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.............................. 2
INTRODUCTION................................ 3
SUMMARY OF CALCULATIONS.................. 4
PARAMETERS, FILES, AND DATA REQUIRED........... 6
$INIT PARAMETER DEFINITIONS................ 7
DATA MATRIX OPTIONS.......................... 8
EXAMPLES OF INPUT PARAMETERS AND DATA ORDERING... 9
SPECIAL OBJECT FORMAT PHRASES................. 10
VAX OPERATING INSTRUCTIONS................. 11
ERROR MESSAGES.......................... 11
PRINTED OUTPUT.......................... 12
REFERENCES.............................. 13

Appendix 1.-- Conversion to other systems.......... 14
Appendix 2.-- Test problem input/output........... 15
Appendix 3.-- Source code availability............. 21
Source code listing........................ 22

Table 1.--Cole-Cole parameter names and order assumed.... 5

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 observation 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

\[ Z(\omega) = R_0 \left[ 1 - m \left( 1 - \frac{1}{1 + i \omega T} \right) \right] \]  

where \( R_0 \) = amplitude of \( Z(\omega) \) at zero frequency (\( R_0 > 0 \)),
\( m \) = chargeability factor (\( 0 \leq |m| \leq 1 \)),
\( \gamma \) = time constant (\( \gamma > 0 \) seconds),
\( c \) = frequency dependency factor (\( 0 < c \leq 1 \)),
\( \omega \) = angular frequency (\( \omega = 2 \pi f, f \geq 0 \) Hertz),
\( i = (-1)^{1/2} \),
and \( Z(\omega) \) = complex impedance at frequency \( \omega \), 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 (\( R_0, m, \gamma, c \)) are to be determined by the NLS inversion program, given an observed data set for \( Z(\omega) \) defined over a range of \( \omega \), 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

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

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 \( B(J), J=1,2,\ldots,K \), 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

<table>
<thead>
<tr>
<th>Parameter name</th>
<th>MODEL</th>
<th>K *</th>
</tr>
</thead>
<tbody>
<tr>
<td>B(1) R0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(2) m</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>B(3) Tau</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(4) c</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(5) m2</td>
<td>2</td>
<td>7</td>
</tr>
<tr>
<td>B(6) Tau2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(7) c2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(8) m3</td>
<td>3</td>
<td>10</td>
</tr>
<tr>
<td>B(9) Tau3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(10) c3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(11) m4</td>
<td>4</td>
<td>13</td>
</tr>
<tr>
<td>B(12) Tau4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B(13) c4</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

* 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=ω/2π) and Y(I) be the corresponding observed real component: amplitude, phase, real-, or imaginary-part of Z(ω), 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 particular 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 augmented 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 $j=1,2,\ldots,L$) can be explicitly written from a general form of equation (2) as

$$\frac{\partial Z}{\partial \tau_j} = \frac{Z}{R_o}, \quad Z(\omega) = R_o[i-m_j f_j] \ldots [i-m_L f_L] \quad (l \leq L \leq 4)$$

$$\frac{\partial Z}{\partial f_j} = \frac{-f_j Z}{[i-m_j f_j]} , \quad f_j = 1 - \frac{1}{1+i\omega \tau_j} \quad (1 \leq j \leq L)$$

$$\frac{\partial Z}{\partial c_j} = \frac{-m_j}{[i-m_j f_j]} \frac{Z}{\partial c_j} , \quad f_j = \frac{(i\omega \tau_j)^f_j (c_j/\tau_j)}{[1+(i\omega \tau_j)^f_j]^2}$$

$$(3)$$

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 $BL(J) \leq B(J) \leq BH(J)$, $J=1,2,\ldots,K$ (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, in 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

$INIT parameters:

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):

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

(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):

