U.S. Department of the Interior
Geological Survey

Adaptive nonlinear least-squares solution for constrained or unconstrained minimization problems (Subprogram NLSOL)

by

Walter L. Anderson

Open-File Report 82-68

1982

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 author 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

INTRODUCTION

Subprogram NLSOL is a general interface routine written for the VAX-11/780 system using a comprehensive adaptive nonlinear least-squares algorithm published by Dennis and others (1979)*. The intent of this interface is to provide a uniform calling approach for any nonlinear function, and to extend Dennis's unconstrained minimization algorithm, external to the original (unchanged) code, with the following additional options:

(1) To perform either unconstrained or constrained adaptive nonlinear regression for arbitrary nonlinear least-squares problems. This includes defining partial or full (lower,higher) parameter bounds and parameter scaling during the regression analysis.

(2) To hold certain parameters fixed (i.e., assigned as constants) during the nonlinear least-squares regression. (This amounts to another form of constraining solution space.)

(3) To provide for a weighted least-squares fit.

(4) To control reading the observed data matrix using any object (run-time) format.

---

* It is suggested that the reader obtain a copy of Dennis and others (1979) before proceeding. We will not repeat details of this algorithm in the present paper.

(5) To include many I/O-options (beyond those already provided by Dennis and others, 1979), such as the parameter correlation matrix, standard (linear) error estimates of the parameter solutions, root-mean-square (RMS) error, and a complete printout of the observed, calculated, residual, and the data matrix row for each observation.

(6) To use either estimated partial derivatives or analytical coded partial derivatives to evaluate the Jacobian matrix.

(7) Plus, a duplicate print-type output file for use in any subsequent processing (e.g., for plot routines, etc.).

Another important goal of the NLSOL interface routine was to provide compatibility with several previously used general least-squares algorithms. Specifically, these are subprograms MARQRT (as used by Anderson, 1980a) and IMSLMQ (as used by Anderson, 1980b). In addition to using the same parameter names, subprograms NLSOL, MARQRT, and IMSLMQ use the same parameter file (containing a title, $PARMS, object-time format, and $INIT parameters) and observed data matrix file. Hence, one may easily switch (if desired) between different algorithms at run time (using the same parameters and data). However, it is anticipated that NLSOL will be more widely used in the future due to its generality and ability to converge to a solution where the other algorithms sometimes fail. [At this point, the reader should read the introduction portion of Dennis and others (1979, p.1).]

SUMMARY OF CALCULATIONS

The complete mathematical details in Dennis and others (1979, p.2-23) describe the logic used by the adaptive nonlinear least-squares algorithm, including some results run from a variety of test problems found in the literature. [The reader should become familiar with the notation and terminology used in Dennis and others (1979), which will be adopted in the remainder of this report.]

The NLSOL interface, after reading and checking the nondefault parameters (see $PARMS below), will switch to either Dennis's analytic derivative routine NL2ITR (if IDER=0) or to the finite-difference routine NL2SNO (if IDER=1). However, the user's code to evaluate the residual vector R(X) and/or the analytical Jacobian matrix J(X) do not follow the prescription given in Dennis and others (1979, p.27-30). The interface routine NLSOL has a slightly different calling sequence, which is defined in Appendix 3 (e.g., see user code requirements for MAIN, FCODE, PCODE, SUBZ, and SUBEND).

In order to modify an unconstrained optimization algorithm to handle simple parameter bounds, such as inequality constraints, a set of parameter transformations (and inverse transformations) can be defined. We could also use more elaborate techniques, such as "gradient projection" methods (e.g., see Bard, 1974, p.146); however, this method would require significant changes to the basic unconstrained algorithm. For program flexibility, it is advantageous to

include any special operational changes external to the original algorithm. Thus various parameter transformations for constrained minimization was included in the interface subroutine NLSOL and associated subprograms, and did not require any modifications to Dennis's original unconstrained algorithm.

A summary of the parameter transformation options (via parameter SP) is given in Table 1.

Table 1.-- NLSOL constrained solution options

SP	Minimization Type*	Desired B-Range	Unconstrained Routine Transform ($-\infty<Xj<\infty$);	Xj Mapping and Inverse
0	unconstrained unscaled	$-\infty<Bj<\infty$	Xj=Bj, j=1,...,K;	Bj=Xj
1	partial constrained, scaled	$0<Bj<\infty$	Xj=ALOG(Bj);	Bj=EXP(Xj)
2	unconstrained scaled	$-\infty<Bj<\infty$	Xj=ASINH(Bj);	Bj=SINH(Xj)
3	full constrained, scaled	$BLj\leq Bj \leq BHj$	Xj=ARCSIN[SQRT ((Bj-BLj)/(BHj-BLj))];	Bj=BLj+ (BHj-BLj)* SIN(Xj)**2
4	full constrained, scaled	$BLj\leq Bj \leq BHj$	Xj=ERFINV[2* ((Bj-BLj)/(BHj-BLj)) -1];	Bj=BLj+0.5* (BHj-BLj)* (1+ERF Xj)

* For all SP, any Bj may be held fixed (see $PARMS IP, IB).

The unconstrained algorithm always uses the mapped or unconstrained parameters Xj (for any value of SP). However, the user's function subprogram (FCODE) must return the nonlinear function evaluation using only the constrained parameters Bj. This forward and inverse transformation is performed automatically within the NLSOL interface, and

therefore, may be ignored by the user.

In addition to the optional transformations given in Table 1, the NLSOL interface routine must calculate the appropriate partial derivative transformations using the derivative chain-rule when analytic derivatives are selected (IDER=0). This is summarized as follows:

SP=0, G=F(Bj), $\partial G/\partial Bj = \partial F/\partial Bj$ (j=1,2,...,K).

SP=1, G=F(Xj), $\partial G/\partial Bj = Bj\ \partial F/\partial Bj$.

SP=2, G=F(Xj), $\partial G/\partial Bj = Pj\ \partial F/\partial Bj$, where

$$Pj = \{[Bj+(Bj^{**}2+1)^{1/2}]+ 1/[Bj+(Bj^{**}2+1)]^{1/2}\}/2.$$

SP=3, G=F(Xj), $\partial G/\partial Bj = Sj\ \partial F/\partial Bj$, where

$$Sj = 2[(Bj-BLj)(BHj-Bj)]^{1/2}.$$

SP=4, G=F(Xj), $\partial G/\partial Bj = Tj\ \partial F/\partial Bj$, where

$$Tj = (BHj-BLj)EXP\{-[ERFINV(2[Bj-BLj]/[BHj-BLj]-1)]^2\}/\pi^{1/2}.$$

For a full (inequality parameter bounds) constrained least-squares, SP=4 may have slightly better "probability qualities" than SP=3, as reported in the IMSL documentation (IMSL, 1979, p. Z-3 and p. Z-4). However, from a practical point of view, SP=3 performs somewhat faster than SP=4, and usually converges to a constrained solution as well as SP=4 in most cases.

The residual vector R(X) required by the original adaptive algorithm (Dennis and others, 1979, p.27) can be expressed as a sum of "weighted residuals" by writing,

$$2f(X)=\sum_{i=1}^{N} Wi\ Ri(Xj)**2=\sum_{i=1}^{N} [Wi^{1/2}\ Ri(Xj)]**2, \text{ where}$$

Ri(Xj)=Yi-F is the i-th observed-calculated residual,

Wi=1/Si**2 is the weight for observation i, and

Si= the standard deviation of observation i (Si**2 is also called the variance).

Note that if any Wi is not unity (assumed by default), then the partial derivative must be scaled as

$$2\ \partial f/\partial Xj = -Wj^{1/2}\ \partial F/\partial Xj; \text{ otherwise,}$$

$$2\ \partial f/\partial Xj = -\partial F/\partial Xj.$$

The method used in NLSOL to incorporate analytical coded partial derivatives requires a user coded subprogram (see PCODE in Appendix 3). In addition to maintaining compatibility with previous codes used by Anderson (1980a, 1980b), this approach, coupled with the adaptive algorithm, can be easily merged using the "reverse communication" option, called NL2ITR (Dennis and others, 1979, p.38). The reverse communication method requires less storage than the original NL2SOL procedure, but also insures that PCODE (when required) will always be called immediately after a call to FCODE. Since some existing inverse programs used values passed in COMMON between FCODE and PCODE at each observation, the reverse technique required no changes to the users logic while using the NLSOL interface (which was one of the goals).

If the user does not have (nor desires to write) a correct PCODE subprogram to evaluate ∂F/∂Bj, then NLSOL can still be used with only the FCODE subprogram by taking the estimated derivative option, IDER=1. In this case, PCODE is a dummy name (and is never called) in the NLSOL interface, which in turn calls the adaptive algorithm using finite-difference Jacobians via NL2SNO (see Dennis and others, 1979, p.35-36). The finite-difference step is controlled by parameters V(31), V(34), and V(35) as described in Dennis and others (1979, p.33-34), and when $PARMS parameter IDER=1 is set. In most cases, the default values supplied are generally sufficient, assuming single-precision arithmetic is used in FCODE, and of course, FCODE is correctly written for the given nonlinear function.

PARAMETERS AND DATA REQUIRED

Parameters required by subprogram NLSOL are read using a FORTRAN NAMELIST simulator on the VAX (currently, VAX FORTRAN-77 Version 2.3 does not contain NAMELIST I/O; see subroutine NAMELIST in Appendix 3 for more details). The namelist names used are $PARMS and $INIT (the latter is an optional NAMELIST that may be used in subroutine SUBZ as described in Appendix 3). Default values are assumed whenever any $PARMS parameter is omitted, except as noted otherwise. Preceding the $PARMS statement is an 80-character title.

The general input order read by subprogram NLSOL is as follows:

1. Title record (always required, maximum of 80-characters).
2. $PARMS --nondefault parameters--$END. Note that $PARMS may begin in column 1 but cannot exceed column 72; parameters may be continued on successive records until the final $ or $END is encountered (the "END" in $END is optional).
3. An object (run-time) format statement defining the format of the input data matrix, where the object format begins in column 1, and ends before column 73. The object format is delimited by left- and right-parentheses; e.g., (2F10.0)
4. Optionally, the data matrix read under the object format may be inserted here if the alternate data file is not used (i.e., if parameter IALT=5 is specified).
5. $INIT --nondefault optional parameters--$END. This step is controlled by subroutine SUBZ (see Appendix 3), and may be omitted in some cases.
6. Optionally, subsequent runs using the same data matrix, but with changed $PARMS and $INIT parameters, may be made by repeating steps 1-3, and step 5, provided parameters ISTOP=0 and IALT is not 5.

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

PROGRAM FILES

FOR005: Title, $PARMS input parameters, and object-time format (defining the input format of the data matrix read on unit IALT).

FOR006: Output on-line terminal file (see parameter IOUT options).

FOR010: Input data matrix file (default IALT=10) read under the object-time format given in FOR005, step 3 above. Parameter IALT may be changed to any file number other than 04,06,13, or 16. Note that IALT=5 means that the data matrix must be included immediately after the object-time format statement on FOR005, step 4 above.

FOR016: Output duplicate (master) print-type disk file, containing all of FOR006, plus other detailed output (unless parameter IOUT=0 is set).

DETAILED PARAMETER DEFINITIONS

The $PARMS parameters described below are for any nonlinear least-squares problem of the general form,

R(F)= SUM[I=1 to N] {WT(I)*(Y(I)-F)**2}, where

R(F)= Residual function of F to be minimized;
WT(I)= Weight of observation I, WT(I)= 1/Si**2,
Si= Standard deviation of observation I;
Y(I)= Observed dependent variable at observation I;

F= F[X(I,L),L=1,2,..,M; B(J),J=1,2,...,K)] is any twice-continuously differentiable nonlinear function of unknown parameters B(J);

X(I,L)= Observed independent variables (L=1,2,..,M) at observation I;

B(J)= Unknown (nonlinear) parameters in F;

N= Number of observations;

M= Number of independent variables; and

K= Number of unknown parameters in F.

Appropriate changes in terminology for some $PARMS may be needed in programs that call NLSOL, insofar as the particular problem definitions are concerned (e.g., see Anderson, 1980a, 1980b).

$PARMS parameters (nondefault parameters must be given):

N= Number of observed dependent real values Y(I), I=1,2,...,N, where N≤500*.

M= Number of observed independent real variables X given in the input data matrix: ((Y(I),X(I,L),L=1,M),I=1,N), where 1≤M≤4*. [Also see the use of X(I,M+1) when using IWT>0 below.]

K= Total number of real parameters in the nonlinear function F(X;B(J),J=1,2,...,K), where 1≤K≤20* and K≤N. (Note that K<N is normal for least-squares problems.)

---

* The maximum limits set for N,M, and K are arbitrary, and can be easily changed in the FORTRAN-77 code (see the PARAMETER statements in Appendix 3 and comment (9) in Appendix 1).

B()= Array of initial (estimated) parameters in the nonlinear function F, given in ascending order B(J),J=1,2,...,K. (The initial B-array must be supplied by the user--and should be a reasonable estimate.)

IP= Number of parameters held fixed (if any) as constants in F, and as specified by index values given in array-IB (see below), where IP<K and N≥K-IP. Note that N-(K-IP) is the number of degrees of freedom in the problem. (The default is IP=0, which means that no parameters are held fixed.)

IB()= Array of IP-indicies (in any order, 1 up to K) corresponding to any parameter B(J) to hold fixed at its input value. For example, IP=2, IB=3,5 will hold fixed B(3), B(5) in F during execution of NLSOL. If the default IP=0 is assumed, then array-IB is not required.

IALT= Logical unit number (default 10) for reading the input data matrix under an object (run-time) format defined in file FOR005. The value of IALT can be any value the operating system supports, but it cannot be 4,6,13,or 16. If IALT=5 is used, then the data matrix ((Y(I),X(I,J),J=1,M),I=1,N) will immediately follow the object format on FOR005 (see EXAMPLES OF INPUT PARAMETERS below).

ISTOP=1 (default) to stop the run after completion of the current problem.

ISTOP=0 to continue processing after completion of the current problem (i.e., a total restart) with the same data matrix, but using on FOR005 a new title, $PARMS, object format, and an optional $INIT input as possibly read by the users initialization subroutine SUBZ (see Appendix 3). Note that ISTOP=0 can only be used whenever IALT is not 5. (This is because file IALT is rewound and read again--possibly with a different object format, etc.). Also, note that all $PARMS and $INIT parameters previously used will be assumed, and the user only needs to supply "override" parameters for each succeeding problem. The very last $PARMS on FOR005 should set ISTOP=1; however, an end-of-file condition on FOR005 will also terminate the run.

IWT=0 (default) for indicating an unweighted least-squares solution is desired; however, in this case, NLSOL will weight all N observations with a value of unity (i.e., with assumed standard deviations 1.0 and WT(I)=1.0 for all I=1,2,...,N).

IWT=1 to indicate a weighted least-squares solution is desired, where WT(I)=1.0/X(I,M+1)**2 and X(I,M+1) is the standard deviation augmented to the data matrix. Internally in NLSOL, WT(I) is stored in X(I,5) since $M \leq 4$. If any X(I,M+1)=0.0 when IWT=1, then WT(I)=1.0 is used to avoid division by 0.

IWT=2 to indicate a weighted least-squares solution is desired, where WT(I)=1.0/ABS(X(I,M+1)) and X(I,M+1)

is the variance augmented to the data matrix. Internally in NLSOL, WT(I) is stored in X(I,5) since M≤4. If any X(I,M+1)=0.0 when IWT=2, then WT(I)=1.0 is used to avoid division by 0. Note that choosing IWT=2, along with X(I,M+1)=Y(I), is equivalent to using a "statistical weight" of 1.0/ABS(Y(I)); this may be useful when an "instrumental weight" (as with IWT=1) is unknown, or the data set Y(I),I=1,...,N contains large variations.

IDER=0 (default) to use analytic partial derivatives in NLSOL, which requires both the user's forward function subroutine (FCODE) and analytic derivative subroutine (PCODE). See Appendix 3 for the proper calling sequences as assumed by NLSOL.

IDER=1 to use estimated partial derivatives in NLSOL, which only calls the user's forward function subroutine (FCODE).

IPRT=0 (default) for minimal printout on FOR006 (but more complete output on FOR016, provided IOUT=1--see below).

IPRT=-1 gives moderate output on FOR006 (but complete output on FOR016), less the data matrix input and output residual vector.

IPRT=-2 same as IPRT=-1, but also gives the data matrix input and output residual vector on FOR006 (and complete output on FOR016).

IPRT=1 same as IPRT=-1, but gives more (and longer) detail

lines from the adaptive algorithm on both FOR006 and FOR016.

IOUT=1 (default) to obtain both FOR006 and FOR016 output (print-type) files. Normally, FOR006 is the VAX terminal output, and FOR016 is a VAX disk file (see VAX operating instruction below).

IOUT=0 to obtain only FOR006 print output file. [Note that it may be convenient to use file FOR016 as deferred printer output, and also it may be used as an input file to other programs (e.g., an X-Y plot program, etc.).]

NITER=10 (default) is the maximum number of iterations allowed in NLSOL before terminating the adaptive algorithm, if one-of-four types of convergence is not obtained (see Dennis and others, 1979, p.11-14); if no convergence occurred after NITER, then all output (except covariance and correlation matrices, and parameter relative errors) will be given, including the last solution vector B obtained. In many cases (e.g., good initial B estimates and data matrix), NITER<10 may suffice. Of course, NITER>>10 will allow the adaptive algorithm to generally converge to a relative minimum R vector; however, a large NITER may be time-consuming for some problems. Obviously, one may restart NLSOL with the last solution vector to continue in smaller (pseudo-interactive) NITER-increments. Note that the users termination

subprogram SUBEND in Appendix 3 could also create a "restart" $PARMS file to automatically contain the last B-vector.

SP=0 (default) to specify an unconstrained (and unscaled) minimization least-squares solution; i.e., the NLSOL interface is essentially identical to the original adaptive algorithm.

SP=i ($1 \leq i \leq 4$) will select various constrained (and scaled) minimization types as indicated in Table 1. When SP=3 or 4, then the corresponding lower and higher parameter bounds are required in arrays BL() and BH(), respectively (see below).

BL()= Array of lower parameter bounds required when SP=3 or 4, where $BL(J) \leq B(J)$, J=1,2,...,K. If array-BL is not given when SP>2, then an error message may occur unless $B(J) \geq 0.0$ for all J=1,K (note BL(J) defaults to 0.0 for J=1,2,...,K).

BH()= Array of higher parameter bounds required when SP=3 or 4, where $B(J) \leq BH(J)$, J=1,2,...,K. If array-BH is not given when SP>2, then an error message may occur unless $B(J) \leq 0.0$ for all J=1,K (note BH(J) defaults to 0.0 for J=1,2,...,K).

Notes on BL,BH: If a very small (or unrealistic) parameter range is defined by $BL(J) \leq B(J) \leq BH(J)$, then the adaptive least-squares algorithm may produce one or more solution B(J)'s equal to the corresponding BL(J) or BH(J). Care should be

exercised in choosing meaningful bounds and an initial estimate to avoid this situation; also, note that any parameter may be held fixed (using IP, IB) for any value of SP≥0, which may be required if a particular parameter cannot be resolved for the given data and/or model function used.

