U.S. DEPARTMENT OF THE INTERIOR
GEOLOGICAL SURVEY

Computation of transient soundings for the time-derivative of Hz near a rectangular loop source on a layered earth (Program FWDTHZ)

by

Walter L. Anderson

Open-File Report 85-270

1985

DISCLAIMER

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

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

CONTENTS

ABSTRACT

The transient solution for the time-derivative of the vertical magnetic field (Hz) near a rectangular loop source on the earth's surface is provided by this program, where the observation point is in the plane of the loop. The method uses fast digital filtering techniques to evaluate Hankel and cosine transforms, as previously published by the author to obtain time-domain solutions for other common loop sources (e.g., central induction and coincident loops). A forward solution interface (designed for arbitrary problems) is used to minimize programming requirements, and to make it possible to easily convert the forward program to an inverse solution.

Numerical comparisons at the center of a rectangular loop with a central induction circular loop solution agrees to about 3-figure accuracy in the transient response. Other tests using a dipole-dipole time-domain program shows good agreement with the far-field limiting form of the new program. Some numerical results are listed and plotted, and a FORTRAN source listing is given in an appendix.

INTRODUCTION

Program FWDTHZ rapidly computes the time-derivative of the Hz-field transient response (voltage or apparent resistivity) about a rectangular loop source lying on an isotropic layered earth model. The solution can be computed at the earth's surface for points inside or outside a rectangular loop source of arbitrary dimensions; however, points very close to a loop side should be avoided (due to accuracy considerations). The quasi-static case, which neglects displacement currents, is assumed throughout this report.

The question of why a rectangular loop was chosen over a more general or arbitrary line segment loop naturally arises. It turns out that in many field applications using a loop source, it is easier to layout a closed loop with four parallel sides, rather than a circular or polygon-sided loop. Furthermore, a technique of "leap-frog" or moving rectangular loops at multiple sites is easily done in the field with this approach. Also, the need to record many line segment end-point coordinates is eliminated with a rectangular loop, as well as consideration of source and observation point symmetries (to be discussed later).

The method for calculating the electromagnetic (EM) fields in the frequency-domain about a rectangular loop source on a layered half-space was given by Anderson (1984a), where a fast Hankel transform (FHT) via lagged convolution (Anderson, 1982) was used to eliminate many repeated integrations over a direct double integration approach as reported by Poddar (1983). The method discussed by Anderson (1984a) was intended to provide a practical tool for studying the frequency response for cases where a dipole source or circular loop cannot always be assumed.

The time-domain solution for a step-function causal system can be expressed as a Fourier cosine integral of a suitable defined frequency response function (e.g., see Kaufman and Keller, 1983). The next section describes how this is done for the rectangular loop source problem.

The remainder of this report contains 1) a summary of the basic computations, 2) a detailed description of the program parameters, and 3) the VAX operating instructions. Appendix 1 offers some suggestions in converting the VAX program to other computer systems; Appendix 2 lists a simple input/output test example; Appendix 3 provides several families of sounding curves computed by varying certain model parameters; and Appendix 4 lists the FORTRAN-77 source code.

SUMMARY OF CALCULATIONS

Figure 1 shows the coordinate system and geometry of the rectangular loop.

