U.S. DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

Nonlinear least-squares inversion of bipole-bipole direct-current data (Program NLSBPBP)

by

Walter L. Anderson

Open-File Report 87-95

1987

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 author regarding program correctness, accuracy, or proper execution on all computer systems.

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

CONTENTS

ABSTRACT

A computer program is presented that inverts direct-current (DC) sounding data for a colinear bipole-bipole array on a layered earth using an adaptive nonlinear least-squares method. An option is provided to use either fixed-length or variable-length source and receiver bipoles for each observed data point. An inversion example using a DC data set for fixed-length bipoles 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 colinear bipole-bipole direct-current (DC) sounding data over layered earth models is provided by program NLSBPBP, which is described in this report. The numerical technique uses a general adaptive nonlinear least-squares (NLS) algorithm originally developed by Dennis and others (1979; 1981), and extended externally for constrained nonlinear regression by Anderson (1982). The DC forward problem, required iteratively in the inverse solution, uses an efficient numerical digital filtering algorithm (called lagged convolution) by Anderson (1975). Note that a stand-alone forward (FWD) program can be obtained easily from the NLSBPBP inversion program; this is described, in general, by Anderson (1984) on how to convert any NLS-program to a FWD-program (or vise-versa).

This report utilizes the general nonlinear least-squares method defined by Anderson (1982), but only as it applies here to DC bipole-bipole sounding data. Either fixed- or variable-length bipole-bipole arrays can be used separately or jointly in the least-squares solution. The joint use of variable-length source and receiver bipoles in the least-squares solution gives added flexability for general field applications. However, fixed-length bipole-bipole arrays are often used in mining and other electrical surveys, with the data commonly plotted as pseudo-sections. In any case, this program provides both types of colinear array configurations as a user option.

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

Figure 1 illustrates the geometry of the colinear bipole-bipole array configuration, where (A,B) are the current electrodes and (M,N) are the potential electrodes.

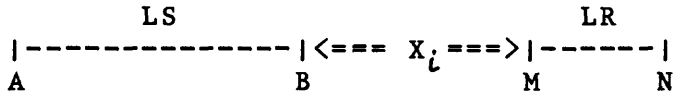


Figure 1.-- Bipole-bipole array on the earth's surface.

Definitions: LS= length (in m) of source bipole (A,B),
LR= length (in m) of receiver bipole (M,N),
X_i= distance (in m) between B and M for each observed data point (i=1 to n).

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, the bipole-bipole apparent resistivity function over a layered earth model, for the array configuration shown in Figure 1, can be written (after some modifications from Niwas and Israil, 1986) as

$$\rho_a(X_i) = \rho_1 \left[\frac{\bar{\rho}(r_1)/r_1 - \bar{\rho}(r_2)/r_2 - \bar{\rho}(r_3)/r_3 + \bar{\rho}(r_4)/r_4}{1/r_1 - 1/r_2 - 1/r_3 + 1/r_4} \right], \quad (1)$$

$$\bar{\rho}(r_k) = r_k \int_0^\infty [K(\lambda)-1]\, J_0(\lambda r_k)\, d\lambda + 1 \quad (k=1,2,3,4), \quad (1a)$$

where ρ_1 = resistivity of layer 1,
r_1 = distance AM = X_i + LS,
r_2 = distance AN = X_i + LS + LR,
r_3 = distance BM = X_i,
r_4 = distance BN = X_i + LR,
X_i = distance BM (A<B<M<N on the x-axis),

and $K(\lambda)$= kernel function determined by the thicknesses h_1, h_2, ..., h_{m-1}, and resistivities $\rho_1, \rho_2, \ldots, \rho_m$ of the horizontally m-layered earth model (0<m<11 in NLSBPBP).

The form of the complete kernel function $[K(\lambda)-1]$ used in the Hankel transform in equation (1a) has been augmented to be a decreasing function as $\lambda \to \infty$ by substracting and adding the known integral term

$$\int_0^{\infty} J_0(\lambda r_k)\, d\lambda = 1/r_k . \qquad (2)$$

This ensures rapid and accurate convergence in the lagged convolution algorithm described by Anderson (1975). Note that equation (1) needs only one direct execution of the lagged convolution algorithm from r-minimum ($=X_1$) to r-maximum ($=X_n+LS+LR$) for all independent variables X_i, i=1,2,...,n in a given data set. All terms in the numerator of equation (1) are obtained by rapid spline interpolation after the initial lagged convolution operation; thus the forward solution is extremely fast for any and all independent variable distances for the entire forward sounding curve. It should also be noted that, using this scheme, special cases such as MN = AB (i.e., $r_1 = r_4$ in equation 1) are not checked nor necessary to save additional integration time.

The unknown m-layered model parameters used in NLSBPBP are denoted by the vector B(J), J=1,2,...,K, where K=2*MM-1 is the total number of parameters, and MM is the number of layers. The order of parameters to be estimated initially in B(J) are given in Table 1.

Table 1.-- Definition of model parameters

layer	resistivities (ohm-m)	thicknesses (m)
1	B(1) = ρ_1	B(MM+1) = h_1
2	B(2) = ρ_2	B(MM+2) = h_2
...	...	...
MM-1	B(MM-1) = ρ_{MM-1}	* B(K) = h_{MM-1}
MM	B(MM)= ρ_{MM}	

* K = 2*MM-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 NLSBPBP. Following the NLS notation in Anderson (1982, p.11-12), we let X(I,1)=X_i (see Figure 1) and Y(I) be the corresponding apparent resistivity for the i-th observation, where each data array X(I,1), Y(I) is given for i=I=1,2,...,n, n>K. In general, a given data set must be given in ascending X(I,1) order for all I; i.e., X(I+1,1)>X(I,1)>0. For the variable source/receiver bipole option, we define LS=X(I,2) and LR=X(I,3) as additional independent variables for each I=1,2,...,n. (See the particular data matrix option, IOPT, available as discussed in the section: DATA MATRIX OPTIONS.)

For the fixed source/receiver bipole option (IOPT=0, LR>0, LS>0; see $INIT parameters definitions below), program NLSBPBP 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 partial derivatives of the apparent resistivity function with respect to each parameter B(J) in Table 1 (with j=J=1,2,...,K) can be explicitly written from differentiating equation (1) as

$$\frac{\partial \rho_a}{\partial B_j} = \left[\frac{\partial/\partial B_j \left[\bar{\rho}(r_1)/r_1 - \bar{\rho}(r_2)/r_2 - \bar{\rho}(r_3)/r_3 + \bar{\rho}(r_4)/r_4 \right] \rho_1}{1/r_1 - 1/r_2 - 1/r_3 + 1/r_4} \right], \quad (3)$$

$$\partial/\partial B_j [\bar{\rho}(r_k)] = r_k \int_0^\infty \partial/\partial B_j [K(\lambda)] \, J_0(\lambda r_k) \, d\lambda \quad (k=1,2,3,4), \quad (3a)$$

where the derivative kernel function in equation (3a) is a recursively defined function similar to $K(\lambda)$ given in equation (1a), and further defined in Niwas and Israil (1986, eqs. 20-21). See Appendix 3 listing of FCODE and PCODE for the coding details, which follows the methods described above in equations (1) and (3), and as selected by $PARMS parameter IDER.

It should be noted that analytic defined partial derivatives in equation (3) using IDER=0 (default) must generally be used in program NLSBPBP to obtain good

convergence in the solution vector B(J) for all J=1,...,K. This is easy to see by examining the form of the forward function in equation (1); due to differencing of individual potential terms in the numerator, use of finite-difference estimated derivatives using equation (1) would produce very poor approximations. Hence, it is recommended that IDER=1 (estimated partials) not be routinely used in this program; this is especially true for small fixed bipoles LS and LR at large values of X(I,1).

Because realizable layered earth 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,...,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 NLSBPBP 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 NLSBPBP must be given exactly as described in Table 1. The $INIT model parameters required by NLSBPBP 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:

MM ... is defined as the number of model layers to use as defined in eq. (1) and Table 1. An arbitrary range of MM is set at $0<MM<11$ (default MM=1). When selecting this $INIT option, it is also required

that $PARMS parameter K=2*MM-1 be explicitly given. (This is a dual requirement that is not checked, however it is necessary because a general purpose nonlinear least-squares routine NLSOL is being used to interface with a layered model problem.)

EPS ... is defined as the lagged convolution integration tolerance in Anderson (1975); default EPS = 1E-8, which is about the best accuracy possible for a 32-bit VAX system. To run somewhat faster, EPS can be increased (e.g., EPS=1E-6).

LS ... is defined as the length (m) of the current source bipole (A,B) in Figure 1. LS>0 is required when using the fixed source/receiver option IOPT=0 defined below.

LR ... is defined as the length (m) of the potential receiver bipole (M,N) in Figure 1. LR>0 is required when using the fixed source/receiver option IOPT=0 defined below.

IOPT= 0 (default) is defined as the fixed source/receiver option, and requires that LS>0 and LR>0 be specifically given as shown in Figure 1. (See DATA MATRIX OPTIONS below for further details.)

