U.S. DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

Nonlinear least-squares inversion of frequency-domain induced polarization data (Program NLSIP)

by

Walter L. Anderson
and
Bruce D. Smith

Open-File Report 86-280

1986

DISCLAIMER

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

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

CONTENTS

ABSTRACT

A computer program is presented that inverts frequency-domain induced polarization (IP) data using an adaptive nonlinear least-squares method. Multiple Cole-Cole relaxation models may be used to define the frequency-domain IP forward function, which can be useful when IP and electromagnetic (EM) coupling effects are present in the observed data. An inversion example using an IP data set is given in numerical and graphical form. Program parameters are defined, and the VAX/VMS operating instructions are summarized. The FORTRAN program source is listed in an appendix.

INTRODUCTION

The inversion of frequency-domain induced polarization (IP) data is provided by program NLSIP, which is described in this report. The numerical technique uses a general adaptive nonlinear least-squares algorithm originally developed by Dennis and others (1979; 1981), and extended externally for constrained nonlinear regression by Anderson (1982). The frequency-domain IP forward problem, required iteratively for the inverse solution, uses Cole-Cole dispersion or relaxation models as defined by Pelton and others (1978), and Major and Silic (1981). Note that a stand-alone forward (FWD) program can be easily obtained directly from the NLSIP inversion program, which is described in general by Anderson (1984) for converting any NLS-program to a FWD-program (or vise-versa).

This report utilizes the general nonlinear least-squares (NLS) method defined by Anderson (1982), but only as it applies here to observed frequency-domain IP data. A similar program for inversion of time-domain IP data was previously published by Anderson and Smith (1984), which uses many aspects of the present frequency-domain program. A Cole-Cole relaxation model is used to evaluate the frequency response over a given range of frequencies. Either amplitude, phase, real, or imaginary spectra of the complex IP impedance can be used separately or jointly in the least-squares solution; the joint uses of spectral components are also termed "mixed obervation types" in this report. Additionally, separate IP data sets from different sites can be processed simultaneously (i.e., concatenated) by NLSIP to obtain a joint inversion. This might be useful, for example, when different measurement arrays are used over a given area in an effort to enhance the results of least-squares interpretation.

Up to four arbitrary relaxation models can be combined (i.e., multiplied as shown by Major and Silic, 1981, p. 917) in one execution of NLSIP. The use of multiple Cole-Cole dispersions may be used to approximate the combined effect of IP and electromagnetic (EM) coupling. This topic was discussed in some detail by Washburne (1982), where he suggests using multiplicative combinations rather than additive models. However, he emphasizes that this is only an intuitive choice based on the behavior of typical frequency-domain responses.

The remainder of this report contains 1) a summary of the general computations, 2) a description of the required program parameters, and 3) the VAX/VMS operating instructions. Appendix 1 offers some suggestions for converting the VAX program to other computer systems; Appendix 2 lists an input/output test example; and Appendix 3 gives a FORTRAN-77 source listing.

SUMMARY OF CALCULATIONS

The NLS method (Anderson, 1982; Dennis and others, 1979, 1981) requires a twice-continuously differentiable nonlinear objective function describing the model equation as a function of the unknown parameters. In our case, a single Cole-Cole relaxation model in the frequency-domain can be written (Major and Silic, 1981) as

$$Z(\omega) = R_0 \left[1 - m \left(1 - \frac{1}{1 + (i\omega\tau)^c} \right) \right], \qquad (1)$$

where R_0 = amplitude of Z( ) at zero frequency ($R_0 > 0$),
m = chargeability factor ($0 \leq |m| \leq 1$),
τ = time constant ($\tau > 0$ seconds),
c = frequency dependency factor ($0 < c \leq 1$),
ω = angular frequency ($\omega = 2\pi f$, $f > 0$ Hertz),
i = $(-1)^{1/2}$,
and $Z(\omega)$ = complex impedance at frequency ω, which can be represented by real numbers with single or joint uses of the amplitude, phase, real-, or imaginary-parts of $Z(\omega)$.

The four model parameters (R_0, m, τ, c) are to be determined by the NLS inversion program, given an observed data set for $Z(\omega)$ defined over a range of ω, and an initial estimate. The physical meaning of these parameters is not specifically addressed here, consequently, we refer the interested reader to Pelton and others (1978), and Major and Silic (1981) for more details. Note that m can be negative in our formulation to account for negative EM coupling cases. Generally we choose phase(Z)<0 so that m>0; e.g., see the test problem in Appendix 2.

To study EM inductive coupling (IC) effects when using IP measurements, multiple Cole-Cole relaxation models can be combined by multiplication (Major and Silic, 1981) as

$$Z_T(\omega) = R_0[1 - m_1 f(\omega, \tau_1, c_1)][1 - m_2 f(\omega, \tau_2, c_2)], \qquad (2)$$

where

$$f(\omega, \tau, c) = 1 - \frac{1}{1 + (i\omega\tau)^c}.$$

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

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

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

* Note K=3*MODEL+1 is the total number of parameters.

Many NLS options are available in the interface subprogram NLSOL (Anderson, 1982, p. 11-21), which the reader should become familiar with before attempting to run NLSIP. Following the NLS notation in Anderson (1982, p.11-12), we let X(I,1)=frequency ($f=\omega/2\pi$) and Y(I) be the corresponding observed real component: amplitude, phase, real-, or imaginary-part of Z(f), where each data array X(I,1), Y(I) is given for I=1,2,...,N, N>K. In general, a given data set should be given in ascending frequency order for all I; however, this is dependent of the paraticular data matrix option selected as discussed in the section: DATA MATRIX OPTIONS. Program NLSIP reads the observed data matrix in N rows in the following order:

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

using an arbitrary object or run-time input format (see any FORTRAN manual).

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

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

where X(I,2) is the standard deviation (IWT=1) of observation Y(I), or X(I,2) is the variance (IWT=2). Note that if X(I,2) is unknown, one may use the statistical weighting factor 1/Y(I) (Bevington, 1969, p.108) by setting

X(I,2)=Y(I) and IWT=2; this procedure is preferable to using unity weights (IWT=0).

An analytical partial derivative subprogram (PCODE) is used by NLSOL whenever the $PARMS parameter IDER=0 (default) is selected; otherwise if IDER=1, then estimated partial derivatives are computed using only the forward problem subprogram (FCODE). For IDER=0, the complex partial derivatives of Z with respect to each parameter in Table 1 (with L=MODEL, and j=1,2,...,L) can be explicitly written from a general form of equation (2) as

$$
\begin{aligned}
\frac{\partial Z_L}{\partial R_0} &= \frac{Z_L}{R_0}, \quad Z_L(\omega) = R_0[1-m_1 f_1]\cdots[1-m_L f_L] \qquad (1 \le L \le 4) \\
\frac{\partial Z_L}{\partial m_j} &= \frac{-f_j Z_L}{[1-m_j f_j]}, \quad f_j = 1 - \frac{1}{1+(i\omega\tau_j)^{c_j}} \qquad (1 \le j \le L) \\
\frac{\partial Z_L}{\partial \tau_j} &= -m_j \frac{\partial f_j}{\partial \tau_j} \frac{Z_L}{[1-m_j f_j]}, \quad \frac{\partial f_j}{\partial \tau_j} = \frac{(i\omega\tau_j)^{c_j}(c_j/\tau_j)}{[1+(i\omega\tau_j)^{c_j}]^2} \qquad (3) \\
\frac{\partial Z_L}{\partial c_j} &= -m_j \frac{\partial f_j}{\partial c_j} \frac{Z_L}{[1-m_j f_j]}, \quad \frac{\partial f_j}{\partial c_j} = \frac{(i\omega\tau_j)^{c_j}\ln(i\omega\tau_j)}{[1+(i\omega\tau_j)^{c_j}]^2}.
\end{aligned}
$$

See Appendix 3 listing of FCODE and PCODE for the coding details, which follows the methods described above in equations (1)-(3), and as selected by $INIT parameter MODEL and $PARMS parameter IDER.

Because realizable Cole-Cole models are sought to fit the given data, a constrained minimization type (SP=3 or 4) is advised, along with an initial guess array B(J) and reasonable lower and higher bound arrays, BL(J) and BH(J) respectively, where BL(J)≤B(J)≤BH(J), J=1,2,...,K (see Anderson,. 1982, p.17). This approach limits parameter space searching, and in some cases may avoid false starts or catastrophic overflow conditions from poor initial estimates and/or noisy data. In addition, individual parameters can be held fixed in the least-squares program by specifying parameters IP and IB (Anderson, 1982, p.13). For example, this should always be done if a parameter is known in advance, such as B(1); also, is some cases, it is helpful to fix a parameter that cannot be adequately resolved with the given data matrix.