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

For any 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 $\text{PARMS}$ 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.
$\text{PARMS N=28,M=1,SP=3,}$
$K=4,\text{IP=1,IB=1, IWT=1, IALT=5,}$
$\text{IDER=0,V(42)=1.E-3,NITER=25,}$
$\text{BL=1,1E-10,.001,1E-10,}$
$\text{BH=1,.9999,1000,.9999,}$
$B=1,.55,10,.75$ (3F10.0)

\begin{verbatim}
6.9  .001  .011
9.98 .002  .095
\end{verbatim}

<etc. for 26 more observations>
$\text{INIT MODEL=1,IOB=2 }$ $\text{END}$

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.
$\text{PARMS N=20,M=2,K=7,SP=3,IALT=5,}$
$\text{IDER=1,IWT=2,NITER=20,}$
$\text{BL=1,3*1E-10,-.9999,2*1E-10,}$
$\text{BH=10,.9999,1E5,.9999,0,1E5,.9999,}$
$B=1.024,.052,.12,.384,-.885,9.4E-5,.779$ (3G12.5,T1,G12.5)

\begin{verbatim}
1.0189  0.11000  1.0000
6.9000  0.11000  2.0000
1.0052  0.33000  1.0000
8.8000  0.33000  2.0000
\end{verbatim}