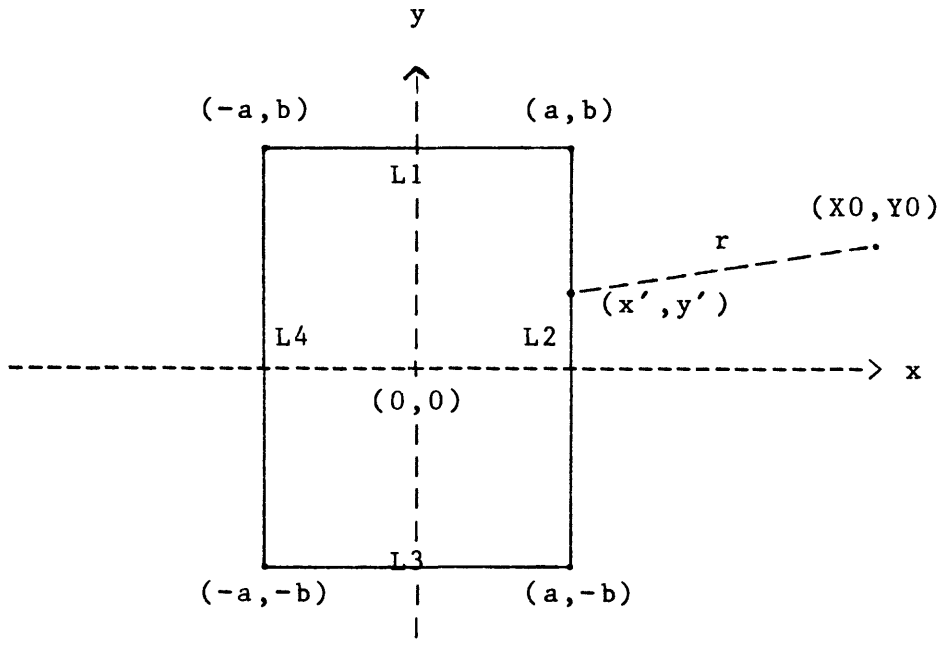


Fig.1.-- Loop geometry at z=0 (earth's surface).

The line segments [(-a,b):(a,b)], [(a,-b):(a,b)], [(-a,-b):(a,-b)], and [(-a,-b):(-a,b)] are denoted respectively as lines L1, L2, L3, and L4. The length of lines L1 and L3 is 2a, and lines L2 and L4 is 2b. The observation point is (X0,Y0), and (x',y') is any point on the rectangular loop source.

Referring to Anderson (1984a), the complex frequency response for the Hz-field inside or outside a rectangular loop at (X0,Y0) with current I exp(iwt) is given by

$$Hz(w) = I\ (H_{L1} + H_{L2} + H_{L3} + H_{L4})/2\pi \quad , \qquad (1)$$

where each term in (1) is the contribution due to each respective line segment in figure 1. The entire process of computing the frequency response over any desired discretation in w (angular frequency) is completely described in Anderson (1984a), and will not be repeated here. However, I wish to emphasize that the fast Hankel transform method (Anderson, 1982) used to compute (1) is nearly as efficient as that used for a dipole source solution, and hence the frequency-domain calculation is reasonably efficient for an arbitrary rectangle.

The process of computing the time-derivative of a frequency-domain function in the time-domain is accomplished using a Fourier cosine transform of the form (Kaufman and

Keller, 1983, p. 360)

$$\partial\, Hz(t)/\partial t = \frac{2\,C}{\pi} \int_0^{\infty} Re\left\{\frac{Hz(w)}{Hzp}\right\} \cos(wt)\, dw, \qquad (2)$$

where both sides of (2) were divided by the primary field (Hzp) for a rectangular loop source (Poddar, 1982, p. 104), such that the constant C in equation (2) contains Hzp times any arbitrary receiver moment (M= no.turns times area of coil) or scaling constant desired. The shape of the voltage curve using (2) will not be affected if C is set to unity (default); however, the actual voltage (assuming any scaling or moment) can be optionally computed (see parameter IOPT below).

Equation (2) is evaluated by a fast lagged convolution (Anderson, 1975) over any selected time range. The kernel function in (2) is interpolated by a cubic spline function, with a specified number of computed points (knots) per frequency decade (default is 8). Lower and upper frequency bounds, or equivalently, a dimensionless induction number range (B0=.01, BM=100), can be overridden for the spline function range; however, this is usually not necessary for typical earth models, since a normalized response Hz(w)/Hzp is used. (Note that the kernel in (2) can be replaced by 1.0 for arguments less than B0, and by 0.0 for arguments greater than BM; this is the primary reason a normalized form is used in (2).)