IOPT= 1 is defined as the variable source/receiver option, and requires that LS=X(I,2) and LR=X(I,3) be specifically given for each observation I=1,2,...,n. Note that $INIT LS=0 and LR=0 can be given or assumed when IOPT=1, since each LS and LR must be explicitly supplied in the data matrix. (See DATA MATRIX OPTIONS below for further details.)

NOTE:

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

Use $PARMS M=1 whenever $INIT IOPT=0;
Use $PARMS M=3 whenever $INIT IOPT=1.

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 MM 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 IOPT parameters as previously defined. The various data matrix options are summarized as follows:

(a) Fixed source/receiver option as defined by $INIT IOPT=0 (requires $PARMS M=1 and $INIT IOPT=0, LS>0, LR>0; max. 3 items per record):

- Y(I)= I-th observed value of apparent resistivity (ohm-m).
- X(I,1)= I-th observed distance (m) between electrodes B and M in Figure 1, where X(I+1,1)>X(I,1)>0 is required for all I (i.e., increasing distances).
- X(I,2)= weight factor of I-th observation (include only if $PARMS IWT>0).

(b) Variable source/receiver option as defined by $INIT IOPT=1 (requires $PARMS M=3 and $INIT IOPT=1; max. 5 items per record):

- Y(I)= I-th observed value of apparent resistivity (ohm-m).
- X(I,1)= I-th observed distance (m) between electrodes B and M in Figure 1, where X(I+1,1)>X(I,1)>0 is required for all I (i.e., increasing distances).
- X(I,2)= I-th observed source bipole length (i.e, LS=X(I,2)>0 m is used for each I).
- X(I,3)= I-th observed receiver bipole length (i.e, LR=X(I,3)>0 m is used for each I).
- X(I,4)= weight factor of I-th observation (include only if $PARMS IWT>0).

For a given data set, (a) or (b) above, the observations must be ordered by increasing distances in X(I,1) for I=1,2,...,n. This also aids in plotting the results after running NLSBPBP. As suggested earlier, it is generally recommended that the weighted observations option IWT>0 (Anderson, 1982, p. 14-15) be used with this program; usually IWT=2 is adequate for most cases. (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. Fixed source/receiver option (IOPT=0), 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.
$PARMS N=28,K=7,M=1,SP=3,
IP=2,IB=1,4, IWT=1, IALT=5,
IDER=0,V(42)=1.E-4,NITER=25,
BL=4*.001, 3*.1,
BH=4*100000, 3*5000,
B=1600,50,100,2000, 400,100,200$
(3F10.0)
1500.      100.       .011
1350.5     120.       .095
<etc. for 26 more observations>
$INIT MM=4,IOPT=0,LS=200,LR=100 $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).

2. Variable source/receiver option (IOPT=1), weighted observations (IWT=2) that rereads Y(I) again as the weight factor X(I,4), and alternate input data file (IALT=5) for reading the data matrix along with the input parameters on file FOR005:

```
EXAMPLE 2.
$PARMS N= 21,K= 5,M=3,
IPRT=-1,SP= 3,IWT=2,IDER=0,
NITER= 40, IALT=5,
IP= 0,IB= 0,
BL=3*.001,2*.01,
BH=3*500000,2*50000,
B=1600,50,2000, 100,35$
(4G16.8,T1,G16.8)
   1504.9453       50.000000    150.          100.
   1500.0944       62.900000    150.          120.
   1489.1576       79.240000    150.          140.
<etc. for 18 more observations>
$INIT MM= 3,LS= 0,LR= 0,IOPT= 1$
```

Note: Since IWT=2 and M=3 (three independent variables), five columns are required in the data matrix row, where in this case, the fifth (implicit) column is reread again as the dependent variable Y(I)

and is defined as the statistical weight in Anderson (1982, p. 14-15). Also note that LR=X(I,3) in the data matrix is increasing for increasing X(I,1), and LS=X(I,2) is fixed in this example; however, LS=X(I,2) and LR=X(I,3) can take on any value greater than zero for each observation, and are not required to be increasing with increasing X(I,1).

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 NLSBPBP 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 NLSBPBP !use $RUN [WANDERSON]NLSBPBP 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 NLSBPBP 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 NLSBPBP !use $RUN [WANDERSON]NLSBPBP 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 NLSBPBP 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 MM in Table 1, and $PARMS M and $INIT IOPT, respectively, are not double checked by program NLSBPBP. 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 (NLSBPBP) 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 IOPT 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 NLSBPBP also provides a summary listing of the final solution vector B and names at the end of the output file.

REFERENCES

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

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

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

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

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

Niwas, Sri, and Israil, M., 1986, Computation of apparent resistivities using an exponential approximation of kernel functions: Geophysics, V. 51, no. 8, p. 1594-1602.

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) VAX non-ANSI DO-ENDDO loops (non-statement number DO format).
(8) 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 NLSBPBP on a VAX system. (This data was generated from a corresponding forward program FWDBPBP using a known model, and with input as described for any FWD-program by Anderson, 1984.) The corresponding NLSBPBP output file (FOR016) is listed following FOR010. In addition, file FOR016.DAT was used to plot the final observed (OBS) and calculated (CAL) curves using an external plotter. The symbol "0" represents Y(I) in the plot, the solid line displays the final layered model solution, and the dashed line represents a curve drawn through the calculated (CAL) points.

FOR005

```
INVERSION TEST
$PARMS N= 21,K= 5,M=1,
IPRT=-1,SP= 3,IWT=2,IDER=0,
NITER= 40,
IP= 1,IB= 1,
BL=3*.001,2*.01,
BH=3*500000,2*50000,
B=3000,50,500, 100,400$
(T17,G16.8,T1,2G16.8)
$INIT MM= 3,LS= 150,LR= 100,IOPT= 0$
```

FOR010

```
  50.000000       3036.1270
  62.946274       3031.6597
  79.244667       3013.9705
  99.763130       2970.4839
  125.59435       2881.3792
  158.11392       2719.1548
  199.05363       2455.5281
  250.59369       2076.9805
  315.47876       1605.8534
  397.16425       1111.2283
  500.00018       685.33105
  629.46295       395.13245
  792.44690       247.57181
  997.63153       200.08124
  1255.9437       202.77417
  1581.1395       226.13190
  1990.5367       258.10400
  2505.9373       293.94037
  3154.7881       333.72064
  3971.6431       373.67569
  5000.0024       407.32465
```

