0.13413793085110098513D-01,          00019700
     *  0.26424337241092676194D 00,0.13536035934956213614D-01,          00019710
```

```
     *  0.23705884558982972721D 00,0.13646518102571291428D-01/
      DATA
     * P(365),P(366),P(367),P(368),P(369),P(370),P(371),
     * P(372),P(373),P(374),P(375),P(376),P(377),P(378),
     * P(379),P(380),P(381)/
     *  0.20966523824318119477D 00,0.13745093443001896632D-01,
     *  0.18208649675925219825D 00,0.13831631909506428676D-01,
     *  0.15434681148137810869D 00,0.13906019601325461264D-01,
     *  0.12647058437230196685D 00,0.13968158806516938516D-01,
     *  0.98482396598119202090D-01,0.14017968039456608810D-01,
     *  0.70406976042855179063D-01,0.14055382072649964277D-01,
     *  0.42269164765363603212D-01,0.14080351962553661325D-01,
     *  0.14093886410782462614D-01,0.14092845069160408355D-01,
     *  0.14094407090096179347D-01/
      ICHECK = 0
C CHECK FOR TRIVIAL CASE.
      IF (A.EQ.B) GO TO 70
C SCALE FACTORS.
      SUM = (B+A)/2.0D0
      DIFF = (B-A)/2.0D0
C 1-POINT GAUSS
      FZERO = F(SUM)
      RESULT(1) = 2.0D0*FZERO*DIFF
      I = 0
      IOLD = 0
      INEW = 1
      K = 2
      ACUM =0.0D0
      GO TO 30
   10 IF (K.EQ.8) GO TO 50
      IF(INEW+IOLD.GE.MEV) GO TO 60
      K = K + 1
      ACUM =0.0D0
C CONTRIBUTION FROM FUNCTION VALUES ALREADY COMPUTED.
      DO 20 J=1,IOLD
        I = I + 1
        ACUM = ACUM + P(I)*FUNCT(J)
   20 CONTINUE
C CONTRIBUTION FROM NEW FUNCTION VALUES.
   30 IOLD = IOLD + INEW
      DO 40 J=INEW,IOLD
        I = I + 1
        X = P(I)*DIFF
        FUNCT(J) = F(SUM+X) + F(SUM-X)
        I = I + 1
        ACUM = ACUM + P(I)*FUNCT(J)
   40 CONTINUE
      INEW = IOLD + 1
      I = I + 1
      RESULT(K) = (ACUM+P(I)*FZERO)*DIFF
C CHECK FOR CONVERGENCE.
      IF(DABS(RESULT(K)-RESULT(K-1))-EPSIL*DABS(RESULT(K)))
     1 60,60,10
C CONVERGENCE NOT ACHIEVED.
   50 ICHECK = 1
```

```
C NORMAL TERMINATION.                                                  00020270
   60 NPTS = INEW + IOLD                                               00020280
      RETURN                                                           00020290
C TRIVIAL CASE                                                         00020300
   70 K = 2                                                            00020310
      RESULT(1) =0.0D0                                                 00020320
      RESULT(2) =0.0D0                                                 00020330
      NPTS = 0                                                         00020340
      RETURN                                                           00020350
      END                                                              00020360
      FUNCTION ERF(X)                                                  00020370
C                                                                      00020380
C  ERF COMPUTES THE ERROR FUNCTION TO ABOUT 7-PLACES.                  00020390
C  SEE MATH. OF COMP., V.22,N.101,JAN,1968.                            00020400
C   ALSO, SEE ERFINV(X).                                               00020410
C                                                                      00020420
      DIMENSION A1(19),A2(19)                                          00020430
      DATA A1/.70322500,.33050152,.20133975,.10863025,                 00020440
     1 .46775523E-1,.15398573E-1,.38015077E-2,.69718379E-3,            00020450
     2 .94490927E-4,.94328117E-5,.69192752E-6,.37225234E-7,            00020460
     3 .14666061E-8,.42261614E-10,.88978652E-12,.13676044E-13,         00020470
     4 .15334234E-15,.12536751E-17,.74517E-20/                         00020480
      DATA A2/.24725517,.14422723,.86989455E-1,.43977338E-1,           00020490
     1 .17243963E-1,.50790696E-2,.11086065E-2,.17822802E-3,            00020500
     2 .21040458E-4,.18206632E-5,.11533099E-6,.53427503E-8,            00020510
     3 .18084859E-9,.44696823E-11,.80606884E-13,.10601364E-14,         00020520
     4 .10164928E-16,.710005E-19,0.0/                                  00020530
      IF(X.EQ.0.0) THEN                                                00020540
        ERF=0.0                                                        00020550
        RETURN                                                         00020560
      ENDIF                                                            00020570