A special feature has been included in program FWDTHZ to approximate soundings during the late-time stage. Since 32-bit words and single-precision digital filters are used, round-off errors during the late stage can seriously affect the curve after the voltage amplitude decays to less than about 10^{-6} from unity. In this case, a late-time asymptotic approximation, developed for a two-layer earth by Kaufman and Keller (1983, p. 371), is optionally available (see parameter TASY below). If the chosen earth model has more than two layers, then the upper layers above the basal half-space is replaced in the approximation as an equivalent layer by using the conductance and accumulated thickness. This method is usually satisfactory for many models and loop geometries commonly used, but can fail for extreme cases (e.g., layers with large conductivity constrasts, or small transmitter loop sides compared to the accumulated layer depths).

Apparent resistivity for a rectangular loop is computed from the voltage curve using an infinite series approximation (Prácser and Frischknecht, 1984, pers. comm.). The expanded series for a rectangular loop source can be written in the form

$$V'(t)=C_1\sum_{k=1}^{\infty} C_k \alpha^{2k+1}\left\{\sum_{j=0}^{k-1}\frac{\binom{k-1}{j}}{2(k-j)+1}\left[\begin{matrix}(a-X0)^{2(k-j)-1} & -(-a-X0)^{2(k-j)-1}\\ (Y0-b)^{2j+1} & -(Y0+b)^{2j+1}\end{matrix}\right]^{*}+\sum_{j=0}^{k-1}\frac{\binom{k-1}{j}}{2(k-j)+1}\left[\begin{matrix}(b-Y0)^{2(k-j)-1} & -(-b-Y0)^{2(k-j)-1}\\ (X0-a)^{2j+1} & -(X0+a)^{2j+1}\end{matrix}\right]^{*}\right\}, \quad (3)$$

where $C_1 = \mu_o M/(4\pi^{3/2} t)$, $V'(t) = \partial Hz(t)/\partial t$,

$$C_k = (-1)^{k+1}[2-6/(2k+3)]/k!, \quad \mu_o = 4\pi 10^{-7},$$

and $$\alpha = [\sigma \mu_o/(4t)]^{1/2}.$$

In order to solve for the apparent resistivity ($\rho_a = 1/\sigma$) in equation (3) at any time t, the voltage response for a layered half-space (from equation (2)) is equated to equation (3), where the receiver moment M cancels on both sides; i.e., assuming C contains a factor M in equation (2). This process can be expressed as a solution for α in the nonlinear equation

$$F(\alpha) = \frac{1}{t}\sum_{k=1}^{\infty} C_k \alpha^{2k+1}\{D_k\} - V'(t) = 0, \quad (4)$$

where D_k are the terms within "{...}" in (3), and $V'(t)$ is the time-derivative of Hz. In practice, only one real solution of (4) is needed, requiring only a few terms of the series (generally no more than 10). This approach, however, does present some computational problems whenever the coefficients in the series (4) exceed the computer maximum floating-point exponent range on the VAX (viz., 10^{+38}). To overcome this problem, a scaling technique was used to reduce this effect internally in the computer code, but this is transparent to the user's input requirements. Given a real root α_t of (4) at any $t>0$, the apparent resistivity is computed by

$$\rho_a = \mu_o/(4t\alpha_t^2) \quad (5)$$

The above method for computing apparent resistivity for a rectangular loop may be far more complicated than required in some instances. For example, it is often convenient to use only a late-time formula for computing apparent resistivity for all $t>0$; i.e., if only the first term of (4) is used, then a trivial solution for ρ_a can be expressed

as

$$\rho_a(t) = \mu_o \left[1.6\ a\ b\ \mu_o/(t\ |V'(t)|)\right]^{2/3}/4\pi t . \qquad (6)$$

An option to select computed apparent resistivity via (5) or (6) is defined by parameter IOPT (see below). Because of the simplicity and speed of computing (6), the late-time method is recommended for most applications, although the early part of the curve will not be representive of the first layer resistivity as when using the series method (e.g., see model 6 in Appendix 3).

Examples of various models using FWDTHZ are provided in Appendix 3. Different model parameters are varied to illustrate the sounding behavior and characteristic form to expect. Note that in all cases, the transient voltage is smoothly varying from the center of the rectangular loop toward any side, and has a positive response. Outside the loop, the transient voltage can change sign, which produces a significant critical value in the corresponding apparent resistivity curve at the point of sign change (e.g., see model 4 in Appendix 3).