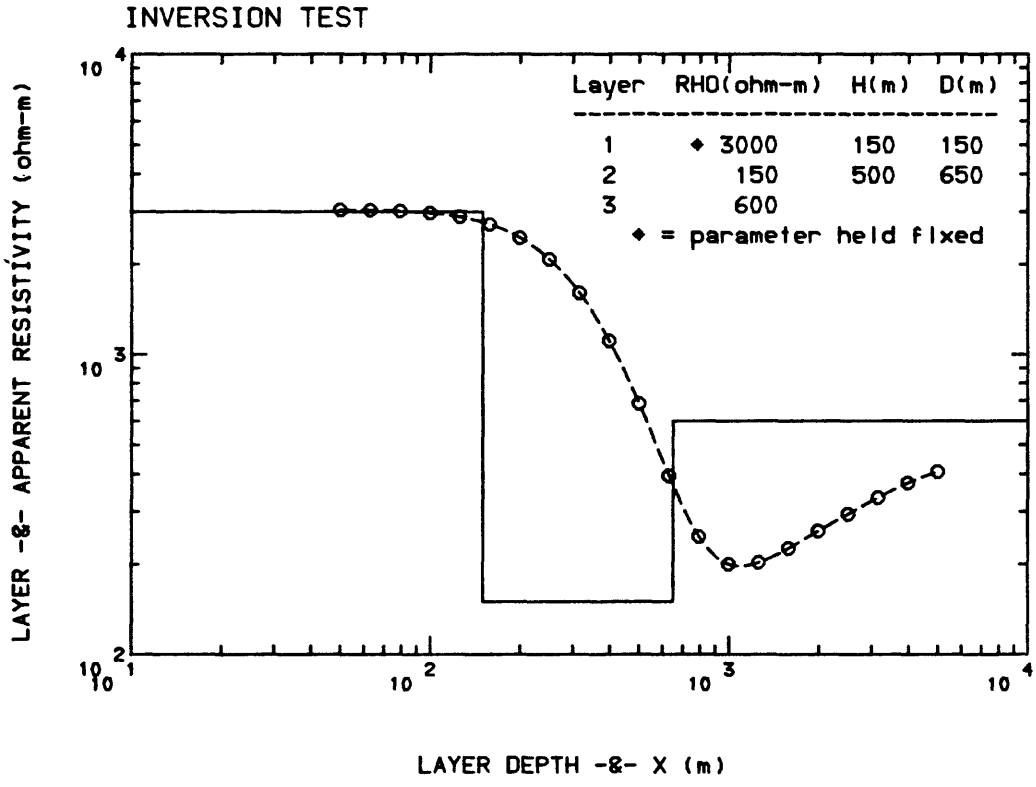
FOR016

```
  <NLSBPBP>:      INVERSION TEST

MM=    3                EPS=   0.99999999E-08
LS=    0.15000000E+03   LR=    0.10000000E+03
IOPT=  0


PARAMETER ORDER--

       1          RHO(   1)
       2          RHO(   2)
       3          RHO(   3)
       4        THICK(   1)
       5        THICK(   2)
```

```
{NLSOL}:        INVERSION TEST

N=        21    K=         5    IP=        1    M=         1    IALT=     10
ISTOP=     1    IWT=       2    IDER=      0    IPRT=     -1    NITER=    40
IOUT=      1    SP=        3

PARAMETERS HELD FIXED: IB=  1

FMT=(T17,G16.8,T1,2G16.8)


PARAMETER LOWER BOUNDS: BL=

 0.10000000E-02  0.10000000E-02  0.10000000E-02  0.99999998E-02  0.99999998E-02

INITIAL PARAMETERS: B=

 0.30000000E+04  0.50000000E+02  0.50000000E+03  0.10000000E+03  0.40000000E+03

PARAMETER HIGHER BOUNDS: BH=

 0.50000000E+06  0.50000000E+06  0.50000000E+06  0.50000000E+05  0.50000000E+05

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

REORDERED PARAMETERS:

 0.50000000E+02  0.50000000E+03  0.10000000E+03  0.40000000E+03

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


    I     INITIAL X(I)       D(I)

    1     0.100001E-01     0.394E+04
    2     0.316280E-01     0.348E+03
    3     0.447340E-01     0.548E+04
    4     0.895613E-01     0.401E+03

   IT    NF      F          DF         COSMAX       VAR

    0     1  0.227E+04                0.821E+00
    1     3  0.968E+03  0.130E+04  0.930E+00  0.235E+01
    2     4  0.617E+02  0.906E+03  0.937E+00  0.149E+02
    3     5  0.274E+02  0.343E+02  0.998E+00  0.170E+02
    4     6  0.218E+02  0.560E+01  0.912E+00  0.237E+02
    5     7  0.413E-01  0.218E+02  0.961E+00  0.170E+02
    6     8  0.175E-04  0.413E-01  0.162E+00  0.170E+02
    7     9  0.175E-04 -0.970E-04  0.162E+00  0.165E+01

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

FUNCTION     0.174640D-04  VARIABILITY   0.165244E+01
FUNC. EVALS        9       GRAD. EVALS        7
GRAD. NORM   0.426188E+01  COSMAX        0.162067E+00

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

    1     0.173210E-01     0.439E+04    -0.419E+01
    2     0.346426E-01     0.992E+03     0.158E+00
    3     0.547980E-01     0.573E+04     0.617E+00
```

```
    4     0.100165E+00      0.521E+03      0.438E+00

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

ROW  1     0.1481E-11
ROW  2     0.1646E-11  0.9024E-11
ROW  3    -0.3928E-12 -0.3863E-12  0.1672E-12
ROW  4     0.1361E-10  0.2671E-10 -0.3502E-11  0.1515E-09

  I     OBS.Y(I)        CAL            RES         %RES.ERR      X(I,1)        X(I,2)        X(I,3)        X(I,4)        WT(I)
  1   0.303613E+04  0.303613E+04 -0.122E-02 -0.402059E-04  0.500000E+02  0.303613E+04  0.000000E+00  0.000000E+00  0.329367E-03
  2   0.303166E+04  0.303166E+04 -0.220E-02 -0.724773E-04  0.629463E+02  0.303166E+04  0.000000E+00  0.000000E+00  0.329852E-03
  3   0.301397E+04  0.301397E+04 -0.195E-02 -0.648023E-04  0.792447E+02  0.301397E+04  0.000000E+00  0.000000E+00  0.331788E-03
  4   0.297048E+04  0.297049E+04 -0.171E-02 -0.575322E-04  0.997631E+02  0.297048E+04  0.000000E+00  0.000000E+00  0.336646E-03
  5   0.288138E+04  0.288138E+04 -0.464E-02 -0.160988E-03  0.125594E+03  0.288138E+04  0.000000E+00  0.000000E+00  0.347056E-03
  6   0.271915E+04  0.271916E+04 -0.488E-03 -0.179571E-04  0.158114E+03  0.271915E+04  0.000000E+00  0.000000E+00  0.367761E-03
  7   0.245553E+04  0.245554E+04 -0.708E-02 -0.288331E-03  0.199054E+03  0.245553E+04  0.000000E+00  0.000000E+00  0.407244E-03
  8   0.207698E+04  0.207700E+04 -0.151E-01 -0.728780E-03  0.250594E+03  0.207698E+04  0.000000E+00  0.000000E+00  0.481468E-03
  9   0.160585E+04  0.160587E+04 -0.131E-01 -0.813363E-03  0.315479E+03  0.160585E+04  0.000000E+00  0.000000E+00  0.622722E-03
 10   0.111123E+04  0.111122E+04  0.342E-02  0.307586E-03  0.397164E+03  0.111123E+04  0.000000E+00  0.000000E+00  0.899905E-03
 11   0.685331E+03  0.685352E+03 -0.204E-01 -0.298340E-02  0.500000E+03  0.685331E+03  0.000000E+00  0.000000E+00  0.145915E-02
 12   0.395132E+03  0.395109E+03  0.231E-01  0.583922E-02  0.629463E+03  0.395132E+03  0.000000E+00  0.000000E+00  0.253080E-02
 13   0.247572E+03  0.247557E+03  0.143E-01  0.578775E-02  0.792447E+03  0.247572E+03  0.000000E+00  0.000000E+00  0.403923E-02
 14   0.200081E+03  0.200106E+03 -0.244E-01 -0.121853E-01  0.997632E+03  0.200081E+03  0.000000E+00  0.000000E+00  0.499797E-02
 15   0.202774E+03  0.202734E+03  0.399E-01  0.196743E-01  0.125594E+04  0.202774E+03  0.000000E+00  0.000000E+00  0.493159E-02
 16   0.226132E+03  0.226146E+03 -0.144E-01 -0.635597E-02  0.158114E+04  0.226132E+03  0.000000E+00  0.000000E+00  0.442220E-02
 17   0.258104E+03  0.258049E+03  0.549E-01  0.212755E-01  0.199054E+04  0.258104E+03  0.000000E+00  0.000000E+00  0.387441E-02
 18   0.293940E+03  0.293927E+03  0.132E-01  0.450609E-02  0.250594E+04  0.293940E+03  0.000000E+00  0.000000E+00  0.340205E-02
 19   0.333721E+03  0.333772E+03 -0.513E-01 -0.153606E-01  0.315479E+04  0.333721E+03  0.000000E+00  0.000000E+00  0.299652E-02
 20   0.373676E+03  0.373676E+03  0.000E+00  0.000000E+00  0.397164E+04  0.373676E+03  0.000000E+00  0.000000E+00  0.267612E-02
 21   0.407325E+03  0.407325E+03  0.000E+00  0.000000E+00  0.500000E+04  0.407325E+03  0.000000E+00  0.000000E+00  0.245504E-02

** RMSERR=  0.24085879E-01  AVE|%RES.ERR|=  0.459620E-02

CORRELATION MATRIX
 2  0.1000E+01
 3  0.4502E+00  0.1000E+01
 4 -0.7893E+00 -0.3145E+00  0.1000E+01
 5  0.9086E+00  0.7223E+00 -0.6959E+00  0.1000E+01

  **PARM_SOL.    STD_ERROR    EEL_ERROR     % ERROR **

 2  0.1500E+03  0.1217E-05  0.7025E-04  0.7025E-02
 3  0.5998E+03  0.3004E-05  0.8672E-04  0.8672E-02
 4  0.1500E+03  0.4089E-06  0.7462E-05  0.7462E-03
 5  0.5000E+03  0.1231E-04  0.1229E-03  0.1229E-01


******** E N D ********      INVERSION TEST

PARAMETER NAME       FINAL SOLUTION        RESISTIVITY   LAYER DEPTH

   1 *    RHO( 1) =  0.30000000E+04    1  0.30000000E+04
   2      RHO( 2) =  0.14999440E+03    2  0.14999440E+03
   3      RHO( 3) =  0.59981726E+03    3  0.59981726E+03
   4    THICK( 1) =  0.15000073E+03                      1  0.15000073E+03
   5    THICK( 2) =  0.49998645E+03                      2  0.64998718E+03

     * FIXED
```

```
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TOTAL "ELAPSED" TIME=            43.13 SEC. (    0 MIN. 43.13 SEC.)
CPU_TIME=          13.07 SEC. (    0 M. 13.07 S.)    CPU % = 30.30%
BUF.I/O_COUNT=       110
DIR.I/O_COUNT=         7
PAGE_FAULTS=         521
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```


INVERSION TEST
Layer RHO(ohm-m) H(m) D(m)
1 ◆ 3000 150 150
2 150 500 650
3 600
◆ = parameter held fixed
LAYER -&- APPARENT RESISTIVITY (ohm-m)
LAYER DEPTH -&- X (m)

Appendix 3.-- Source code availability and listing

Source Code Availability

The current version of the source code may be obtained by writing directly to the author*, 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 "NLSBPBP.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 <NLSBPBP>: INVERSION OF BIPOLE-BIPOLE DC SOUNDINGS                    00000010
C  BY LAGGED-CONVOLUTION.                                               00000020
C                                                                       00000030
C** VAX-11/780 VERSION:  11/14/86 **                                    00000040
C                                                                       00000050
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.           00000060
C                                                                       00000070
C-----------------------------------------------------------------      00000080
C                                                                       00000090
C--REFERENCES:                                                          00000100
C                                                                       00000110
C  ANDERSON, W.L, 1982, ADAPTIVE NONLINEAR LEAST-SQUARES SOLUTION       00000120
C    FOR CONSTRAINED OR UNCONSTRAINED MINIMIZATION PROBLEMS             00000130
C    (SUBPROGRAM NLSOL): USGS OPEN-FILE REPT. 82-68, 65 P.              00000140
C                                                                       00000150
C  -------, 1984, A GENERAL INTERFACE FOR PRODUCING FORWARD              00000160
C    SOLUTION PROGRAMS (SUBPROGRAM FWDSOL): USGS OPEN-FILE REPT.        00000170
C    84-348, 43 P.                                                      00000180
C                                                                       00000190
C-----------------------------------------------------------------      00000200
C                                                                       00000210
      EXTERNAL FCODE,SUBZ,PCODE,RHOSUBEND                               00000220
      CALL SETTIME                                                      00000230
      CALL NLSOL2(FCODE,PCODE,SUBZ,RHOSUBEND)                           00000240
      CALL CPUTIME(6,16)                                                00000250
      CALL EXIT                                                         00000260
      END                                                               00000270
      SUBROUTINE FCODE(Y,X,B,PRNT,F,IN,IDER)                            00000280
C--FUNCT. EVAL. FOR 'NLSBPBP' USING FAST LAGGED-CONVOLUTION.            00000290
C                                                                       00000300
C--PARAMETERS--                                                         00000310
C                                                                       00000320
C       Y=      OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)              00000330
C       X=      OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)          00000340
C       B=      CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)              00000350
C       PRNT=   WORK AND PRINT ARRAY (DIM. 5)                           00000360
C       F=      OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. IN  00000370
C       IN=     OBSERVATION NO. TO EVAL. F (1<=IN<=N)                   00000380
C       IDER=   0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED) 00000390
C               1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED) 00000400