PARAMETERS, FILES AND DATA REQUIRED

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

$INIT PARAMETER DEFINITIONS

<u>$INIT parameters</u>:

MODEL=1 (default) uses a single Cole-Cole model defined by eq. (1) and Table 1. When selecting this $INIT option, it is required that $PARMS parameter K=4 is also explicitly given.

MODEL>1 but <5 uses the combined Cole-Cole models as defined by eq. (2) and Table 1. Note that the corresponding $PARMS K value must be supplied independently and correctly as indicated in Table 1, otherwise unpredictable results would occur.

IOB ... is defined as the "observation type" given in data array Y(I), where Y(I) is a real-valued component of the complex impedance Z(f). The IOB options available are:

IOB=1 (default) defines Y(I) as the amplitude of Z.

IOB=2 defines Y(I) as the phase (in milliradians) of Z.

IOB=3 defines Y(I) as the real-part of Z.

IOB=4 defines Y(I) as the imaginary-part of Z.

IOB=5 means that "mixed observation types" are given in the data matrix column vector Y(I), I=1,...,N, where the corresponding I-th type for Y(I) is specified in X(I,2), which in turn can be equal to 1.0 (for amplitude), 2.0 (for phase), 3.0 (for real-part), or 4.0 (for imaginary-part). See DATA MATRIX OPTIONS and EXAMPLES below for specific cases.

The number of independent variables (M) must be explicitly specified in $PARMS for each IOB option as follows:

Use $PARMS M=1 whenever $INIT IOB<5;
Use $PARMS M=2 whenever $INIT IOB=5.

These are dual NAMELIST input requirements that are not cross-checked by the general purpose NLSOL subprogram, similar to the dual requirements between $PARMS K and $INIT MODEL in Table 1.

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

DATA MATRIX OPTIONS

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

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

where M*=M if IWT=0 (default), or M*=M+1 if IWT=1 or 2. The data matrix is read on logical unit IALT (default 10) using a run-time format statement (see any FORTRAN manual). The number of items read per record depends on $PARMS M,IWT and $INIT IOB parameters as previously defined. The various data matrix options are summarized as follows:

(a) Single observation type as defined by $INIT IOB<5 (requires $PARMS M=1 and $INIT IOB<5; max. 3 items per record):

- Y(I)= I-th observed value, where IOB<5 defines the particular type.
- X(I,1)= I-th observed frequency (Hertz), where X(I,1)>0.0 is required. Normally, this array should be in ascending frequency order, but is not strictly required nor checked. Generally, for plotting the results after obtaining a least-squares solution, ascending order is recommended.
- X(I,2)= weight factor of I-th observation (include only if $PARMS IWT>0).

(b) Mixed observation types, including single or multiple data sets, as defined by $INIT IOB=5 (requires $PARMS M=2 and $INIT IOB=5; max. 4 items per record):

- Y(I)= I-th observed value, where the actual type is defined by X(I,2).
- X(I,1)= I-th observed frequency (Hertz), where X(I,1)>0.0 is required; generally, the data set is ordered with changing X(I,2) within constant X(I,1).
- X(I,2)= Observation type in Y(I); use X(I,2)=1.0 for amplitude, 2.0 for phase (in milliradians), 3.0 for real-part, or 4.0 for imaginary-part.
- X(I,3)= weight factor of I-th observation (include only if $PARMS IWT>0).

For a given data set, the observations need not be ordered by increasing frequency in X(I,1). However, for efficiency in program operation (and subsequent plotting, etc.), it is advised that the data be ordered and sorted by increasing frequency with respect to each observation type (see EXAMPLES in the next section, and in Appendix 2).

EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING

(In this section we assume that the reader is familiar with all the $PARMS definitions as given in Anderson, 1982, p.11-19.)

1. Specific observation type: phase (IOB=2), weighted observations (IWT=1), and alternate input data file (IALT=5) for reading the data matrix along with the input parameters on file FOR005:

```
EXAMPLE 1.
$PARMS N=28,M=1,SP=3,
K=4,IP=1,IB=1, IWT=1, IALT=5,
IDER=0,V(42)=1.E-3,NITER=25,
BL=1,1E-10,.001,1E-10,
BH=1,.9999,1000,.9999,
B=1,.55,10,.75$
(3F10.0)
6.9         .001        .011
9.98        .002        .095
<etc. for 26 more observations>
$INIT MODEL=1,IOB=2 $END
```

Note: Since IWT=1 and M=1, three columns are required in the data matrix row, where in this case, the last column represents the standard deviation of Y(I)=phase of Z[X(I,1)].

2. Mixed observation types (IOB=5), weighted observations (IWT=2) that rereads Y(I) again as the weight factor X(I,3), two Cole-Cole models (MODEL=2), and alternate input data file (IALT=5) for reading the data matrix along with the input parameters on file FOR005:

```
EXAMPLE 2.
$PARMS N=20,M=2,K=7,SP=3,IALT=5,
IDER=1,IWT=2,NITER=20,
BL=.1,3*1E-10, -.9999,2*1E-10,
BH=10,.9999,1E5,.9999, 0,1E5,.9999,
B=1.024,.052,.12,.384,-.885,9.4E-5,.779$
(3G12.5,T1,G12.5)
  1.0189     0.11000      1.0000
  6.9000     0.11000      2.0000
  1.0052     0.33000      1.0000
  8.8000     0.33000      2.0000
<etc. for 18 more observations>
$INIT IOB=5,MODEL=2$
```

Note: Since IWT=2 and M=2, four columns are required in the data matrix row, where in this case, the fourth (implicit) column is reread again as Y(I) and is the statistical weight in Anderson (1982, p. 14-15).

3. Mixed observation types and two joint data sets (IOB=5), unweighted observations (IWT=0), and alternate input data file (IALT=5) for reading the data matrix along with the input parameters on file FOR005:

```
EXAMPLE 3.
$PARMS N=40,M=2,K=4,SP=3,IALT=5,
IWT=0,NITER=20,
BL=.1,3*1E-10,
BH=10,.9999,1E5,.9999,
B=1.024,.52,.12,.5$
(3G12.5)
  1.0189     0.11000      1.0000
  6.9000     0.11000      2.0000
  1.0052     0.33000      1.0000
  8.8000     0.33000      2.0000
<etc. for 16 more observations for first data set>
  2.045       .05         1.0000
  5.2034      .05         2.0000
  1.058       .5          1.0000
  10.715      .5          2.0000
<etc. for 16 more observations for second data set>
$INIT IOB=5,MODEL=1$
```

Note: Since IWT=0 (no weight factor) and M=2, only three columns are required in the data matrix row, where N=40 is the total number of observations from both data sets. A joint inversion is implied using two independent data sets, where amplitude and phase data are given via X(I,2) sorted by increasing frequency in X(I,1); the latter in not required, but convenient for subsequent plotting, etc.

SPECIAL OBJECT FORMAT PHRASES

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

VAX OPERATING INSTRUCTIONS

In general, the basic steps described to run NLSOL (Anderson, 1982, p.22-24) can be followed to run NLSIP in either on-line or batch modes. That is, the parameter and data matrix files may be associated with the logical names FOR005 and FOR010, respectively, using the VAX-DCL statements:

```
$ASSIGN parameterfilename FOR005
$ASSIGN datamatrixfilename FOR010
$RUN NLSIP !use $RUN [WANDERSON]NLSIP on USGS VAX
```

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

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

```
$ASSIGN FOR005.TMP FOR005
$RUN NLSIP !use $RUN [WANDERSON]NLSIP on USGS VAX
```

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

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

ERROR MESSAGES

Almost all $PARMS syntactical errors are flagged and printed on files FOR006 and FOR016 and the job is aborted (see Anderson, 1982, p.24). However, some cross references (or dual inputs) are not checked; for example, the relationships between $PARMS K and $INIT MODEL in Table 1, and $PARMS M and $INIT IOB, respectively, are not double checked by program NLSIP. This is because a general-purpose nonlinear least-squares algorithm (NLSOL) is being used as a control program, but the model input is external to the particular nonlinear problem requirements (NLSIP) read by subprogram SUBZ (see Anderson, 1982, p.38). Therefore, the user is responsible for providing exactly K parameter estimates in B(I),I=1,2,...,K (see Table 1), and that $INIT IOB and $PARMS M are properly set (otherwise, unpredictable results could occur that are unchecked).

PRINTED OUTPUT

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

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

REFERENCES

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

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

Anderson, W.L., and Smith, B.D., 1984, Nonlinear least-squares inversion of transient induced polarization data (Program NLSTIP): USGS Open-File Rept. 84-514, 63 p.

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

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

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

Major, J., and Silic, J., 1981, Restrictions on the use of Cole-Cole dispersion models in complex resistivity interpretation: Geophysics, v. 46, no. 6, p. 916-931.