IV()= Integer array (dimension 80--but only the first 24 are input) defining the control parameters and options used in the adaptive algorithm NL2ITR (IDER=0) or NL2SNO (IDER=1). For most cases, the standard default values for array-IV are adequate as supplied by subprogram DFAULT (see Dennis and others, 1979, p.31-32). However, a few IV values are automatically overridden by the NLSOL interface routine; these are as follows:

IV(15)= 3 to select the non-Hessian form of computing covariance matrices. This was chosen to conform to methods previously used by Anderson (1980a, 1980b) to compute correlation matrices and standard errors. Of course, other IV(15) values can be supplied, if desired.

IV(18)= NITER (unless IV(1) is not 10) simply sets the maximum iterations to NITER (default is 10, instead of 150 as given by DFAULT).

IV(19)= -1 if -3<IPRT<1 (default IPRT=0); otherwise, IV(19)=IPRT (i.e., when IPRT>0). Note that long summary lines are printed if IPRT>0, and short

lines if IPRT<0.

IV(21)= 16 if IOUT=1 (default). This is the primary output unit for the adaptive algorithm. In order for all IPRT options to work properly, file FOR016 is reused in NLSOL (and/or appended) to produce specific output on FOR006. Therefore, it is recommended that IV(21)=16 and IOUT=1 are always assumed.

IV(21)= 6 if IOUT=0. This will ignore any IPRT options, since only FOR006 is "on-line", and is usually a terminal output file.

V()= Real array (dimension 2906--but only the first 42 are of primary interest as input) defining other control parameters and options used in the adaptive algorithm NL2ITR (IDER=0) or NL2SNO (IDER=1). For most cases, the standard default values for array-V are adequate as supplied by subprogram DFAULT (see Dennis and others, 1979, p.33-35). For the VAX-11/780 system, the constants MACHEP= 5.960464E-8, ETA= 2.939E-39, and BIG= 1.7E+38 are the machine-dependent values assumed (see Dennis and others, 1979, p.33-38).

$END [end of $PARMS parameters; the "END" in $END may be omitted, if desired.]

DISCUSSION OF IV, V PARAMETERS

For many simple problems, arrays IV and V may be ignored; i.e., the assumed defaults are generally adequate. However, for some long and/or complex problems, computer time can often be conserved (at least initially while looking for a best model, etc.) if using some of the suggestions given in Table 2.

Table 2.-- Some tuning parameters of interest

Default $PARMS	Optional $PARMS	Purpose of optional $PARMS; see Dennis and others on given page
IV(14)=1, IV(15)=3	IV(14)=0, IV(15)=0	To suppress an attempt to compute and print a covariance matrix. (p.31: COVPRT, COVREQ)
IV(17)=200	IV(17)<200	To limit the maximum number of function calls, not including covariance matrix calls, if any. (p.31: MXFCAL)
V(24)=.1	$0 \leq V(24) \leq .5$	To control function reduction to "decrease" the trust region radius. (p.39: TUNER1)
V(25)=10	$V(25) \geq 1$	To control function reduction to "increase" the trust region radius. (p.39: TUNER2)
V(26)=.75	$.001 \leq V(26)$ $V(26) \leq 1$	To control function reduction to "increase" the trust region radius. (p.40: TUNER3)
V(28)= 5.96E-5	V(28)> 6.E-5	To relax the "X-convergence" tolerance (p.35: XCONCR)
V(29)= 5.96E-5	V(29)> 6.E-5	To relax the "cosine convergence" tolerance (p.33: CCONCR)
V(36)=.001	V(36)=0	Will set the scale vector D=1.0, which usually gives good performance on well-scaled problems (p.34: D0).
V(39)=100	V(39)<100 or V(39)>100	To limit (or increase) the maximum 2-norm step length on the first iteration (p.34-35, p.36: LMAX0).

V(40)= 1.E-9	V(40)> 1.E-9	To relax the "residual convergence" tolerance (p.35: RCONCR).
V(42)= 1.E-4	V(42)> 1.E-4	To relax the "variability convergence tolerance (p.35: VCONCR).

For example, to relax all 4-types of convergence tolerances, one could use the nondefault values:

$PARMS V(28)=.001,V(29)=.001,V(40)=.001,V(42)=.005,...

The adaptive algorithm NL2ITR or NL2SNO will only print the nondefault values as overridden in array V, but will not print any nondefault IV values. All IV and V definitions, and assumed default values (except as noted in $PARMS IV() above), are given in Dennis and others (1979, p.31-40).

EXAMPLES OF INPUT PARAMETERS

The following example uses the general input file ordering described in the section PARAMETERS and DATA REQUIRED:

```
EXAMPLE TITLE WITH OBJECT DATA ON FOR005 (IALT=5)
$PARMS N=30,M=1,K=3,IP=1,IB=2,
IALT=5,IPRT=-1, SP=3,IWT=1,
BL=2*1,10,
B=10,1,100,
BH=100,1,1000, V(36)=0$
(3F10.0)
1.          .25         .05
1.5         .3          .08
---<etc. for 28 more observations>---
```

Note that in this example, IWT=1 and M=1; therefore, three columns are required in the data matrix row: Y(I),(X(I,L),L=1,M+1). Any desired FORTRAN object-time format can be used to read one observation row of the data

matrix, which we define 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. When IWT>0, the last column X(I,M+1) is the standard deviation (IWT=1) or variance (IWT=2) augumented to the data matrix row.

Refer to Appendix 2 for another example where IALT=10 (default) and the data matrix is separated from the parameter file--which is recommended so that ISTOP=0 can be used (if desired).

VAX OPERATING INSTRUCTIONS

Assuming subprogram NLSOL (and all associated subprograms) was previously compiled and linked using the VAX/VMS operating system, along with a user written MAIN program and subprograms FCODE, PCODE, SUBZ, and SUBEND (as required--see Appendix 3 for details), the following steps are general execution guidelines (note that many variations are possible using VMS in either time-sharing or batch modes):

1. Either assign (via $ASSIGN command) an input parameter file name to the logical name FOR005, or let FOR005 default to the users terminal input (if logged-in on-line). The order of the parameters on FOR005 must be given exactly as defined in the section PARAMETERS and DATA REQUIRED above--and also in Appendix 3. To assign FOR005, use the DCL command:

$ASSIGN parameterfilename FOR005

2. If IALT=10 (default), then assign the data matrix file name to FOR010 using the DCL command:

$ASSIGN datamatrixfilename FOR010

[If IALT=5 (and the data matrix is included in FOR005), then this step should be skipped.]

3. The MAIN program (called "main") may be executed with the DCL command:

$RUN main

The above execution steps could also be submitted (via a $SUBMIT command) to be run in batch mode. For this reason, it was convenient to exclude any prompting messages and user responses in subprogram NLSOL; also, VAX system-dependent commands and calls have been minimized in NLSOL for ease of program conversion to other systems (see Appendix 1 for information on conversion problems).

Because prompt messages are not given in step 3 on FOR006 (terminal output), it is recommended that FOR005 always be assigned in step 1 (otherwise, the user must remember the complete parameter order to type on-line). Also, in case of parameter errors (see ERROR MESSAGES below), it is easier to

edit the parameter file and return to step 3.

Note that FOR016 is a duplicate (print) disk file (normally called FOR016.DAT, unless assigned otherwise), and file FOR006 is usually the on-line terminal print file (or LOG file if $SUBMIT was used).

ERROR MESSAGES

Most $PARMS syntactical errors are flagged and printed on files FOR006 and FOR016 by the VAX-NAMELIST simulator subroutine (see Appendix 3), and the job is aborted. If FOR005 was assigned to a disk parameter file, then correct the parameter file using any VAX editor and rerun the job (e.g., use $RUN or $SUBMIT). Other parameter errors (or omissions) are also flagged and the job is terminated.

A VAX-system overflow condition will terminate the run. Usually, when SP<3, an overflow condition can result from a very poor initial parameter estimate in array B, or the given data matrix is incomplete (or inaccurate) for the particular nonlinear model chosen. One can sometimes obtain a solution using SP>2 and supply reasonable parameter bounds in arrays BL and BH; however, some parameters may still not be resolvable (i.e., large percent errors) by using incomplete or inaccurate data and (or) an inappropriate model.

We would like to quote a paragraph from Bard (1974) concerning nonlinear parameter estimation problems and errors:

> "The reader should realize that the state of the art of nonlinear optimization is such that one cannot as yet write a computer program that will produce the correct answer to every parameter estimation problem in a single computer run. All too often, the first run produces unacceptable results. By studying these results one can perhaps obtain better starting guesses; one can choose to impose bounds or a prior distribution on the variables, or to relax previously imposed bounds; one can search for errors in the coding of the model equations or their derivatives. By careful coaxing, the computer may be made to yield acceptable results in subsequent runs. An interactive computer system can be particularly useful for this purpose."

PRINTED OUTPUT

Results are printed on files FOR006 and FOR016 (if IOUT=1). Refer to Appendix 2 for a sample output listing of file FOR006.

The following list defines additional names (or terms) used in the printed output files, other than $PARMS as previously defined:

NAMES/TERMS	PRINTED OUTPUT DEFINITIONS
FMT	The object (run-time) format used to read the data matrix row.
INITIAL X(I)	The unconstrained X-vector (i.e., the transformed initial B-vector via Table 1) as used in NLSOL with respect to the parameter SP option.
D(I)	The initial (or final) scale vector D; see Dennis and others (1979, p.34, and p.36).
IT,NF,F,DF,..	The iteration count (IT), number of function evaluations (NF), current function value (F=

	half the weighted residual sum-of-squares), difference between previous and current function values (DF),.. etc. The complete short or long print lines used by the adaptive algorithm is described in Dennis and others (1979, p.36-37).
G(I)	The gradient vector G (Dennis and others, 1979, p.33) corresponding to the final X(I),D(I) vectors.
OBS.Y(I)	The observed dependent variable Y(I), for I=1,2,...,N.
CAL	The calculated function F corresponding to Y(I), I=1,2,...,N.
RES	The residual defined as [Y(I)-F], I=1,2,...,N.
%RES.ERR	The percent residual error defined as 100*RES/CAL for I=1,2,...,N.
X(I,L)	The observed independent variables X(I,L), L=1,2,...,M; I=1,2,...,N.
WT(I)	The weight of observation I (see $PARMS IWT formula used).
RMSERR	The root-mean-square error defined as RMSERR=SQRT(SUMRES**2/(N-K+IP)), where SUMRES is the sum-of-squares of the residual vector.
PARM.SOL	The FINAL X(I) solution vector, after an inverse transform if SP>0, and excluding any parameters held fixed (via IP, IB); the solution vector B is always passed to the users termination subroutine SUBEND (see Appendix 3), complete with any parameters held fixed, and whether or not convergence was obtained in the adaptive algorithm.
STD.ERROR*	The parameter standard error derived as the square-root of the corresponding diagonal element of the covariance matrix (if computed) defined by eq.(6.6)-(6.8) in Dennis and others (1979, p.14, p.31).
REL.ERROR*	The parameter relative error defined as STD.ERROR/PARM.SOL.
% ERROR*	The parameter percent error defined as 100*REL.ERROR.

* These values are given only if a covariance matrix was computed, and if it is positive-definite. Note that all statistics assume a linear model in the neighborhood of the nonlinear function minimum. (See Bard, 1974, for a discussion of statistical inference of nonlinear model solutions.)

REFERENCES

Anderson, W.L., 1980a, Program MARQHXY: Marquardt inversion of Hx and Hy frequency soundings from a grounded wire source: U.S. Geological Survey Open-File Report 80-901, 111p.

--------, 1980b, Program IMSLEXY: Marquardt inversion of Ex and Ey frequency soundings from a grounded wire source: U.S. Geological Survey Open-File Report 80-1073, 87p.

Bard, Y., 1974, Nonlinear parameter estimation, Academic Press, Inc., N.Y., 341p.

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), 40p.

IMSL (International Mathematical and Statistical Libraries), 1979, IMSL-LIB-0007 (Revised Jan, 1979), 7500 Bellaire Blvd., 6th Floor, GNB Bldg., Houston, Texas 77036.

Appendix 1.-- Conversion to other systems

This subprogram (and associated subprograms) was written in 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.

Since the FORTRAN-77 ANSI-standard presently does not provide for a NAMELIST I/O capability, a VAX-11 NAMELIST simulator subprogram is included in this program package. For most large main-frame systems (e.g., IBM/370, CYBER, etc.), a NAMELIST READ/WRITE is usually available; in this case, the VAX NAMELIST subprogram and associated routines (DECODEIX, DECODEX) can be eliminated; also, appropriate changes can be made where COMMON/NAME_LIST/ and CALL NAMELIST is used in the source program.

Other changes for non-VAX systems might include some (or all) of the following:

(1) Variables with more than 6-characters.
(2) Use of the underscore character or dollar character in some variables and/or COMMON names.
(3) Character strings delimited by single-quote characters (e.g., 'STRING'); also, character string concatination (e.g., 'STRING1'//'STRING2').
(4) Passing variable-length character strings in subroutine calls; e.g., CHARACTER*(*) passed length character arguments.

(5) Need to suppress 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.

(6) Replacement of any special VAX-dependent CALLS or statements (e.g., CALL LIB$INDEX, ACCEPT, TYPE, CALL SYS$anyname, etc.--note that we have minimized machine-dependent calls, where possible).

(7) Hexidecimal constants (e.g., '4A'X) if used in any DATA statements.

(8) Virtual-sized arrays, if any (i.e, DIMENSION statements greater than physical memory).

(9) To increase the default dimensions in NLSOL (defined in Appendix 3 as NDIM=500, MDIM=5, and KDIM=20), change the PARAMETER statements in <u>each</u> subprogram: NLSOL, NLITR, INTRAN, CALCR, and NAMELIST. However, in most cases, the default limits are anticipated to be sufficient and would not require any changes, unless of course the target machine does not support PARAMETER statements.

Appendix 2.-- Test problem code example

A simple 7-parameter nonlinear least-squares test problem (defined below) was run on a VAX system using several NLSOL options as described in this report.

The MAIN program (called T1NLSOL) used the following nonlinear function (F) and corresponding analytic partial derivatives Pj= ∂F/∂Bj (j=1,2,...,7):

```
      F=B1+B2*t+B3*t**2+B4*SIN(B5*t)+B6*EXP(-(B7*t)),

      P1=1,
      P2=t,
      P3=t*t,
      P4=SIN(B5*t),
      P5=B4*t*COS(B5*t),
      P6=EXP(-B7*t),
      P7=-B6*t*P6,
```

where t is the independent variable, and Bj (j=1,2,...,7) are the unknown model parameters.

The following code example follows the requirements given in Appendix 3 (to be linked with NLSOL):

```
C {T1NLSOL}: TEST1 FOR NLSOL USING A 7-PARAMETER PROBLEM
C
      EXTERNAL T1FCODE,T1PCODE,T1SUBZ,T1SUBEND
      CALL NLSOL(T1FCODE,T1PCODE,T1SUBZ,T1SUBEND)
      CALL EXIT
      END
      SUBROUTINE T1FCODE(Y,X,B,PASS,F,IN,IDER)
C--FUNCTION EVALUATION FOR 7-PARAMETER PROBLEM
      DIMENSION Y(1),X(500,5),B(1),PASS(5)
      PASS(1)=X(IN,1)
      T=PASS(1)
      F=B(1)+B(2)*T+B(3)*T**2+B(4)*SIN(B(5)*T)+B(6)*
     1 EXP(-B(7)*T)
      RETURN
      END
      SUBROUTINE T1PCODE(P,X,B,PASS,F,IN,IP,IB)
C--ANALYTIC DERIVATIVES FOR 7-PARAMETER PROBLEM
      DIMENSION P(1),X(500,5),B(1),PASS(5),IB(1)
```

```
      T=PASS(1)
      P(1)=1.0
      P(2)=T
      P(3)=T**2
      P(4)=SIN(B(5)*T)
      P(5)=B(4)*T*COS(B(5)*T)
      P(6)=EXP(-B(7)*T)
      P(7)=-B(6)*T*P(6)
      IF(IP.EQ.0) RETURN
      DO 10 I=1,IP
      DO 10 J=1,7
        IF(IB(I).NE.J) GO TO 10
        P(J)=0.0
10      CONTINUE
      RETURN
      END
      SUBROUTINE T1SUBZ(Y,X,B,PASS,NPASS,N,TITLE,IOUT)
C--INITIALIZATION FOR 7-PARAMETER PROBLEM
C  ($INIT INPUT NOT NEEDED IN THIS EXAMPLE)
      DIMENSION Y(1),X(500,5),B(1),PASS(5)
      CHARACTER*80 TITLE
      NPASS=1
      IF(IOUT.EQ.1) WRITE(16,10) TITLE
10    FORMAT('0{T1NLSOL}:',5X,A)
      RETURN
      END
      SUBROUTINE T1SUBEND(Y,X,B,K,N,TITLE,IOUT)
C--TERMINATION FOR 7-PARAMETER PROBLEM
      DIMENSION Y(1),X(500,5),B(1)
      CHARACTER*80 TITLE
      WRITE(6,10)
10      FORMAT(/' ********  E N D  ********'/
     1' ** FINAL SOLUTION VECTOR:'/)
      IF(IOUT.EQ.1) WRITE(16,10)
      DO 30 I=1,K
      WRITE(6,20) I,B(I)
20      FORMAT(2X,I3,E16.8)
      IF(IOUT.EQ.1) WRITE(16,20) I,B(I)
30      CONTINUE
      RETURN
      END
```

Test problem input/output

The following input files (FOR005, FOR010) were used to run the test program T1NLSOL (and subprogram NLSOL) on a VAX system. The corresponding output file (FOR006) is given following file FOR010.