        B=2.*X/5.                                                      00020580
        S=SIN(B)                                                       00020590
        C=COS(B)                                                       00020600
        C2=C+C                                                         00020610
        ALP=C2*C-1.                                                    00020620
        SUM=0.0                                                        00020630
        DO 10 N=1,19                                                   00020640
          SUM=SUM+(A1(N)+C2*A2(N))*ALP**(N-1)                          00020650
10      CONTINUE                                                       00020660
        ERF=B/3.1415927+S*SUM                                          00020670
        RETURN                                                         00020680
      END                                                              00020690
      FUNCTION ERFINV(Y)                                               00020700
C                                                                      00020710
C  ERFINV COMPUTES THE INVERSE ERROR FUNCTION TO ABOUT 7-PLACES.       00020720
C  SEE MATH. OF COMP., V.22,N.101,JAN,1968.                            00020730
C  ALSO, SEE ERF(X).                                                   00020740
C                                                                      00020750
      CHARACTER*16 XX                                                  00020760
      DIMENSION T3(1:38),T4(0:26),T5(0:37),T6(0:25)                    00020770
      DATA T3/.12046752,.16078199E-1,.26867044E-2,.49963473E-3,        00020780
     1 .98898219E-4,.20391813E-4,.43272716E-5,.93808141E-6,            00020790
     2 .20673472E-6,.46159699E-7,.10416680E-7,.23715100E-8,            00020800
     3 .54392841E-9,.12554899E-9,.29138180E-10,.67949422E-11,          00020810
```

```
     4 .15912343E-11,.37402505E-12,.88208776E-13,.20865090E-13,          00020820
     5 .49488041E-14,.11766395E-14,.28038557E-15,.66950664E-16,          00020830
     6 .16016550E-16,.38382583E-17,.9212851E-18,.2214615E-18,            00020840
     7 .533091E-19,.128488E-19,.31006E-20,.7491E-21,.1812E-21,           00020850
     8 .439E-22,.106E-22,.26E-23,.6E-24,.2E-24/                          00020860
      DATA T4/.91215880,-.16266282E-1,.43355647E-3,.21443857E-3,         00020870
     1 .26257511E-5,-.30210911E-5,-.12406061E-7,.62406609E-7,            00020880
     2 -.54012479E-9,-.14232079E-8,.34384028E-10,.33584870E-10,          00020890
     3 -.14584289E-11,-.81021743E-12,.52532409E-13,.19711541E-13,        00020900
     4 -.17494334E-14,-.48005966E-15,.55730299E-16,.11632605E-16,        00020910
     5 -.17262489E-17,-.2784973E-18,.524481E-19,.65270E-20,              00020920
     6 -.15707E-20,-.1475E-21,.450E-22/                                  00020930
      DATA T5/.95667971,-.23107004E-1,-.43742361E-2,-.57650342E-3,       00020940
     1 -.10961022E-4,.25108547E-4,.10562336E-4,.27544123E-5,             00020950
     2 .43248450E-6,-.20530336E-7,-.43891537E-7,-.17684010E-7,           00020960
     3 -.39912890E-8,-.18693241E-9,.27292274E-9,.13281721E-9,            00020970
     4 .31834248E-10,.16700608E-11,-.20364650E-11,-.96484681E-12,        00020980
     5 -.21956727E-12,-.95689813E-14,.13703257E-13,.62538505E-14,        00020990
     6 .14584615E-14,.10781240E-15,-.70922999E-16,-.39141178E-16,        00021000
     7 -.11165921E-16,-.15770366E-17,.2853149E-18,.2716662E-18,          00021010
     8 .957770E-19,.176835E-19,-.9828E-21,-.20464E-20,-.802E-21,         00021020
     9 -.1650E-21/                                                       00021030
      DATA T6/.98857506,.10857705E-1,-.17511651E-2,.21196993E-4,         00021040
     1 .15664871E-4,-.51904169E-5,-.37135790E-7,.12174309E-8,            00021050
     2 -.17681155E-9,-.11937218E-10,.38025054E-12,-.66018832E-13,        00021060
     3 -.87917055E-14,-.35068693E-15,-.69722150E-16,-.10956794E-16,      00021070
     4 -.11536390E-17,-.1326235E-18,-.263938E-19,.5341E-21,              00021080
     5 -.2261E-20,.9552E-21,-.525E-21,.2487E-21,-.1134E-21,.42E-22/      00021090
      X=Y                                                                00021100