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

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

Appendix 1.-- Conversion to other systems

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

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

(1) Variables with more than 6-characters.
(2) Character strings delimited by single-quote characters (e.g., 'STRING'); also, character string concatination (e.g., 'STRING1'//'STRING2').
(3) Passing variable-length character strings in subroutine calls; e.g., CHARACTER*(*) passed length character arguments.
(4) Suppression of arithmetic or exponential underflow messages; note that a VAX-11 result is automatically set to 0.0 after any underflow--which is assumed for this program package. If the target system does not set underflows to 0.0, and suppress warning messages, then a suitable conversion procedure must be used for proper operation of this program package.
(5) Replacement of any special VAX-dependent CALLS or statements (e.g., CALL SETTIME, CALL CPUTIME, CALL SYS$GETJPI in module PROCINFO, etc.)
(6) VAX non-ANSI NAMELIST input and output statements.
(7) Replacement of machine-dependent constants in module RMDCON, which is currently set for a VAX-11/780 32-bit machine. See Dennis and others (1979, p. 37-38) for a discussion of constants BIG, ETA, and MACHEP; also see comments in the source code for RMDCON and IMDCON.

Appendix 2.-- Test problem input/output listing

The following input files (FOR005 and FOR010) were used to run a test problem for program NLSIP on a VAX system. (This data was taken from an example given in Washburne, 1982, p. 217-219.) The corresponding output file (FOR016) is listed following FOR010. In addition, file FOR016.DAT was used to plot the final observed (OBS) and calculated (CAL) amplitude and phase curves using an external plotter. The symbol "0" represents Y(I) in the plot, and the solid line represents a curve drawn through the calculated (CAL) points.

FOR005

```
MODEL=2 FIT [T]
$PARMS N=34,M=2,K=7,SP=3,IDER= 0,
IWT=2,NITER=20,
IP= 1,IB= 4,
BL=7*1E-10,
BH=1000, .9999,1000,.9999, .9999,1000,.9999,
B=1.5, .5,1,.5, .5,1E-3,.3$
(3G12.5,T1,G12.5)
$INIT IOB=5,MODEL=2$
```

FOR010

```
  1.9700      0.10000E-02  1.0000
 -14.100      0.10000E-02  2.0000
  1.9500      0.31600E-02  1.0000
 -23.100      0.31600E-02  2.0000
  1.9100      0.10000E-01  1.0000
 -35.700      0.10000E-01  2.0000
  1.8500      0.31600E-01  1.0000
 -50.100      0.31600E-01  2.0000
  1.7700      0.10000      1.0000
 -61.400      0.10000      2.0000
  1.6800      0.31600      1.0000
 -64.100      0.31600      2.0000
  1.6000       1.0000      1.0000
 -59.100       1.0000      2.0000
  1.5400       3.1600      1.0000
 -53.300       3.1600      2.0000
  1.4900       10.000      1.0000
 -53.500       10.000      2.0000
  1.4300       31.600      1.0000
 -61.800       31.600      2.0000
  1.3600       100.00      1.0000
 -75.400       100.00      2.0000
  1.2800       316.00      1.0000
 -86.300       316.00      2.0000
  1.1900       1000.0      1.0000
 -85.100       1000.0      2.0000
  1.1200       3160.0      1.0000
```

```
-70.800       3160.0       2.0000
 1.0700       10000.       1.0000
-51.000       10000.       2.0000
 1.0400       31600.       1.0000
-33.200       31600.       2.0000
 1.0200       0.10000E+06  1.0000
-20.300       0.10000E+06  2.0000
```

FOR016

```
<NLSIP>:     MODEL=2 FIT [T]

MODEL= 2   IOB=5


PARAMETER ORDER & NAMES--

       1      B( 1)=    R0
       2      B( 2)=    m
       3      B( 3)=    Tau
       4      B( 4)=    c
       5      B( 5)=    m2
       6      B( 6)=    Tau2
       7      B( 7)=    c2
```

```
{NLSOL}:          MODEL=2 FIT [T]

N=          34    K=          7     IP=          1     M=           2     IALT=       10
ISTOP=       1    IWT=        2     IDER=        0     IPRT=        0     NITER=      20
IOUT=        1    SP=         3

PARAMETERS HELD FIXED: IB=  4

FMT=(3G12.5,T1,G12.5)


PARAMETER LOWER BOUNDS: BL=

 0.10000000E-09  0.10000000E-09  0.10000000E-09  0.10000000E-09  0.10000000E-09
 0.10000000E-09  0.10000000E-09

INITIAL PARAMETERS: B=

 0.15000000E+01  0.50000000E+00  0.10000000E+01  0.50000000E+00  0.50000000E+00
 0.10000000E-02  0.30000001E+00

PARAMETER HIGHER BOUNDS: BH=

 0.10000000E+04  0.99989998E+00  0.10000000E+04  0.99989998E+00  0.99989998E+00
 0.10000000E+04  0.99989998E+00

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

REORDERED PARAMETERS:

 0.15000000E+01  0.50000000E+00  0.10000000E+01  0.50000000E+00  0.10000000E-02
 0.30000001E+00

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


    I     INITIAL X(I)      D(I)

    1     0.387395E-01     0.147E+03
    2     0.785448E+00     0.116E+03
    3     0.316281E-01     0.656E+03
    4     0.785448E+00     0.857E+02
    5     0.100000E-02     0.839E+04
    6     0.579673E+00     0.655E+02

   IT    NF       F          DF         COSMAX       VAR

    0     1  0.548E+03              0.972E+00
    1     2  0.121E+02  0.536E+03  0.846E+00  0.279E+02
    2     3  0.675E-01  0.120E+02  0.874E+00  0.280E+02
    3     4  0.168E-03  0.673E-01  0.291E+00  0.279E+02
    4     5  0.151E-03  0.172E-04  0.141E-03  0.287E+01
    5     5  0.151E-03  0.172E-04  0.141E-03  0.113E-05

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

FUNCTION      0.150782D-03  VARIABILITY   0.112723E-05
FUNC. EVALS        5        GRAD. EVALS        5
GRAD. NORM    0.436892E-01  COSMAX        0.140971E-03

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

1	0.447353E-01	0.225E+03	-0.167E-03
2	0.521939E+00	0.685E+02	0.114E-04
3	0.346471E-01	0.260E+03	0.192E-04
4	0.616849E+00	0.754E+02	-0.437E-04
5	0.632961E-03	0.179E+05	0.437E-01
6	0.785637E+00	0.370E+02	0.901E-04

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

```
ROW  1    0.2142E-09
ROW  2    0.2899E-10  0.4756E-08
ROW  3    0.3871E-11 -0.5421E-09  0.2571E-09
ROW  4    0.1291E-10 -0.2869E-08  0.5854E-09  0.4546E-08
ROW  5    0.3544E-13 -0.1199E-10  0.2449E-11  0.1173E-10  0.7500E-13