FOR005

```
A 7-PARAMETER PROBLEM
$PARMS N=21,K=7,M=1,IPRT=-2,
NITER=50,
IP=2,IB=3,1,
SP=3,
BL=1,.5,-.5,3*1,.1,
BH=1,5,-.5,3*10,10,
B=1,.75,-.5,4.5,1.8, 5.5,.87$
(2F10.0)
```

FOR010

```
  7.000000  0.000000
  7.809257  0.250000
  8.380069  0.500000
  8.292930  0.750000
  7.344467  1.000000
  5.581667  1.250000
  3.278261  1.500000
  0.858261  1.750000
 -1.215198  2.000000
 -2.558975  2.250000
 -2.968187  2.500000
 -2.469844  2.750000
 -1.318940  3.000000
  0.061875  3.250000
  1.184131  3.500000
  1.611856  3.750000
  1.067327  4.000000
 -0.501716  4.250000
 -2.909872  4.500000
 -5.779944  4.750000
 -8.635657  5.000000
```

```
(NLSOL):        A 7-PARAMETER PROBLEM

N=        21    K=         7    IP=        2    M=         1    IALT=     10
ISTOP=     1    IWT=       0    IDER=      0    IPRT=     -2    NITER=    50
IOUT=      1    SP=        3

PARAMETERS HELD FIXED: IB=  3  1

FMT=(2F10.0)


PARAMETER LOWER BOUNDS: BL=

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

INITIAL PARAMETERS: B=

 0.10000000E+01  0.75000000E+00 -0.50000000E+00  0.45000000E+01  0.18000000E+01
 0.55000000E+01  0.87000000E+00

PARAMETER HIGHER BOUNDS: BH=

 0.10000000E+01  0.50000000E+01 -0.50000000E+00  0.10000000E+02  0.10000000E+02
 0.10000000E+02  0.10000000E+02

PARAMETER INDEX:  1  2  3  4  5  6  7
REORDERED AS...:  2  4  5  6  7

REORDERED PARAMETERS:

 0.75000000E+00  0.45000000E+01  0.18000000E+01  0.55000000E+01  0.87000000E+00

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


    I     INITIAL X(I)        D(I)
    1     0.237941E+00      0.276E+02
    2     0.673352E+00      0.254E+02
    3     0.302746E+00      0.213E+03
    4     0.755398E+00      0.152E+02
    5     0.282635E+00      0.358E+02

   IT    NF       F           DF          COSMAX       VAR

    0     1  0.341E+02                  0.894E+00
    1     2  0.247E+01  0.317E+02  0.774E+00  0.158E+02
    2     3  0.394E-01  0.243E+01  0.739E+00  0.160E+02
    3     4  0.236E-05  0.394E-01  0.818E+00  0.160E+02
    4     5  0.346E-10  0.236E-05  0.890E+00  0.160E+02
    5     6  0.306E-10  0.397E-11  0.922E+00  0.149E+02
    6     7  0.306E-10 -0.255E-10  0.922E+00  0.154E+02

***** X-CONVERGENCE *****

FUNCTION      0.306103D-10  VARIABILITY    0.153756E+02
FUNC. EVALS          7      GRAD. EVALS          6
GRAD. NORM    0.166560E-02  CUSMAX         0.922324E+00

    I       FINAL X(I)          D(I)           G(I)
    1     0.339537E+00      0.379E+02     0.144E-03
    2     0.615480E+00      0.265E+02    -0.447E-04
```

```
   3     0.339837E+00     0.230E+03    -0.166E-02
   4     0.841069E+00     0.143E+02    -0.439E-06
   5     0.306277E+00     0.341E+02    -0.167E-04

 I    OBS.Y(I)       CAL            RES          %RES.ERR      X(I,1)         X(I,2)         X(I,3)         X(I,4)         WT(I)
 1  0.700000E+01  0.700000E+01 -0.477E-06 -0.681196E-05  0.000000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
 2  0.780926E+01  0.780926E+01  0.000E+00  0.000000E+00  0.250000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
 3  0.838007E+01  0.838007E+01  0.954E-06  0.113803E-04  0.500000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
 4  0.829293E+01  0.829293E+01  0.954E-06  0.114998E-04  0.750000E+00  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
 5  0.734447E+01  0.734447E+01  0.954E-06  0.129849E-04  0.100000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
 6  0.558167E+01  0.558167E+01 -0.954E-06 -0.170858E-04  0.125000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
 7  0.327826E+01  0.327826E+01 -0.954E-06 -0.290909E-04  0.150000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
 8  0.858261E+00  0.858262E+00 -0.954E-06 -0.111117E-03  0.175000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
 9 -0.121520E+01 -0.121520E+01 -0.107E-05 -0.882889E-04  0.200000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
10 -0.255897E+01 -0.255897E+01 -0.119E-05 -0.465848E-04  0.225000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
11 -0.296819E+01 -0.296819E+01 -0.238E-06 -0.803247E-05  0.250000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
12 -0.246984E+01 -0.246984E+01  0.715E-06  0.289595E-04  0.275000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
13 -0.131894E+01 -0.131894E+01  0.834E-06  0.632678E-04  0.300000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
14  0.618750E-01  0.618736E-01  0.145E-05  0.233608E-02  0.325000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
15  0.118413E+01  0.118413E+01  0.179E-05  0.151009E-03  0.350000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
16  0.161186E+01  0.161186E+01  0.596E-06  0.369789E-04  0.375000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
17  0.106733E+01  0.106733E+01  0.238E-06  0.223379E-04  0.400000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
18 -0.501716E+00 -0.501714E+00 -0.203E-05 -0.403927E-03  0.425000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
19 -0.290987E+01 -0.290987E+01 -0.358E-05 -0.122902E-03  0.450000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
20 -0.577994E+01 -0.577994E+01 -0.381E-05 -0.659989E-04  0.475000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01
21 -0.863566E+01 -0.863565E+01 -0.381E-05 -0.441738E-04  0.500000E+01  0.000000E+00  0.000000E+00  0.000000E+00  0.100000E+01

** RMSERR=  0.19560896E-05

COVARIANCE MATRIX
 2  0.4465E-14
 4 -0.5830E-15  0.6153E-14
 5  0.2111E-15 -0.4422E-16  0.8302E-16
 6  0.2992E-14 -0.3627E-14  0.1135E-15  0.3401E-13
 7  0.3175E-14 -0.1904E-15  0.9849E-16  0.9812E-14  0.7589E-14

CORRELATION MATRIX
 2  0.1000E+01
 4 -0.1112E+00  0.1000E+01
 5  0.3468E+00 -0.6186E-01  0.1000E+01
 6  0.2428E+00 -0.2508E+00  0.6754E-01  0.1000E+01
 7  0.5454E+00 -0.2786E-01  0.1241E+00  0.6107E+00  0.1000E+01

  **PARM_SOL.    STD_ERROR    REL_ERROR     % ERROR **

 2  0.1000E+01  0.6682E-07  0.6682E-07  0.6682E-05
 4  0.4000E+01  0.7844E-07  0.1961E-07  0.1961E-05
 5  0.2000E+01  0.9112E-08  0.4556E-08  0.4556E-06
 6  0.6000E+01  0.1844E-06  0.3074E-07  0.3074E-05
 7  0.1000E+01  0.8712E-07  0.8712E-07  0.8712E-05

********  E N D  ********
** FINAL SOLUTION VECTOR:

   1  0.10000000E+01
   2  0.10000001E+01
   3 -0.50000000E+00
   4  0.39999993E+01
   5  0.19999999E+01
   6  0.60000005E+01
   7  0.10000001E+01
```

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 author*. A magnetic tape copy can be sent to requestors 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.]