<etc. for 18 more observations>
$\text{INIT IOB=5,MODEL=2}$

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.**

```plaintext
$PARMS N=40,M=2,K=4,SP=3,IALT=5,
IWT=0,NITER=20,
BL=.1,3*1E-10,
BH=10,.9999,1E5,.9999,
B=1.024,.52,.12,.5$

\[3G12.5\]

\[
\begin{array}{ccc}
1.0189 & 0.11000 & 1.0000 \\
6.9000 & 0.11000 & 2.0000 \\
1.0052 & 0.33000 & 1.0000 \\
8.8000 & 0.33000 & 2.0000 \\
\end{array}
\]

<etc. for 16 more observations for first data set>

\[
\begin{array}{ccc}
2.045 & .05 & 1.0000 \\
5.2034 & .05 & 2.0000 \\
1.058 & .5 & 1.0000 \\
10.715 & .5 & 2.0000 \\
\end{array}
\]

<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 **not** 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
The 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 $B(I), I=1,2,\ldots,K$ (see Table 1), and that $\text{INIT IOB}$ and $\text{PARMS M}$ 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 $\text{INIT}$ parameters given first, followed by all $\text{PARMS}$ 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


----------, 1984, A general interface for producing forward solution programs (Subprogram FWDSOL): USGS Open-File Rept. 84-348, 43 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 concatenation (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
<NLSIP>: MODEL-2 FIT [T]

MODELL = 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

INITIAL PARAMETERS: B=
0.15000000E+01 0.50000000E+00 0.10000000E+01 0.50000000E+00 0.50000000E+00 0.10000000E-02 0.30000000E+00

PARAMETER HIGHER BOUNDS: BH=
0.10000000E+04 0.99989999E+00 0.10000000E+04 0.99989999E+00 0.99989999E+00 0.10000000E+04 0.99989999E+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 0.30000000E+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  0.546E+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)
<table>
<thead>
<tr>
<th>X(I,J)</th>
<th>0.447233E-01</th>
<th>0.223E+03</th>
<th>-0.167E-01</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.366471E-01</td>
<td>0.760E+03</td>
<td>0.192E-04</td>
</tr>
<tr>
<td>2</td>
<td>0.632981E-01</td>
<td>0.179E+05</td>
<td>0.437E-01</td>
</tr>
<tr>
<td>3</td>
<td>0.185638E+00</td>
<td>0.370E+02</td>
<td>0.901E-04</td>
</tr>
</tbody>
</table>

**Covariance = SCALE * (J*J' + J**2)**

<table>
<thead>
<tr>
<th>ROW 1</th>
<th>0.21432-09</th>
<th>0.28999-10</th>
<th>0.47564-08</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROW 2</td>
<td>0.38711-10</td>
<td>-0.54218-09</td>
<td>0.25711-09</td>
</tr>
<tr>
<td>ROW 3</td>
<td>0.12910-10</td>
<td>-0.28496-08</td>
<td>0.50494-09</td>
</tr>
</tbody>
</table>

**Program NLSIP**

<table>
<thead>
<tr>
<th>ROW 5</th>
<th>0.35444-13</th>
<th>-0.1198-10</th>
<th>0.2449-11</th>
<th>0.11723-10</th>
<th>0.75000-13</th>
</tr>
</thead>
<tbody>
<tr>
<td>ROW 6</td>
<td>0.53778-08</td>
<td>-0.92358-08</td>
<td>-0.59578-08</td>
<td>-0.2107E-10</td>
<td>0.16498-07</td>
</tr>
</tbody>
</table>

**Observed** = \( X(I,J) \)

<table>
<thead>
<tr>
<th>X(I,J)</th>
<th>0.1870000001</th>
<th>0.156903X01</th>
<th>0.95000000</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.71410000001</td>
<td>0.241898X01</td>
<td>0.30000000</td>
</tr>
</tbody>
</table>

**Covariance**

<table>
<thead>
<tr>
<th>X(I,J)</th>
<th>0.530748E+00</th>
<th>0.370E+02</th>
<th>0.901E-04</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.750748E+00</td>
<td>0.370E+02</td>
<td>0.901E-04</td>
</tr>
</tbody>
</table>

**Correlation Matrix**

<table>
<thead>
<tr>
<th>I</th>
<th>0.100000001</th>
<th>0.100000001</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.38738-01</td>