Some of the FWDTHZ results in Appendix 3 for a square loop at the loop center (X0=Y0=0) were compared with a transient central-induction circular loop program TCILOOP (Anderson, 1981). Agreements were generally good to about 3-place accuracy, where the radius of an equivalent circular loop was taken to be $L/\sqrt{\pi}$, L=2a=2b in Figure 1. Similar checks were also made using FWDTHZ far outside a rectangular loop with a dipole-dipole transient program TRANS-HCLOOP (Anderson, 1979); again, the results agreed to approximately 3-place accuracy for most models tested.

PARAMETERS, FILES, AND INPUT ORDER REQUIRED

Parameters required by program FWDTHZ are read using FORTRAN NAMELIST input statements on the VAX/VMS system. The namelist names used are $FWD and $INIT. All $FWD parameters, program files (FOR005, FOR006, FOR012, FOR016, FOR0098-99), and input order required are identical to those described in detail for the general forward program interface FWDSOL (Anderson, 1984b, p. 6-8). (Familiarity with the $FWD parameters in the latter reference is assumed; definitions of these parameters are not repeated here in the interest of brevity.) The example in Appendix 2 can be followed as a typical set of forward model parameters. The $INIT model parameters described below are for a rectangular loop, and are similar to those for the frequency-domain program HRZRECT (Anderson, 1984a).

$INIT PARAMETER DEFINITIONS

$INIT parameters (nondefault parameters must always be given):

MM= Number of layers in the model ($1 \le MM \le 10$; default MM=1 for a homogeneous half-space). Note that $FWD MM must also be given to agree with $INIT MM value; this is because a general purpose forward program interface (FWDSOL) is used, and some duplicate information must be given that cannot be easily cross-checked.

AX,BY= Coordinates (a,b) of corner of rectangular loop source in the first quadrant in Fig. 1. (Both AX and BY must be given >0.)

X0,Y0= Coordinates of the observation point. Note that (X0,Y0) must not lie on the rectangular source, and in general, it should not be extremely close to any line segment L1, L2, L3, or L4 in Fig. 1; e.g., a guideline is to choose (X0,Y0) such that $r > MIN(AX,BY)/10$. Symmetry (e.g., $X0 \ge 0$, Y0=0) should be used whenever possible. (Default is X0=0, Y0=0.)

EPS= Requested integration tolerance used to compute all Hankel transforms by related and lagged convolutions as described for subprogram HANKEL in Anderson (1982, p.352-353). Default is EPS=0.1E-9, which is about the optimum request for a 32-bit word machine to give relative errors $\ge$0.1E-7.

EPS2= Requested relative error for all finite spline-interpolated integrations using adaptive Guassian quadrature subprogram ZSUBA1 (see code in Appendix 4). Default is EPS2=0.1E-3, which gives about 3 or more figure accuracy. EPS2 should not be chosen smaller than about 0.1E-6 due to increased computer time and accuracy problems while using single-precision 32-bit floating-point words.

MXEVAL= The maximum function evaluations permitted in any

spline interpolation during adaptive Gaussian quadrature using subprogram ZSUBA1. (Default is MXEVAL=1000; MXEVAL should be increased if EPS2 is decreased from its default value, or if attempting near source problems.)

RFAC= An heuristic factor ($\geq$1.0) to use to expand the distance-to-rectangle range [RMIN,RMAX] in order to avoid pertubations on the ends of each spline-defined function. (Default is RFAC=5, which should be more than adequate for typical field applications.)

B0= The lower induction number for which the Hz/Hzp frequency response approaches 1.0 for B<B0. Default is B0=.01, which is usually adequate for most models; for more accuracy in the late-time transient, B0<.01 can be tried (but use of TASY may be more appropriate here).

BM= The upper induction number for which the Hz/Hzp frequency response approaches 0.0 for B>BM. Default is BM=100, which is usually adequate for most models; for more accuracy in the early-time transient, BM>100 can be used, but at the expense of longer computing times.