C                                                                       00000410
```

```
      REAL Y(1),X(500,5),B(1),PRNT(5),RHO(10),H(9),BR(20), LS,LR        00000420
      EXTERNAL RKERN                                                    00000430
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                               00000440
      COMMON/RPASS/RS(500), LS,LR,RO,RM,NN,IFIRST,IOPT                  00000450
      IF(IN.GT.1.OR.MM.EQ.1) GO TO 20                                   00000460
      DO 10 J=2,MM                                                      00000470
      IF(B(J).EQ.B(J-1))CALL ERRMSG('SOME RHO(J)=RHO(J-1)',4,6,16)      00000480
10    CONTINUE                                                          00000490
20    DO 30 J=1,5                                                       00000500
30    PRNT(J)=X(IN,J)                                                   00000510
      IF(IN.GT.1) GO TO 800                                             00000520
      IF(IDER.EQ.1) GO TO 8001                                          00000530
35    IF(MM.EQ.1) GO TO 45                                              00000540
      DO 40 J=1,M1                                                      00000550
      RHO(J)=B(J)                                                       00000560
40    H(J)=B(J+MM)                                                      00000570
45    RHO(MM)=B(MM)                                                     00000580
C--GET LAGGED-CONVOLUTION RHOA-FUNCTION (ONLY WHEN IN=1 OR IDER=1)      00000590
      NEW=1                                                             00000600
      DO 50 I=1,NN                                                      00000610
      IF(IOPT.EQ.1) THEN                                                00000620
        LS=X(I,2)                                                       00000630
        LR=X(I,3)                                                       00000640
      ENDIF                                                             00000650
      R1=X(I,1)+LS                                                      00000660
      R2=R1+LR                                                          00000670
      R3=X(I,1)                                                         00000680
      R4=X(I,1)+LR                                                      00000690
      TERM1=R1*RHLAG0(RKERN,EPS,RO,RM,ALOG(R1),NEW)+1.0                 00000700
      NEW=0                                                             00000710
      TERM2=R2*RHLAG0(RKERN,EPS,RO,RM,ALOG(R2),NEW)+1.0                 00000720
      TERM3=R3*RHLAG0(RKERN,EPS,RO,RM,ALOG(R3),NEW)+1.0                 00000730
      TERM4=R4*RHLAG0(RKERN,EPS,RO,RM,ALOG(R4),NEW)+1.0                 00000740
      RS(I)=RHO(1)*ABS((TERM1/R1-TERM2/R2-TERM3/R3+TERM4/R4)/           00000750
     1   (1./R1-1./R2-1./R3+1./R4))                                     00000760
50    CONTINUE                                                          00000770
      IF(IDER.EQ.0) GO TO 600                                           00000780
      DO 60 J=1,M21                                                     00000790
60    BR(J)=B(J)                                                        00000800
      IFIRST=0                                                          00000810
C--GET PRE-SPLINED SOUNDING                                             00000820
600   F=RS(IN)                                                          00000830
      RETURN                                                            00000840
800   IF(IDER.EQ.0) GO TO 600                                           00000850
C--IDER=1 EST.DER.OPTION                                                00000860
8001  IF(IFIRST.EQ.1) GO TO 35                                          00000870
      DO 8002 J=1,M21                                                   00000880
        IF(B(J).NE.BR(J)) GO TO 35                                      00000890
8002  CONTINUE                                                          00000900
      GO TO 600                                                         00000910
      END                                                               00000920
      SUBROUTINE PCODE(P,X,B,PRNT,F,IN,IP,IB)                           00000930
C--ANALYTIC PARTIALS FOR 'NLSBPBP' USING FAST LAGGED-CONVOLUTION        00000940
C                                                                       00000950
```

```
C  (PCODE ONLY CALLED IF IDER=0--DEFAULT)                                  00000960
C                                                                          00000970
C--PARAMETERS--                                                            00000980
C                                                                          00000990
C       P=       OUTPUT PARTIAL DERIVATIVE ARRAY (DIM. K)                  00001000
C                EVALUATED FOR GIVEN X(IN,),B(K) AT OBS. IN                00001010
C       X=       OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)            00001020
C       B=       CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)                00001030
C       PRNT=    WORK AND PRINT ARRAY (DIM. 5)                             00001040
C       F=       LAST FUNCTION VALUE FROM FCODE AT GIVEN IN.               00001050
C                F MAY OR MAY NOT BE NEEDED--BUT AVAILABLE ANYWAY.         00001060
C       IN=      OBSERVATION NO. TO EVAL. P ARRAY (1<=IN<=N)               00001070
C       IP=      NO. PARAMETERS HELD FIXED (IF ANY--IF NONE IP=0).         00001080
C       IB=      ARRAY OF PARAMETER INDICES HELD FIXED IF IP.GT.0          00001090
C                (DIM. 19).                                                00001100
C                                                                          00001110
      INTEGER IB(1)                                                        00001120
      REAL P(1),X(500,5),B(1),PRNT(5),RHO(10),H(9),PM(500,20),             00001130
     1 LS,LR                                                               00001140
      EXTERNAL RFVP                                                        00001150
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                                  00001160
      COMMON/RPASS/RS(500), LS,LR,R0,RM,NN,IFIRST,IOPT                     00001170
      IF(IN.GT.1) GO TO 50                                                 00001180
      DO 30 J=1,M21                                                        00001190
      JJ=J                                                                 00001200
      IFIX=0                                                               00001210
      IF(IP.LE.0) GO TO 11                                                 00001220
      DO 1 I=1,IP                                                          00001230
      IF(IB(I).EQ.J) IFIX=1                                                00001240
1     CONTINUE                                                             00001250
      IF(IFIX.EQ.1) GO TO 6                                                00001260
C--GET LAGGED-CONVOLUTION PARTIALS OF RHOA-FUNCTION (ONLY WHEN IN=1)       00001270
11    NEW=1                                                                00001280
      DO 5 I=1,NN                                                          00001290
      IF(IOPT.EQ.1) THEN                                                   00001300
        LS=X(I,2)                                                          00001310
        LR=X(I,3)                                                          00001320
      ENDIF                                                                00001330
      R1=X(I,1)+LS                                                         00001340
      R2=R1+LR                                                             00001350
      R3=X(I,1)                                                            00001360
      R4=X(I,1)+LR                                                         00001370
      TERM1=R1*RHLAG0(RFVP,EPS,R0,RM,ALOG(R1),NEW)                         00001380
      NEW=0                                                                00001390
      TERM2=R2*RHLAG0(RFVP,EPS,R0,RM,ALOG(R2),NEW)                         00001400
      TERM3=R3*RHLAG0(RFVP,EPS,R0,RM,ALOG(R3),NEW)                         00001410
      TERM4=R4*RHLAG0(RFVP,EPS,R0,RM,ALOG(R4),NEW)                         00001420
      PP=RHO(1)*(TERM1/R1-TERM2/R2-TERM3/R3+TERM4/R4)/                     00001430
     1   (1./R1-1./R2-1./R3+1./R4)                                         00001440
      IF(J.EQ.1) PP=PP+RS(I)/RHO(1)                                        00001450
        PM(I,J)=PP                                                         00001460
5     CONTINUE                                                             00001470
      GO TO 30                                                             00001480
6     DO 7 I=1,NN                                                          00001490
```

```
7      PM(I,J)=0.0                                                      00001500
30     CONTINUE                                                         00001510
C--GET PRE-SPLINED PARTIALS                                             00001520
50     DO 60 J=1,M21                                                    00001530
60     P(J)=PM(IN,J)                                                    00001540
       RETURN                                                           00001550
       END                                                              00001560
       REAL FUNCTION RFVP(X)                                            00001570
C--RESISTIVITY KERNEL USED IN INTEGRAL OF PARTIAL RHOA W/R B(JJ),        00001580
C  JJ=1,2*MM-1 GIVEN IN COMMON/RESIS/.                                  00001590
C  (RFVP BY RECURRENCE METHOD).                                         00001600
C                                                                       00001610
       REAL RHO(10),H(9),K1                                             00001620
       COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                              00001630
       X2=-2.0*X                                                        00001640
       JJMM=JJ-MM                                                       00001650