<td>0.1490000001</td>
</tr>
</tbody>
</table>

**Ensemble = 0.2177426-01 AVSIES.REE = 0.564447E-01**

**Page 18**
**Program NLSIP**

**VAX Documentation**

---

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 **

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.2000E+01</td>
<td>0.1464E-04</td>
<td>0.3272E-03</td>
<td>0.3272E-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.2485E+00</td>
<td>0.6896E-04</td>
<td>0.1321E-03</td>
<td>0.1321E-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.1200E+01</td>
<td>0.1604E-04</td>
<td>0.4628E-03</td>
<td>0.4628E-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.0346E+00</td>
<td>0.6742E-04</td>
<td>0.1093E-03</td>
<td>0.1093E-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.4006E-03</td>
<td>0.2739E-06</td>
<td>0.4327E-03</td>
<td>0.4327E-01</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.5002E+00</td>
<td>0.1292E-03</td>
<td>0.1645E-03</td>
<td>0.1645E-01</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>PARAMETER NAME</th>
<th>FINAL SOLUTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>B( 1) R0</td>
<td>0.19999168E+01</td>
</tr>
<tr>
<td>B( 2) m</td>
<td>0.24853931E+00</td>
</tr>
<tr>
<td>B( 3) Tau</td>
<td>0.11999412E+01</td>
</tr>
<tr>
<td>B( 4) c</td>
<td>0.50000000E+00</td>
</tr>
<tr>
<td>B( 5) m2</td>
<td>0.33439151E+00</td>
</tr>
<tr>
<td>B( 6) Tau2</td>
<td>0.40063914E-03</td>
</tr>
<tr>
<td>B( 7) c2</td>
<td>0.50018924E+00</td>
</tr>
</tbody>
</table>

* FIXED

$\text{TOTAL "ELAPSED" TIME=} 20.61 \text{ SEC. ( 0 MIN. 20.61 SEC.)}$

$\text{CPU TIME=} 2.72 \text{ SEC. ( 0 M. 2.72 S.) CPU Z = 13.20\%}$

$\text{BUF.I/O COUNT=} 86$

$\text{DIR.I/O COUNT=} 35$

$\text{PAGE_FAULTS=} 253$
MODEL=2 FIT [T]

L.SQ. SOLUTION

\[ \begin{align*}
R_0 &= 2.00 \\
m_1 &= 0.25 \\
m_2 &= 0.33 \\
\tau_1 &= 1.20 \\
\tau_2 &= 0.4E-3 \\
\varphi_1 &= 0.50 \\
\varphi_2 &= 0.50 \\
\end{align*} \]

* = parameter held fixed

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:

00000010  [MAIN PROGRAM]
00000310  SUBROUTINE FCODE
00000950  SUBROUTINE PCODE
00001670  SUBROUTINE SUBZ
00002330  SUBROUTINE TIP_SUBEND
00003150  SUBROUTINE ERRMSG
00004130  SUBROUTINE NLSOL2
00004580  SUBROUTINE NLSITR
00005180  SUBROUTINE INTRAN
00005660  SUBROUTINE INTRN
00006020  SUBROUTINE INTRAN
00006180  SUBROUTINE PROCINFO
00006540  REAL FUNCTION ASINH
00006740  FUNCTION ERF
00006940  FUNCTION ERFINV
00007140  SUBROUTINE ERRMSGI
00007590  INTEGER FUNCTION LOC
00008030  SUBROUTINE NL2SOL
00008440  SUBROUTINE NL2SNO
00008850  SUBROUTINE NL2ITR
00009190  SUBROUTINE ASSESS
00009590  SUBROUTINE COVCLC
00009990  SUBROUTINE DFAULT
00010390  REAL FUNCTION DOTPRD
00010790  SUBROUTINE DUPDAT
00011190  SUBROUTINE GQTSTP
00011670  SUBROUTINE ITSMRY
00012070  SUBROUTINE LINVRT
00012430  SUBROUTINE LITVNU
00012930  SUBROUTINE LIBMUL
00013430  SUBROUTINE LMSTEP
00013830  SUBROUTINE LSQRT
00014230  REAL FUNCTION LSVMIN
00014630  SUBROUTINE LTSQAR
00015030  SUBROUTINE PARCHK
00015430  SUBROUTINE QAPPLY
00015830  SUBROUTINE QRFACT
00016230  SUBROUTINE RPTMUL
00016630  SUBROUTINE SLUFPD
00017030  SUBROUTINE SLVMUL
00017430  LOGICAL FUNCTION STOPX
00017830  SUBROUTINE VAXPY
00018230  SUBROUTINE VCOPY
SUBROUTINE VSCOPY
REAL FUNCTION V2NORM
INTEGER FUNCTION IMDCON
REAL FUNCTION RMDCON
FUNCTION TCHEB
SUBROUTINE WARN

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.
C <NLSIP>: INVERSION OF FREQUENCY IP SOUNDINGS.
C** VAX-11/780 VERSION: 4/18/86 **
C--BY W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.