NB= The number of induction numbers to compute per frequency decade in the interval [B0,BM]. Default is NB=8, which is usually adequate for many moderate layered earth models (e.g., total thicknesses commersurate with the loop size, and relatively small conductivity constrasts). In general, NB should not be smaller than about 5, but can be greater than 8; it is recommended that NB not be larger than about 25 because the computer time would significantly increase as NB increases (also see parameter NBSKIP). If NB=0 is used, then a direct mode of computing the frequency function is used in the lagged cosine digital filter (i.e., the spline function is bypassed only if NB=0). The latter is generally not recommeded.

NBSKIP= The number of induction numbers to skip in the first decade [B0,10*B0] and last decade [BM/10,BM] when NB>>8. This technique will save considerable time when NB is large and/or B0 is decreased and/or BM is increased. Note that almost all frequency response curves in these respective subinterval are nearly constant (e.g, see Hz/Z0 curves in Appendix 3); this suggests that fewer spline knots are needed to adequately define the spline function in the beginning and end intervals in [B0,BM]. Default is NBSKIP=1, which means no points are skipped; NBSKIP>1 will skip the number of points requested (up to NB), but will always include the first and last point in each subinterval.

TASY= The selection of the late-time asymptotic approximation is controlled whenever TASY is not zero (default TASY=0.0 turns off this option). If TASY>0 is selected, then FWDTHZ will "force" the

asymptotic approximation starting at TASY seconds; however, it should be noted that no checks are made to insure the approximation is valid at this selected point. A safer use of this option is selected whenever TASY=-1.0 (i.e. TASY<0). When TASY<0, the program will automatically apply the asymptotic approximation after a time when the voltage amplitude drops to less than 1.E-6, which is about the best relative error possible with the single-precision 32-bit word filters. (In some extreme models and/or geometries, the asymptotic approximation will fail with an unwanted peak and shift in the voltage and/or apparent resistivity curve.)

TOFF= A ramp-correction option to apply to the voltage curve only when TOFF>0.0 seconds. Default is TOFF=0.0, which ignores this option. When TOFF>0.0, an EM-37 type ramp off-time is used to integrate the step function voltage at each t to t+TOFF. If TOFF<0.0, a SIROTEM type ramp off-time is used from t-|TOFF| to t for all t.

NK= The number of terms in the series (4) to use when IOPT=1 (see below). Default is NK=10, which is usually adequate; maximum is NK=30, but this may give overflow conditions in some cases. It is advised that NK<21 be generally used.

IOPT= The ouput sounding option (called Y(I) in FWDSOL):

-1 to select V'(t) output in units of volts/amp;

0 (default) to select V'(t)/SIG(1) output in unscaled units;

1 to select apparent resistivity output via eq. (5) with NK terms in eq. (4);

2 to select apparent resistivity output via eq. (6)--i.e., the late-time approximation method (this is generally recommended). Note that the default (IOPT=0) is set to be compatable with other time-domain programs previously used by USGS, but IOPT=2 is commonly overridden in practice.

IALL= A special USGS output plot file option defined as follows:

0 (default) to ignore this option (recommended for non-USGS use);

1 to write USGS plot files FWDTHZ.F, FWDTHZ.V, and FWDTHZ.R to disk for later plotting of stacked results (e.g., as used in Appendix 3); these files are written in ASCII mode, and where the format is similar to that described in Anderson (1984b, p. 16-17). (Note: if converting this forward program to an inverse program using NLSOL2 as described in Anderson, 1984b, then IALL=0 should always be used.)

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

EXAMPLES OF INPUT PARAMETERS

```
Example Title (see Anderson, 1984b for $FWD definitions)
$FWD MM=2,SIG=1E-3,2.E-2, H=200,
X1=1E-7,NX=8,XM=.1, SHIFT=1,
IPLT=1,XT='TIME (SEC.)',YT='APP.RES. RHO-A (OHM-M.)'$
$INIT IALL=1,NBSKIP=2,TASY=-1,NB=15,
IOPT=2,AX=200,BY=200,X0=0,Y0=0,MM=2$
2/ same model using IOPT=1
$FWD $
$INIT IOPT=1$
```

(See Appendix 2 for a complete input/output example.)