       VM=1.0                                                           00001660
       PV1=0.0                                                          00001670
       IF(MM.EQ.1) GO TO 40                                             00001680
C--INITIALIZE PARTIAL INDEX J1=MM-1 (NUM. INDEX)                        00001690
       J=MM                                                             00001700
C--LOOP ON J1 INDEX                                                     00001710
   10  J1=J-1                                                           00001720
       E=X2*H(J1)                                                       00001730
       E1=0.0                                                           00001740
       IF(E.GT.-88.028) E1=EXP(E)                                       00001750
       DENK1=1.0/(RHO(J1)+RHO(J)*VM)                                    00001760
       K1=DENK1*(RHO(J1)-RHO(J)*VM)                                     00001770
       DENV1=1.0/(1.0+K1*E1)                                            00001780
       V1=DENV1*(1.0-K1*E1)                                             00001790
       IF(JJ.LE.MM) GO TO 20                                            00001800
C--RECUR FOR PARTIAL W/R H(JJ)                                          00001810
       PEH=0.0                                                          00001820
       IF(JJMM.EQ.J1) PEH=X2*E1                                         00001830
       PKH=-DENK1*RHO(J)*PV1*(1.0+K1)                                   00001840
       PV1=-DENV1*(K1*PEH+E1*PKH)*(1.0+V1)                              00001850
       GO TO 30                                                         00001860
C--RECUR FOR PARTIAL W/R RHO(JJ)                                        00001870
   20  PR1=0.0                                                          00001880
       IF(JJ.EQ.J1) PR1=1.0                                             00001890
       PRM=0.0                                                          00001900
       IF(JJ.EQ.J) PRM=1.0                                              00001910
       PKR=DENK1*(PR1*(1.0-K1)-(1.0+K1)*(RHO(J)*PV1+VM*PRM))            00001920
       PV1=-DENV1*E1*PKR*(1.0+V1)                                       00001930
   30  IF(J.LE.2) GO TO 40                                              00001940
       VM=V1                                                            00001950
       J=J1                                                             00001960
       GO TO 10                                                         00001970
   40  RFVP=PV1                                                         00001980
       RETURN                                                           00001990
       END                                                              00002000
       SUBROUTINE SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)                   00002010
C-- INITIALIZATION ROUTINE (CALLED ONCE)                                00002020
C                                                                       00002030
```

```
C  SUBZ IS CALLED AFTER THE DATA Y(I),X(I,5) ARE READ.                  00002040
C  SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL $INIT                   00002050
C  PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE...             00002060
C                                                                        00002070
C--PARAMETERS--                                                          00002080
C                                                                        00002090
C       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.                          00002100
C       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY              00002110
C               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES               00002120
C               PRINTED BY PGM MARQRT...                                 00002130
C       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)                    00002140
C       TITLE=  ALPHA TITLE ARRAY READ IN BY PGM MARQRT.                 00002150
C       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED                      00002160
C               0 IF ONLY UNIT 6 PRINT FILE USED.                        00002170
C                                                                        00002180
      CHARACTER*80 TITLE                                                 00002190
      REAL Y(1),X(500,5),B(1),PRNT(1),RHO(10),H(9), LS,LR                00002200
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                                00002210
      COMMON/RPASS/RS(500), LS,LR,R0,RM,NN,IFIRST,IOPT                   00002220
      NAMELIST/INIT/MM,EPS,LS,LR,IOPT                                    00002230
      DATA ISUBZ/0/                                                      00002240
      IF(ISUBZ.NE.0) GO TO 10                                            00002250
C--PRESET                                                                00002260
      ISUBZ=1                                                            00002270
      MM=1                                                               00002280
      EPS=.1E-7                                                          00002290
      IOPT=0                                                             00002300
      LS=0.0                                                             00002310
      LR=0.0                                                             00002320
10    READ(99,INIT,END=11)                                               00002330
C//10 CALL NAMELIST(5,'$INIT',*11)                                       00002340
11    CALL NONBLANK(TITLE,NB)                                            00002350
      WRITE(6,20) TITLE,MM,EPS,LS,LR,IOPT                                00002360
20    FORMAT('1<NLSBPBP>:',5X,A<NB>//' MM=',I4,13X,'  EPS=',E16.8/       00002370
     1 ' LS=',E17.8,'  LR=',E17.8/' IOPT=',I2)                           00002380
      IF(IOUT.EQ.1) WRITE(16,20) TITLE,MM,EPS,LS,LR,IOPT                 00002390
C--TEST $INIT PARMS                                                      00002400
      IF(MM.LT.1.OR.MM.GT.10)CALL ERRMSG('MM<1 OR >10    ',3,6,16)       00002410
      IF(IOPT.EQ.0) THEN                                                 00002420
        IF(LS.LE.0.0)CALL ERRMSG('LS<=0.0 AND IOPT=0',0,6,16)            00002430
        IF(LR.LE.0.0)CALL ERRMSG('LR<=0.0 AND IOPT=0',0,6,16)            00002440
      ELSE IF(IOPT.EQ.1) THEN                                            00002450
        SUMMAX=0.0                                                       00002460
        DO I=1,N                                                         00002470
          IF(X(I,2).LE.0.0)CALL ERRMSG(                                  00002480
     1    'SOME LS=X(I,2)<=0.0 AND IOPT=1',0,6,16)                       00002490
          IF(X(I,3).LE.0.0)CALL ERRMSG(                                  00002500
     1    'SOME LR=X(I,3)<=0.0 AND IOPT=1',0,6,16)                       00002510
          SUM=X(I,2)+X(I,3)                                              00002520
          IF(SUM.GT.SUMMAX) SUMMAX=SUM                                   00002530
        ENDDO                                                            00002540
      ELSE                                                               00002550
        CALL ERRMSG('IOPT<0 OR >1',0,6,16)                               00002560
      ENDIF                                                              00002570
```

```
C--TEST X(I,1) DATA BEFORE PROCEEDING                                   00002580
      IF(X(1,1).LE.0.0)CALL ERRMSG('X(1,1)<=0.',2,6,16)                 00002590
      DO 40 I=2,N                                                       00002600
      IF(X(I,1).LE.X(I-1,1).OR.X(I,1).LE.0.0)                           00002610
     * CALL ERRMSG('SOME X(I,1)<=0.0 OR NOT INCREASING.',7,6,16)         00002620
40    CONTINUE                                                          00002630
C--PRESET SOME GLOBAL CONSTANTS                                         00002640
      IFIRST=1                                                          00002650
      NPRNT=2                                                           00002660
      NN=N                                                              00002670
      R0=.5*X(1,1)      !! GLOBAL RMIN FOR LAGGED CONVOLUTION           00002680
      IF(IOPT.EQ.0) THEN                                                00002690
          RM=X(N,1)+LS+LR  !! GLOBAL RMAX FOR LAGGED CONVOLUTION        00002700
      ELSE                                                              00002710
        RM=X(N,1)+SUMMAX                                                00002720
      ENDIF                                                             00002730
      M1=MM-1                                                           00002740
      M21=2*MM-1                                                        00002750
      WRITE(6,60) (I,I,I=1,MM)                                          00002760
      IF(IOUT.EQ.1) WRITE(16,60) (I,I,I=1,MM)                           00002770
60    FORMAT(////' PARAMETER ORDER--'//(5X,I3,6X,'  RHO(',I3,')'))      00002780
      IF(MM.EQ.1) GO TO 90                                              00002790
      DO 70 I=1,M1                                                      00002800
      J=MM+I                                                            00002810
      IF(IOUT.EQ.1) WRITE(16,80) J,I                                    00002820
70    WRITE(6,80) J,I                                                   00002830
80    FORMAT(5X,I3,6X,'THICK(',I3,')')                                  00002840
90    RETURN                                                            00002850
      END                                                               00002860
      REAL FUNCTION RKERN(X)                                            00002870
C--KERNEL FUNCTION USED IN FCODE INTEGRAL                               00002880
C  FOR BIPOLE-BIPOLE APPARENT RESISTIVITY                               00002890
C                                                                       00002900