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

> 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 "NLSOL.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:


```
      SUBROUTINE NLSOL(FCODE,PCODE,SUBZ,SUBEND)                          00000010
C                                                                        00000020
C {NLSOL}: GENERAL NONLINEAR LEAST-SQUARES SOLUTION   {11/9/81}          00000030
C          USING DENNIS ET AL (1979; SEE REF1 BELOW)                     00000040
C          ADAPTIVE NONLINEAR LEAST-SQUARES ALGORITHM.                   00000050
C                                                                        00000060
C** THIS IS AN INTERFACE ROUTINE WRITTEN FOR THE VAX-11/780 BY            00000070
C   W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.               00000080
C                                                                        00000090
C** THIS INTERFACE (NLSOL) HAS ADDITIONAL OPTIONS (BESIDE REF1) TO:       00000100
C   (1) PERFORM EITHER UNCONSTRAINED OR UP TO 4-TYPES OF CONSTRAINED     00000110
C       ADAPTIVE NONLINEAR REGRESSION FOR ARBITRARY NONLINEAR PROBLEMS.  00000120
C       (I.E., PARTIAL OR FULL LOWER/HIGHER PARAMETER BOUNDS, ETC.)      00000130
C   (2) HOLDING CERTAIN PARAMETERS FIXED (I.E., AS CONSTANTS) IN THE     00000140
C       LEAST-SQUARES (THIS IS ANOTHER FORM OF CONSTRAINING SOLUTION     00000150
C       SPACE).                                                          00000160
C   (3) PROVIDE FOR WEIGHTED OBSERVATIONS (I.E., WEIGHTED LEAST-SQUARES)00000170
C   (4) OBJECT (RUN)-TIME CONTROL OF READING THE DATA MATRIX, PLUS       00000180
C       MANY OTHER I/O OPTIONS, ETC.                                     00000190
C   (5) OPTIONALLY, ONE CAN USE EITHER ESTIMATED PARTIAL DERIVATIVES, OR00000200
C       ANALYTICAL PARTIAL DERIVATIVES (IF SUBROUTINE PCODE AVAILABLE).  00000210
C                                                                        00000220
C** THE USER ONLY NEEDS TO WRITE SUBROUTINES FCODE, PCODE, SUBZ, AND     00000230
C   SUBEND (SEE DETAILS BELOW) EXACTLY AS USED IN SUBROUTINE 'MARQRT'    00000240
C   (SEE REF2) OR 'IMSLMQ' (SEE REF3).  ALSO, THE SAME PARAMETER FILE    00000250
C   FOR005 AND OBJECT (RUN)-TIME DATA MATRIX FILE FOR010 AS USED BY      00000260
C   EITHER MARQRT OR IMSLMQ MAY BE USED IN 'NLSOL'.                      00000270
C                                                                        00000280
C** NLSOL CALLS NLITR WHICH CALLS 'NL2ITR' AS PUBLISHED BY DENNIS ET AL,00000290
C   (SEE REF1, P. 38), OR 'NL2SNO' (SEE REF1, P. 35).                    00000300
C                                                                        00000310
C** REF1:  DENNIS, J.E., ET AL, 1979, AN ADAPTIVE NONLINEAR LEAST-       00000320
```

```
C           SQUARES ALGORITHM, NTIS REPORT AD-A079-716.                    00000330
C                                                                          00000340
C    REF2:  ANDERSON, W.L., 1980, PROGRAM MARQHXY: INVERSION OF HX AND HY00000350
C           FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN-    00000360
C           FILE REPT. 80-901.                                             00000370
C                                                                          00000380
C    REF3:  ANDERSON, W.L., 1980, PROGRAM IMSLEXY: INVERSION OF EX AND EY00000390
C           FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN-    00000400
C           FILE REPT. 80-1073.                                            00000410
C                                                                          00000420
C*************************************************************************00000430
C                                                                          00000440
C****  THE USER MUST DECLARE THE CALLING PARAMETERS AS EXTERNAL IN THE     00000450
C      CALLING PROGRAM (ANY DESIRED NAMES MAY BE USED).                    00000460
C E.G.,                                                                    00000470
C                                                                          00000480
C [MAIN]:                                                                  00000490
C      EXTERNAL MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND                        00000500
C      CALL NLSOL(MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND)                     00000510
C      STOP   !<OR USE>: CALL EXIT                                         00000520
C      END                                                                 00000530
C [FCODE]:                                                                 00000540
C      SUBROUTINE MY_FCODE(Y,X,B,W,F,IN,IDER)                              00000550
C      USER WRITTEN TO EVALUATE THE NONLINEAR OBJECTIVE FUNCTION (F)       00000560
C      USED IN NLSOL AS THE WEIGHTED SUM OF (Y(IN)-F)**2, WHERE            00000570
C      Y= OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N, WHERE N IS            00000580
C       GIVEN IN $PARMS NAMELIST INPUT--SEE BELOW).                        00000590
C      X= OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,M, WHERE             00000600
C       M IS IN $PARMS INPUT).                                             00000610
C      B= CURRENT PARAMETER ESTIMATES (DIM. K, WHERE                       00000620
C       K IS IN $PARMS INPUT).                                             00000630
C      W= WORK ARRAY (DIM. 5)--MAY BE USED TO PASS DATA TO PCODE.          00000640
C      F= (OUTPUT) THE FUNCTION VALUE EVALUATED FOR THE GIVEN              00000650
C       Y,X, AND B ARRAYS AT THE OBSERVATION NO. 'IN'.                     00000660
C      IN= (INPUT) OBSERVATION NO. TO EVALUATE F (1.LE.IN.LE.N),           00000670
C       WHICH IS CONTROLLED EXTERNALLY BY 'NLSOL'. USUALLY,                00000680
C       IN=1,2,...,N--BUT NOT ALWAYS.                                      00000690
C      IDER= 0 IF ANALYTICAL DERIVATIVES ARE USED (PCODE CALLED            00000700
C         AFTER FCODE).                                                    00000710
C          = 1 IF ESTIMATED DERIVATIVES ARE USED (PCODE NOT CALLED         00000720
C         AFTER FCODE).                                                    00000730
C      DIMENSION Y(1),X(500,5),B(1),W(5)                                   00000740
C>>>>> INSERT USER CODE HERE TO EVALUATE F  <<<<<                          00000750
C      END                                                                 00000760
C [PCODE]: >> PCODE MAY BE A DUMMY NAME IF ONLY IDER=1 IS TO BE USED. <<00000770
C      SUBROUTINE MY_PCODE(P,X,B,W,F,IN,IP,IB)                             00000780
C      USER WRITTEN TO EVALUATE THE ANALYTICAL PARTIAL DERIVATIVES OF      00000790
C      F WITH RESPECT TO B(J),J=1,2,...,K, AT OBSERVATION 'IN', WHERE      00000800
C      P= (OUTPUT) PARTIAL DERIVATIVE ARRAY (DIM. K, WHERE                 00000810
C       K IS IN $PARMS INPUT).                                             00000820
C      X,B,W ARE THE SAME AS USED IN FCODE (SEE ABOVE).                    00000830
C      F= LAST FUNCTION VALUE FROM FCODE AT OBSERVATION IN.                00000840
C       (NOTE THAT F MAY NOT BE NEEDED, BUT IS AVAILABLE ANYWAY)           00000850
C      IN= (INPUT) OBSERVATION NO. TO EVALUATE P ARRAY, WHICH IS           00000860
C       CONTROLLED EXTERNALLY BY 'NLSOL' (1.LE.IN.LE.N).                   00000870
C      IP= (INPUT) THE NO. OF B-PARAMETERS HELD FIXED IN THE LEAST-        00000880
C       SQUARES (0.LE.IP.LE.K-1; USE IP=0 IF NONE).                        00000890
```

```
C      IB= ARRAY OF B-PARAMETER INDICES HELD FIXED IF IP.GT.0.          00000900
C       NOTE THAT THE INDICES IN IB ARRAY MAY BE IN ANY ORDER,          00000910
C       BUT MUST BE BETWEEN 1 AND K (K IS IN $PARMS INPUT).             00000920
C      DIMENSION P(1),X(500,5),B(1),W(5),IB(1)                          00000930
C>>>>>  INSERT USER CODE HERE TO EVALUATE P  <<<<<                      00000940
C      END                                                              00000950
C [SUBZ]:                                                               00000960
C      SUBROUTINE MY_SUBZ(Y,X,B,W,NW,N,TITLE,IOUT)                      00000970
C      USER WRITTEN INITIALIZATION ROUTINE (CALLED ONCE BY 'NLSOL').    00000980
C      SUBZ MAY BE USED TO CHECK Y(IN),X(IN,M) AFTER INPUT VIA          00000990
C      OBJECT (RUN)-TIME INPUT (SEE BELOW) ON UNIT IALT. ALSO, SUBZ     00001000
C      MAY BE USED TO READ ADDITIONAL $INIT PARAMETERS, AND TO LOAD     00001010
C       ANY COMMON BLOCKS IF NEEDED IN THE USERS FCODE,PCODE.           00001020
C      Y,X,B,W  ARE THE SAME AS USED IN FCODE (SEE ABOVE).              00001030
C      NW= USE ANY DUMMY INTEGER VARIABLE (THIS IS                      00001040
C       TO MAINTAIN COMPATIBILITY WITH 'MARQRT' OR 'IMSLMQ').           00001050
C      N= NO. OF OBSERVATIONS IN Y(N),X(N,M) ARRAYS, WHERE              00001060
C       K.GE.N.LE.500 (N,M,K  ARE IN $PARMS INPUT).                     00001070
C      TITLE= (INPUT) 80-CHARACTER HEADING (SEE INPUT FOR005 BELOW).    00001080
C      IOUT= 1 IF TO WRITE OUTPUT ON BOTH FOR006 AND FOR016.            00001090
C          = 0 IF TO WRITE OUTPUT ONLY ON FOR006.                       00001100
C      DIMENSION Y(1),X(500,5),B(1),W(5)                                00001110
C      CHARACTER*80 TITLE                                               00001120
C>>>>> INSERT USER CODE HERE FOR ANY INITIALIZATION DESIRED  <<<<<      00001130
C      END                                                              00001140
C [SUBEND]:                                                             00001150
C      SUBROUTINE MY_SUBEND(Y,X,B,K,N,TITLE,IOUT)                       00001160
C      USER WRITTEN TERMINATION ROUTINE (CALLED ONCE BY 'NLSOL').       00001170
C      SUBEND MAY BE USED TO OUTPUT THE FINAL SOLUTION VECTOR B(I),     00001180
C      I=1,2,...,K, IN OTHER FORMS, ETC., AS DESIRED. [OR IT MAY BE A   00001190
C      DUMMY ROUTINE; I.E., JUST RETURNS.]                              00001200
C      Y,X,K,N,TITLE,IOUT  ARE THE SAME AS IN SUBZ AND FCODE.           00001210
C      B= (INPUT) IS THE FINAL SOLUTION VECTOR AS DETERMINED BY         00001220
C       'NLSOL' (SEE REF1 FOR DETAILS).                                 00001230
C      DIMENSION Y(1),X(500,5),B(1)                                     00001240
C      CHARACTER*80 TITLE                                               00001250
C>>>>>  INSERT USER CODE HERE FOR ANY TERMINATION SUMMARY DESIRED  <<<<<00001260
C      END                                                              00001270
C                                                                       00001280
C***********************************************************************00001290
C                                                                       00001300
C** INPUT ORDER ON FOR005 (PARAMETER FILE LOGICAL NAME):                00001310
C                                                                       00001320
C 1.  TITLE (MAX. 80-CHARACTERS--ALWAYS READ BEFORE $PARMS INPUT).      00001330
C 2.  $PARMS (SAME DEFINITIONS AS IN 'MARQRT', REF2, OR IN 'IMSLMQ',    00001340
C      REF3), WHICH INCLUDES: N,K,IP,M,IALT,ISTOP,IWT,IDER,             00001350
C      IPRT,NITER,IOUT,SP,B(),IB(); PLUS THE FOLLOWING PARAMETERS FROM  00001360
C      REF1 (NL2SOL), P.31-35: IV(),V();  IN ADDITION, THE LOWER AND    00001370
C      UPPER BOUND ARRAYS BL(),BH(), RESPECTIVELY, ARE REQUIRED IF      00001380
C      SP>2.                                                            00001390
C 3.  (OBJECT-RUN-TIME FORMAT STATEMENT) TO DESCRIBE THE FORMAT OF THE  00001400
C      DATA MATRIX ROW Y(I),(X(I,J),J=1,M*) READ ON FILE IALT, WHERE    00001410
C      M*=M (IF IWT=0) OR M*=M+1 (IF IWT>0), M.LE.4, AND I=1,2,...,N.   00001420
C (3A). INSERT DATA MATRIX HERE ONLY IF IALT=5.                         00001430
C 4.  $INIT OPTIONAL NAMELIST USED FOR READING PROBLEM-DEPENDENT        00001440
C      PARAMETERS USED IN SUBROUTINE SUBZ (SEE ABOVE).  CURRENTLY,      00001450
C      THE FOLLOWING $INIT NAMES (AND DIM.) CAN BE USED: IOB,MM,XO,YO,  00001460
```

```
C      L,EP,EPS,NEPS,METHOD,NFIN,IER,MEV,IOPT,NSIG,MAXFN,DELTA,PARM(4),  00001470
C      AND IRATIO(2).                                                    00001480
C 5.   OPTIONALLY, REPEAT STEPS 1-4, IF PARAMETER ISTOP=0 WAS USED       00001490
C      IN THE LAST STEP 2.                                               00001500
C                                                                        00001510
C** OUTPUT IS GIVEN ON FOR006 (ON-LINE USUALLY) AND ON FOR016(IF IOUT=1)00001520
C   FOR016 CONTAINS ALL PRINTABLE OUTPUT SELECTED VIA $PARMS IPRT,IOUT. 00001530
C   NOTE: IPRT=0 GIVES ABBREVIATED OUTPUT ON FOR006 (BUT MORE ON FOR016)00001540
C         IPRT=1 OR -2 GIVES DETAILED OUTPUT ON BOTH 6 AND 16.          00001550
C         IPRT=-1 GIVES MODERATE OUTPUT ON 6 (DETAILED ON 16).          00001560
C                                                                        00001570
C** TO RUN ON VAX (ELIMINATE <> DELIMITERS IN SUBSTITUTIONS):            00001580
C                                                                        00001590
C   $ASSIGN <PARAMETER FILE NAME> FOR005                                 00001600
C   $ASSIGN <DATA MATRIX FILE NAME> FOR010                               00001610
C   $RUN <MAIN NAME>                                                     00001620
C                                                                        00001630
C************************************************************************00001640
C                                                                        00001650
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00001660
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF          00001670
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                         00001680
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                                00001690
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS.   00001700
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:             00001710
      PARAMETER (K1DIM=KDIM-1,K2DIM=KDIM+KDIM,M1DIM=MDIM-1,              00001720
     1 IVDIM=KDIM+60,NKVDIM=96+2*NDIM+(KDIM*(7*KDIM+41))/2)              00001730
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00001740
C                                                                        00001750
      REAL*4 L                                                           00001760
      DIMENSION B(KDIM),SQWT(NDIM),IB(K1DIM),C(KDIM),INDEX(KDIM),        00001770
     1 IV(IVDIM),V(NKVDIM),CBOUND(K2DIM),                                00001780
     2 BL(KDIM),BH(KDIM),CL(KDIM),CH(KDIM),SE(KDIM),                     00001790
     3 W(KDIM),PARM(4),IRATIO(2),PRNT(5)                                 00001800
      INTEGER SP,SCALEP,SY,SCALEY                                        00001810
      CHARACTER*3 CHAR3                                                  00001820
      CHARACTER*6 CALLED                                                 00001830
      CHARACTER*80 TITLE                                                 00001840
      CHARACTER*132 LINE132                                              00001850
      CHARACTER*72 FMT                                                   00001860
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,      00001870
     1 IDER_,K_,ISP                                                      00001880
      COMMON/BOUNDS/BL_(KDIM),BH_(KDIM)                                  00001890
      COMMON/REVCOM/R(NDIM)                                              00001900
      EQUIVALENCE (SQWT(1),X(1,MDIM)),(N,NOBS),(K,KPARMS),(M,MVARS),     00001910
     1 (CL(1),CBOUND(1)),(CH(1),CBOUND(KDIM+1))                          00001920
      EXTERNAL FCODE,PCODE,CALCR                                         00001930
C**                                                                      00001940
C  THE FOLLOWING COMMON/NAME_LIST/ IS TO SIMULATE ON VAX-11/780:         00001950
C      NAMELIST/PARMS/ & READ(5,PARMS) VIA 'CALL NAMELIST(5,'$PARMS',*)' 00001960
C      NAMELIST/INIT/ & READ(5,INIT)  VIA 'CALL NAMELIST(5,'$INIT',*)'   00001970
C** SEE SUBROUTINE NAMELIST FOR MORE DETAILS, AND ALSO REF1-REF3 FOR     00001980
C   DETAILS ON EACH PARAMETER DEFINITION.                                00001990
C**                                                                      00002000
      COMMON/NAME_LIST/N,K,IP,M,IALT,ISTOP,IWT,IDER,IPRT,NITER,INON,     00002010
     1 FF,T,E,TAU,XL,MODLAM,GAMCR,DEL,ZETA,IOUT,SP,SCALEP,SY,SCALEY,     00002020
     2 B,IB, IOB,MM,X0,Y0,L,EP,EPS,NEPS,METHOD,NFIN,IER,MEV,             00002030
```

```
     3 IV,V,BL,BH,                                                      00002040
     4 IOPT,NSIG,MAXFN,DELTA,PARM, H,IRATIO                             00002050
C**                                                                     00002060
C  NOTE THAT COMMON/NAME_LIST/ CONTAINS SOME PARAMETERS ONLY FOR        00002070
C  COMPATIBILITY WITH 'MARQRT' OR 'IMSLMQ'; I.E., THE FOLLOWING LIST    00002080
C  OF PARAMETERS ARE CURRENTLY NOT USED DIRECTLY BY 'NLSOL':            00002090
C    INON,FF,T,TAU,XL,MODLAM,GAMCR,DEL,E,ZETA,SY,SCALEY,SCALEP,         00002100
C    IOPT,NSIG,MAXFN,DELTA,PARM.                                        00002110
C**                                                                     00002120
C                                                                       00002130
C** READ NLSOL TITLE LINE                                              00002140
      READ(5,10,ERR=9000,END=9010) TITLE                                00002150
10    FORMAT(A80)                                                       00002160
C                                                                       00002170
C**PRESET DEFAULT PARMS (SOME MUST BE GIVEN IN $PARMS ELSE AN ERROR)    00002180
C                                                                       00002190
      N=0                                                               00002200
      K=0                                                               00002210
      IP=0                                                              00002220
      M=0                                                               00002230
      IALT=10                                                           00002240
      ISTOP=1                                                           00002250
      ICALL=1                                                           00002260
      IWT=0                                                             00002270
      IDER=0                                                            00002280
      IPRT=0                                                            00002290
      NITER=10                                                          00002300
      IOUT=1                                                            00002310
      SP=0                                                              00002320
      DO 20 I=1,KDIM                                                    00002330
      IF(I.LT.KDIM) IB(I)=0                                             00002340
      BL(I)=0.0                                                         00002350
      B(I)=0.0                                                          00002360
      BH(I)=0.0                                                         00002370
20    CONTINUE                                                          00002380
22    IV(1)=10                                                          00002390
C**                                                                     00002400
C  PRESET NLITR                                                         00002410
C**                                                                     00002420
      CALL DFAULT(IV,V)                                                 00002430
C**                                                                     00002440
C** OVERRIDE FOR IV(15)=3 DEFAULT (MAY BE CHANGED VIA $PARMS INPUT)     00002450
C**                                                                     00002460
      IV(15)=3                                                          00002470
C**                                                                     00002480
C  READ $PARMS ON FOR005 VIA 'CALL NAMELIST' ON VAX                     00002490
C**                                                                     00002500
30    CALL NAMELIST(5,'$PARMS',*9020)                                   00002510
C**                                                                     00002520
C  SET EQUIVALENT PARAMETERS IN DIFFERENT COMMON'S                      00002530
C**                                                                     00002540
      ISP=SP                                                            00002550
      DO 32 I=1,KDIM                                                    00002560
      BFIX(I)=B(I)                                                      00002570
      BL_(I)=BL(I)                                                      00002580
      BH_(I)=BH(I)                                                      00002590
      IF(I.LT.KDIM) IIB(I)=IB(I)                                        00002600
```

```
32     CONTINUE                                                          00002610
       IIP=IP                                                            00002620
       IDER_=IDER                                                        00002630
       K_=K                                                              00002640
C**                                                                      00002650
C  TEST $PARMS BEFORE PROCEEDING                                         00002660
C**                                                                      00002670
       IF(IP.LT.0.OR.IP.GT.K1DIM)CALL ERRMSG('IP<0 OR IP>19',0,6,16)     00002680
       KIP=K-IP                                                          00002690
       IF(N.LT.1.OR.N.GT.NDIM.OR.N.LT.KIP)                               00002700
      1 CALL ERRMSG('N<1,N>500,OR N<K-IP',0,6,16)                        00002710
       IF(K.LT.1.OR.K.GT.KDIM.OR.KIP.LT.1)                               00002720
      1 CALL ERRMSG('K<1,K>20,OR K-IP<1',0,6,16)                         00002730
       IF(M.LT.1.OR.M.GT.M1DIM)CALL ERRMSG('M<1 OR M>4',0,6,16)          00002740
       IF(IALT.EQ.6.OR.IALT.EQ.13.OR.IALT.EQ.16.OR.IALT.EQ.4)            00002750
      1 CALL ERRMSG('IALT=4,6,13,OR 16',0,6,16)                          00002760
       IF(ISTOP.EQ.0.AND.IALT.EQ.5)                                      00002770
      1   CALL ERRMSG('ISTOP=0 BUT IALT=5',0,6,16)                       00002780
       IF(IWT.LT.0.OR.IWT.GT.2)CALL ERRMSG('IWT<0 OR IWT>2',0,6,16)      00002790
       IF(IDER.LT.0.OR.IDER.GT.1)CALL ERRMSG('IDER<0 OR IDER>1',0,6,16)  00002800
       IF(SP.LT.0.OR.SP.GT.4)CALL ERRMSG('SP<0 OR SP>4',0,6,16)          00002810
       IF(IP.GT.0) THEN                                                  00002820
         DO J=1,IP                                                       00002830
           IF(IB(J).LT.1.OR.IB(J).GT.K) THEN                             00002840
             ENCODE(3,43,CHAR3) J                                        00002850
             CALL ERRMSG('IP>0 AND IB(J)<1 OR IB(J)>K FOR J='//          00002860
      1      CHAR3,0,6,16)                                               00002870
           ENDIF                                                         00002880
         ENDDO                                                           00002890
       ENDIF                                                             00002900
       IF(SP.EQ.0.OR.SP.EQ.2) GO TO 41                                   00002910
       DO 40 I=1,KPARMS                                                  00002920
         IF(SP.EQ.1) THEN                                                00002930
             IF(IP.GT.0) THEN                                            00002940
             DO 42 J=1,IP                                                00002950
               IF(I.EQ.IB(J)) GO TO 40                                   00002960
42           CONTINUE                                                    00002970
             ENDIF                                                       00002980
           IF(B(I).LE.0.) THEN                                           00002990
             ENCODE(3,43,CHAR3) I                                        00003000
43            FORMAT(I2,'.')                                             00003010
                CALL ERRMSG('SP=1 AND B(I)<=0 FOR I='//CHAR3,0,6,16)     00003020
              ENDIF                                                      00003030
         ELSE IF(SP.GT.2) THEN                                           00003040
           IF(B(I).LT.BL(I).OR.B(I).GT.BH(I).OR.BL(I).GT.BH(I)) THEN     00003050
            ENCODE(3,43,CHAR3) I                                         00003060
             CALL ERRMSG('SP>2 AND B(I)<BL(I), '//                       00003070
      1       'B(I)>BH(I), OR BL(I)>BH(I)'//                             00003080
      2       ' FOR I='//CHAR3,0,6,16)                                   00003090
           ENDIF                                                         00003100
          IF(BL(I).EQ.BH(I)) THEN                                        00003110
             IF(IP.GT.0) THEN                                            00003120
               DO 45 J=1,IP                                              00003130
                 IF(I.EQ.IB(J)) GO TO 40                                 00003140
45           CONTINUE                                                    00003150
              ENDIF                                                      00003160
             ENCODE(3,43,CHAR3) I                                        00003170
```

```
            CALL ERRMSG('SP>2 AND BL(I)=BH(I) BUT B(I) NOT HELD '//     00003180
     1      'FIXED FOR I='//CHAR3,0,6,16)                               00003190
           ENDIF                                                        00003200
         ENDIF                                                          00003210
40     CONTINUE                                                        00003220
41     IF(IV(1).EQ.10) THEN                                            00003230
C**                                                                    00003240
C  NOTE CALL DFAULT(IV,V) WAS PRESET BEFORE $PARMS READ                00003250
C**                                                                    00003260
         IV(18)=NITER                                                  00003270
         IF(IPRT.GT.-3.AND.IPRT.LT.1) THEN                             00003280
           IV(19)=-1                                                   00003290
         ELSE                                                          00003300
           IV(19)=IPRT                                                 00003310