ROW  6   -0.3634E-11  0.5377E-08 -0.9235E-09 -0.5057E-08 -0.2107E-10
          0.1669E-07
```

I	OBS.Y(I)	CAL	RES	%RES.ERR	X(I,1)	X(I,2)	X(I,3)	X(I,4)	WT(I)
1	0.197000E+01	0.196905E+01	0.950E-03	0.402698E-01	0.100000E-02	0.100000E+01	0.197000E+01	0.000000E+00	0.507614E+00
2	-0.141000E+02	-0.140811E+02	-0.189E-01	-0.134371E+00	0.100000E-02	0.200000E+01	-0.141000E+02	0.000000E+00	0.709220E-01
3	0.195000E+01	0.194590E+01	0.410E-02	0.210649E+00	0.316000E-02	0.100000E+01	0.195000E+01	0.000000E+00	0.512820E+00
4	-0.231000E+02	-0.230960E+02	-0.402E-02	-0.174251E-01	0.316000E-02	0.200000E+01	-0.231000E+02	0.000000E+00	0.432900E-01
5	0.191000E+01	0.190726E+01	0.274E-02	0.143726E+00	0.100000E-01	0.100000E+01	0.191000E+01	0.000000E+00	0.523560E+00
6	-0.357000E+02	-0.357138E+02	0.138E-01	0.387303E-01	0.100000E-01	0.200000E+01	-0.357000E+02	0.000000E+00	0.280112E-01
7	0.185000E+01	0.184780E+01	0.220E-02	0.119190E+00	0.316000E-01	0.100000E+01	0.185000E+01	0.000000E+00	0.540541E+00
8	-0.501000E+02	-0.501443E+02	0.443E-01	0.883072E-01	0.316000E-01	0.200000E+01	-0.501000E+02	0.000000E+00	0.199601E-01
9	0.177000E+01	0.176821E+01	0.179E-02	0.101107E+00	0.100000E+00	0.100000E+01	0.177000E+01	0.000000E+00	0.564972E+00
10	-0.614000E+02	-0.613760E+02	-0.240E-01	-0.391564E-01	0.100000E+00	0.200000E+01	-0.614000E+02	0.000000E+00	0.162066E-01
11	0.168000E+01	0.168123E+01	-0.123E-02	-0.731466E-01	0.316000E+00	0.100000E+01	0.168000E+01	0.000000E+00	0.595238E+00
12	-0.641000E+02	-0.640767E+02	-0.233E-01	-0.363153E-01	0.316000E+00	0.200000E+01	-0.641000E+02	0.000000E+00	0.156006E-01
13	0.160000E+01	0.160333E+01	-0.333E-02	-0.207848E+00	0.100000E+01	0.100000E+01	0.160000E+01	0.000000E+00	0.625000E+00
14	-0.591000E+02	-0.590835E+02	-0.165E-01	-0.280081E-01	0.100000E+01	0.200000E+01	-0.591000E+02	0.000000E+00	0.169205E-01
15	0.154000E+01	0.154087E+01	-0.871E-03	-0.565305E-01	0.316000E+01	0.100000E+01	0.154000E+01	0.000000E+00	0.649351E+00
16	-0.533000E+02	-0.533631E+02	0.631E-01	0.118244E+00	0.316000E+01	0.200000E+01	-0.533000E+02	0.000000E+00	0.187617E-01
17	0.149000E+01	0.148727E+01	0.273E-02	0.183390E+00	0.100000E+02	0.100000E+01	0.149000E+01	0.000000E+00	0.671141E+00
18	-0.535000E+02	-0.534737E+02	-0.263E-01	-0.491446E-01	0.100000E+02	0.200000E+01	-0.535000E+02	0.000000E+00	0.186916E-01
19	0.143000E+01	0.143107E+01	-0.107E-02	-0.746293E-01	0.316000E+02	0.100000E+01	0.143000E+01	0.000000E+00	0.699301E+00
20	-0.618000E+02	-0.617621E+02	-0.379E-01	-0.614309E-01	0.316000E+02	0.200000E+01	-0.618000E+02	0.000000E+00	0.161812E-01
21	0.136000E+01	0.136200E+01	-0.200E-02	-0.147051E+00	0.100000E+03	0.100000E+01	0.136000E+01	0.000000E+00	0.735294E+00
22	-0.754000E+02	-0.754467E+02	0.467E-01	0.619378E-01	0.100000E+03	0.200000E+01	-0.754000E+02	0.000000E+00	0.132626E-01
23	0.128000E+01	0.127862E+01	0.138E-02	0.108206E+00	0.316000E+03	0.100000E+01	0.128000E+01	0.000000E+00	0.781250E+00
24	-0.863000E+02	-0.862843E+02	-0.157E-01	-0.181441E-01	0.316000E+03	0.200000E+01	-0.863000E+02	0.000000E+00	0.115875E-01
25	0.119000E+01	0.119244E+01	-0.244E-02	-0.204341E+00	0.100000E+04	0.100000E+01	0.119000E+01	0.000000E+00	0.840336E+00
26	-0.851000E+02	-0.851073E+02	0.726E-02	0.853415E-02	0.100000E+04	0.200000E+01	-0.851000E+02	0.000000E+00	0.117509E-01
27	0.112000E+01	0.112067E+01	-0.675E-03	-0.602176E-01	0.316000E+04	0.100000E+01	0.112000E+01	0.000000E+00	0.892857E+00
28	-0.708000E+02	-0.708077E+02	0.771E-02	0.108826E-01	0.316000E+04	0.200000E+01	-0.708000E+02	0.000000E+00	0.141243E-01
29	0.107000E+01	0.107096E+01	-0.958E-03	-0.894383E-01	0.100000E+05	0.100000E+01	0.107000E+01	0.000000E+00	0.934579E+00
30	-0.510000E+02	-0.509765E+02	-0.235E-01	-0.461118E-01	0.100000E+05	0.200000E+01	-0.510000E+02	0.000000E+00	0.196078E-01
31	0.104000E+01	0.104047E+01	-0.468E-03	-0.450156E-01	0.316000E+05	0.100000E+01	0.104000E+01	0.000000E+00	0.961538E+00
32	-0.332000E+02	-0.332041E+02	0.412E-02	0.124077E-01	0.316000E+05	0.200000E+01	-0.332000E+02	0.000000E+00	0.301205E-01
33	0.102000E+01	0.102280E+01	-0.280E-02	-0.273385E+00	0.100000E+06	0.100000E+01	0.102000E+01	0.000000E+00	0.980392E+00
34	-0.203000E+02	-0.203076E+02	0.760E-02	0.374283E-01	0.100000E+06	0.200000E+01	-0.203000E+02	0.000000E+00	0.492611E-01

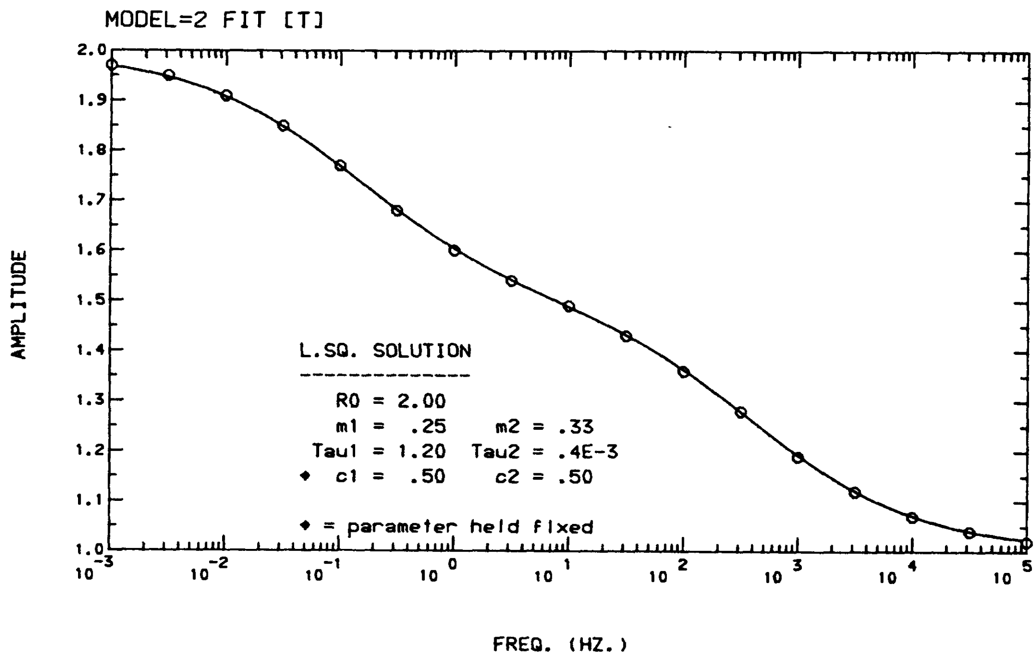
** RMSERR= 0.21774426E-01 AVE|%RES.ERR|= 0.868447E-01

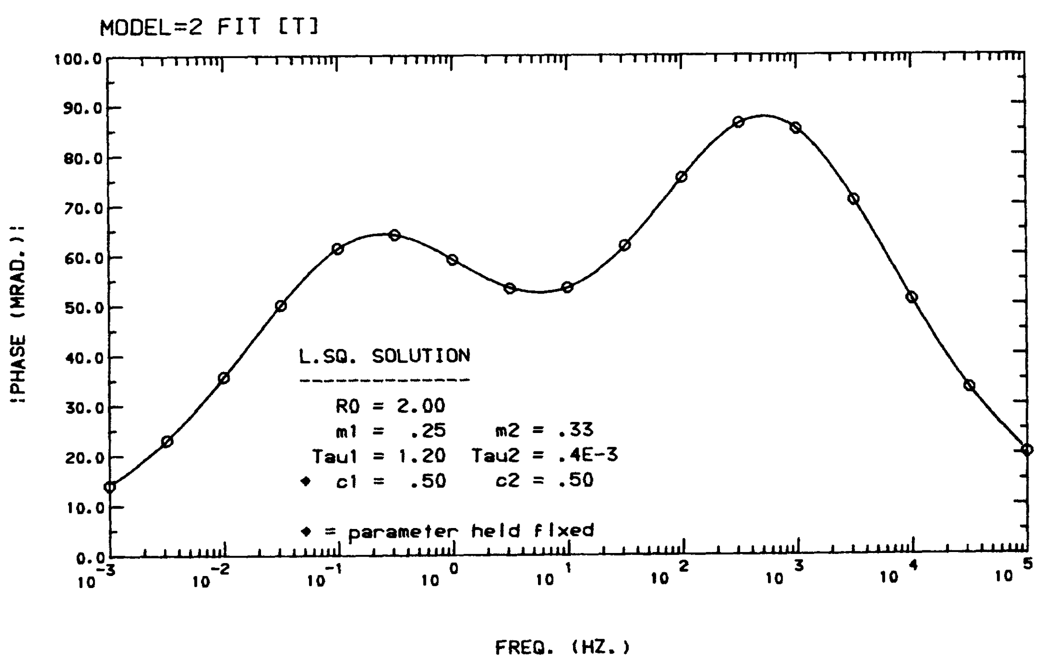
CORRELATION MATRIX