VAX OPERATING INSTRUCTIONS

Assuming program FWDTHZ and all associated subprograms have been compiled and linked using the VAX/VMS operating system, the following steps are general execution guidelines (note that many variations are possible using VMS in either time-sharing or batch modes):

1. Either assign (via $ASSIGN command) an input parameter file name to the logical name FOR005, or let FOR005 default to the users terminal input (if logged-in on-line). The order of the parameters on FOR005 must be given exactly as defined in Anderson (1984b, p. 6). To assign FOR005, use the DCL command:

   ```
   $ASSIGN parameter_file_name FOR005
   ```

2. If $FWD IPLT>0 is selected, then a specific output file name may be assigned to FOR012 (as in step 1); otherwise, the system will assume a file name of FOR012.DAT. When IPLT=0 (default), this step may be ignored. (For a description of the format of FOR012, refer to Anderson, 1984b, p. 16-17).

3. Program FWDTHZ may be executed with the DCL command:

   ```
   $RUN FWDTHZ
   ```

   On the USGS system, use the command:

   ```
   $RUN [WANDERSON]FWDTHZ
   ```

The above execution steps can also be submitted (via a $SUBMIT command) to be run in batch mode. For this reason, prompting messages and user responses have been excluded from program FWDTHZ. VAX system-dependent commands and calls have been minimized for ease of program conversion to other systems (see Appendix 1 for information on conversion considerations).

PRINTED OUTPUT

File FOR016 is the print (disk) file, normally called FOR016.DAT unless assigned otherwise, and file FOR006 is usually the on-line terminal print file (or LOG file if $SUBMIT was used). (On the USGS VAX system, the on-line file FOR006 is used only for messages, and file FOR016.DAT is the master print-out disk file.) Refer to Appendix 2 for a sample output listing of file FOR016 and corresponding input file FOR005. For each problem (title, $FWD, $INIT) set, a title line is printed and a complete NAMELIST write is given for all default and/or initial values. This is followed by a numerical listing of X(I), Y(I), where X is the selected time range ($FWD X1 to XM), and Y is the selected sounding curve via $INIT parameter IOPT.

ERROR MESSAGES

Most $FWD and $INIT syntactical errors are flagged and printed on files FOR006 and FOR016, and the job is aborted. If FOR005 was previously assigned to a disk parameter file, then correct the parameter file using any VAX editor and rerun the job (e.g., use $RUN or $SUBMIT). Other parameter errors (or omissions) are also flagged by program FWDTHZ, and the job is terminated. The messages "ICK<0...AFTER ZSUBA1" or "NEVAL()>MXEVAL...AFTER ZSUBA1" may result if MXEVAL or EPS2 are too small when attempting near source problems.

REFERENCES

Anderson, W.L., 1975, Improved digital filters for evaluating Fourier and Hankel transform integrals: U.S. Geologcial Survey Report USGS-GD-75-012, avail. as NTIS report PB-242-800/1WC, 223 p.

---------, 1979, Programs TRANS-HCLOOP and TRANS-HZWIRE: Calculation of transient horizontal coplanar loop soundings and transient wire-loop soundings: U.S. Geological Survey Open-File Report 79-590, 46 p.

---------, 1981, Calculation of transient soundings for a central induction loop system (Program TCILOOP): U.S. Geological Survey Open-File Report 81-1309, 80 p.

---------, 1982, Fast Hankel transforms using related and lagged convolutions: Assoc. for Computing Mach. Trans. on Math. Software, v. 8, no. 4, p. 344-368.

----------, 1984a, Fast evaluation of Hr and Hz field soundings near a rectangular loop source on a layered earth (Program HRZRECT): U.S. Geological Survey Open-File Report 84-257, 80 p.

----------, 1984b, A general interface for producing forward solution programs (Subprogram FWDSOL): U.S. Geological Survey Open-File Report 84-348, 43 p.