         ENDIF                                                         00003320
         IF(IOUT.EQ.0) THEN                                            00003330
           IV(21)=6                                                    00003340
         ELSE                                                          00003350
           IV(21)=16                                                   00003360
         ENDIF                                                         00003370
       ENDIF                                                           00003380
       IF(IP.GT.0) THEN                                                00003390
         DO 50 I=1,IP                                                  00003400
           IF(IB(I).LE.0)CALL ERRMSG('IP>0 BUT SOME IB(I)<=0',0,6,16)  00003410
50       CONTINUE                                                      00003420
       ENDIF                                                           00003430
C                                                                      00003440
C  READ OBJECT(RUN)-TIME FORMAT FOR DATA MATRIX FROM FILE IALT.        00003450
C                                                                      00003460
       READ(5,60,ERR=9000,END=9010) FMT                                00003470
60     FORMAT(A72)                                                     00003480
       IF(IWT.EQ.0) THEN                                               00003490
         M1=MVARS                                                      00003500
       ELSE                                                            00003510
         M1=MVARS+1                                                    00003520
       ENDIF                                                           00003530
       DO 70 I=1,NOBS                                                  00003540
         READ(IALT,FMT,ERR=9030,END=9040) Y(I),(X(I,J),J=1,M1)         00003550
         SQWT(I)=1.0                                                   00003560
         IF(IWT.EQ.0.OR.X(I,M1).EQ.0.0) THEN                           00003570
           GO TO 70                                                    00003580
         ELSE IF(IWT.EQ.1) THEN                                        00003590
           SQWT(I)=1.0/X(I,M1)                                         00003600
         ELSE                                                          00003610
           SQWT(I)=1.0/SQRT(ABS(X(I,M1)))                              00003620
         ENDIF                                                         00003630
70     CONTINUE                                                        00003640
C                                                                      00003650
C  INITIALIZE VIA CALL SUBZ (READ $INIT AND TEST, LOAD COMMON, ETC.)   00003660
C                                                                      00003670
       CALL SUBZ(Y,X,BFIX,PRNT,NPRNT,N,TITLE,IOUT)                     00003680
C      ****************************************                        00003690
C                                                                      00003700
C  WRITE $PARMS ON FOR006 AND FOR016 (THE LATTER IF IOUT=1)            00003710
C                                                                      00003720
       CALL NONBLANK(TITLE,NB)                                         00003730
       WRITE(6,80) TITLE,N,K,IP,M,IALT,ISTOP,IWT,IDER,IPRT,NITER,IOUT,SP 00003740
```

```
80     FORMAT('1{NLSOL}:',8X,A<NB>//' N=',4X,I6,T18,'K=',4X,I6,T34,'IP=',00003750
      1 3X,I6,T50,'M=',4X,I6,T66,'IALT=',1X,I6/' ISTOP=',I6,T18,'IWT=', 00003760
      2 2X,I6,T34,'IDER=',I7,T50,'IPRT=',I7,T66,'NITER=',I6/' IOUT=',   00003770
      3 5X,I2,T18,'SP=',3X,I6)                                          00003780
       IF(IOUT.NE.0)                                                    00003790
      1WRITE(16,80)TITLE,N,K,IP,M,IALT,ISTOP,IWT,IDER,IPRT,NITER,IOUT,SP 00003800
       IF(IP.GT.0) THEN                                                 00003810
         WRITE(6,90) (IB(I),I=1,IP)                                     00003820
90       FORMAT(/' PARAMETERS HELD FIXED: IB=',20I3)                    00003830
         IF(IOUT.NE.0) WRITE(16,90) (IB(I),I=1,IP)                      00003840
       ENDIF                                                            00003850
       CALL NONBLANK(FMT,NB)                                            00003860
       WRITE(6,100) FMT                                                 00003870
100    FORMAT(/' FMT=',A<NB>/)                                          00003880
       IF(IOUT.NE.0) WRITE(16,100) FMT                                  00003890
       IF(SP.GT.2) THEN                                                 00003900
         WRITE(6,111) (BL(I),I=1,KPARMS)                                00003910
111      FORMAT(/' PARAMETER LOWER BOUNDS: BL='//(5E16.8))              00003920
         IF(IOUT.NE.0) WRITE(16,111) (BL(I),I=1,KPARMS)                 00003930
       ENDIF                                                            00003940
       WRITE(6,110) (B(I),I=1,KPARMS)                                   00003950
110    FORMAT(/' INITIAL PARAMETERS: B='//(5E16.8))                     00003960
       IF(IOUT.NE.0) WRITE(16,110) (B(I),I=1,KPARMS)                    00003970
       IF(SP.GT.2) THEN                                                 00003980
         WRITE(6,112) (BH(I),I=1,KPARMS)                                00003990
112      FORMAT(/' PARAMETER HIGHER BOUNDS: BH='//(5E16.8))             00004000
         IF(IOUT.NE.0) WRITE(16,112) (BH(I),I=1,KPARMS)                 00004010
       ENDIF                                                            00004020
       DO 120 I=1,KDIM                                                  00004030
120      INDEX(I)=I                                                     00004040
       IF(IP.EQ.0) THEN                                                 00004050
         DO 130 I=1,KPARMS                                              00004060
         IF(SP.GT.2) THEN                                               00004070
           CL(I)=BL(I)                                                  00004080
           CH(I)=BH(I)                                                  00004090
         ENDIF                                                          00004100
130      C(I)=B(I)                                                      00004110
       ELSE                                                             00004120
C                                                                       00004130
C  REORDER B TO C WHEN IP>0 (AND BL,BH TO CL,CH, RESPECTIVELY)          00004140
C                                                                       00004150
         IM=0                                                           00004160
         DO 150 I=1,KPARMS                                              00004170
         DO 140 J=1,IP                                                  00004180
           IF(I.EQ.IB(J)) GO TO 150                                     00004190
140      CONTINUE                                                       00004200
         IM=IM+1                                                        00004210
         C(IM)=B(I)                                                     00004220
         IF(SP.GT.2) THEN                                               00004230
           CL(IM)=BL(I)                                                 00004240
           CH(IM)=BH(I)                                                 00004250
         ENDIF                                                          00004260
         INDEX(IM)=I                                                    00004270
150      CONTINUE                                                       00004280
         WRITE(6,160) (I,I=1,KPARMS)                                    00004290
160      FORMAT(/' PARAMETER INDEX:',20I3)                              00004300
         IF(IOUT.NE.0) WRITE(16,160) (I,I=1,KPARMS)                     00004310
```

```
        WRITE(6,170) (INDEX(I),I=1,KIP)                                 00004320
170     FORMAT(' REORDERED AS...:',20I3)                                00004330
        IF(IOUT.NE.0) WRITE(16,170) (INDEX(I),I=1,KIP)                  00004340
        WRITE(6,180) (C(I),I=1,KIP)                                     00004350
180     FORMAT(/' REORDERED PARAMETERS:'//(5E16.8))                     00004360
        IF(IOUT.NE.0) WRITE(16,180) (C(I),I=1,KIP)                      00004370
      ENDIF                                                             00004380
C                                                                       00004390
C  PERFORM INITIAL PARAMETER TRANSFORMS VIA SP (SCALEP)                 00004400
C                                                                       00004410
      IF(SP.EQ.0) GO TO 220                                             00004420
      DO 210 I=1,KIP                                                    00004430
        GO TO (201,202,203,203),SP                                      00004440
201     C(I)=ALOG(C(I))                                                 00004450
        GO TO 210                                                       00004460
202     C(I)=ASINH(C(I))                                                00004470
        GO TO 210                                                       00004480
203     TEM=(C(I)-CL(I))/(CH(I)-CL(I))                                  00004490
        IF(SP.EQ.3) THEN                                                00004500
          C(I)=ASIN(SQRT(TEM))                                          00004510
        ELSE                                                            00004520
          C(I)=ERFINV(2.0*TEM-1.0)                                      00004530
        ENDIF                                                           00004540
210   CONTINUE                                                          00004550
C                                                                       00004560
C  INTERFACE WITH NL2ITR USING MARQRT FCODE AND PCODE (IF IDER=0)       00004570
C                                                                       00004580
220   ENCODE(6,222,CALLED) ICALL                                        00004590
222   FORMAT(I3,' **')                                                  00004600
      WRITE(6,221) CALLED                                               00004610
221   FORMAT('0** NLITR (IDER=0) OR NL2SNO (IDER=1) CALLED:',A6/)       00004620
      IF(IOUT.NE.0) WRITE(16,221) CALLED                                00004630
      IF(IDER.EQ.0) THEN                                                00004640
        CALL NLITR(NOBS,KIP,C,IV,V,CBOUND,FCODE,PCODE)                  00004650
C       ********************************************                    00004660
      ELSE                                                              00004670
        CALL NL2SNO(NOBS,KIP,C,CALCR,IV,V,IDUMMY,CBOUND,FCODE)          00004680
C       ****************************************************            00004690
      ENDIF                                                             00004700
C                                                                       00004710
C  GET INVERSE PARAMETER TRANSFORMATION OF SOLUTION VECTOR C            00004720
C                                                                       00004730
      IF(SP.EQ.0) GO TO 229                                             00004740
      DO 228 I=1,KIP                                                    00004750
        GO TO (224,225,226,226),SP                                      00004760
224     C(I)=EXP(C(I))                                                  00004770
        GO TO 228                                                       00004780
225     C(I)=SINH(C(I))                                                 00004790
        GO TO 228                                                       00004800
226     TEM=CH(I)-CL(I)                                                 00004810
        IF(SP.EQ.3) THEN                                                00004820
          C(I)=CL(I)+TEM*SIN(C(I))**2                                   00004830
        ELSE                                                            00004840
          C(I)=CL(I)+0.5*TEM*(1.0+ERF(C(I)))                            00004850
        ENDIF                                                           00004860
228   CONTINUE                                                          00004870
C                                                                       00004880
```

```
C  OUTPUT SELECTED RESULTS ON FOR006 (ALL RESULTS ON FOR016 IF IOUT=1)  00004890
C                                                                       00004900
229    IF(IOUT.NE.0.AND.IPRT.NE.0) THEN                                 00004910
         I=1                                                            00004920
         REWIND 16                                                      00004930
230      READ(16,232,END=240) LINE132                                   00004940
232      FORMAT(A)                                                      00004950
         IF(I.EQ.1) THEN                                                00004960
C                                                                       00004970
C  VAX FUNCTION 'LIB$INDEX' USED TO DISTINGUISH FROM ARRAY 'INDEX'      00004980
C                                                                       00004990
           IF(LIB$INDEX(LINE132,'CALLED:'//CALLED).EQ.0) GO TO 230      00005000
           I=0                                                          00005010
           GO TO 230                                                    00005020
         ENDIF                                                          00005030
         IF(LIB$INDEX(LINE132,'OBS.Y(I)').NE.0) GO TO 236               00005040
         IF(LIB$INDEX(LINE132,'COVARIANCE = SCALE').NE.0) GO TO 236     00005050
         CALL NONBLANK(LINE132,J)                                       00005060
         IF(J.LE.0) GO TO 230                                           00005070
         WRITE(6,234) LINE132                                           00005080
234      FORMAT(A<J>)                                                   00005090
         GO TO 230                                                      00005100
236      READ(16,232,END=240) LINE132                                   00005110
         GO TO 236                                                      00005120
       ENDIF                                                            00005130
240    IF(IOUT.NE.0) WRITE(16,250)                                      00005140
250    FORMAT(/3X,'I',4X,'OBS.Y(I)',6X,'CAL',11X,'RES',8X,              00005150
     1 '%RES.ERR',6X,'X(I,1)',8X,                                       00005160
     2 'X(I,2)',8X,'X(I,3)',8X,'X(I,4)',8X,'WT(I)')                     00005170
       IF(IPRT.EQ.-2) WRITE(6,250)                                      00005180
       SUMF2=0.0                                                        00005190
       IF(IDER.NE.0) IADR=IV(50)-1                                      00005200
       DO 270 I=1,NOBS                                                  00005210
         IF(IDER.EQ.0) THEN                                             00005220
           F2=R(I)                                                      00005230
         ELSE                                                           00005240
           F2=V(IADR+I)                                                 00005250
         ENDIF                                                          00005260
         RES=F2/SQWT(I)                                                 00005270
         CAL=Y(I)-RES                                                   00005280
         IF(CAL.NE.0.0) THEN                                            00005290
           PERR=100.0*RES/ABS(CAL)                                      00005300
         ELSE                                                           00005310
           PERR=0.0                                                     00005320
         ENDIF                                                          00005330
         WT=SQWT(I)**2                                                  00005340
         SUMF2=SUMF2+RES**2                                             00005350
         IF(IPRT.EQ.-2)WRITE(6,260) I,Y(I),CAL,RES,PERR,                00005360
     1 (X(I,J),J=1,4),WT                                                00005370
260      FORMAT(1X,I3,2E14.6,E11.3,6E14.6)                              00005380
         IF(IOUT.NE.0) WRITE(16,260) I,Y(I),CAL,RES,PERR,               00005390
     1 (X(I,J),J=1,4),WT                                                00005400
270    CONTINUE                                                         00005410
       IF(NOBS.EQ.KIP) THEN                                             00005420
         RMSERR=0.0                                                     00005430
       ELSE                                                             00005440
         RMSERR=SQRT(SUMF2/(NOBS-KIP))                                  00005450
```

```
      ENDIF                                                              00005460
      WRITE(6,280) RMSERR                                                00005470
280   FORMAT(/' ** RMSERR=',E16.8)                                       00005480
      IF(IOUT.NE.0) WRITE(16,280) RMSERR                                 00005490
      IF(IV(26).LE.0) GO TO 380                                          00005500
C                                                                        00005510
C  A COVARIANCE MATRIX WAS COMPUTED (GET ADDITIONAL STATISTICS)           00005520
C                                                                        00005530
      IADR=IV(26)-1                                                      00005540
      IF(IPRT.LT.-1) WRITE(6,290)                                        00005550
290   FORMAT(/' COVARIANCE MATRIX')                                      00005560
      DO 320 I=1,KIP                                                     00005570
      DO 300 J=1,I                                                       00005580
300     W(J)=V(IADR+LOC(J,I))                                            00005590
        SE(I)=SQRT(ABS(W(I)))                                            00005600
        IF(IPRT.LT.-1) WRITE(6,310) INDEX(I),(W(J),J=1,I)                00005610
310     FORMAT(1X,I2,10E12.4/(3X,10E12.4))                               00005620
320   CONTINUE                                                           00005630
C                                                                        00005640
C  GET CORRELATION COEFFICIENT MATRIX                                    00005650
C                                                                        00005660
      IF(IOUT.NE.0) WRITE(16,330)                                        00005670
330   FORMAT(/' CORRELATION MATRIX')                                     00005680
      IF(IPRT.LT.0) WRITE(6,330)                                         00005690
      DO 350 I=1,KIP                                                     00005700
        IF(SE(I).EQ.0.0) THEN                                            00005710
          W(I)=1.0                                                       00005720
        ENDIF                                                            00005730
      DO 340 J=1,I                                                       00005740
        IF(SE(J).NE.0.0) W(J)=V(IADR+LOC(J,I))/(SE(I)*SE(J))             00005750
340   CONTINUE                                                           00005760
      IF(IOUT.NE.0) WRITE(16,310) INDEX(I),(W(J),J=1,I)                  00005770
      IF(IPRT.LT.0) WRITE(6,310) INDEX(I),(W(J),J=1,I)                   00005780
350   CONTINUE                                                           00005790
C                                                                        00005800
C  PRINT PARAMETER STANDARD ERRORS (SE) AND RELATIVE ERRORS              00005810
C                                                                        00005820
      WRITE(6,360)                                                       00005830
360   FORMAT(/'   **PARM_SOL.   STD_ERROR   REL_ERROR    % ERROR **'/)    00005840
      IF(IOUT.NE.0) WRITE(16,360)                                        00005850
      DO 370 I=1,KIP                                                     00005860
C  NOTE SE(I)=RMSERR*SQRT(COV(I,I)) FORM -- SEE REF1, P.14               00005870
        RELERR=SE(I)/C(I)                                                00005880
        PERR=100.*RELERR                                                 00005890
        WRITE(6,310) INDEX(I),C(I),SE(I),RELERR,PERR                     00005900
        IF(IOUT.NE.0) WRITE(16,310) INDEX(I),C(I),SE(I),RELERR,PERR      00005910
370   CONTINUE                                                           00005920
C                                                                        00005930
C  PUT SOLUTION C AND BFIX TOGETHER (IF IP>0)                            00005940
C                                                                        00005950
380   DO 390 I=1,KIP                                                     00005960
390   W(I)=C(I)                                                          00005970
      IF(IP.EQ.0) GO TO 420                                              00005980
      IM=0                                                               00005990
      DO 410 I=1,KPARMS                                                  00006000
        W(I)=BFIX(I)                                                     00006010
      DO 400 J=1,IP                                                      00006020
```

```
        IF(I.EQ.IB(J)) GO TO 410                                        00006030
400     CONTINUE                                                        00006040
          IM=IM+1                                                       00006050
          W(I)=C(IM)                                                    00006060
410     CONTINUE                                                        00006070
420     CALL SUBEND(Y,X,W,K,N,TITLE,IOUT)                               00006080
C         *********************************                             00006090
        IF(ISTOP.NE.1) THEN                                             00006100
          READ(5,10,ERR=9000,END=9010) TITLE                            00006110
          IF(IALT.NE.5) REWIND IALT                                     00006120
          ICALL=ICALL+1                                                 00006130
          GO TO 22                                                      00006140
        ENDIF                                                           00006150
C                                                                       00006160
C** RETURN FROM NLSOL                                                   00006170
C                                                                       00006180
        RETURN                                                          00006190
9000    CALL ERRMSG('ERR=9000 READING FOR005',0,6,16)                   00006200
9010    CALL ERRMSG('PREMATURE E.O.F (END=9010) READING FOR005',0,6,16) 00006210
9020    CALL ERRMSG('END *9020 READING FOR005 IN {NAMELIST}',0,6,16)    00006220
9030    CALL ERRMSG('END=9030 READING FILE IALT',0,6,16)                00006230
9040    CALL ERRMSG('PREMATURE E.O.F (END=9040) READING FILE IALT',     00006240
     1 0,6,16)                                                          00006250
C                                                                       00006260
C** END OF SUBROUTINE NLSOL                                             00006270
C                                                                       00006280
        END                                                             00006290
        SUBROUTINE NLITR(N,KIP,C,IV,V,CBOUND,FCODE,PCODE)               00006300
C                                                                       00006310
C**CALCULATES BOTH THE RESIDUAL VECTOR R(N) & ANALYTICAL JACOBIAN       00006320
C  JAC(N,KIP) BY 'REVERSE COMMUNICATION VIA INTERNAL CALL NL2ITR'       00006330
C  (SEE REF1, P. 38).                                                   00006340
C                                                                       00006350
C     N = NO. OBSERVATIONS <=500 (SEE NDIM BELOW)                       00006360
C     KIP = NO. ADJUSTABLE PARAMETERS =K-IIP WHERE                      00006370
C        K=TOTAL PARAMETERS, IIP=NO. PARAMETERS HELD FIXED              00006380
C          IN IIB(IIP) VIA COMMON/FIXDAT/                               00006390
C     C() = I/O PARAMETER VECTOR (SUPPLIED BY NL2ITR)                   00006400
C        WHICH ARE THE UNCONSTRAINED PARAMETERS IN NL2ITR.              00006410
C     IV() = SAME CONTROL INFORMATION SET BY NLSOL (OR NL2ITR).         00006420
C     V() = SAME CONTROL INFORMATION SET BY NLSOL (OR NL2ITR).          00006430
C     CBOUND = INPUT ARRAY OF LOW AND HIGH BOUNDS USED ONLY WHEN SP>2.  00006440
C     FCODE = EXTERNAL FUNCTION NAME (SAME AS USED IN 'MARQRT' OR       00006450
C        'IMSLMQ' TO COMPUTE THE NONLINEAR OBJECTIVE FUNCTION).         00006460
C     PCODE = EXTERNAL ANALYTIC DERIVATIVE NAME (SAME AS USED IN        00006470
C        'MARQRT' WHEN IDER=0) CORRESPONDING TO EACH FCODE CALL.        00006480
C                                                                       00006490
C**SEE REF1 (P.38) FOR MORE DETAILS ON CALLING NL2ITR.                  00006500
C                                                                       00006510
C**OTHER DATA IN COMMON/FIXDAT/ MUST BE PRESET.                         00006520
C                                                                       00006530
C                                                                       00006540
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00006550
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF         00006560
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                        00006570
        PARAMETER (NDIM=500,MDIM=5,KDIM=20)                             00006580
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS.  00006590
```

```
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:                00006600
      PARAMETER (K1DIM=KDIM-1)                                              00006610
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00006620
C                                                                           00006630
      INTEGER SP                                                            00006640
      DIMENSION C(1),IV(1),V(1),CBOUND(1),PRNT(5),SQWT(NDIM),               00006650
     1 BIP(KDIM),D(KDIM),R(NDIM),PART(KDIM),W(KDIM)                         00006660
      REAL*4 JAC(NDIM,KDIM)                                                 00006670
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,         00006680
     1 IDER,KPARMS,SP                                                       00006690