```
 1  0.1000E+01
 2  0.2873E-01  0.1000E+01
 3  0.1649E-01 -0.4902E+00  0.1000E+01
 5  0.1308E-01 -0.6170E+00  0.5415E+00  0.1000E+01
```

```
 6  0.8843E-02 -0.6350E+00  0.5576E+00  0.6353E+00  0.1000E+01
 7 -0.1922E-02  0.6035E+00 -0.4457E+00 -0.6723E+00 -0.5954E+00  0.1000E+01

  **PARM_SOL.    STD_ERROR    REL_ERROR    % ERROR **

 1  0.2000E+01  0.1464E-04  0.3272E-03  0.3272E-01
 2  0.2485E+00  0.6896E-04  0.1321E-03  0.1321E-01
 3  0.1200E+01  0.1604E-04  0.4628E-03  0.4628E-01
 5  0.3346E+00  0.6742E-04  0.1093E-03  0.1093E-01
 6  0.4006E-03  0.2739E-06  0.4327E-03  0.4327E-01
 7  0.5002E+00  0.1292E-03  0.1645E-03  0.1645E-01


********  E N D  ********     MODEL=2 FIT [T]

PARAMETER NAME   FINAL SOLUTION

B( 1)      R0   =  0.19999168E+01
B( 2)      m    =  0.24853931E+00
B( 3)      Tau  =  0.11999412E+01
B( 4)   *  c    =  0.50000000E+00
B( 5)      m2   =  0.33459151E+00
B( 6)      Tau2=   0.40063914E-03
B( 7)      c2   =  0.50018924E+00

       * FIXED
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TOTAL "ELAPSED" TIME=           20.61 SEC. (    0 MIN. 20.61 SEC.)
CPU_TIME=           2.72 SEC. (    0 M.   2.72 S.)    CPU % = 13.20%
BUF.I/O_COUNT=        86
DIR.I/O_COUNT=        35
PAGE_FAULTS=         253
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```


MODEL=2 FIT [T]
AMPLITUDE
2.0
1.9
1.8
1.7
1.6
1.5
1.4
1.3
1.2
1.1
1.0
L.SQ. SOLUTION
R0 = 2.00
m1 = .25 m2 = .33
Tau1 = 1.20 Tau2 = .4E-3
◆ c1 = .50 c2 = .50
◆ = parameter held fixed
10^-3 10^-2 10^-1 10^0 10^1 10^2 10^3 10^4 10^5
FREQ. (HZ.)


MODEL=2 FIT [T]
|PHASE (MRAD.)|
100.0
90.0
80.0
70.0
60.0
50.0
40.0
30.0
20.0
10.0
0.0
L.SQ. SOLUTION
R0 = 2.00
m1 = .25 m2 = .33
Tau1 = 1.20 Tau2 = .4E-3
◆ c1 = .50 c2 = .50
◆ = parameter held fixed
10^-3 10^-2 10^-1 10^0 10^1 10^2 10^3 10^4 10^5
FREQ. (HZ.)

Appendix 3.-- Source code availability and listing

Source Code Availability

The current version of the source code may be obtained by writing directly to the first author*, and enclosing a magnetic tape to be copied and returned. This method of releasing the source code was selected in order to satisfy requests for the latest (e.g., possibly updated) version. The attached listing does not include the adaptive nonlinear least-squares algorithm (Anderson, 1982; Dennis and others, 1979) due to its length; however, the complete algorithm is available on the distributed tape.