C-------------------------------------------------------------------
C--REFERENCES:
C--
C 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.
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)
SAVE
COMPLEX Z,Z1,ZF,ONE, FL,FM,WTC
REAL Y(1),X(500,5),B(1),PRNT(5)

COMMON/COLSEIP/MODEL,IOB
COMMON/PASS/FL(4),FM(4),WTC(4),ZF,AMP,OMEGA

DATA ONE/(1.0,0.0)/,TWOP/(6.283185308/
DATA PI2/1.570796327/

DO 10 J=1,5
10 PRNT(J)=X(IN,J)

FREQ=PRNT(1)

OMEGA=TWOP/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
   11=3*IMODEL-1
   12=11+1
   13=11+2

   Z=(OMEGA*B(12))**B(13)

   Z1=Z*(CEXP(CMPLX(0.0,B(13)*PI2)))

   WTC(IMODEL)=Z1

   FL(IMODEL)=ONE-ONE/(ONE+Z1)

   Z=ONE-CMPLX(B(11),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.

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.0).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 J-J1,I3
    IF(IP.GT.O) THEN
      DO I=1,IP
        IF(IB(I).EQ.J) THEN
          P(J)=0.0
          GO TO 100
        ENDIF
      ENDDO
      IF(SKIP) GO TO 5
      IF(J.EQ.1) THEN
        PZ(1)=ZF/B(1)
      ELSE IF(J.EQ.11) THEN
        PZ(1)=FL(IMODEL)*ZF/FM(IMODEL)
      ELSE IF(J.EQ.12) THEN
        PZ(1)=(WTC(IMODEL)*B(11)/B(12))*/2
      ELSE IF(J.EQ.13) THEN
        PZ(1)=(WTC(IMODEL)*CLOG(CMPLX(0.0,OMEGA*B(12))))*/2
      ENDIF
      P(J)=B(11)*PARTF*ZF/FM(IMODEL)
    ELSE IF(J.EQ.11) THEN
      PZ(1)=FL(IMODEL)*ZF/FM(IMODEL)
    ELSE IF(J.EQ.12) THEN
      PZ(1)=(WTC(IMODEL)*B(13)/B(12))*/2
    ELSE IF(J.EQ.13) THEN
      PZ(1)=(WTC(IMODEL)*CLOG(CMPLX(0.0,OMEGA*B(12))))*/2
    ENDIF
    P(J)=-B(11)*PARTF*ZF/FM(IMODEL)
  ENDIF
  GO TO 100
  P(J)=-1000.*(REAL(ZF)*AIMAG(PZ(J))-AIMAG(ZF)*REAL(PZ(J)))/(AMP*AMP))
Program NLSIP
VAX Documentation

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
C CHARACTER*80 TITLE
C CHARACTER*4 PARNAM(13)
C REAL Y(1),X(500,5),B(1),PRNT(1)
C COMMON/COLEIP/MODEL,IOB
C NAMELIST/INIT/MODEL,IOB
C DATA ISUBZ/0/,PARNAM/'RO','M','TAU','C','M2','TAU2','C2',
C 'M3','TAU3','C3','M4','TAU4','C4'/
C IF(ISUBZ.NE.0) GO TO 10
C
C--PRESET
C ISUBZ=1
C MODEL=1
C IOB=1
C
C READ(99,INIT)
CALL NONBLANK(TITLE,NONBLK)
WRITE(6,20) TITLE
20 FORMAT('KNLSIP>: ' , 5X, A<NONBLK>/)
IF(IOUT.EQ.1) WRITE(16,20) TITLE
WRITE(6,30) MODEL,IOB
30 FORMAT('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-XU.1X-0',0,6,16)
  IF(IOB.LT.5) GO TO 70
  IF(IFIX(X(I,2)).LT.1.R.IFIX(X(I,2)).GT.4) CALL ERRMSG(
    1 '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)
  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/"RO" ,'M' ,'TAU' ,"C" ,"M2" ,"TAU2" ,"C2" ,
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(IFG.T.E 0) THEN
    DO J=1,IP
      IF(IB(J).NE.0) FLAG(IB(J))='*
    ENDDO
  ENDF
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:)=' ,'
  ENDIF
ELSE
  IDOL=2
  LINE(I:)=' ,'
ENDIF
ENDIF
ENDIF
CALL NONBLANK(LINE,NB)
WRITE(4,66) LINE
IFCIDOL.LT.2) GO TO 70
DO 80 I=1,K
ENCODE(16,90,LINE(3:18)) B(I)
80 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 CPUPTIME(I1,I2)
C
CPUTIME WRITES "ELAPSED & CPU" TIME FROM PREVIOUS "CALL SETTIME" ON
FORTRAN UNITS II (IF NOT 0) AND I2 (IF NOT 0).
C WILL EJECT FIRST IF I1>0 (OR I2>0).
C DOUBLE SPACE FIRST IF I1<0 (OR I2<0).
C E.G., USE TO TIME ELAPSED & CPU TIME FOR PROGRAM OR CODE SEGMENTS AS:
CALL SETTIME I DON'T FORGET TO DO THIS!
THE CODE TO TIME IS HERE USUALLY A COMPLETE PROGRAM!
CALL CPUTIME(-6,16) ! OR USE II OR I2=0 TO OMIT WRITE.
ALSO CAN USE CALL GETTIME(CPU) TO GET JUST THE CPU (SEC)
SINCE THE LAST CALL SETTIME WAS DONE.
SAVE
INTEGER*4 ABSVAL(4),INCRVAL(4)
CALL PROCFINFO(ABSVAL,INCRVAL)