      REAL RHO(10),H(9)                                                 00002910
      COMMON/RESIS/RHO,H,EPS,MM,M1,M21,JJ                               00002920
      X2=-2.0*X                                                         00002930
      V=1.0                                                             00002940
      IF(MM.LE.1) GO TO 30                                              00002950
      I=MM                                                              00002960
   10 I1=I-1                                                            00002970
      T=V/RHO(I1)                                                       00002980
      TR=T*RHO(I)                                                       00002990
      E=X2*H(I1)                                                        00003000
      IF(E.LT.-88.028) GO TO 40                                         00003010
      T=((1.0-TR)/(1.0+TR))*EXP(E)                                      00003020
      V=(1.0-T)/(1.0+T)                                                 00003030
   20 IF(I.LE.2) GO TO 30                                               00003040
      I=I-1                                                             00003050
      GO TO 10                                                          00003060
   30 RKERN=V-1.0                                                       00003070
      RETURN                                                            00003080
   40 V=1.0                                                             00003090
      GO TO 20                                                          00003100
      END                                                               00003110
```

```
      SUBROUTINE RHOSUBEND(Y,X,B,K,N,TITLE,IOUT)                        00003120
C**GENERAL SUBEND TERMINATION ROUTINE WITH 'RHO' NAMES.                 00003130
C  ALSO GIVES RESTART $PARMS ON UNIT=4 AS 'FOR005.TMP'                  00003140
C                                                                       00003150
      CHARACTER*132 LINE                                                00003160
      CHARACTER*80 TITLE                                                00003170
      CHARACTER*1 FLAG(19)                                              00003180
      COMMON/FIXDAT/DUM(3020),IB(19),IP,IDUM(3)                         00003190
      REAL Y(1),X(500,5),B(1)                                           00003200
      DO I=1,K                                                          00003210
        FLAG(I)=' '                                                     00003220
        IF(IP.GT.0) THEN                                                00003230
          DO J=1,IP                                                     00003240
            IF(IB(J).NE.0) FLAG(IB(J))='*'                              00003250
          ENDDO                                                         00003260
        ENDIF                                                           00003270
      ENDDO                                                             00003280
      CALL NONBLANK(TITLE,NB)                                           00003290
      WRITE(6,10) TITLE                                                 00003300
10    FORMAT(//' ********  E N D  ********',5X,A<NB>//                  00003310
     1 ' PARAMETER NAME',6X,'FINAL SOLUTION',8X,                        00003320
     2 'RESISTIVITY   LAYER DEPTH'/)                                    00003330
      IF(IOUT.EQ.1) WRITE(16,10) TITLE                                  00003340
      MM=(K+1)/2                                                        00003350
      DO 30 I=1,MM                                                      00003360
      WRITE(6,20) I,FLAG(I),I,B(I),I,B(I)                               00003370
20    FORMAT(2X,I3,1X,A1,3X,'RHO(',I2,') =',E16.8,2X,I2,E16.8)          00003380
      IF(IOUT.EQ.1) WRITE(16,20) I,FLAG(I),I,B(I),I,B(I)                00003390
30    CONTINUE                                                          00003400
      IF(K.LE.1) GO TO 60                                               00003410
      M2=MM+1                                                           00003420
      D=0.0                                                             00003430
      DO 50 I=M2,K                                                      00003440
      D=D+B(I)                                                          00003450
      L=I-MM                                                            00003460
      WRITE(6,40) I,FLAG(I),L,B(I),L,D                                  00003470
40    FORMAT(2X,I3,1X,A1,1X,'THICK(',I2,') =',E16.8,22X,I2,E16.8)       00003480
      IF(IOUT.EQ.1) WRITE(16,40) I,FLAG(I),L,B(I),L,D                   00003490
50    CONTINUE                                                          00003500
C** GENERATE RESTART $PARMS ON FOR005.TMP                               00003510
60    REWIND 5                                                          00003520
      OPEN(UNIT=4,FILE='FOR005.TMP',STATUS='NEW',                       00003530
     1 CARRIAGECONTROL='LIST')                                          00003540
      READ(5,65,END=999) LINE                                           00003550
65    FORMAT(A)                                                         00003560
      CALL NONBLANK(LINE,NB)                                            00003570
      WRITE(4,66) LINE                                                  00003580
66    FORMAT(A<NB>)                                                     00003590
      IDOL=0                                                            00003600
70    READ(5,65,END=999) LINE                                           00003610
      I=INDEX(LINE,'$')                                                 00003620
      IF(I.NE.0) THEN                                                   00003630
        IF(IDOL.EQ.0) THEN                                              00003640
          IDOL=1                                                        00003650
```

```
          J=INDEX(LINE(I+1:),'$')                                        00003660
          IF(J.NE.0) THEN                                                00003670
            IDOL=2                                                       00003680
            LINE(J:J)=','                                                00003690
          ENDIF                                                          00003700
        ELSE                                                             00003710
          IDOL=2                                                         00003720
          LINE(I:I)=','                                                  00003730
        ENDIF                                                            00003740
      ENDIF                                                              00003750
      CALL NONBLANK(LINE,NB)                                             00003760
      WRITE(4,66) LINE                                                   00003770
      IF(IDOL.LT.2) GO TO 70                                             00003780
      LINE(1:)='B='                                                      00003790
      DO 80 I=1,K                                                        00003800
      ENCODE(16,90,LINE(3:18)) B(I)                                      00003810
90    FORMAT(G16.8)                                                      00003820
      IF(I.LT.K) THEN                                                    00003830
        LINE(19:19)=','                                                  00003840
      ELSE                                                               00003850
        LINE(19:19)='$'                                                  00003860
      ENDIF                                                              00003870
      CALL NONBLANK(LINE,NB)                                             00003880
      WRITE(4,66) LINE                                                   00003890
      LINE(1:2)='  '                                                     00003900
80    CONTINUE                                                           00003910
100   READ(5,65,END=999) LINE                                            00003920
      CALL NONBLANK(LINE,NB)                                             00003930
      IF(NB.EQ.0) GO TO 100                                              00003940
      WRITE(4,66) LINE                                                   00003950
      GO TO 100                                                          00003960
999   IF(IP.GT.0) THEN                                                   00003970
        WRITE(6,110)                                                     00003980
110     FORMAT(/6X,'* FIXED')                                            00003990
        IF(IOUT.EQ.1) WRITE(16,110)                                      00004000
      ENDIF                                                              00004010
      RETURN                                                             00004020
      END                                                                00004030
      SUBROUTINE CPUTIME(I1,I2)                                          00004040
C                                                                        00004050
C  CPUTIME WRITES "ELAPSED & CPU" TIME FROM PREVIOUS "CALL SETTIME" ON   00004060
C  FORTRAN UNITS I1 (IF NOT 0) AND I2 (IF NOT 0).                        00004070
C                                                                        00004080
C  WILL EJECT FIRST IF I1>0 (OR I2>0).                                   00004090
C  DOUBLE SPACE FIRST IF I1<0 (OR I2<0).                                 00004100
C                                                                        00004110
C  E.G., USE TO TIME ELAPSED & CPU TIME FOR PROGRAM OR CODE SEGMENTS AS: 00004120
C                                                                        00004130
C     CALL SETTIME    ! DON'T FORGET TO DO THIS!                         00004140
C     >>>>> THE CODE TO TIME IS HERE <<<<<  ! USUALLY A COMPLETE PROGRAM 00004150
C     CALL CPUTIME(-6,16) ! OR USE I1 OR I2=0 TO OMIT WRITE.             00004160
C     >>>>> ALSO CAN USE CALL GETTIME(CPU) TO GET JUST THE CPU (SEC)     00004170