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

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

```
C <NLSIP>: INVERSION OF FREQUENCY IP SOUNDINGS.
C
C** VAX-11/780 VERSION:  4/18/86  **
C
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.
C
C--------------------------------------------------------------
C
C--REFERENCES:
C
C  ANDERSON, W.L., 1982, ADAPTIVE NONLINEAR LEAST-SQUARES SOLUTION
C    FOR CONSTRAINED OR UNCONSTRAINED MINIMIZATION PROBLEMS
C    (SUBPROGRAM NLSOL): USGS OPEN-FILE REPT. 82-68, 65 P.
C
C  -------, 1984, A GENERAL INTERFACE FOR PRODUCING FORWARD
C    SOLUTION PROGRAMS (SUBPROGRAM FWDSOL): USGS OPEN-FILE
C    REPT. 84-348, 43 P.
C
C  ANDERSON, W.L., AND SMITH, B.D., 1986, NONLINEAR LEAST-SQUARES
C    INVERSION OF FREQUENCY-DOMAIN INDUCED POLARIZATION DATA
C    (PROGRAM NLSIP): USGS OPEN-FILE REPT. [THIS REPORT]
C
C--------------------------------------------------------------
C
      EXTERNAL FCODE,PCODE,SUBZ,TIP_SUBEND
      CALL SETTIME
      CALL NLSOL2(FCODE,PCODE,SUBZ,TIP_SUBEND)
      CALL CPUTIME(6,16)
      CALL EXIT
      END
      SUBROUTINE FCODE(Y,X,B,PRNT,F,IN,IDER)
C--FUNCTION EVALUATION FOR 'FWDIP' OR 'NLSIP'
C
C--PARAMETERS--
C
C      Y=      OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)
C      X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)
C      B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)
C      PRNT=   WORK AND PRINT ARRAY (DIM. 5)
C      F=      OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. IN
C      IN=     OBSERVATION NO. TO EVAL. F (1<=IN<=N)
C      IDER=   0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED)
C              1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED)
C
      SAVE
      COMPLEX Z,Z1,ZF,ONE, FL,FM,WTC
```

```
      REAL Y(1),X(500,5),B(1),PRNT(5)
      COMMON/COLEIP/MODEL,IOB
      COMMON/PASS/FL(4),FM(4),WTC(4),ZF,AMP,OMEGA
      DATA ONE/(1.0,0.0)/,TWOPI/6.283185308/
      DATA PI2/1.570796327/
      DO 10 J=1,5
10    PRNT(J)=X(IN,J)
      FREQ=PRNT(1)
      OMEGA=TWOPI*FREQ
      IF(IN.EQ.1.OR.FREQ.NE.FREQL) GO TO 20
      IF(IOB.EQ.5) GO TO 40
20    Z=(OMEGA*B(3))**B(4)
      Z1=Z*(CEXP(CMPLX(0.0,B(4)*PI2)))
      WTC(1)=Z1
      FL(1)=ONE-ONE/(ONE+Z1)
      ZF=ONE-CMPLX(B(2),0.0)*FL(1)
      FM(1)=ZF
      IF(MODEL.EQ.1) GO TO 30
      DO IMODEL=2,MODEL
        I1=3*IMODEL-1
        I2=I1+1
        I3=I1+2
        Z=(OMEGA*B(I2))**B(I3)
        Z1=Z*(CEXP(CMPLX(0.0,B(I3)*PI2)))
          WTC(IMODEL)=Z1
          FL(IMODEL)=ONE-ONE/(ONE+Z1)
        Z=ONE-CMPLX(B(I1),0.0)*FL(IMODEL)
          FM(IMODEL)=Z
        ZF=ZF*Z
      ENDDO
30    ZF=CMPLX(B(1),0.0)*ZF
40    GO TO (41,42,43,44,45),IOB
41    F=CABS(ZF)
      AMP=F
      GO TO 50
C--GET PHZ IN MILLIRADIANS
42    CALL POLAR2(ZF,AMP,DEG)
      F=17.45329252*DEG
      GO TO 50
43    F=REAL(ZF)
      GO TO 50
44    F=AIMAG(ZF)
      GO TO 50
45    IOBS=PRNT(2)
      GO TO (41,42,43,44),IOBS
50    FREQL=FREQ
      RETURN
      END
      SUBROUTINE PCODE(P,X,B,PRNT,F,IN,IP,IB)
C--ANALYTIC PARTIALS W/R PARAMETERS IN B(K) AND IN COMMON
C  (PCODE CALLED ONLY IF $PARMS IDER=0--DEFAULT.)
C
C--PARAMETERS
C
```

```
C  P=   OUTPUT REAL ARRAY (DIM.K) OF PARTIAL DERIVATIVES W/R B()
C  X,B,PRNT,F,IN-- SAME AS IN FCODE DEFINITIONS
C  IP=  NO. PARAMETERS HELD FIXED IN INPUT ARRAY IB(I),I=1,IP
C  IB=  ARRAY OF INDICES OF PARAMETERS IN B() HELD FIXED.
C
      SAVE
      DIMENSION P(1),X(500,5),B(1),PRNT(5),IB(1)
      COMPLEX FL,FM,WTC,ZF,ONE,PZ(13),PARTF
      LOGICAL SKIP
      COMMON/COLEIP/MODEL,IOB
      COMMON/PASS/FL(4),FM(4),WTC(4),ZF,AMP,OMEGA
      DATA ONE/(1.0,0.0)/
      IF(PRNT(1).EQ.FREQL.AND.IN.GT.1.AND.IOB.EQ.5) THEN
        SKIP=.TRUE.
      ELSE
        SKIP=.FALSE.
      ENDIF
      DO 200 IMODEL=1,MODEL
        I1=3*IMODEL-1
        I2=I1+1
        I3=I1+2
        IF(IMODEL.EQ.1) THEN
          J1=1
        ELSE
          J1=I1
        ENDIF
        DO 100 J=J1,I3
          IF(IP.GT.0) THEN
            DO I=1,IP
              IF(IB(I).EQ.J) THEN
                P(J)=0.0
                GO TO 100
              ENDIF
            ENDDO
          ENDIF
          IF(SKIP) GO TO 5
          IF(J.EQ.1) THEN             ! DER W/R RO = B(1)
            PZ(1)=ZF/B(1)
          ELSE IF(J.EQ.I1) THEN       ! DER W/R M IN B(I1)
            PZ(J)=-FL(IMODEL)*ZF/FM(IMODEL)
          ELSE IF(J.EQ.I2) THEN       ! DER W/R TAU IN B(I2)
            PARTF=(WTC(IMODEL)*B(I3)/B(I2))/
     1      (ONE+WTC(IMODEL))**2
            PZ(J)=-B(I1)*PARTF*ZF/FM(IMODEL)
          ELSE IF(J.EQ.I3) THEN       ! DER W/R C IN B(I3)
            PARTF=(WTC(IMODEL)*CLOG(CMPLX(0.0,OMEGA*B(I2))))/
     1      (ONE+WTC(IMODEL))**2
            PZ(J)=-B(I1)*PARTF*ZF/FM(IMODEL)
          ENDIF
5         GO TO (10,20,30,40,50),IOB
10        P(J)=(REAL(ZF)*REAL(PZ(J))+AIMAG(ZF)*AIMAG(PZ(J)))/AMP
          GO TO 100
20        P(J)=1000.*((REAL(ZF)*AIMAG(PZ(J))-
     1    AIMAG(ZF)*REAL(PZ(J)))/(AMP*AMP))
```

```
          GO TO 100
30        P(J)=REAL(PZ(J))
          GO TO 100
40        P(J)=AIMAG(PZ(J))
          GO TO 100
50        IOBS=PRNT(2)
          GO TO (10,20,30,40),IOBS
100     CONTINUE
200   CONTINUE
      FREQL=PRNT(1)
      RETURN
      END
      SUBROUTINE SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)
C-- INITIALIZATION ROUTINE (CALLED ONCE)
C  SUBZ IS CALLED AFTER THE DATA Y(I),X(I,5) ARE READ.
C  SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL $INIT
C  PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE...
C
C--PARAMETERS--
C
C       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.
C       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY
C               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES
C               PRINTED BY PGM MARQRT...
C       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)
C       TITLE=  ALPHA TITLE ARRAY READ IN BY PGM MARQRT.
C       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED
C               0 IF ONLY UNIT 6 PRINT FILE USED.
C
      CHARACTER*80 TITLE
      CHARACTER*4 PARNAM(13)
      REAL   Y(1),X(500,5),B(1),PRNT(1)
      COMMON/COLEIP/MODEL,IOB
      NAMELIST/INIT/MODEL,IOB
      DATA ISUBZ/0/,PARNAM/'RO','M','TAU','C','M2','TAU2','C2',
     1 'M3','TAU3','C3','M4','TAU4','C4'/
      IF(ISUBZ.NE.0) GO TO 10
C--PRESET
      ISUBZ=1
      MODEL=1
      IOB=1
   10 READ(99,INIT)
      CALL NONBLANK(TITLE,NONBLK)
      WRITE(6,20) TITLE
   20 FORMAT('1<NLSIP>:',5X,A<NONBLK>/)
      IF(IOUT.EQ.1) WRITE(16,20) TITLE
      WRITE(6,30) MODEL,IOB
      IF(IOUT.EQ.1)
     1 WRITE(16,30) MODEL,IOB
   30 FORMAT(7H MODEL=, I2,2X,5H IOB=,I1)
C--TEST $INIT PARMS
      IF(MODEL.LT.1.OR.MODEL.GT.4)
     &CALL ERRMSG('MODEL<1 OR >4',0,6,16)
      IF(IOB.LT.1.OR.IOB.GT.5)CALL ERRMSG('IOB<1 OR >5',0,6,16)
```

```
      MPARM=3*MODEL+1
      DO I=1,MPARM
        IF(B(I).EQ.0.0)CALL ERRMSG(
     1 'SOME B(I)=0 FOR I=1,3*MODEL+1',0,6,16)
      ENDDO
C--TEST X(I, ) DATA FOR GIVEN IOB BEFORE PROCEEDING--