      COMMON/BOUNDS/BL(KDIM),BH(KDIM)                                       00006700
      COMMON/REVCOM/R                                                       00006710
      EQUIVALENCE (SQWT(1),X(1,MDIM))                                       00006720
      DATA NN/NDIM/                                                         00006730
C                                                                           00006740
C  GET INVERSE PARAMETER TRANSFORMATION (C TO BIP)                          00006750
C                                                                           00006760
10    CALL INTRAN(KIP,C,CBOUND,BIP)                                         00006770
C                                                                           00006780
C  DETERMINE FROM IV(1) HOW TO CALL NL2ITR                                  00006790
      IV1=IV(1)                                                             00006800
        DO 120 I=1,N                                                        00006810
          CALL FCODE(Y,X,BIP,PRNT,F,I,IDER)                                 00006820
C         ********************************                                  00006830
          IF(IV1.NE.2) R(I)=SQWT(I)*(Y(I)-F)                                00006840
          IF(IV1.EQ.1) GO TO 120                                            00006850
          CALL PCODE(PART,X,BIP,PRNT,F,I,IIP,IIB)                           00006860
C         ****************************************                          00006870
C                                                                           00006880
C  SCALE PART(J) VIA SP AND THE DERIVATIVE CHAIN-RULE.                      00006890
C                                                                           00006900
         IF(SP.EQ.0) GO TO 80                                               00006910
          IF(SP.EQ.1) THEN                                                  00006920
            DO 11 K=1,KPARMS                                                00006930
11          PART(K)=BIP(K)*PART(K)                                          00006940
          ELSE IF(SP.EQ.2) THEN                                             00006950
            DO 12 K=1,KPARMS                                                00006960
            IF(PART(K).EQ.0.0) GO TO 12                                     00006970
            TEM=BIP(K)+SQRT(BIP(K)**2+1.0)                                  00006980
            PART(K)=0.5*(TEM+1.0/TEM)*PART(K)                               00006990
12          CONTINUE                                                        00007000
          ELSE IF (SP.EQ.3) THEN                                            00007010
            DO 13 K=1,KPARMS                                                00007020
            IF(PART(K).EQ.0.0) GO TO 13                                     00007030
            PART(K)=2.*PART(K)*SQRT((BIP(K)-BL(K))*                         00007040
     1         (BH(K)-BIP(K)))                                              00007050
13          CONTINUE                                                        00007060
          ELSE IF(SP.EQ.4) THEN                                             00007070
            DO 14 K=1,KPARMS                                                00007080
            IF(PART(K).EQ.0.0) GO TO 14                                     00007090
            TEM=BH(K)-BL(K)                                                 00007100
            PART(K)=0.56418958*PART(K)*TEM*EXP(-(ERFINV(2.*(BIP(K)-         00007110
     1       BL(K))/TEM-1.))**2)                                            00007120
14          CONTINUE                                                        00007130
          ENDIF                                                             00007140
80        IF(IIP.EQ.0) THEN                                                 00007150
            DO 90 J=1,KIP                                                   00007160
```

```
90          JAC(I,J)=-SQWT(I)*PART(J)                                    00007170
          ELSE                                                           00007180
            IM=0                                                         00007190
            DO 110 K=1,KPARMS                                            00007200
            DO 100 J=1,IIP                                               00007210
              IF(K.EQ.IIB(J)) GO TO 110                                  00007220
100         CONTINUE                                                     00007230
            IM=IM+1                                                      00007240
            JAC(I,IM)=-SQWT(I)*PART(K)                                   00007250
110         CONTINUE                                                     00007260
          ENDIF                                                          00007270
120     CONTINUE                                                         00007280
C                                                                        00007290
C                                                                        00007300
      CALL NL2ITR(D,IV,JAC,N,NN,KIP,R,V,C)                               00007310
C     ************************************                               00007320
      IF(IV(1).EQ.1.OR.IV(1).EQ.2) GO TO 10                              00007330
      RETURN                                                             00007340
      END                                                                00007350
      SUBROUTINE INTRAN(KIP,C,CBOUND,BIP)                                00007360
C                                                                        00007370
C**INVERSE PARAMETER TRANSFORMATION USED IN 'NLSOL','NLITR'.             00007380
C                                                                        00007390
C  CALCULATES CONSTRAINED PARAMETERS FOR FCODE OR PCODE BACK FROM THE    00007400
C  UNCONSTRAINED PARAMETERS IN 'NL2ITR' OR 'NL2SNO'                      00007410
C                                                                        00007420
C     KIP = NO. ADJUSTABLE PARAMETERS = K-IIP (IIP IN COMMON/FIXDAT)     00007430
C     C() = INPUT UNCONSTRAINED VECTOR (DIM. KIP)                        00007440
C     CBOUND = INPUT CONSTRAINED BOUNDS, IF ANY.                         00007450
C     BIP() = OUTPUT CONSTRAINED VECTOR (DIM. KPARMS--IN COMMON).        00007460
C                                                                        00007470
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00007480
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF          00007490
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                         00007500
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                                00007510
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS.   00007520
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:             00007530
      PARAMETER (K1DIM=KDIM-1)                                           00007540
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00007550
C                                                                        00007560
      INTEGER SP                                                         00007570
      DIMENSION C(1),CBOUND(1),BIP(1),CTEM(KDIM)                         00007580
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,      00007590
     1 IDER,KPARMS,SP                                                    00007600
      IF(SP.EQ.0) THEN                                                   00007610
        DO 10 I=1,KIP                                                    00007620
10      CTEM(I)=C(I)                                                     00007630
      ELSE                                                               00007640
        DO 50 I=1,KIP                                                    00007650
          GO TO (20,30,40,40),SP                                         00007660
20        CTEM(I)=EXP(C(I))                                              00007670
          GO TO 50                                                       00007680
30        CTEM(I)=SINH(C(I))                                             00007690
          GO TO 50                                                       00007700
40        DIF=CBOUND(KDIM+I)-CBOUND(I)                                   00007710
          IF(SP.EQ.3) THEN                                               00007720
            CTEM(I)=CBOUND(I)+DIF*SIN(C(I))**2                           00007730
```

```
          ELSE                                                          00007740
            CTEM(I)=CBOUND(I)+0.5*DIF*(1.0+ERF(C(I)))                   00007750
          ENDIF                                                         00007760
50      CONTINUE                                                        00007770
      ENDIF                                                             00007780
      IF(IIP.EQ.0) THEN                                                 00007790
        DO 60 I=1,KIP                                                   00007800
60      BIP(I)=CTEM(I)                                                  00007810
      ELSE                                                              00007820
        IM=0                                                            00007830
        DO 80 I=1,KPARMS                                                00007840
          BIP(I)=BFIX(I)                                                00007850
        DO 70 J=1,IIP                                                   00007860
          IF(I.EQ.IIB(J)) GO TO 80                                      00007870
70      CONTINUE                                                        00007880
        IM=IM+1                                                         00007890
        BIP(I)=CTEM(IM)                                                 00007900
80      CONTINUE                                                        00007910
      ENDIF                                                             00007920
      RETURN                                                            00007930
      END                                                               00007940
      SUBROUTINE CALCR(N,KIP,C,NF,R,LASTNF,CBOUND,FCODE)                00007950
C                                                                       00007960
C**CALCULATES RESIDUAL VECTOR R(N) FOR 'NL2SNO' WHEN IDER=1.            00007970
C                                                                       00007980
C     N = NO. OBSERVATIONS <=500 (SEE NDIM BELOW)                       00007990
C     KIP = NO. ADJUSTABLE PARAMETERS =K-IIP WHERE                      00008000
C        K=TOTAL PARAMETERS, IIP=NO. PARAMETERS HELD FIXED              00008010
C          IN IIB(IIP) VIA COMMON/FIXDAT/                               00008020
C     C() = INPUT PARAMETER VECTOR (SUPPLIED BY NL2SNO)                 00008030
C        WHICH ARE THE UNCONSTRAINED PARAMETERS IN NL2SNO.              00008040
C     NF = INVOCATION COUNT (INPUT)FOR USE BY NL2SNO OR NL2SOL.         00008050
C     R() = OUTPUT WEIGHTED RESIDUAL VECTOR (DIM. N)                    00008060
C     LASTNF = LAST NF (ON EXIT FOR POSSIBLE USE IN CALCJ OR NL2SOL).   00008070
C     CBOUND = INPUT ARRAY OF LOW AND HIGH BOUNDS USED ONLY WHEN SP>2.  00008080
C     FCODE = EXTERNAL FUNCTION NAME (SAME AS USED IN 'MARQRT' OR       00008090
C        'IMSLMQ' TO COMPUTE THE NONLINEAR OBJECTIVE FUNCTION).         00008100
C                                                                       00008110
C**OTHER DATA IN COMMON/FIXDAT/ MUST BE PRESET.                         00008120
C                                                                       00008130
C                                                                       00008140
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00008150
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF         00008160
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                        00008170
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                               00008180
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS. 00008190
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:            00008200
      PARAMETER (K1DIM=KDIM-1)                                          00008210
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00008220
C                                                                       00008230
      INTEGER SP                                                        00008240
      DIMENSION C(1),R(1),CBOUND(1),PRNT(5),SQWT(NDIM),BIP(KDIM)        00008250
      COMMON/FIXDAT/Y(NDIM),X(NDIM,MDIM),BFIX(KDIM),IIB(K1DIM),IIP,     00008260
     1 IDER,KPARMS,SP                                                   00008270
      EQUIVALENCE (SQWT(1),X(1,MDIM))                                   00008280
C                                                                       00008290
C  GET INVERSE PARAMETER TRANSFORMATION (C TO BIP)                      00008300
```

```
C                                                                       00008310
      CALL INTRAN(KIP,C,CBOUND,BIP)                                     00008320
C                                                                       00008330
C  COMPUTE RESIDUAL VECTOR R(N) USING BIP IN FCODE                      00008340
C                                                                       00008350
      DO 10 I=1,N                                                       00008360
        CALL FCODE(Y,X,BIP,PRNT,F,I,IDER)                               00008370
C       ********************************                                00008380
        R(I)=SQWT(I)*(Y(I)-F)                                           00008390
10    CONTINUE                                                          00008400
      LASTNF=NF                                                         00008410
      RETURN                                                            00008420
      END                                                               00008430
      SUBROUTINE NAMELIST(IUNIT,NAME,*)                                 00008440
C                                                                       00008450
C {NAMELIST INPUT ON VAX-11/780} VIA "CALL NAMELIST" {VERSION: 12/10/80}00008460
C                                                                       00008470
C--A SIMULATED 'NAMELIST/NAME/' PROCESSOR FOR VAX-11 FORTRAN-77 TO      00008480
C  IMPLEMENT "CALL NAMELIST(IUNIT,'$NAME',*EOF)" ON VAX, WHICH          00008490
C  IS SIMILAR TO "READ(IUNIT,NAME,END=EOF)" ON MOST LARGE SYSTEMS.      00008500
C                                                                       00008510
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.           00008520
C                                                                       00008530
C--THIS IS A SUBSET OF THE ACTUAL NAMELIST/NAME/ AVAILABLE ON           00008540
C  MOST LARGE MAIN-FRAME SYSTEMS. CURRENT OPTIONS ARE:                  00008550
C                                                                       00008560
C  (1) ALL VARNAM'S ARE RESTRICTED TO 1 TO 6 CHAR'S (ALP,NUM, AND '_')  00008570
C      BUT MUST BEGIN WITH AN ALP CHAR (E.G., A3_, BVAR, C_2, ETC.)     00008580
C  (2) ONLY VARIABLE TYPES REAL*4 *8 (NAMTYP=1) AND INTEGER*2 *4        00008590
C      (NAMTYP=0). SEE C==== EXAMPLE STATEMENTS FOR NAMTYP BELOW =====. 00008600
C      {NOTE: COMPLEX,LOGICAL, OR CHARACTER VARIABLE TYPES ARE "NOT"    00008610
C      CODED IN THIS VERSION.}                                          00008620
C  (3) MAX. 60 VARNAM'S ALLOWED IN NAMELIST (FOR ALL '$NAMES' USED).    00008630
C  (4) MAX. NUMBER FIELD (FLOAT OR FIXED) IS 20 CHAR WIDE, WHERE        00008640
C      BLANK CHAR'S ARE IGNORED, AND TYPE CONVERSION IS AUTOMATIC.      00008650
C      FLOAT NUMBERS WITH OPTIONAL E+XX OR D-XX AND WITH OR WITHOUT '.' 00008660
C      IN THE MANTISSA IS ALLOWED (E.G., 123E-3, .123D+02, -3.14, ETC.).00008670
C  (5) PARTIAL ARRAY'S ALLOWED; E.G., A(10)=25.1,                       00008680
C      AND B=1,3.2,...                                                  00008690
C  (6) REPEAT FACTORS ALLOWED; E.G., C=2*1,3,..                         00008700
C  (7) ONLY 1-DIM ARRAYS ALLOWED WITH MAX SIZE 99999.                   00008710
C  (8) THE NAMELIST '$NAME' MUST BE 2 TO 7 CHAR'S, AND MUST BEGIN WITH  00008720
C      A "$" CHAR (E.G., '$P', '$PARMS', ETC.);  ALSO, THE FIRST CHAR IN00008730
C      IFILE MAY BEGIN IN COL. 1 BUT LESS THAN COL. 72 (BUFFER IS 80).  00008740
C      LINES IN IFILE MAY BE CONTINUED TO COL. 1 ON NEXT LINE, AND      00008750
C      TERMINATE THE NAMELIST BY "$[END]"--THE "END" IS OPTIONAL. E.G., 00008760
C                                                                       00008770
C      $PARMS A=1,B=2.3,7*1,C(3)=-.123E-10,                             00008780
C      D=1800, E=5*20$END                                               00008790
C      $NEXNAM F=123, G=-10,C(2)=15.02 $                                00008800
C      ...END-OF-IFILE...                                               00008810
C  (9) ABOUT 98% OF ALL THE POSSIBLE ERRORS ARE DETECTED AND AN         00008820
C      ERROR MESSAGE IS PRINTED ON UNIT 06, FOLLOWED BY CALL EXIT.      00008830
C      {NOTE: WATCH OUT FOR THE REMAINING 2% UNDETECTED ERRORS!}        00008840
C                                                                       00008850
C--SUBROUTINES CALLED:                                                  00008860
C                                                                       00008870
```

```
C  DECODEIX, DECODEX, AND NONBLANK.                                    00008880
C                                                                      00008890
C--USAGE:                                                              00008900
C                                                                      00008910
C  1. MODIFY FILE 'INCLNAMES.FOR' AS REQUIRED (USE ANY EDITOR).        00008920
C     (SEE C==== EXAMPLE STATEMENTS BELOW =====.)                      00008930
C  2. RECOMPILE SUBROUTINE 'NAMELIST' WITH THE DESIRED INCLNAMES.FOR.   00008940
C  3. IN USERS CALLING PROGRAM, USE:                                    00008950
C     CALL NAMELIST(IUNIT,'$NAME',*N)  --ON VAX, WHERE N=E.O.F RETURN   00008960
C     STATEMENT LABEL.  THIS SIMULATES ON VAX:                         00008970
C      'READ(IUNIT,NAME,END=N)' ON SYSTEMS WITH NAMELIST/NAME/...      00008980
C                                                                      00008990
C*********************************************************************00009000
C                                                                      00009010
      CHARACTER*(*) NAME                                               00009020
      CHARACTER*1 C(47),BUFI                                           00009030
      CHARACTER*6 VARNAM                                               00009040
      CHARACTER*20 NUMFLD                                              00009050
      CHARACTER*80 BUF                                                 00009060
C                                                                      00009070
C=====================================================================00009080
C====== THE USER MUST CHANGE THE FOLLOWING STATEMENTS FOR THE SPECIFIC  00009090
C====== NAMELIST VARIABLES DESIRED (E.G., USE TECO OR EDT, ETC.)=======00009100
C====== DIMENSION NO_NAM VARIABLES TO AGREE WITH CHANGED DATA STATEMENTS00009110
C==                                                                    00009120
C==ON VAX USE THE FOLLOWING INCLUDE STATEMENT (OPTIONALLY, USE /LIST):  00009130
C==                                                                    00009140
C>>   INCLUDE 'INCLNAMES.FOR/NOLIST'                                   00009150
C                                                                      00009160
C========================= INCLNAME2.FT ==============================00009170
C==================== FOR USE IN CALL NAMELIST ======================00009180
C NORMALLY, ONE SHOULD COPY 'INCLNAME2.FT' TO 'INCLNAMES.FT'; THEN     00009190
C   EDIT 'INCLNAMES.FT' AS DESIRED FOR USERS CALL NAMELIST. NOTE THAT  00009200
C   ONE MUST RECOMPILE 'NAMELIST.FT' WITH USERS CALLING PROGRAM,       00009210
C   WHERE 'NAMELIST.FT' CONTAINS THE FOLLOWING STATEMENT:              00009220
C                                                                      00009230
C     INCLUDE 'INCLNAMES.FT/LIST'                                      00009240
C=====================================================================00009250
C                                                                      00009260
C*********************************************************************00009270
C  THIS IS $PARMS AND $INIT INPUT FOR "MOST" INVERSION PROGRAMS ON VAX. 00009280
C*********************************************************************00009290
C                                                                      00009300
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00009310
C$$ CHANGE THE FOLLOWING FORTRAN-77 PARAMETER STATEMENT ONLY IF        00009320
C$$ INCREASING THE DEFAULT DIMENSIONS FOR NLSOL:                       00009330
      PARAMETER (NDIM=500,MDIM=5,KDIM=20)                              00009340
C$$ WHERE NDIM=MAX.OBS., MDIM=MAX.INDEP.VARS., KDIM=MAX.UNKNOWN PARMS. 00009350
C$$ DO NOT CHANGE THE FOLLOWING RELATED PARAMETER STATEMENT:           00009360
      PARAMETER (K1DIM=KDIM-1,                                         00009370
     1 IVDIM=KDIM+60,NKVDIM=96+2*NDIM+(KDIM*(7*KDIM+41))/2)            00009380
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00009390
C                                                                      00009400
      COMMON/NAME_LIST/V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,                 00009410
     * V11,V12,V13,V14,V15,V16,V17,V18,V19,V20,                        00009420
     * V21,V22,V23,V24,V25,V26,V27,V28,V29,V30,                        00009430
     * V31,V32,V33,V34,V35,V36,V37,V38,V39,                            00009440
```

```
     * V40,V41,V42,V43,V44,V45,V46,V47,V48,V49,V50                        00009450
      INTEGER V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,V11,                         00009460
     * V17, V21,V22,V23,V24,V25, V27,V28,V29, V35,V36,V37,V38,V39,        00009470
     * V40,V44,V45,V46,V50                                               00009480
      DIMENSION V1(1),V2(1),V3(1),V4(1),                                 00009490
     * V5(1),V6(1),V7(1),V8(1),V9(1),V10(1),                            00009500
     * V11(1),V12(1),V13(1),V14(1),V15(1),                              00009510
     * V16(1),V17(1),V18(1),V19(1),V20(1),                              00009520
     * V21(1),V22(1),V23(1),V24(1),V25(1),                              00009530
     * V26(KDIM),V27(K1DIM),V28(1),V29(1),V30(1),                       00009540
     * V31(1),V32(1),V33(1),V34(1),V35(1),                              00009550
     * V36(1),V37(1),V38(1),V39(1),V40(IVDIM),                          00009560
     * V41(NKVDIM),V42(KDIM),V43(KDIM),V44(1),V45(1),                   00009570
     * V46(1),V47(1),V48(4),V49(1),V50(2),                              00009580
     * V51(1),V52(1),V53(1),V54(1),V55(1),                              00009590
     * V56(1),V57(1),V58(1),V59(1),V60(1)                               00009600
      DIMENSION NAMDIM(60),NAMLEN(60),NAMTYP(60)                         00009610
      CHARACTER*6 NAM(60)                                                00009620
      DATA NAM/'N','K','IP','M','IALT','ISTOP','IWT','IDER',             00009630
     * 'IPRT','NITER','INON','FF','T','E','TAU','XL','MODLAM',           00009640
     * 'GAMCR','DEL','ZETA','IOUT','SP','SCALEP','SY','SCALEY',          00009650
     * 'B','IB','IOB','MM','X0','Y0','L','EP','EPS','NEPS',              00009660
     * 'METHOD','NFIN','IER','MEV','IV','V','BL','BH',                   00009670
     * 'IOPT','NSIG','MAXFN','DELTA','PARM','H','IRATIO',10*' '/         00009680
      DATA NAMDIM/25*1,KDIM,K1DIM,12*1,IVDIM,NKVDIM,2*KDIM,4*1,          00009690
     1 4,1,2,10*0/                                                       00009700
      DATA NAMLEN/2*1,2,1,4,5,3,2*4,5,4,2,2*1,3,2,6,5,3,2*4,             00009710
     * 2,6,2,6,1,2,3,3*2,1,2,3,4,6,4,2*3,2,1,2*2,2*4,2*5,4,1,6,          00009720
     * 10*0/                                                             00009730
      DATA NAMTYP/11*0,5*1,0,3*1,5*0,1,3*0,5*1,5*0,0,3*1,3*0,3*1,0,10*0/00009740
      DATA NO_NAM/50/                                                    00009750
C====== END OF INCLUDE STATEMENTS ======================================00009760
C                                                                        00009770
C==                                                                      00009780
C== FOR EXAMPLE, FILE 'INCLNAMES.FOR' MAY CONTAIN (WITHOUT "C=="):       00009790
C==                                                                      00009800
C==       COMMON/NAME_LIST/V1,V2,V3,V4                                   00009810
C==       REAL*8 V1                                                      00009820
C==       INTEGER V3                                                     00009830
C==       DIMENSION V1(1),V2(2),V3(3),V4(4),                             00009840
C==      * V5(1),V6(1),V7(1),V8(1),V9(1),V10(1),                         00009850
C==      * V11(1),V12(1),V13(1),V14(1),V15(1),                           00009860
C==      * V16(1),V17(1),V18(1),V19(1),V20(1),                           00009870
C==      * V21(1),V22(1),V23(1),V24(1),V25(1),                           00009880
C==      * V26(1),V27(1),V28(1),V29(1),V30(1),                           00009890
C==      * V31(1),V32(1),V33(1),V34(1),V35(1),                           00009900
C==      * V36(1),V37(1),V38(1),V39(1),V40(1),                           00009910
C==      * V41(1),V42(1),V43(1),V44(1),V45(1),                           00009920
C==      * V46(1),V47(1),V48(1),V49(1),V50(1),                           00009930
C==      * V51(1),V52(1),V53(1),V54(1),V55(1),                           00009940
C==      * V56(1),V57(1),V58(1),V59(1),V60(1)                            00009950
C==       DIMENSION NAMDIM(60),NAMLEN(60),NAMTYP(60)                     00009960
C==       CHARACTER*6 NAM(60)                                            00009970
C==       DATA NAM/'A','BB','ICC','DDD_4',56*' '/                        00009980
C==       DATA NAMDIM/1,2,3,4,56*0/                                      00009990
C==       DATA NAMLEN/1,2,3,5,56*0/                                      00010000
C==       DATA NAMTYP/2*1,0,1,56*0/                                      00010010
```

```
C==      DATA NO_NAM/4/                                                  00010020
C====== END OF EXAMPLE INCLUDE STATEMENTS ==============================00010030
C                                                                        00010040