C           SINCE THE LAST CALL SETTIME WAS DONE.                        00004180
C                                                                        00004190
```

```
      SAVE                                                              00004200
      INTEGER*4 ABSVAL(4),INCRVAL(4)                                    00004210
      CALL PROCINFO(ABSVAL,INCRVAL)                                     00004220
      TIMES=SECNDS(TIME0)                                               00004230
      MIN=TIMES/60.0                                                    00004240
      SEC=AMOD(TIMES,60.0)                                              00004250
      CPUSEC=INCRVAL(1)*.01                                             00004260
      IMIN=CPUSEC/60.0                                                  00004270
      CSEC=AMOD(CPUSEC,60.0)                                            00004280
      PCPU=100.*(CPUSEC/TIMES)                                          00004290
      IF(I1.NE.0) THEN                                                  00004300
        IF(I1.GT.0) THEN                                                00004310
          J=1                                                           00004320
        ELSE                                                            00004330
          J=0                                                           00004340
        ENDIF                                                           00004350
      WRITE(IABS(I1),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,         00004360
     1 (INCRVAL(I),I=2,4)                                               00004370
60    FORMAT(I1,65('$')/' TOTAL "ELAPSED" TIME=',F16.2,' SEC. (',       00004380
     1 I4,' MIN.',F6.2,' SEC.)'/                                        00004390
     2 ' CPU_TIME=',F15.2,' SEC. (',I4,' M. ',F5.2,                     00004400
     1 ' S.)   CPU % =',F6.2,'%'/                                       00004410
     3 ' BUF.I/O_COUNT=',I10/                                           00004420
     4 ' DIR.I/O_COUNT=',I10/                                           00004430
     5 ' PAGE_FAULTS=',2X,I10/                                          00004440
     6 ' ',65('$')//)                                                   00004450
      ENDIF                                                             00004460
      IF(I2.NE.0) THEN                                                  00004470
        IF(I2.GT.0) THEN                                                00004480
          J=1                                                           00004490
        ELSE                                                            00004500
          J=0                                                           00004510
        ENDIF                                                           00004520
      WRITE(IABS(I2),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,         00004530
     1 (INCRVAL(I),I=2,4)                                               00004540
      ENDIF                                                             00004550
      RETURN                                                            00004560
C** ENTRY 'CALL SETTIME'--MUST BE DONE BEFORE 'CALL CPUTIME(I1,I2)'     00004570
      ENTRY SETTIME()                                                   00004580
      TIME0=SECNDS(0.0)                                                 00004590
      CALL PROCINFO(ABSVAL,INCRVAL)                                     00004600
      RETURN                                                            00004610
C** ENTRY 'CALL GETTIME(CPU)'--TO GET CPU(SEC) SINCE LAST CALL SETTIME  00004620
      ENTRY GETTIME(CPU)                                                00004630
      CALL PROCINFO(ABSVAL,INCRVAL)                                     00004640
      CPU=INCRVAL(1)*.01                                                00004650
      RETURN                                                            00004660
      END                                                               00004670
      SUBROUTINE ERRMSG(MSG,ISKIP,IUNIT1,IUNIT2)                        00004680
C                                                                       00004690
C  GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780                  00004700
C                                                                       00004710
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                                  00004720
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2         00004730
```

```
C          > 0 FOR ONE BLANK LINE BEFORE.                                00004740
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).      00004750
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).      00004760
C                                                                        00004770
C  MESSAGES ARE WRITTEN IN THE FORM:                                     00004780
C                                                                        00004790
C  {ERRMSG}: _MSG_HERE_                                                  00004800
C                                                                        00004810
      CHARACTER*(*) MSG                                                 00004820
      I=LEN(MSG)                                                        00004830
      DO 1 J=1,2                                                        00004840
        IF(J.EQ.1) THEN                                                 00004850
       JUNIT=IUNIT1                                                     00004860
          ELSE                                                          00004870
       JUNIT=IUNIT2                                                     00004880
        ENDIF                                                           00004890
        IF(JUNIT.GT.0) THEN                                             00004900
          IF(ISKIP.EQ.0) THEN                                           00004910
            WRITE(JUNIT,2) MSG                                          00004920
            ELSE                                                        00004930
       WRITE(JUNIT,3) MSG                                               00004940
          ENDIF                                                         00004950
      ENDIF                                                             00004960
1     CONTINUE                                                          00004970
      CALL EXIT                                                         00004980
2     FORMAT(1X,'{ERRMSG}: ',A<I>)                                      00004990
3     FORMAT(/1X,'{ERRMSG}: ',A<I>)                                     00005000
      END                                                               00005010
      SUBROUTINE NLSOL2(FCODE,PCODE,SUBZ,SUBEND)                        00005020
C                                                                       00005030
C>> NLSOL2 IS A REVISED VERSION OF NLSOL WITHOUT CALL NAMELIST;         00005040
C   I.E., NLSOL2 USES THE CURRENT VAX-11/780 NAMELIST VERSION.  NOTE    00005050
C   SUBZ MUST BE CHANGED TO READ(99,INIT) FROM CALL NAMELIST(5,'$INIT'),00005060
C   WHERE CALL PRENAM(5,99) IS USED TO ENSURE UNIT=5 NAMELIST IS IN     00005070
C   PROPER FORMAT ON SCRATCH UNIT=99 (FOR099 DELETED ON RETURN TO VMS). 00005080
C                                                                       00005090
C {NLSOL2}: GENERAL NONLINEAR LEAST-SQUARES SOLUTION   {2/19/86}        00005100
C           USING DENNIS ET AL (1979; SEE REF1 BELOW)    -------        00005110
C           ADAPTIVE NONLINEAR LEAST-SQUARES ALGORITHM.                 00005120
C                                                                       00005130
C** THIS IS AN INTERFACE ROUTINE WRITTEN FOR THE VAX-11/780 BY          00005140
C   W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.              00005150
C                                                                       00005160
C** THIS INTERFACE (NLSOL) HAS ADDITIONAL OPTIONS (BESIDE REF1) TO:     00005170
C   (1) PERFORM EITHER UNCONSTRAINED OR UP TO 4-TYPES OF CONSTRAINED    00005180
C       ADAPTIVE NONLINEAR REGRESSION FOR ARBITRARY NONLINEAR PROBLEMS. 00005190
C       (I.E., PARTIAL OR FULL LOWER/HIGHER PARAMETER BOUNDS, ETC.)     00005200
C   (2) HOLDING CERTAIN PARAMETERS FIXED (I.E., AS CONSTANTS) IN THE    00005210
C       LEAST-SQUARES (THIS IS ANOTHER FORM OF CONSTRAINING SOLUTION    00005220
C       SPACE).                                                         00005230
C   (3) PROVIDE FOR WEIGHTED OBSERVATIONS (I.E., WEIGHTED LEAST-SQUARES)00005240
C   (4) OBJECT (RUN)-TIME CONTROL OF READING THE DATA MATRIX, PLUS      00005250
C       MANY OTHER I/O OPTIONS, ETC.                                    00005260
C   (5) OPTIONALLY, ONE CAN USE EITHER ESTIMATED PARTIAL DERIVATIVES, OR00005270
```

```
C       ANALYTICAL PARTIAL DERIVATIVES (IF SUBROUTINE PCODE AVAILABLE). 00005280
C                                                                       00005290
C** THE USER ONLY NEEDS TO WRITE SUBROUTINES FCODE, PCODE, SUBZ, AND    00005300
C   SUBEND (SEE DETAILS BELOW) EXACTLY AS USED IN SUBROUTINE 'MARQRT'   00005310
C   (SEE REF2) OR 'IMSLMQ' (SEE REF3).  ALSO, THE SAME PARAMETER FILE   00005320
C   FOR005 AND OBJECT (RUN)-TIME DATA MATRIX FILE FOR010 AS USED BY     00005330
C   EITHER MARQRT OR IMSLMQ MAY BE USED IN 'NLSOL'.                     00005340
C                                                                       00005350
C** NLSOL CALLS NLITR WHICH CALLS 'NL2ITR' AS PUBLISHED BY DENNIS ET AL,00005360
C   (SEE REF1, P. 38), OR 'NL2SNO' (SEE REF1, P. 35).                   00005370
C                                                                       00005380
C** REF1:  DENNIS, J.E., ET AL, 1979, AN ADAPTIVE NONLINEAR LEAST-      00005390
C          SQUARES ALGORITHM, NTIS REPORT AD-A079-716.                  00005400
C                                                                       00005410
C   REF2:  ANDERSON, W.L., 1980, PROGRAM MARQHXY: INVERSION OF HX AND HY00005420
C          FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN-  00005430
C          FILE REPT. 80-901.                                           00005440