   40 DO 70 I=1,N
      IF(X(I,1).LE.0.0) CALL ERRMSG(
     1 'SOME FREQ=X(I,1)<=0',0,6,16)
      IF(IOB.LT.5) GO TO 70
      IF(IFIX(X(I,2)).LT.1.OR.IFIX(X(I,2)).GT.4) CALL ERRMSG(
     &   'SOME X(I,2) OUT OF RANGE WHEN IOB=5',0,6,16)
   70 CONTINUE
C--PRINT PARAMETER NAMES AND ORDER
      WRITE(6,50)
50    FORMAT(////' PARAMETER ORDER & NAMES--'/)
      IF(IOUT.EQ.1) WRITE(16,50)
      DO I=1,MPARM
        WRITE(6,60) I,I,PARNAM(I)
60      FORMAT(5X,I3,5X,'B(',I2,')=',4X,A4)
        IF(IOUT.EQ.1) WRITE(16,60) I,I,PARNAM(I)
      ENDDO
      RETURN
      END
      SUBROUTINE TIP_SUBEND(Y,X,B,K,N,TITLE,IOUT)
C** SUBEND TERMINATION ROUTINE
C  ALSO GIVES RESTART $PARMS ON UNIT=4 AS 'FOR005.TMP'
C
      CHARACTER*132 LINE
      CHARACTER*80 TITLE
      CHARACTER*4 PARNAM(13)
      DATA PARNAM/'R0','M','TAU','C','M2','TAU2','C2',
     1 'M3','TAU3','C3','M4','TAU4','C4'/
      CHARACTER*1 FLAG(19)
      COMMON/FIXDAT/DUM(3020),IB(19),IP,IDUM(3)
      REAL Y(1),X(500,5),B(1)
      DO I=1,K
        FLAG(I)=' '
        IF(IP.GT.0) THEN
          DO J=1,IP
            IF(IB(J).NE.0) FLAG(IB(J))='*'
          ENDDO
        ENDIF
      ENDDO
      CALL NONBLANK(TITLE,NB)
      WRITE(6,10) TITLE
10    FORMAT(//' ********  E N D  ********',5X,A<NB>//
     1 ' PARAMETER NAME',3X,'FINAL SOLUTION'/)
      IF(IOUT.EQ.1) WRITE(16,10) TITLE
      DO 30 I=1,K
      WRITE(6,20) I,FLAG(I),PARNAM(I),B(I)
20    FORMAT(1X,'B(',I2,')',2X,A,2X,A,'=',E16.8)
      IF(IOUT.EQ.1) WRITE(16,20) I,FLAG(I),PARNAM(I),B(I)
30    CONTINUE
```

```
C** GENERATE RESTART $PARMS ON FOR005.TMP
60    REWIND 5
      OPEN(UNIT=4,FILE='FOR005.TMP',STATUS='NEW',
     1 CARRIAGECONTROL='LIST')
      READ(5,65,END=999) LINE
65    FORMAT(A)
      CALL NONBLANK(LINE,NB)
      WRITE(4,66) LINE
66    FORMAT(A<NB>)
      IDOL=0
70    READ(5,65,END=999) LINE
      I=INDEX(LINE,'$')
      IF(I.NE.0) THEN
        IF(IDOL.EQ.0) THEN
          IDOL=1
          J=INDEX(LINE(I+1:),'$')
          IF(J.NE.0) THEN
            IDOL=2
            LINE(J:J)=','
          ENDIF
        ELSE
          IDOL=2
          LINE(I:I)=','
        ENDIF
      ENDIF
      CALL NONBLANK(LINE,NB)
      WRITE(4,66) LINE
      IF(IDOL.LT.2) GO TO 70
      LINE(1:)='B='
      DO 80 I=1,K
      ENCODE(16,90,LINE(3:18)) B(I)
90    FORMAT(G16.8)
      IF(I.LT.K) THEN
        LINE(19:19)=','
      ELSE
        LINE(19:19)='$'
      ENDIF
      CALL NONBLANK(LINE,NB)
      WRITE(4,66) LINE
      LINE(1:2)='  '
80    CONTINUE
100   READ(5,65,END=999) LINE
      CALL NONBLANK(LINE,NB)
      WRITE(4,66) LINE
      GO TO 100
999   IF(IP.GT.0) THEN
        WRITE(6,110)
110     FORMAT(/8X,'* FIXED')
        IF(IOUT.EQ.1) WRITE(16,110)
      ENDIF
      RETURN
      END
      SUBROUTINE CPUTIME(I1,I2)
C
```

```
C  CPUTIME WRITES "ELAPSED & CPU" TIME FROM PREVIOUS "CALL SETTIME" ON
C  FORTRAN UNITS I1 (IF NOT 0) AND I2 (IF NOT 0).
C
C  WILL EJECT FIRST IF I1>0 (OR I2>0).
C  DOUBLE SPACE FIRST IF I1<0 (OR I2<0).
C
C  E.G., USE TO TIME ELAPSED & CPU TIME FOR PROGRAM OR CODE SEGMENTS AS:
C
C     CALL SETTIME    ! DON'T FORGET TO DO THIS!
C     >>>>> THE CODE TO TIME IS HERE <<<<<  ! USUALLY A COMPLETE PROGRAM
C     CALL CPUTIME(-6,16) ! OR USE I1 OR I2=0 TO OMIT WRITE.
C     >>>>> ALSO CAN USE CALL GETTIME(CPU) TO GET JUST THE CPU (SEC)
C           SINCE THE LAST CALL SETTIME WAS DONE.
C
      SAVE
      INTEGER*4 ABSVAL(4),INCRVAL(4)
      CALL PROCINFO(ABSVAL,INCRVAL)
      TIMES=SECNDS(TIME0)
      MIN=TIMES/60.0
      SEC=AMOD(TIMES,60.0)
      CPUSEC=INCRVAL(1)*.01
      IMIN=CPUSEC/60.0
      CSEC=AMOD(CPUSEC,60.0)
      PCPU=100.*(CPUSEC/TIMES)
      IF(I1.NE.0) THEN
        IF(I1.GT.0) THEN
          J=1
        ELSE
          J=0
        ENDIF
      WRITE(IABS(I1),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,
     1 (INCRVAL(I),I=2,4)
60    FORMAT(I1,65('$')/' TOTAL "ELAPSED" TIME=',F16.2,' SEC. (',
     1 I4,' MIN.',F6.2,' SEC.)'/
     2 ' CPU_TIME=',F15.2,' SEC. (',I4,' M. ',F5.2,
     1 ' S.)   CPU % =',F6.2,'%'/
     3 ' BUF.I/O_COUNT=',I10/
     4 ' DIR.I/O_COUNT=',I10/
     5 ' PAGE_FAULTS=',2X,I10/
     6 ' ',65('$')//)
      ENDIF
      IF(I2.NE.0) THEN
        IF(I2.GT.0) THEN
          J=1
        ELSE
          J=0
        ENDIF
      WRITE(IABS(I2),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,
     1 (INCRVAL(I),I=2,4)
      ENDIF
      RETURN
C** ENTRY 'CALL SETTIME'--MUST BE DONE BEFORE 'CALL CPUTIME(I1,I2)'
      ENTRY SETTIME()
      TIME0=SECNDS(0.0)
```

```
      CALL PROCINFO(ABSVAL,INCRVAL)
      RETURN
C** ENTRY 'CALL GETTIME(CPU)'--TO GET CPU(SEC) SINCE LAST CALL SETTIME
      ENTRY GETTIME(CPU)
      CALL PROCINFO(ABSVAL,INCRVAL)
      CPU=INCRVAL(1)*.01
      RETURN
      END
      SUBROUTINE ERRMSG(MSG,ISKIP,IUNIT1,IUNIT2)
C
C  GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780
C
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2
C          > 0 FOR ONE BLANK LINE BEFORE.
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).
C
C  MESSAGES ARE WRITTEN IN THE FORM:
C
C  {ERRMSG}: _MSG_HERE_
C
      CHARACTER*(*) MSG
      I=LEN(MSG)
      DO 1 J=1,2
        IF(J.EQ.1) THEN
       JUNIT=IUNIT1
          ELSE
       JUNIT=IUNIT2
        ENDIF
        IF(JUNIT.GT.0) THEN
          IF(ISKIP.EQ.0) THEN
            WRITE(JUNIT,2) MSG
            ELSE
       WRITE(JUNIT,3) MSG
          ENDIF
      ENDIF
1     CONTINUE
      CALL EXIT
2     FORMAT(1X,'{ERRMSG}: ',A<I>)
3     FORMAT(/1X,'{ERRMSG}: ',A<I>)
      END
      SUBROUTINE NLSOL2(FCODE,PCODE,SUBZ,SUBEND)
C
C>> NLSOL2 IS A REVISED VERSION OF NLSOL WITHOUT CALL NAMELIST;
C   I.E., NLSOL2 USES THE CURRENT VAX-11/780 NAMELIST VERSION.  NOTE
C   SUBZ MUST BE CHANGED TO READ(99,INIT) FROM CALL NAMELIST(5,'$INIT'),
C   WHERE CALL PRENAM(5,99) IS USED TO ENSURE UNIT=5 NAMELIST IS IN
C   PROPER FORMAT ON SCRATCH UNIT=99 (FOR099 DELETED ON RETURN TO VMS).
C
C {NLSOL2}: GENERAL NONLINEAR LEAST-SQUARES SOLUTION   {2/19/86}
C          USING DENNIS ET AL (1979; SEE REF1 BELOW)    -------
C          ADAPTIVE NONLINEAR LEAST-SQUARES ALGORITHM.
C
```

```
C** THIS IS AN INTERFACE ROUTINE WRITTEN FOR THE VAX-11/780 BY
C   W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.
C
C** THIS INTERFACE (NLSOL) HAS ADDITIONAL OPTIONS (BESIDE REF1) TO:
C   (1) PERFORM EITHER UNCONSTRAINED OR UP TO 4-TYPES OF CONSTRAINED
C       ADAPTIVE NONLINEAR REGRESSION FOR ARBITRARY NONLINEAR PROBLEMS.
C       (I.E., PARTIAL OR FULL LOWER/HIGHER PARAMETER BOUNDS, ETC.)
C   (2) HOLDING CERTAIN PARAMETERS FIXED (I.E., AS CONSTANTS) IN THE
C       LEAST-SQUARES (THIS IS ANOTHER FORM OF CONSTRAINING SOLUTION
C       SPACE).
C   (3) PROVIDE FOR WEIGHTED OBSERVATIONS (I.E., WEIGHTED LEAST-SQUARES)
C   (4) OBJECT (RUN)-TIME CONTROL OF READING THE DATA MATRIX, PLUS
C       MANY OTHER I/O OPTIONS, ETC.
C   (5) OPTIONALLY, ONE CAN USE EITHER ESTIMATED PARTIAL DERIVATIVES, OR
C       ANALYTICAL PARTIAL DERIVATIVES (IF SUBROUTINE PCODE AVAILABLE).