C*************************************************************************00010050
C NOTE: THE ABOVE EXAMPLE SIMULATES                                      00010060
C       'NAMELIST/NAME/A,BB,ICC,DDD_4'                                   00010070
C       'READ(IUNIT,NAME,END=EOF)'                                       00010080
C       'READ(IUNIT,ANYNAM,END=EOF)'                                     00010090
C       IN THE CALLING PROGRAM USING:                                    00010100
C       ...                                                              00010110
C       REAL*8 A                                                         00010120
C       ...                                                              00010130
C       COMMON/NAME_LIST/A,BB(2),ICC(3),DDD_4(4)                         00010140
C       ...                                                              00010150
C       CALL NAMELIST(IUNIT,'$NAME',*EOF)                                00010160
C       ...                                                              00010170
C       CALL NAMELIST(IUNIT,'$ANYNAM',*EOF)                              00010180
C       ...                                                              00010190
C*************************************************************************00010200
C                                                                        00010210
      DATA C/'A','B','C','D','E','F','G','H','I','J','K','L','M','N',   00010220
     * 'O','P','Q','R','S','T','U','V','W','X','Y','Z','_',             00010230
     * '1','2','3','4','5','6','7','8','9','0',                         00010240
     * ' ','$','=',',','(','*',')','.','+','-'/                         00010250
      J=LEN(NAME)                                                        00010260
      IF(J.LT.2.OR.J.GT.7) THEN                                          00010270
        CALL ERRMSG('CALL NAMELIST ILLEGAL WITH NAME= '//                00010280
     1 NAME//'   (LENGTH<2 OR >7 CHAR''S)',1,6,0)                        00010290
      ENDIF                                                              00010300
      IF(NAME(1:1).NE.'$')                                               00010310
     1 CALL ERRMSG('CALL NAMELIST ILLEGAL WITH NAME= '//                 00010320
     2 NAME//'   (1ST CHAR MUST BE "$" CHAR)',1,6,0)                     00010330
C--INITIALIZE                                                            00010340
      INAME=0                                                            00010350
10    READ(IUNIT,11,END=99991,ERR=99992) BUF                             00010360
11    FORMAT(A80)                                                        00010370
      IF(INAME.EQ.1) GO TO 20                                            00010380
C--LOOK FOR "$NAME"                                                      00010390
        I=INDEX(BUF,NAME)                                                00010400
      IF(I.EQ.0) GO TO 10                                                00010410
      INAME=1                                                            00010420
      ICOL=I+J                                                           00010430
      JNAM=0                                                             00010440
      ILEN=0                                                             00010450
      VARNAM=' '                                                         00010460
      NUMLEN=0                                                           00010470
      IELE=1                                                             00010480
      GO TO 30                                                           00010490
20    ICOL=1                                                             00010500
30    CALL NONBLANK(BUF,LENBUF)                                          00010510
C==BEGIN PARSER LOOP (THE BIG 20000 LOOP)                                00010520
      IEND=0                                                             00010530
      DO 20000 I=ICOL,LENBUF                                             00010540
         BUFI=BUF(I:I)                                                   00010550
      DO 40 IC=1,27                                                      00010560
         IF(BUFI.EQ.C(IC)) GO TO 100                                     00010570
40    CONTINUE                                                           00010580
```

```
      DO 50 IC=28,37                                                    00010590
      IF(BUFI.EQ.C(IC)) GO TO 200                                       00010600
50    CONTINUE                                                          00010610
      DO 60 IC=38,47                                                    00010620
      IC_=IC-37                                                         00010630
      IF(BUFI.EQ.C(IC)) GO TO 70                                        00010640
60    CONTINUE                                                          00010650
61    WRITE(6,66) I,BUF                                                 00010660
66    FORMAT(/' {NAMELIST}: ERROR IN FOLLOWING RECORD AT COL(',I2,'):'/ 00010670
     1 1X,A80/<I>X,'^')                                                 00010680
      CALL ERRMSG('ILLEGAL CHAR="'//BUFI//'" FOUND',0,6,0)              00010690
67    WRITE(6,66) I,BUF                                                 00010700
      CALL ERRMSG('NUMLEN<1 IN DECODEIX    ',0,6,0)                     00010710
68    WRITE(6,66) I,BUF                                                 00010720
      CALL ERRMSG('NUMLEN<1 IN DECODEX',0,6,0)                          00010730
70    GO TO (20000,72,73,74,75,76,77,78,79,79),IC_                      00010740
C--'$' CHAR                                                             00010750
72    IEND=1                                                            00010760
      IF(NUMLEN.GT.0) GO TO 798                                         00010770
      IF(JNAM.EQ.0) GO TO 99990                                         00010780
      WRITE(6,66) I,BUF                                                 00010790
      CALL ERRMSG('MISPLACED "$" CHAR',0,6,0)                           00010800
C--'=' CHAR                                                             00010810
73    IEQ=1                                                             00010820
C--CHECK FOR VALID VARNAM, LENGTH ILEN, ETC.                            00010830
      IF(ILEN.LT.1) GO TO 733                                           00010840
      DO 732 J=1,NO_NAM                                                 00010850
      JNAM=J                                                            00010860
      JLEN=NAMLEN(J)                                                    00010870
      IF(JLEN.NE.ILEN) GO TO 732                                        00010880
      DO 731 K=1,JLEN                                                   00010890
      IF(VARNAM(K:K).NE.NAM(JNAM)(K:K)) GO TO 732                       00010900
731   CONTINUE                                                          00010910
C--VARNAM VERIFIED OK TO PROCEED TO NUMFLD(S)                           00010920
C                                                                       00010930
      IDIM=NAMDIM(JNAM)                                                 00010940
      NUMLEN=0                                                          00010950
      NDEC=0                                                            00010960
      NREP=1                                                            00010970
      NEXP=0                                                            00010980
      GO TO 20000                                                       00010990
732   CONTINUE                                                          00011000
      WRITE(6,66) I,BUF                                                 00011010
      CALL ERRMSG('ILLEGAL VARNAM='//VARNAM//' FOUND',0,6,0)            00011020
733   WRITE(6,66) I,BUF                                                 00011030
      CALL ERRMSG('MISPLACED "=" CHAR  ',0,6,0)                         00011040
C--',' CHAR                                                             00011050
74    IF(NUMLEN.GT.0) GO TO 799                                         00011060
      WRITE(6,66) I,BUF                                                 00011070
      CALL ERRMSG('MISPLACED "," CHAR',0,6,0)                           00011080
C--'(' CHAR                                                             00011090
75    IELE=0                                                            00011100
      GO TO 20000                                                       00011110
C--'*' CHAR                                                             00011120
76    IF(JNAM.EQ.0.OR.NUMLEN.LT.1.OR.NUMLEN.GT.5) GO TO 767             00011130
760   CALL DECODEIX(NUMFLD,NUMLEN,NREP,*67)                             00011140
      NUMLEN=0                                                          00011150
```

```
      IF(NREP.GT.0.AND.NREP.LE.NAMDIM(JNAM)) GO TO 20000                00011160
      WRITE(6,66) I,BUF                                                 00011170
      CALL ERRMSG('REPEAT FACTOR <1 OR >NAMDIM    ',0,6,0)              00011180
767   WRITE(6,66) I,BUF                                                 00011190
      CALL ERRMSG('REPEAT WIDTH > 5 OR MISPLACED "*" CHAR',0,6,0)       00011200
C--')' CHAR                                                             00011210
77    IF(IELE.NE.0) GO TO 772                                           00011220
      CALL DECODEIX(NUMFLD,NUMLEN,IELE,*67)                             00011230
      IF(IELE.LT.1) GO TO 773                                           00011240
      NREP=1                                                            00011250
      GO TO 20000                                                       00011260
772   WRITE(6,66) I,BUF                                                 00011270
      CALL ERRMSG('MISPLACED ")" CHAR',0,6,0)                           00011280
773   WRITE(6,66) I,BUF                                                 00011290
      CALL ERRMSG('ARRAY IELE<1 OR >NAMDIM  ',0,6,0)                    00011300
C--'.' CHAR                                                             00011310
78    IF(JNAM.EQ.0.OR.NEXP.GT.0.OR.NDEC.GT.0) GO TO 781                 00011320
      NDEC=NUMLEN+1                                                     00011330
      IF(NAMTYP(JNAM).EQ.1) GO TO 200                                   00011340
781   WRITE(6,66) I,BUF                                                 00011350
      CALL ERRMSG('MISPLACED "." CHAR',0,6,0)                           00011360
C--'-' OR '+' CHAR                                                      00011370
79    IF(IELE.GT.0.OR.NEXP.GT.0) GO TO 210                              00011380
      WRITE(6,66) I,BUF                                                 00011390
      CALL ERRMSG('MISPLACED "-" OR "+" CHAR',0,6,0)                    00011400
C--<ALP> CHAR                                                           00011410
100   IF(NUMLEN.GT.0) GO TO 209                                         00011420
      IF(ILEN.GT.0) GO TO 102                                           00011430
      IEQ=0                                                             00011440
      IELE=1                                                            00011450
102   ILEN=ILEN+1                                                       00011460
      IF(ILEN.GT.6) GO TO 101                                           00011470
      VARNAM(ILEN:ILEN)=BUFI                                            00011480
      GO TO 20000                                                       00011490
101   WRITE(6,66) I,BUF                                                 00011500
      CALL ERRMSG('VARNAM>6 CHAR''S',0,6,0)                             00011510
C--<+-NUM> CHAR                                                         00011520
200   IF(IELE.EQ.0) GO TO 210                                           00011530
      IF(IEQ.EQ.0) GO TO 102                                            00011540
      GO TO 210                                                         00011550
209   IF(BUFI.EQ.'E'.OR.BUFI.EQ.'D') THEN                               00011560
        NEXP=NUMLEN+1                                                   00011570
      ELSE                                                              00011580
        GO TO 61                                                        00011590
      ENDIF                                                             00011600
210   NUMLEN=NUMLEN+1                                                   00011610
      IF(NUMLEN.GT.20) GO TO 211                                        00011620
      NUMFLD(NUMLEN:NUMLEN)=BUFI                                        00011630
      GO TO 20000                                                       00011640
211   WRITE(6,66) I,BUF                                                 00011650
      CALL ERRMSG('NUM FIELD>20 CHAR''S',0,6,0)                         00011660
C--PROCESS NUMBER FIELD                                                 00011670
799   IDIM=IDIM-1                                                       00011680
      IF(IDIM.LT.0) GO TO 10004                                         00011690
798   IF(NEXP.GT.0) GO TO 1000                                          00011700
C--[NEXP=0]                                                             00011710
      IF(NDEC.GT.0) GO TO 899                                           00011720
```

```
C--[NEXP=0, NDEC=0]                                                    00011730
      CALL DECODEIX(NUMFLD,NUMLEN,IX,*67)                              00011740
C--CONVERT IX AND STORE IN COMMON                                      00011750
800   X=IX                                                             00011760
      IF(IELE.GT.NAMDIM(JNAM)) GO TO 773                               00011770
8000  GO TO (801,802,803,804,805,806,807,808,809,810,                  00011780
     * 811,812,813,814,815,816,817,818,819,820,                       00011790
     * 821,822,823,824,825,826,827,828,829,830,                       00011800
     * 831,832,833,834,835,836,837,838,839,840,                       00011810
     * 841,842,843,844,845,846,847,848,849,850,                       00011820
     * 851,852,853,854,855,856,857,858,859,860),JNAM                  00011830
801   V1(IELE)=X                                                       00011840
      GO TO 10000                                                      00011850
802   V2(IELE)=X                                                       00011860
      GO TO 10000                                                      00011870
803   V3(IELE)=X                                                       00011880
      GO TO 10000                                                      00011890
804   V4(IELE)=X                                                       00011900
      GO TO 10000                                                      00011910
805   V5(IELE)=X                                                       00011920
      GO TO 10000                                                      00011930
806   V6(IELE)=X                                                       00011940
      GO TO 10000                                                      00011950
807   V7(IELE)=X                                                       00011960
      GO TO 10000                                                      00011970
808   V8(IELE)=X                                                       00011980
      GO TO 10000                                                      00011990
809   V9(IELE)=X                                                       00012000
      GO TO 10000                                                      00012010
810   V10(IELE)=X                                                      00012020
      GO TO 10000                                                      00012030
811   V11(IELE)=X                                                      00012040
      GO TO 10000                                                      00012050
812   V12(IELE)=X                                                      00012060
      GO TO 10000                                                      00012070
813   V13(IELE)=X                                                      00012080
      GO TO 10000                                                      00012090
814   V14(IELE)=X                                                      00012100
      GO TO 10000                                                      00012110
815   V15(IELE)=X                                                      00012120
      GO TO 10000                                                      00012130
816   V16(IELE)=X                                                      00012140
      GO TO 10000                                                      00012150
817   V17(IELE)=X                                                      00012160
      GO TO 10000                                                      00012170
818   V18(IELE)=X                                                      00012180
      GO TO 10000                                                      00012190
819   V19(IELE)=X                                                      00012200
      GO TO 10000                                                      00012210
820   V20(IELE)=X                                                      00012220
      GO TO 10000                                                      00012230
821   V21(IELE)=X                                                      00012240
      GO TO 10000                                                      00012250
822   V22(IELE)=X                                                      00012260
      GO TO 10000                                                      00012270
823   V23(IELE)=X                                                      00012280
      GO TO 10000                                                      00012290
```

```
824   V24(IELE)=X                                                    00012300
      GO TO 10000                                                    00012310
825   V25(IELE)=X                                                    00012320
      GO TO 10000                                                    00012330
826   V26(IELE)=X                                                    00012340
      GO TO 10000                                                    00012350
827   V27(IELE)=X                                                    00012360
      GO TO 10000                                                    00012370
828   V28(IELE)=X                                                    00012380
      GO TO 10000                                                    00012390
829   V29(IELE)=X                                                    00012400
      GO TO 10000                                                    00012410
830   V30(IELE)=X                                                    00012420
      GO TO 10000                                                    00012430
831   V31(IELE)=X                                                    00012440
      GO TO 10000                                                    00012450
832   V32(IELE)=X                                                    00012460
      GO TO 10000                                                    00012470
833   V33(IELE)=X                                                    00012480
      GO TO 10000                                                    00012490
834   V34(IELE)=X                                                    00012500
      GO TO 10000                                                    00012510
835   V35(IELE)=X                                                    00012520
      GO TO 10000                                                    00012530
836   V36(IELE)=X                                                    00012540
      GO TO 10000                                                    00012550
837   V37(IELE)=X                                                    00012560
      GO TO 10000                                                    00012570
838   V38(IELE)=X                                                    00012580
      GO TO 10000                                                    00012590
839   V39(IELE)=X                                                    00012600
      GO TO 10000                                                    00012610
840   V40(IELE)=X                                                    00012620
      GO TO 10000                                                    00012630
841   V41(IELE)=X                                                    00012640
      GO TO 10000                                                    00012650
842   V42(IELE)=X                                                    00012660
      GO TO 10000                                                    00012670
843   V43(IELE)=X                                                    00012680
      GO TO 10000                                                    00012690
844   V44(IELE)=X                                                    00012700
      GO TO 10000                                                    00012710
845   V45(IELE)=X                                                    00012720
      GO TO 10000                                                    00012730
846   V46(IELE)=X                                                    00012740
      GO TO 10000                                                    00012750
847   V47(IELE)=X                                                    00012760
      GO TO 10000                                                    00012770
848   V48(IELE)=X                                                    00012780
      GO TO 10000                                                    00012790
849   V49(IELE)=X                                                    00012800
      GO TO 10000                                                    00012810
850   V50(IELE)=X                                                    00012820
      GO TO 10000                                                    00012830
851   V51(IELE)=X                                                    00012840
      GO TO 10000                                                    00012850
852   V52(IELE)=X                                                    00012860
```

```
      GO TO 10000                                                    00012870
853   V53(IELE)=X                                                    00012880
      GO TO 10000                                                    00012890
854   V54(IELE)=X                                                    00012900
      GO TO 10000                                                    00012910
855   V55(IELE)=X                                                    00012920
      GO TO 10000                                                    00012930
856   V56(IELE)=X                                                    00012940
      GO TO 10000                                                    00012950
857   V57(IELE)=X                                                    00012960
      GO TO 10000                                                    00012970
858   V58(IELE)=X                                                    00012980
      GO TO 10000                                                    00012990
859   V59(IELE)=X                                                    00013000
      GO TO 10000                                                    00013010
860   V60(IELE)=X                                                    00013020
      GO TO 10000                                                    00013030
C--[NEXP=0, NDEC>0]                                                  00013040
899   CALL DECODEX(NUMFLD,NUMLEN,NDEC,X,*68)                         00013050
C--CONVERT X AND STORE IN COMMON                                     00013060
900   IF(IELE.GT.NAMDIM(JNAM)) GO TO 773                             00013070
      GO TO 8000                                                     00013080
C--[NEXP>0]                                                          00013090
1000  IF(NDEC.GT.0) GO TO 2000                                       00013100
C--[NEXP>0, NDEC=0]                                                  00013110
      CALL DECODEIX(NUMFLD,NEXP-1,IX,*67)                            00013120
      X=IX                                                           00013130
1002  J=1                                                            00013140
      DO 1001 K=NEXP+1,NUMLEN                                        00013150
      NUMFLD(J:J)=NUMFLD(K:K)                                        00013160
1001  J=J+1                                                          00013170
      CALL DECODEIX(NUMFLD,NUMLEN-NEXP,IE,*67)                       00013180
      X=X*10.**IE                                                    00013190
C** {LATER INSERT A CALL TO A OVERFLOW HANDLER, ETC.}                 00013200
      GO TO 900                                                      00013210
C--[NEXP>0, NDEC>0]                                                  00013220
2000  CALL DECODEX(NUMFLD,NEXP-1,NDEC,X,*68)                         00013230
      GO TO 1002                                                     00013240
C--NEXT IELE?                                                        00013250
10000 IELE=IELE+1                                                    00013260
      IF(IELE.GT.NAMDIM(JNAM)) GO TO 10002                           00013270
      IF(NREP.GT.1) GO TO 10003                                      00013280
10001 IF(IEND.EQ.1) GO TO 99990                                      00013290
      NUMLEN=0                                                       00013300
      NDEC=0                                                         00013310
      NEXP=0                                                         00013320
      NREP=1                                                         00013330
      ILEN=0                                                         00013340
      VARNAM=' '                                                     00013350
      GO TO 20000                                                    00013360
10002 IELE=1                                                         00013370
      GO TO 10001                                                    00013380
10003 NREP=NREP-1                                                    00013390
      IDIM=IDIM-1                                                    00013400
      IF(IDIM.GE.0) GO TO 8000                                       00013410
10004 WRITE(6,66) I,BUF                                              00013420
      CALL ERRMSG('TOO MANY ELEMENTS FOR GIVEN NAMDIM.',0,6,0)       00013430
```

```
C==END OF DO 20000   CONTINUE PARSER -OR- READ IN NEXT BUF, ETC.        00013440
20000 CONTINUE                                                          00013450
                GO TO 10                                                00013460