      X1=ABS(X)                                                          00021110
      IF(X1.GE.1.0) THEN                                                 00021120
        ENCODE(16,1,XX) X1                                               00021130
1       FORMAT(E16.8)                                                    00021140
        IF(X1.GT.1.000001)CALL ERRMSG('ABS(X)='//XX//                    00021150
     1    ' >1.000001 IN [ERFINV]',0,6,0)                                00021160
        CALL WARN('ABS(X)='//XX//                                        00021170
     2 ' >=1.0 IN [ERFINV]; X=0.9999998*SIGN(1.,X) USED.',0,6,0,*2)      00021180
2       X=0.9999998*SIGN(1.,X)                                           00021190
      ENDIF                                                              00021200
      X1=1.-X                                                            00021210
      IF(X.GE.0.8.AND.X.LE.0.9975) THEN                                  00021220
        BETA=SQRT(-ALOG(1.-X*X))                                         00021230
        R=0.0                                                            00021240
        DO 10 N=0,26                                                     00021250
10        R=R+T4(N)*TCHEB(N,-1.54881304*BETA+2.5654901)                  00021260
        ERFINV=BETA*R                                                    00021270
      ELSE IF(X1.GE.5E-16.AND.X1.LE.25E-4) THEN                          00021280
        BETA=SQRT(-ALOG(1.-X*X))                                         00021290
        R=0.0                                                            00021300
        DO 20 N=0,37                                                     00021310
20        R=R+T5(N)*TCHEB(N,-.55945763*BETA+2.2879157)                   00021320
        ERFINV=BETA*R                                                    00021330
      ELSE IF(X1.LT.5E-16) THEN                                          00021340
        BETA=SQRT(-ALOG(1.-X*X))                                         00021350
        SBETA=SQRT(BETA)                                                 00021360
```

```
        R=0.0                                                          00021370
        DO 30 N=0,25                                                   00021380
30        R=R+T6(N)*TCHEB(N,-9.1999924/SBETA+2.7949908)                00021390
        ERFINV=BETA*R                                                  00021400
      ELSE                                                             00021410
        R=0.0                                                          00021420
        A=X*X/.32-1.                                                   00021430
        DO 40 N=1,38                                                   00021440
40        R=R+T3(N)*TCHEB(N,A)                                         00021450
        ERFINV=X*(.99288538+R)                                         00021460
      ENDIF                                                            00021470
      RETURN                                                           00021480
      END                                                              00021490
      INTEGER FUNCTION LOC(I,J)                                        00021500
C--GETS ACTUAL ADDR OF A(I,J)=A(J,I) SYMMETRIC MATRIX                  00021510
C  STORED AS THE VECTOR A(LOC(I,J)) OF N*(N+1)/2 ELEMENTS--            00021520
C  WHERE ANY I,J.LE.N MAY BE USED (N NOT EXPLICITLY NEEDED)...         00021530
C                                                                      00021540
      IF(I-J) 10,20,20                                                 00021550
   10 LOC=I+(J*J-J)/2                                                  00021560
      RETURN                                                           00021570
   20 LOC=J+(I*I-I)/2                                                  00021580
      RETURN                                                           00021590
      END                                                              00021600
      SUBROUTINE NL2SOL(N, P, X, CALCR, CALCJ, IV, V, UIPARM, URPARM,  00021610
     1                  UFPARM)                                        00021620
```

$$$ Because of the length of NL2SOL and related subprograms, the rest of the listing has been suppressed; however, the complete code is available on the distributed tape.