C
C** THE USER ONLY NEEDS TO WRITE SUBROUTINES FCODE, PCODE, SUBZ, AND
C   SUBEND (SEE DETAILS BELOW) EXACTLY AS USED IN SUBROUTINE 'MARQRT'
C   (SEE REF2) OR 'IMSLMQ' (SEE REF3).  ALSO, THE SAME PARAMETER FILE
C   FOR005 AND OBJECT (RUN)-TIME DATA MATRIX FILE FOR010 AS USED BY
C   EITHER MARQRT OR IMSLMQ MAY BE USED IN 'NLSOL'.
C
C** NLSOL CALLS NLITR WHICH CALLS 'NL2ITR' AS PUBLISHED BY DENNIS ET AL,
C   (SEE REF1, P. 38), OR 'NL2SNO' (SEE REF1, P. 35).
C
C** REF1:  DENNIS, J.E., ET AL, 1979, AN ADAPTIVE NONLINEAR LEAST-
C          SQUARES ALGORITHM, NTIS REPORT AD-A079-716.
C
C   REF2:  ANDERSON, W.L., 1980, PROGRAM MARQHXY: INVERSION OF HX AND HY
C          FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN-
C          FILE REPT. 80-901.
C
C   REF3:  ANDERSON, W.L., 1980, PROGRAM IMSLEXY: INVERSION OF EX AND EY
C          FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN-
C          FILE REPT. 80-1073.
C
C**********************************************************************
C
C****  THE USER MUST DECLARE THE CALLING PARAMETERS AS EXTERNAL IN THE
C      CALLING PROGRAM (ANY DESIRED NAMES MAY BE USED).
C E.G.,
C
C [MAIN]:
C     EXTERNAL MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND
C     CALL NLSOL2(MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND)
C     STOP   !<OR USE>: CALL EXIT
C     END
C [FCODE]:
C     SUBROUTINE MY_FCODE(Y,X,B,W,F,IN,IDER)
C     USER WRITTEN TO EVALUATE THE NONLINEAR OBJECTIVE FUNCTION (F)
C     USED IN NLSOL AS THE WEIGHTED SUM OF (Y(IN)-F)**2, WHERE
C     Y= OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N, WHERE N IS
C      GIVEN IN $PARMS NAMELIST INPUT--SEE BELOW).
C     X= OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,M, WHERE
```

```
C        M IS IN $PARMS INPUT).
C       B= CURRENT PARAMETER ESTIMATES (DIM. K, WHERE
C        K IS IN $PARMS INPUT).
C       W= WORK ARRAY (DIM. 5)--MAY BE USED TO PASS DATA TO PCODE.
C       F= (OUTPUT) THE FUNCTION VALUE EVALUATED FOR THE GIVEN
C        Y,X, AND B ARRAYS AT THE OBSERVATION NO. 'IN'.
C       IN= (INPUT) OBSERVATION NO. TO EVALUATE F (1.LE.IN.LE.N),
C        WHICH IS CONTROLLED EXTERNALLY BY 'NLSOL'. USUALLY,
C        IN=1,2,...,N--BUT NOT ALWAYS.
C       IDER= 0 IF ANALYTICAL DERIVATIVES ARE USED (PCODE CALLED
C          AFTER FCODE).
C           = 1 IF ESTIMATED DERIVATIVES ARE USED (PCODE NOT CALLED
C          AFTER FCODE).
C       DIMENSION Y(1),X(500,5),B(1),W(5)
C>>>>> INSERT USER CODE HERE TO EVALUATE F  <<<<<
C       END
C [PCODE]: >> PCODE MAY BE A DUMMY NAME IF ONLY IDER=1 IS TO BE USED. <<
C       SUBROUTINE MY_PCODE(P,X,B,W,F,IN,IP,IB)
C       USER WRITTEN TO EVALUATE THE ANALYTICAL PARTIAL DERIVATIVES OF
C       F WITH RESPECT TO B(J),J=1,2,...,K, AT OBSERVATION 'IN', WHERE
C       P= (OUTPUT) PARTIAL DERIVATIVE ARRAY (DIM. K, WHERE
C        K IS IN $PARMS INPUT).
C       X,B,W ARE THE SAME AS USED IN FCODE (SEE ABOVE).
C       F= LAST FUNCTION VALUE FROM FCODE AT OBSERVATION IN.
C        (NOTE THAT F MAY NOT BE NEEDED, BUT IS AVAILABLE ANYWAY)
C       IN= (INPUT) OBSERVATION NO. TO EVALUATE P ARRAY, WHICH IS
C        CONTROLLED EXTERNALLY BY 'NLSOL' (1.LE.IN.LE.N).
C       IP= (INPUT) THE NO. OF B-PARAMETERS HELD FIXED IN THE LEAST-
C        SQUARES (0.LE.IP.LE.K-1; USE IP=0 IF NONE).
C       IB= ARRAY OF B-PARAMETER INDICES HELD FIXED IF IP.GT.0.
C        NOTE THAT THE INDICES IN IB ARRAY MAY BE IN ANY ORDER,
C        BUT MUST BE BETWEEN 1 AND K (K IS IN $PARMS INPUT).
C       DIMENSION P(1),X(500,5),B(1),W(5),IB(1)
C>>>>>  INSERT USER CODE HERE TO EVALUATE P  <<<<<
C       END
C [SUBZ]:
C       SUBROUTINE MY_SUBZ(Y,X,B,W,NW,N,TITLE,IOUT)
C       USER WRITTEN INITIALIZATION ROUTINE (CALLED ONCE BY 'NLSOL').
C       SUBZ MAY BE USED TO CHECK Y(IN),X(IN,M) AFTER INPUT VIA
C       OBJECT (RUN)-TIME INPUT (SEE BELOW) ON UNIT IALT. ALSO, SUBZ
C       MAY BE USED TO READ ADDITIONAL $INIT PARAMETERS, AND TO LOAD
C        ANY COMMON BLOCKS IF NEEDED IN THE USERS FCODE,PCODE.
C       Y,X,B,W  ARE THE SAME AS USED IN FCODE (SEE ABOVE).
C       NW= USE ANY DUMMY INTEGER VARIABLE (THIS IS
C        TO MAINTAIN COMPATIBILITY WITH 'MARQRT' OR 'IMSLMQ').
C       N= NO. OF OBSERVATIONS IN Y(N),X(N,M) ARRAYS, WHERE
C        K.GE.N.LE.500 (N,M,K  ARE IN $PARMS INPUT).
C       TITLE= (INPUT) 80-CHARACTER HEADING (SEE INPUT FOR005 BELOW).
C       IOUT= 1 IF TO WRITE OUTPUT ON BOTH FOR006 AND FOR016.
C           = 0 IF TO WRITE OUTPUT ONLY ON FOR006.
C       DIMENSION Y(1),X(500,5),B(1),W(5)
C       CHARACTER*80 TITLE
C>>>>> INSERT USER CODE HERE FOR ANY INITIALIZATION DESIRED  <<<<<
C       END
```

```
C [SUBEND]:
C     SUBROUTINE MY_SUBEND(Y,X,B,K,N,TITLE,IOUT)
C     USER WRITTEN TERMINATION ROUTINE (CALLED ONCE BY 'NLSOL').
C     SUBEND MAY BE USED TO OUTPUT THE FINAL SOLUTION VECTOR B(I),
C     I=1,2,...,K, IN OTHER FORMS, ETC., AS DESIRED. [OR IT MAY BE A
C     DUMMY ROUTINE; I.E., JUST RETURNS.]
C     Y,X,K,N,TITLE,IOUT  ARE THE SAME AS IN SUBZ AND FCODE.
C     B= (INPUT) IS THE FINAL SOLUTION VECTOR AS DETERMINED BY
C      'NLSOL' (SEE REF1 FOR DETAILS).
C     DIMENSION Y(1),X(500,5),B(1)
C     CHARACTER*80 TITLE
C>>>>>  INSERT USER CODE HERE FOR ANY TERMINATION SUMMARY DESIRED  <<<<<
C     END
C
C************************************************************************
C
C** INPUT ORDER ON FOR005 (PARAMETER FILE LOGICAL NAME):
C
C 1.  TITLE (MAX. 80-CHARACTERS--ALWAYS READ BEFORE $PARMS INPUT).
C 2.  $PARMS  -- POSSIBLE NAMES ARE: N,M,K,B(),IP,IB(),IALT,IWT,IDER,
C       BL(),BH(),IPRT,IOUT,NITER,SP, -- PLUS FOLLOWING PARAMETERS FROM
C     REF1 (NL2SOL), P.31-35: IV(),V().
C 3.  (OBJECT-RUN-TIME FORMAT STATEMENT) TO DESCRIBE THE FORMAT OF THE
C     DATA MATRIX ROW Y(I),(X(I,J),J=1,M*) READ ON FILE IALT, WHERE
C     M*=M (IF IWT=0) OR M*=M+1 (IF IWT>0), M.LE.4, AND I=1,2,...,N.
C (3A). INSERT DATA MATRIX HERE ONLY IF IALT=5.
C 4.  $INIT OPTIONAL NAMELIST USED FOR READING PROBLEM-DEPENDENT
C     PARAMETERS USED IN SUBROUTINE SUBZ (SEE ABOVE).
C 5.  OPTIONALLY, REPEAT STEPS 1-4, IF PARAMETER ISTOP=0 WAS USED
C     IN THE LAST STEP 2.
C
C** OUTPUT IS GIVEN ON FOR006 (ON-LINE USUALLY) AND ON FOR016(IF IOUT=1)
C   FOR016 CONTAINS ALL PRINTABLE OUTPUT SELECTED VIA $PARMS IPRT,IOUT.
C   NOTE: IPRT=0 GIVES ABBREVIATED OUTPUT ON FOR006 (BUT MORE ON FOR016)
C         IPRT=1 OR -2 GIVES DETAILED OUTPUT ON BOTH 6 AND 16.
C         IPRT=-1 GIVES MODERATE OUTPUT ON 6 (DETAILED ON 16).
C
C** TO RUN ON VAX (ELIMINATE <> DELIMITERS IN SUBSTITUTIONS):
C
C   $ASSIGN <PARAMETER FILE NAME> FOR005
C   $ASSIGN <DATA MATRIX FILE NAME> FOR010
C   $RUN <MAIN NAME>
C
C [NOTE: NLSOL2 USES SCRATCH UNIT FOR099 VIA CALL PRENAM(5,99)]
C
C************************************************************************
C
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

Because of the length of NLSOL2 and related subprograms, the rest of the listing has been suppressed; however, the complete code is available on the distributed tape.