C--'$' CHAR (DELIMITER $[END] FOR THIS $NAME --$)                       00013470
99990 RETURN                                                            00013480
C--E.O.F. ON FILE IUNIT ENCOUNTERED.                                    00013490
99991 RETURN 1                                                          00013500
99992 CALL ERRMSG('CANNOT OPEN/READ CALL NAMELIST(IFILE,...)',1,6,0)    00013510
      END                                                               00013520
      SUBROUTINE DECODEIX(NUMFLD,NUMLEN,IX,*)                           00013530
C--USED IN CALL NAMELIST(IUNIT,'$NAME',*)                               00013540
      CHARACTER*9 FMT                                                   00013550
      CHARACTER*20 NUMFLD                                               00013560
      IF(NUMLEN.LT.1) RETURN 1                                          00013570
      IDIFF=20-NUMLEN                                                   00013580
      IF(IDIFF.EQ.0) THEN                                               00013590
        ENCODE(9,991,FMT) NUMLEN                                        00013600
      ELSE                                                              00013610
        ENCODE(9,992,FMT) NUMLEN,IDIFF                                  00013620
      ENDIF                                                             00013630
991   FORMAT('(I',I2,'    )')                                          00013640
992   FORMAT('(I',I2,',',I2,'X)')                                       00013650
      DECODE(9,FMT,NUMFLD) IX                                           00013660
      RETURN                                                            00013670
      END                                                               00013680
      SUBROUTINE DECODEX(NUMFLD,NUMLEN,NDEC,X,*)                        00013690
C--USED IN CALL NAMELIST(IUNIT,'$NAME',*)                               00013700
      CHARACTER*12 FMT                                                  00013710
      CHARACTER*20 NUMFLD                                               00013720
      IF(NUMLEN.LT.1) RETURN 1                                          00013730
      LENDEC=NUMLEN-NDEC                                                00013740
      IDIFF=20-NUMLEN                                                   00013750
      IF(IDIFF.EQ.0) THEN                                               00013760
       ENCODE(12,991,FMT) NUMLEN,LENDEC                                 00013770
      ELSE                                                              00013780
       ENCODE(12,992,FMT) NUMLEN,LENDEC,IDIFF                           00013790
      ENDIF                                                             00013800
991   FORMAT('(F',I2,'.',I2,'    )')                                    00013810
992   FORMAT('(F',I2,'.',I2,',',I2,'X)')                                00013820
      DECODE(12,FMT,NUMFLD) X                                           00013830
      RETURN                                                            00013840
      END                                                               00013850
      SUBROUTINE ERRMSG(MSG,ISKIP,IUNIT1,IUNIT2)                        00013860
C                                                                       00013870
C  GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780                  00013880
C                                                                       00013890
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                                  00013900
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2         00013910
C          > 0 FOR ONE BLANK LINE BEFORE.                               00013920
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).     00013930
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).     00013940
C                                                                       00013950
C  MESSAGES ARE WRITTEN IN THE FORM:                                    00013960
C                                                                       00013970
C  {ERRMSG}: _MSG_HERE_                                                 00013980
C                                                                       00013990
      CHARACTER*(*) MSG                                                 00014000
```

```
      I=LEN(MSG)                                                        00014010
      DO 1 J=1,2                                                        00014020
        IF(J.EQ.1) THEN                                                 00014030
       JUNIT=IUNIT1                                                     00014040
          ELSE                                                          00014050
       JUNIT=IUNIT2                                                     00014060
        ENDIF                                                           00014070
        IF(JUNIT.GT.0) THEN                                             00014080
          IF(ISKIP.EQ.0) THEN                                           00014090
            WRITE(JUNIT,2) MSG                                          00014100
            ELSE                                                        00014110
       WRITE(JUNIT,3) MSG                                               00014120
          ENDIF                                                         00014130
      ENDIF                                                             00014140
1     CONTINUE                                                          00014150
      CALL EXIT                                                         00014160
2     FORMAT(1X,'{ERRMSG}: ',A<I>)                                      00014170
3     FORMAT(/1X,'{ERRMSG}: ',A<I>)                                     00014180
      END                                                               00014190
      SUBROUTINE NONBLANK(C,NB)                                         00014200
C--DETERMINE NON-BLANK CHAR LENGTH (=NB ON EXIT) OF C*(*)               00014210
C  NOTE THAT NB WILL BE IN [0,LEN(C)].                                  00014220
C                                                                       00014230
      CHARACTER*(*) C                                                   00014240
      L=LEN(C)                                                          00014250
      DO 10 I=L,1,-1                                                    00014260
        NB=I                                                            00014270
        IF(C(I:I).NE.' ') RETURN                                        00014280
10    CONTINUE                                                          00014290
      NB=0                                                              00014300
      RETURN                                                            00014310
      END                                                               00014320
      REAL FUNCTION ASINH(X)                                            00014330
C--INVERSE HYPERBOLIC SIN FUNCTION                                      00014340
C                                                                       00014350
      REAL*8 X2                                                         00014360
      X2=X                                                              00014370
      ASINH=DLOG(X2+DSQRT(X2*X2+1.0DQ))                                 00014380
      RETURN                                                            00014390
      END                                                               00014400
      FUNCTION ERF(X)                                                   00014410
C                                                                       00014420
C  ERF COMPUTES THE ERROR FUNCTION TO ABOUT 7-PLACES.                   00014430
C  SEE MATH. OF COMP., V.22,N.101,JAN,1968.                             00014440
C   ALSO, SEE ERFINV(X).                                                00014450
C                                                                       00014460
      DIMENSION A1(19),A2(19)                                           00014470
      DATA A1/.70322500,.33050152,.20133975,.10863025,                  00014480
     1 .46775523E-1,.15398573E-1,.38015077E-2,.69718379E-3,             00014490
     2 .94490927E-4,.94328117E-5,.69192752E-6,.37225234E-7,             00014500
     3 .14666061E-8,.42261614E-10,.88978652E-12,.13676044E-13,          00014510
     4 .15334234E-15,.12536751E-17,.74517E-20/                          00014520
      DATA A2/.24725517,.14422723,.86989455E-1,.43977338E-1,            00014530
     1 .17243963E-1,.50790696E-2,.11086065E-2,.17822802E-3,             00014540
     2 .21040458E-4,.18206632E-5,.11533099E-6,.53427503E-8,             00014550
     3 .18084859E-9,.44696823E-11,.80606884E-13,.10601364E-14,          00014560
     4 .10164928E-16,.710005E-19,0.0/                                   00014570
```

```
      IF(X.EQ.0.0) THEN                                                 00014580
        ERF=0.0                                                         00014590
        RETURN                                                          00014600
      ENDIF                                                             00014610
        B=2.*X/5.                                                       00014620
        S=SIN(B)                                                        00014630
        C=COS(B)                                                        00014640
        C2=C+C                                                          00014650
        ALP=C2*C-1.                                                     00014660
        SUM=0.0                                                         00014670
        DO 10 N=1,19                                                    00014680
          SUM=SUM+(A1(N)+C2*A2(N))*ALP**(N-1)                           00014690
10      CONTINUE                                                        00014700
        ERF=B/3.1415927+S*SUM                                           00014710
        RETURN                                                          00014720
        END                                                             00014730
      FUNCTION ERFINV(Y)                                                00014740
C                                                                       00014750
C  ERFINV COMPUTES THE INVERSE ERROR FUNCTION TO ABOUT 7-PLACES.        00014760
C  SEE MATH. OF COMP., V.22,N.101,JAN,1968.                             00014770
C  ALSO, SEE ERF(X).                                                    00014780
C                                                                       00014790
      CHARACTER*16 XX                                                   00014800
      DIMENSION T3(1:38),T4(0:26),T5(0:37),T6(0:25)                     00014810
      DATA T3/.12046752,.16078199E-1,.26867044E-2,.49963473E-3,         00014820
     1 .98898219E-4,.20391813E-4,.43272716E-5,.93808141E-6,             00014830
     2 .20673472E-6,.46159699E-7,.10416680E-7,.23715100E-8,             00014840
     3 .54392841E-9,.12554899E-9,.29138180E-10,.67949422E-11,           00014850
     4 .15912343E-11,.37402505E-12,.88208776E-13,.20865090E-13,         00014860
     5 .49488041E-14,.11766395E-14,.28038557E-15,.66950664E-16,         00014870
     6 .16016550E-16,.38382583E-17,.9212851E-18,.2214615E-18,           00014880
     7 .533091E-19,.128488E-19,.31006E-20,.7491E-21,.1812E-21,          00014890
     8 .439E-22,.106E-22,.26E-23,.6E-24,.2E-24/                         00014900
      DATA T4/.91215880,-.16266282E-1,.43355647E-3,.21443857E-3,        00014910
     1 .26257511E-5,-.30210911E-5,-.12406061E-7,.62406609E-7,           00014920
     2 -.54012479E-9,-.14232079E-8,.34384028E-10,.33584870E-10,         00014930
     3 -.14584289E-11,-.81021743E-12,.52532409E-13,.19711541E-13,       00014940
     4 -.17494334E-14,-.48005966E-15,.55730299E-16,.11632605E-16,       00014950
     5 -.17262489E-17,-.2784973E-18,.524481E-19,.65270E-20,             00014960
     6 -.15707E-20,-.1475E-21,.450E-22/                                 00014970
      DATA T5/.95667971,-.23107004E-1,-.43742361E-2,-.57650342E-3,      00014980
     1 -.10961022E-4,.25108547E-4,.10562336E-4,.27544123E-5,            00014990
     2 .43248450E-6,-.20530336E-7,-.43891537E-7,-.17684010E-7,          00015000
     3 -.39912890E-8,-.18693241E-9,.27292274E-9,.13281721E-9,           00015010
     4 .31834248E-10,.16700608E-11,-.20364650E-11,-.96484681E-12,       00015020
     5 -.21956727E-12,-.95689813E-14,.13703257E-13,.62538505E-14,       00015030
     6 .14584615E-14,.10781240E-15,-.70922999E-16,-.39141178E-16,       00015040
     7 -.11165921E-16,-.15770366E-17,.2853149E-18,.2716662E-18,         00015050
     8 .957770E-19,.176835E-19,-.9828E-21,-.20464E-20,-.802E-21,        00015060
     9 -.1650E-21/                                                      00015070
      DATA T6/.98857506,.10857705E-1,-.17511651E-2,.21196993E-4,        00015080
     1 .15664871E-4,-.51904169E-5,-.37135790E-7,.12174309E-8,           00015090
     2 -.17681155E-9,-.11937218E-10,.38025054E-12,-.66018832E-13,       00015100
     3 -.87917055E-14,-.35068693E-15,-.69722150E-16,-.10956794E-16,     00015110
     4 -.11536390E-17,-.1326235E-18,-.263938E-19,.5341E-21,             00015120
     5 -.2261E-20,.9552E-21,-.525E-21,.2487E-21,-.1134E-21,.42E-22/     00015130
      X=Y                                                               00015140
```

```
      X1=ABS(X)                                                          00015150
      IF(X1.GE.1.0) THEN                                                 00015160
        ENCODE(16,1,XX) X1                                               00015170
1       FORMAT(E16.8)                                                    00015180
        IF(X1.GT.1.000001)CALL ERRMSG('ABS(X)='//XX//                    00015190
     1    ' >1.000001 IN [ERFINV]',0,6,0)                                00015200
        CALL WARN('ABS(X)='//XX//                                        00015210
     2 ' >=1.0 IN [ERFINV]; X=0.9999998*SIGN(1.,X) USED.',0,6,0,*2)      00015220
2       X=0.9999998*SIGN(1.,X)                                           00015230
      ENDIF                                                              00015240
      X1=1.-X                                                            00015250
      IF(X.GE.0.8.AND.X.LE.0.9975) THEN                                  00015260
        BETA=SQRT(-ALOG(1.-X*X))                                         00015270
        R=0.0                                                            00015280
        DO 10 N=0,26                                                     00015290
10        R=R+T4(N)*TCHEB(N,-1.54881304*BETA+2.5654901)                  00015300
        ERFINV=BETA*R                                                    00015310
      ELSE IF(X1.GE.5E-16.AND.X1.LE.25E-4) THEN                          00015320
        BETA=SQRT(-ALOG(1.-X*X))                                         00015330
        R=0.0                                                            00015340
        DO 20 N=0,37                                                     00015350
20        R=R+T5(N)*TCHEB(N,-.55945763*BETA+2.2879157)                   00015360
        ERFINV=BETA*R                                                    00015370
      ELSE IF(X1.LT.5E-16) THEN                                          00015380
        BETA=SQRT(-ALOG(1.-X*X))                                         00015390
        SBETA=SQRT(BETA)                                                 00015400
        R=0.0                                                            00015410
        DO 30 N=0,25                                                     00015420
30        R=R+T6(N)*TCHEB(N,-9.1999924/SBETA+2.7949908)                  00015430
        ERFINV=BETA*R                                                    00015440
      ELSE                                                               00015450
        R=0.0                                                            00015460
        A=X*X/.32-1.                                                     00015470
        DO 40 N=1,38                                                     00015480
40        R=R+T3(N)*TCHEB(N,A)                                           00015490
        ERFINV=X*(.99288538+R)                                           00015500
      ENDIF                                                              00015510
      RETURN                                                             00015520
      END                                                                00015530
      INTEGER FUNCTION LOC(I,J)                                          00015540
C--GETS ACTUAL ADDR OF A(I,J)=A(J,I) SYMMETRIC MATRIX                    00015550
C  STORED AS THE VECTOR A(LOC(I,J)) OF N*(N+1)/2 ELEMENTS--              00015560
C  WHERE ANY I,J.LE.N MAY BE USED (N NOT EXPLICITLY NEEDED)...           00015570
C                                                                        00015580
      IF(I-J) 10,20,20                                                   00015590
   10 LOC=I+(J*J-J)/2                                                    00015600
      RETURN                                                             00015610
   20 LOC=J+(I*I-I)/2                                                    00015620
      RETURN                                                             00015630
      END                                                                00015640
      FUNCTION TCHEB(N,X)                                                00015650
C                                                                        00015660
C  TCHEBYSHEV POLYNOMIAL OR ORDER N.GE.0.AND.N.LE.100, ARGUMENT X.       00015670
C                                                                        00015680
      DIMENSION Y(101)                                                   00015690
      IF(N.GT.100)CALL ERRMSG('N>100 IN {TCHEB}',0,6,0)                  00015700
      IF(N.LE.0) THEN                                                    00015710
```

```
        TCHEB=1.                                                    00015720
        RETURN                                                      00015730
      ELSE IF(N.EQ.1) THEN                                          00015740
        TCHEB=X                                                     00015750
        RETURN                                                      00015760
      ELSE                                                          00015770
        Y(1)=1.                                                     00015780
        Y(2)=X                                                      00015790
3       F=X+X                                                       00015800
        DO 4 I=2,N                                                  00015810
4       Y(I+1)=F*Y(I)-Y(I-1)                                        00015820
        TCHEB=Y(N+1)                                                00015830
      ENDIF                                                         00015840
      RETURN                                                        00015850
      END                                                           00015860
      SUBROUTINE WARN(MSG,ISKIP,IUNIT1,IUNIT2,*)                    00015870
C                                                                   00015880
C  GENERAL WARNING MESSAGE OUTPUT AND RETURN 1 ON VAX-11/780        00015890
C                                                                   00015900
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                              00015910
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2     00015920
C          > 0 FOR ONE BLANK LINE BEFORE.                           00015930
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1). 00015940
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2). 00015950
C                                                                   00015960
C  MESSAGES ARE WRITTEN IN THE FORM:                                00015970
C                                                                   00015980
C  {WARN}: _MSG_HERE_                                               00015990
C                                                                   00016000
      CHARACTER*(*) MSG                                             00016010
      I=LEN(MSG)                                                    00016020
      DO 1 J=1,2                                                    00016030
        IF(J.EQ.1) THEN                                             00016040
       JUNIT=IUNIT1                                                 00016050
          ELSE                                                      00016060
       JUNIT=IUNIT2                                                 00016070
        ENDIF                                                       00016080
        IF(JUNIT.GT.0) THEN                                         00016090
          IF(ISKIP.EQ.0) THEN                                       00016100
            WRITE(JUNIT,2) MSG                                      00016110
            ELSE                                                    00016120
       WRITE(JUNIT,3) MSG                                           00016130
          ENDIF                                                     00016140
      ENDIF                                                         00016150
1     CONTINUE                                                      00016160
      RETURN 1                                                      00016170
2     FORMAT(1X,'{WARN}: ',A<I>)                                    00016180
3     FORMAT(/1X,'{WARN}: ',A<I>)                                   00016190
      END                                                           00016200
C {T1NLSOL}: TEST1 FOR NLSOL USING A 7-PARAMETER PROBLEM            00016210
C                                                                   00016220
      EXTERNAL T1FCODE,T1PCODE,T1SUBZ,T1SUBEND                      00016230
      CALL NLSOL(T1FCODE,T1PCODE,T1SUBZ,T1SUBEND)                   00016240
      CALL EXIT                                                     00016250
      END                                                           00016260
      SUBROUTINE T1FCODE(Y,X,B,PASS,F,IN,IDER)                      00016270
C--FUNCTION EVALUATION FOR 7-PARAMETER PROBLEM                      00016280
```

```
      DIMENSION Y(1),X(500,5),B(1),PASS(5)                              00016290
      PASS(1)=X(IN,1)                                                   00016300
      T=PASS(1)                                                         00016310
      F=B(1)+B(2)*T+B(3)*T**2+B(4)*SIN(B(5)*T)+B(6)*                    00016320
     1 EXP(-B(7)*T)                                                     00016330
      RETURN                                                            00016340
      END                                                               00016350
      SUBROUTINE T1PCODE(P,X,B,PASS,F,IN,IP,IB)                         00016360
C--ANALYTIC DERIVATIVES FOR 7-PARAMETER PROBLEM                         00016370
      DIMENSION P(1),X(500,5),B(1),PASS(5),IB(1)                        00016380
      T=PASS(1)                                                         00016390
      P(1)=1.0                                                          00016400
      P(2)=T                                                            00016410
      P(3)=T**2                                                         00016420
      P(4)=SIN(B(5)*T)                                                  00016430
      P(5)=B(4)*T*COS(B(5)*T)                                           00016440
      P(6)=EXP(-B(7)*T)                                                 00016450
      P(7)=-B(6)*T*P(6)                                                 00016460
      IF(IP.EQ.0) RETURN                                                00016470
      DO 10 I=1,IP                                                      00016480
      DO 10 J=1,7                                                       00016490
        IF(IB(I).NE.J) GO TO 10                                         00016500
        P(J)=0.0                                                        00016510
10    CONTINUE                                                          00016520
      RETURN                                                            00016530
      END                                                               00016540
      SUBROUTINE T1SUBZ(Y,X,B,PASS,NPASS,N,TITLE,IOUT)                  00016550
C--INITIALIZATION FOR 7-PARAMETER PROBLEM                               00016560
C  ($INIT INPUT NOT NEEDED IN THIS EXAMPLE)                             00016570
      DIMENSION Y(1),X(500,5),B(1),PASS(5)                              00016580
      CHARACTER*80 TITLE                                                00016590
      NPASS=1                                                           00016600
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                                  00016610
10    FORMAT('0{T1NLSOL}:',5X,A)                                        00016620
      RETURN                                                            00016630
      END                                                               00016640
      SUBROUTINE T1SUBEND(Y,X,B,K,N,TITLE,IOUT)                         00016650
C--TERMINATION FOR 7-PARAMETER PROBLEM                                  00016660
      DIMENSION Y(1),X(500,5),B(1)                                      00016670
      CHARACTER*80 TITLE                                                00016680
      WRITE(6,10)                                                       00016690
10    FORMAT(/' ********  E N D  ********'/                             00016700
     1' ** FINAL SOLUTION VECTOR:'/)                                    00016710
      IF(IOUT.EQ.1) WRITE(16,10)                                        00016720
      DO 30 I=1,K                                                       00016730
      WRITE(6,20) I,B(I)                                                00016740
20    FORMAT(2X,I3,E16.8)                                               00016750
      IF(IOUT.EQ.1) WRITE(16,20) I,B(I)                                 00016760
30    CONTINUE                                                          00016770
      RETURN                                                            00016780
      END                                                               00016790
      SUBROUTINE NL2SOL(N, P, X, CALCR, CALCJ, IV, V, UIPARM, URPARM,   00016800
     1                  UFPARM)                                         00016810
C** VAX-11/780 VERSION {12/18/80} MODIFIED BY                           00016820
C** W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.             00016830
C                                                                       00016840
>>>>>  The rest of NL2SOL listing has been suppressed,  <<<<<<
>>>>>  but is available on the distributed tape.        <<<<<<
```