TIMES=SECONDS(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)
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
C** ENTRY 'CALL SETTIME'--MUST BE DONE BEFORE 'CALL CPUTIME(I1,I2)'
CALL PROCI NFO(ABSVAL, INCRVAL)
RETURN

C** ENTRY 'CALL GETTIME(CPU)' -- TO GET CPU(SEC) SINCE LAST CALL SETTIME
ENTRY GETTIME(CPU)
CALL PROCI NFO(ABSVAL, INCRVAL)
CPU=INCRVAL(1)*.01
RETURN
END

SUBROUTINE ERRMSG(MSG, ISKIP, IUNIT1, IUNIT2)

GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780

MSG(*) - VARIABLE-LENGTH 'MESSAGE'
ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2
> 0 FOR ONE BLANK LINE BEFORE.
IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).
IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).

MESSAGES ARE WRITTEN IN THE FORM:

{ERRMSG}: _MSG_HERE_

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

1 FORMAT(IX, '{ERRMSG}: ', A(I))
2 FORMAT(/1X, '{ERRMSG}: ', A(I))
3 END

SUBROUTINE NLSOL2(FCODE, PCODE, SUBZ, SUBEND)

{NLSOL2}: GENERAL NONLINEAR LEAST-SQUARES SOLUTION {2/19/86}

USING DENNIS ET AL (1979; SEE REF1 BELOW)

NLSOL2 IS A REVISED VERSION OF NLSOL WITHOUT CALL NAMELIST;
I.E., NLSOL2 USES THE CURRENT VAX-11/780 NAMELIST VERSION. NOTE
SUBZ MUST BE CHANGED TO READ(99,INIT) FROM CALL NAMELIST(5,'$INIT'),00004170
WHERE CALL PRENAM(5,99) IS USED TO ENSURE UNIT=5 NAMELIST IS IN
PROPER FORMAT ON SCRATCH UNIT=99 (FOR099 DELETED ON RETURN TO VMS).

SUBROUTINE NLSOL2(FCODE, PCODE, SUBZ, SUBEND)
C** THIS IS AN INTERFACE ROUTINE WRITTEN FOR THE VAX-11/780 BY 00004250
C W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO. 00004260
C C** THIS INTERFACE (NLSOL) HAS ADDITIONAL OPTIONS (BESIDE REF1) TO: 00004270
C (1) PERFORM EITHER UNCONSTRAINED OR UP TO 4-TYPES OF CONSTRAINED 00004280
C ADAPTIVE NONLINEAR REGRESSION FOR ARBITRARY NONLINEAR PROBLEMS. 00004300
C (I.E., PARTIAL OR FULL LOWER/HIGHER PARAMETER BOUNDS, ETC.) 00004310
C (2) HOLDING CERTAIN PARAMETERS FIXED (I.E., AS CONSTANTS) IN THE 00004320
C LEAST-SQUARES (THIS IS ANOTHER FORM OF CONSTRAINTING SOLUTION 00004330
C SPACE). 00004340
C (3) PROVIDE FOR WEIGHTED OBSERVATIONS (I.E., WEIGHTED LEAST-SQUARES) 00004350
C (4) OBJECT (RUN)-TIME CONTROL OF READING THE DATA MATRIX, PLUS 00004360
C MANY OTHER I/O OPTIONS, ETC. 00004370
C (5) OPTIONALLY, ONE CAN USE EITHER ESTIMATED PARTIAL DERIVATIVES, OR 00004380
C ANALYTICAL PARTIAL DERIVATIVES (IF SUBROUTINE FCODE AVAILABLE). 00004390
C C** THE USER ONLY NEEDS TO WRITE SUBROUTINES FCODE, FCODE, SUBZ, AND 00004400
C SUBEND (SEE DETAILS BELOW) EXACTLY AS USED IN SUBROUTINE 'MARQRT' 00004420
C (SEE REF2) OR 'IMSLMQ' (SEE REF3). ALSO, THE SAME PARAMETER FILE 00004430
C FOR005 AND OBJECT (RUN)-TIME DATA MATRIX FILE FOR010 AS USED BY 00004440
C EITHER MARQRT OR IMSLMQ MAY BE USED IN 'NLSOL'. 00004450
C C** NLSOL CALLS NLITR WHICH CALLS 'NL2ITR' AS PUBLISHED BY DENNIS ET AL,00004470
C (SEE REF1, P. 38), OR 'NL2SNO' (SEE REF1, P. 35). 00004480
C C** REF1: DENNIS, J.E., ET AL, 1979, AN ADAPTIVE NONLINEAR LEAST- 00004490
C SQUARES ALGORITHM, NTIS REPORT AD-A079-716. 00004500
C C** REF2: ANDERSON, W.L., 1980, PROGRAM MARQHXY: INVERSION OF HX AND HY00004520
C FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN- 00004530
C FILE REPT. 80-901. 00004540
C C** REF3: ANDERSON, W.L., 1980, PROGRAM IMSLEXY: INVERSION OF EX AND EY00004560
C FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN- 00004570
C FILE REPT. 80-1073. 00004580
C C***********************************************************************00004610
C C**** THE USER MUST DECLARE THE CALLING PARAMETERS AS EXTERNAL IN THE 00004630