C                                                                       00005450
C   REF3:  ANDERSON, W.L., 1980, PROGRAM IMSLEXY: INVERSION OF EX AND EY00005460
C          FREQUENCY SOUNDINGS FROM A GROUNDED WIRE SOURCE, USGS OPEN-  00005470
C          FILE REPT. 80-1073.                                          00005480
C                                                                       00005490
C***********************************************************************00005500
C                                                                       00005510
C****  THE USER MUST DECLARE THE CALLING PARAMETERS AS EXTERNAL IN THE  00005520
C      CALLING PROGRAM (ANY DESIRED NAMES MAY BE USED).                 00005530
C E.G.,                                                                 00005540
C                                                                       00005550
C [MAIN]:                                                               00005560
C     EXTERNAL MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND                      00005570
C     CALL NLSOL2(MY_FCODE,MY_PCODE,MY_SUBZ,MY_SUBEND)                  00005580
C     STOP   !<OR USE>: CALL EXIT                                       00005590
C     END                                                               00005600
C [FCODE]:                                                              00005610
C     SUBROUTINE MY_FCODE(Y,X,B,W,F,IN,IDER)                            00005620
C     USER WRITTEN TO EVALUATE THE NONLINEAR OBJECTIVE FUNCTION (F)     00005630
C     USED IN NLSOL AS THE WEIGHTED SUM OF (Y(IN)-F)**2, WHERE          00005640
C     Y= OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N, WHERE N IS          00005650
C      GIVEN IN $PARMS NAMELIST INPUT--SEE BELOW).                      00005660
C     X= OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,M, WHERE           00005670
C      M IS IN $PARMS INPUT).                                           00005680
C     B= CURRENT PARAMETER ESTIMATES (DIM. K, WHERE                     00005690
C      K IS IN $PARMS INPUT).                                           00005700
C     W= WORK ARRAY (DIM. 5)--MAY BE USED TO PASS DATA TO PCODE.        00005710
C     F= (OUTPUT) THE FUNCTION VALUE EVALUATED FOR THE GIVEN            00005720
C      Y,X, AND B ARRAYS AT THE OBSERVATION NO. 'IN'.                   00005730
C     IN= (INPUT) OBSERVATION NO. TO EVALUATE F (1.LE.IN.LE.N),         00005740
C      WHICH IS CONTROLLED EXTERNALLY BY 'NLSOL'. USUALLY,              00005750
C      IN=1,2,...,N--BUT NOT ALWAYS.                                    00005760
C     IDER= 0 IF ANALYTICAL DERIVATIVES ARE USED (PCODE CALLED          00005770
C        AFTER FCODE).                                                  00005780
C         = 1 IF ESTIMATED DERIVATIVES ARE USED (PCODE NOT CALLED       00005790
C        AFTER FCODE).                                                  00005800
C     DIMENSION Y(1),X(500,5),B(1),W(5)                                 00005810
```

```
C>>>>> INSERT USER CODE HERE TO EVALUATE F  <<<<<                        00005820
C     END                                                                00005830
C [PCODE]: >> PCODE MAY BE A DUMMY NAME IF ONLY IDER=1 IS TO BE USED. <<00005840
C     SUBROUTINE MY_PCODE(P,X,B,W,F,IN,IP,IB)                            00005850
C     USER WRITTEN TO EVALUATE THE ANALYTICAL PARTIAL DERIVATIVES OF     00005860
C     F WITH RESPECT TO B(J),J=1,2,...,K, AT OBSERVATION 'IN', WHERE     00005870
C     P= (OUTPUT) PARTIAL DERIVATIVE ARRAY (DIM. K, WHERE                00005880
C      K IS IN $PARMS INPUT).                                            00005890
C     X,B,W ARE THE SAME AS USED IN FCODE (SEE ABOVE).                   00005900
C     F= LAST FUNCTION VALUE FROM FCODE AT OBSERVATION IN.               00005910
C      (NOTE THAT F MAY NOT BE NEEDED, BUT IS AVAILABLE ANYWAY)          00005920
C     IN= (INPUT) OBSERVATION NO. TO EVALUATE P ARRAY, WHICH IS          00005930
C      CONTROLLED EXTERNALLY BY 'NLSOL' (1.LE.IN.LE.N).                  00005940
C     IP= (INPUT) THE NO. OF B-PARAMETERS HELD FIXED IN THE LEAST-       00005950
C      SQUARES (0.LE.IP.LE.K-1; USE IP=0 IF NONE).                       00005960
C     IB= ARRAY OF B-PARAMETER INDICES HELD FIXED IF IP.GT.0.            00005970
C      NOTE THAT THE INDICES IN IB ARRAY MAY BE IN ANY ORDER,            00005980
C      BUT MUST BE BETWEEN 1 AND K (K IS IN $PARMS INPUT).               00005990
C     DIMENSION P(1),X(500,5),B(1),W(5),IB(1)                            00006000
C>>>>>  INSERT USER CODE HERE TO EVALUATE P  <<<<<                       00006010
C     END                                                                00006020
C [SUBZ]:                                                                00006030
C     SUBROUTINE MY_SUBZ(Y,X,B,W,NW,N,TITLE,IOUT)                        00006040
C     USER WRITTEN INITIALIZATION ROUTINE (CALLED ONCE BY 'NLSOL').      00006050
C     SUBZ MAY BE USED TO CHECK Y(IN),X(IN,M) AFTER INPUT VIA            00006060
C     OBJECT (RUN)-TIME INPUT (SEE BELOW) ON UNIT IALT. ALSO, SUBZ       00006070
C     MAY BE USED TO READ ADDITIONAL $INIT PARAMETERS, AND TO LOAD       00006080
C      ANY COMMON BLOCKS IF NEEDED IN THE USERS FCODE,PCODE.             00006090
C     Y,X,B,W  ARE THE SAME AS USED IN FCODE (SEE ABOVE).                00006100
C     NW= USE ANY DUMMY INTEGER VARIABLE (THIS IS                        00006110
C      TO MAINTAIN COMPATIBILITY WITH 'MARQRT' OR 'IMSLMQ').             00006120
C     N= NO. OF OBSERVATIONS IN Y(N),X(N,M) ARRAYS, WHERE                00006130
C      K.GE.N.LE.500 (N,M,K  ARE IN $PARMS INPUT).                       00006140
C     TITLE= (INPUT) 80-CHARACTER HEADING (SEE INPUT FOR005 BELOW).      00006150
C     IOUT= 1 IF TO WRITE OUTPUT ON BOTH FOR006 AND FOR016.              00006160
C         = 0 IF TO WRITE OUTPUT ONLY ON FOR006.                         00006170
C     DIMENSION Y(1),X(500,5),B(1),W(5)                                  00006180
C     CHARACTER*80 TITLE                                                 00006190
C>>>>> INSERT USER CODE HERE FOR ANY INITIALIZATION DESIRED  <<<<<       00006200
C     END                                                                00006210
C [SUBEND]:                                                              00006220
C     SUBROUTINE MY_SUBEND(Y,X,B,K,N,TITLE,IOUT)                         00006230
C     USER WRITTEN TERMINATION ROUTINE (CALLED ONCE BY 'NLSOL').         00006240
C     SUBEND MAY BE USED TO OUTPUT THE FINAL SOLUTION VECTOR B(I),       00006250
C     I=1,2,...,K, IN OTHER FORMS, ETC., AS DESIRED. [OR IT MAY BE A     00006260
C     DUMMY ROUTINE; I.E., JUST RETURNS.]                                00006270
C     Y,X,K,N,TITLE,IOUT  ARE THE SAME AS IN SUBZ AND FCODE.             00006280
C     B= (INPUT) IS THE FINAL SOLUTION VECTOR AS DETERMINED BY           00006290
C      'NLSOL' (SEE REF1 FOR DETAILS).                                   00006300
C     DIMENSION Y(1),X(500,5),B(1)                                       00006310
C     CHARACTER*80 TITLE                                                 00006320
C>>>>>  INSERT USER CODE HERE FOR ANY TERMINATION SUMMARY DESIRED  <<<<<00006330
C     END                                                                00006340
C                                                                        00006350
```

```
C*************************************************************************00006360
C                                                                         00006370
C** INPUT ORDER ON FOR005 (PARAMETER FILE LOGICAL NAME):                   00006380
C                                                                         00006390
C 1.  TITLE (MAX. 80-CHARACTERS--ALWAYS READ BEFORE $PARMS INPUT).        00006400
C 2.  $PARMS  -- POSSIBLE NAMES ARE: N,M,K,B(),IP,IB(),IALT,IWT,IDER,     00006410
C       BL(),BH(),IPRT,IOUT,NITER,SP, -- PLUS FOLLOWING PARAMETERS FROM   00006420
C     REF1 (NL2SOL), P.31-35: IV(),V().                                   00006430
C 3.  (OBJECT-RUN-TIME FORMAT|STATEMENT) TO DESCRIBE THE FORMAT OF THE    00006440
C     DATA MATRIX ROW Y(I),(X(I,J),J=1,M*) READ ON FILE IALT, WHERE       00006450
C     M*=M (IF IWT=0) OR M*=M+1 (IF IWT>0), M.LE.4, AND I=1,2,...,N.      00006460
C (3A). INSERT DATA MATRIX HERE ONLY IF IALT=5.                           00006470
C 4.  $INIT OPTIONAL NAMELIST USED FOR READING PROBLEM-DEPENDENT          00006480
C     PARAMETERS USED IN SUBROUTINE SUBZ (SEE ABOVE).                     00006490
C 5.  OPTIONALLY, REPEAT STEPS 1-4, IF PARAMETER ISTOP=0 WAS USED         00006500
C     IN THE LAST STEP 2.                                                 00006510
C                                                                         00006520
C** OUTPUT IS GIVEN ON FOR006 (ON-LINE USUALLY) AND ON FOR016(IF IOUT=1)00006530
C   FOR016 CONTAINS ALL PRINTABLE OUTPUT SELECTED VIA $PARMS IPRT,IOUT.   00006540
C   NOTE: IPRT=0 GIVES ABBREVIATED OUTPUT ON FOR006 (BUT MORE ON FOR016)00006550
C         IPRT=1 OR -2 GIVES DETAILED OUTPUT ON BOTH 6 AND 16.            00006560
C         IPRT=-1 GIVES MODERATE OUTPUT ON 6 (DETAILED ON 16).            00006570
C                                                                         00006580
C** TO RUN ON VAX (ELIMINATE <> DELIMITERS IN SUBSTITUTIONS):             00006590
C                                                                         00006600
C   $ASSIGN <PARAMETER FILE NAME> FOR005                                  00006610
C   $ASSIGN <DATA MATRIX FILE NAME> FOR010                                00006620
C   $RUN <MAIN NAME>                                                      00006630
C                                                                         00006640
C [NOTE: NLSOL2 USES SCRATCH UNIT FOR099 VIA CALL PRENAM(5,99)]           00006650
C                                                                         00006660
C*************************************************************************00006670
C                                                                         00006680
C$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$00006690
```

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.