Kaufman, A.A., and Keller, G.V., 1983, Frequency and transient soundings: Elsevier, N.Y., 685 p.

Poddar, M., 1982, A rectangular loop source of current on a two-layered earth: Geophysical Prospecting, v. 30, p. 101-114.

----------, 1983, A rectangular loop source of current on multilayered earth: Geophysics, v. 48, no. 1, p. 107-109.

Appendix 1.-- Conversion to other systems

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

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

(1) 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) VAX non-ANSI NAMELIST input and output statements.

Appendix 2.-- Test problem input/output listing

The following input file (FOR005) was used to run a test problem for program FWDTHZ on a VAX system. The corresponding output file (FOR016) is given following FOR005.

FOR005

```
MODEL 6 (IOPT=2) -- (SEE APPENDIX 3 FOR PLOT.)
$FWD MM=3,SIG=.001,.02,.002, H=200,50,
X1=.1E-5,NX=8,XM=.1, SHIFT=1,
IPLT=1,XT='TIME (sec.)',YT='APP.RES. (ohm-m.)'$
$INIT IALL=0,NBSKIP=1,TASY=-1,NB=8,IOPT=2,
AX=200,BY=100,X0=100,Y0=50,MM=3$
```

FOR016

```
<FWDTHZ>:     MODEL 6 (IOPT=2) -- (SEE APPENDIX 3 FOR PLOT.)

 $INIT
 MM      =           3,
 AX      =   200.0000    ,
 BY      =   100.0000    ,
 X0      =   100.0000    ,
 Y0      =   50.00000    ,
 EPS     =  1.0000000E-10,
 EPS2    =  9.9999997E-05,
 MXEVAL  =        1000,
 RFAC    =   5.000000    ,
 B0      =  9.9999998E-03,
 BM      =   100.0000    ,
 NB      =           8,
 NBSKIP  =           1,
 TASY    =  -1.000000    ,
 TOFF    =  0.0000000E+00,
 NK      =          10,
 IOPT    =           2,
 IALL    =           0
 $END


 PARAMETER ORDER--

       1       SIGMA(  1)
       2       SIGMA(  2)
       3       SIGMA(  3)
       4       THICK(  1)
       5       THICK(  2)
       6           B(  6) SHIFT parameter in: B(2*MM)*APP.RES.2
```

<FWDSOL>: MODEL 6 (IOPT=2) -- (SEE APPENDIX 3 FOR PLOT.)

```
$FWD
MM      =            3,
MODE    =            1,
SIG     =  1.0000000E-03,  2.0000000E-02,  2.0000001E-03, 7*0.0000000E+00,
RHO     =   999.9999    ,   50.00000    ,   500.0000    , 7*0.0000000E+00,
H       =   200.0000    ,   50.00000    , 7*0.0000000E+00,
SHIFT   =   1.000000    ,
B       =  1.0000000E-03,  2.0000000E-02,  2.0000001E-03,   200.0000    ,
           50.00000     ,   1.000000    , 14*0.0000000E+00,
X1      =  1.0000000E-06,
NX      =            8,
XM      =  0.1000000    ,
XNX     = 500*0.0000000E+00,
X2      =   1.000000    ,
X3      =   1.000000    ,
X4      =   1.000000    ,
IPLT    =            1,
XT      = 'TIME (sec.)                          ',
YT      = 'APP.RES. (ohm-m.)                    '
$END
```

FWDSOL>: MODEL 6 (IOPT=2) -- (SEE APPENDIX 3 FOR PLOT.)