C CALLING PROGRAM (ANY DESIRED NAMES MAY BE USED). 00004640
C E.G., 00004650
C [MAIN]: 00004660
C EXTERNAL MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND 00004670
C CALL NLSOL2(MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND) 00004680
C STOP 1(COR USE): CALL EXIT 00004690
C END 00004700
C [FCODE]: 00004710
C SUBROUTINE MY_FCODE(Y,X,B,W,F,IN,IDER) 00004720
C USER WRITTEN TO EVALUATE THE NONLINEAR OBJECTIVE FUNCTION (F) 00004730
C USED IN NLSOL AS THE WEIGHTED SUM OF (Y(IN)-F)**2, WHERE 00004740
C Y= OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N, WHERE N IS 00004760
C GIVEN IN SPARMS NAMELIST INPUT--SEE BELOW). 00004770
C X= OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,M, WHERE 00004780
C
M IS IN $PARMS INPUT).
C
K IS IN $PARMS INPUT).
C
W= WORK ARRAY (DIM. 5)—MAY BE USED TO PASS DATA TO FCODE.
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
C IPCODE]: » 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,IBM)
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 P= LAST FUNCTION VALUE FROM FCODE AT OBSERVATION IN.
C (NOTE THAT P 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 IBM= ARRAY OF B-PARAMETER INDICES HELD FIXED IF IP.GT.0.
C NOTE THAT THE INDICES IN IBM 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),IBM(1)
C>»» INSERT USER CODE HERE TO EVALUATE P <««
C END
C
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
C
C [SUBEND]:
C SUBROUTINE SUBEND(Y,X,B,K,N,TITLE,IOUT)
C USER WRITTEN TERMINATION ROUTINE (CALLED ONCE BY 'NELSOL').
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 'NELSOL' (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 <<<<00005440
C END 00005450
C 00005460
C***********************************************************************00005470
C***********************************************************************00005480
C** INPUT ORDER ON FOR005 (PARAMETER FILE LOGICAL NAME): 00005490
C 00005500
C 1. TITLE (MAX. 80-CHARACTERS--ALWAYS READ BEFORE $PARMS INPUT). 00005510
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 00005520
C REF1 (NEL2SOL), P.31-35: IV(),V(). 00005530
C 3. (OBJECT-RUN-TIME FORMAT STATEMENT) TO DESCRIBE THE FORMAT OF THE 00005540
C DATA MATRIX ROW Y(I),(X(I,J),J=1,M) READ ON FILE IALT, WHERE 00005550
C M=M (IF IWT=0) OR M=M+1 (IF IWT>0), N.LE.4, AND I=1,2,...,N. 00005560
C (3A). INSERT DATA MATRIX HERE ONLY IF IALT=5. 00005570
C 4. $INIT OPTIONAL NAMELIST USED FOR READING PROBLEM-DEPENDENT 00005580
C PARAMETERS USED IN SUBROUTINE SUBZ (SEE ABOVE). 00005590
C 5. Optionally, repeat steps 1-4, if parameter ISTOP=0 was used 00005600
C in the last step 2. 00005610
C 00005620
C** OUTPUT IS GIVEN ON FOR006 (ON-LINE USUALLY) AND ON FOR016(IF IOUT=1) 00005630
C FOR016 CONTAINS ALL PRINTABLE OUTPUT SELECTED VIA $PARMS IPRT,IOUT. 00005640
C NOTE: IPRT=0 GIVES ABBREVIATED OUTPUT ON FOR006 (BUT MORE ON FOR016)00005650
C IPRT=1 OR -2 GIVES DETAILED OUTPUT ON BOTH 6 AND 16. 00005660
C IPRT=-1 GIVES MODERATE OUTPUT ON 6 (DETAILED ON 16). 00005670
C 00005680
C** TO RUN ON VAX (ELIMINATE <> DELIMITERS IN SUBSTITUTIONS): 00005690
C 00005700
C $ASSIGN <PARAMETER FILE NAME> FOR005 00005710
C $ASSIGN <DATA MATRIX FILE NAME> FOR010 00005720
C $RUN <MAIN NAME> 00005730
C 00005740
C [NOTE: NLSOL2 USES SCRATCH UNIT FOR099 VIA CALL PRENAM(5,99)] 00005750
C 00005760
C***********************************************************************00005780
C***********************************************************************00005800

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.