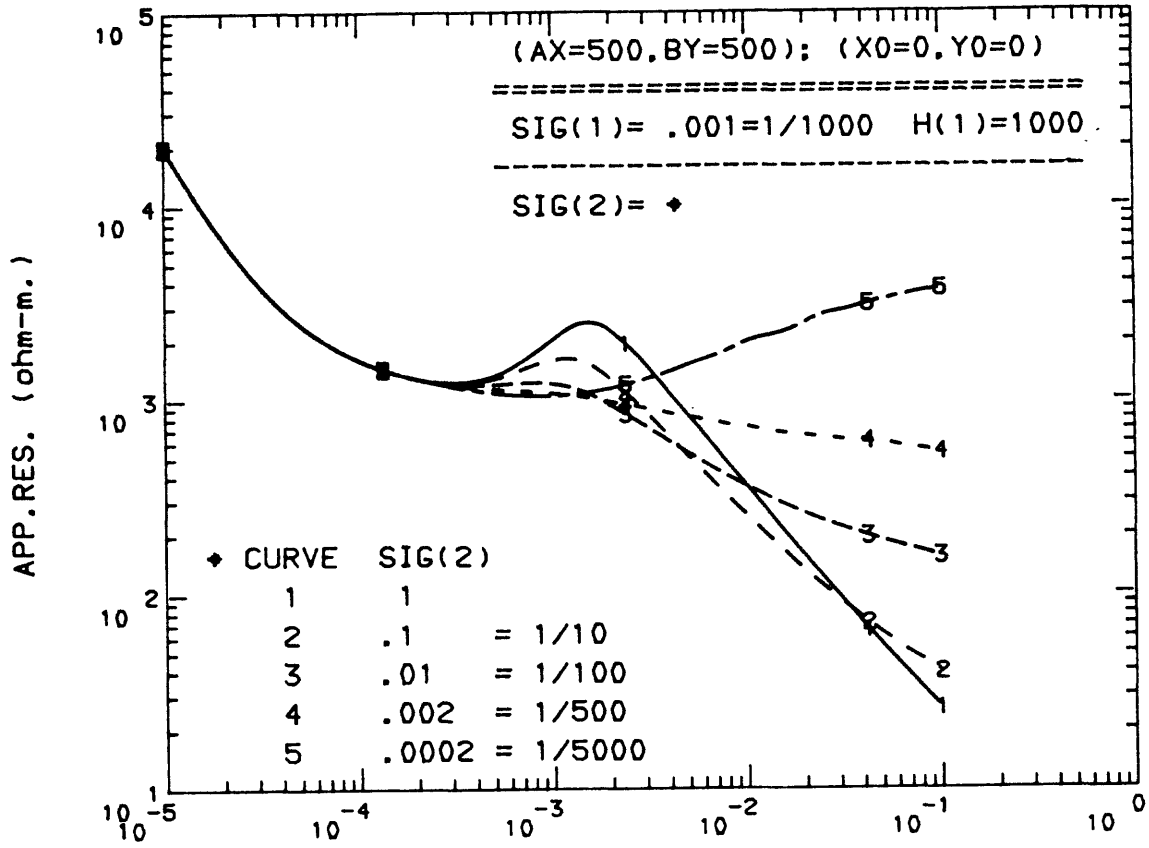
I	X(I)	Y(I)
1	0.10000000E-05	0.90444932E+04
2	0.13335215E-05	0.70037998E+04
3	0.17782795E-05	0.55184165E+04
4	0.23713740E-05	0.44159102E+04
5	0.31622781E-05	0.36009814E+04
6	0.42169659E-05	0.29903235E+04
7	0.56234144E-05	0.25246536E+04
8	0.74989439E-05	0.21702017E+04
9	0.10000002E-04	0.19077126E+04
10	0.13335218E-04	0.17277395E+04
11	0.17782799E-04	0.16254583E+04
12	0.23713745E-04	0.15980015E+04
13	0.31622789E-04	0.15961777E+04
14	0.42169668E-04	0.15455844E+04
15	0.56234159E-04	0.13765162E+04
16	0.74989461E-04	0.11045560E+04
17	0.10000006E-03	0.83933972E+03
18	0.13335222E-03	0.64729407E+03
19	0.17782806E-03	0.50782016E+03
20	0.23713753E-03	0.41799625E+03
21	0.31622799E-03	0.35881497E+03
22	0.42169681E-03	0.31911615E+03
23	0.56234177E-03	0.29897656E+03
24	0.74989483E-03	0.28487073E+03
25	0.10000009E-02	0.28508337E+03
26	0.13335226E-02	0.28605850E+03
27	0.17782811E-02	0.29464081E+03

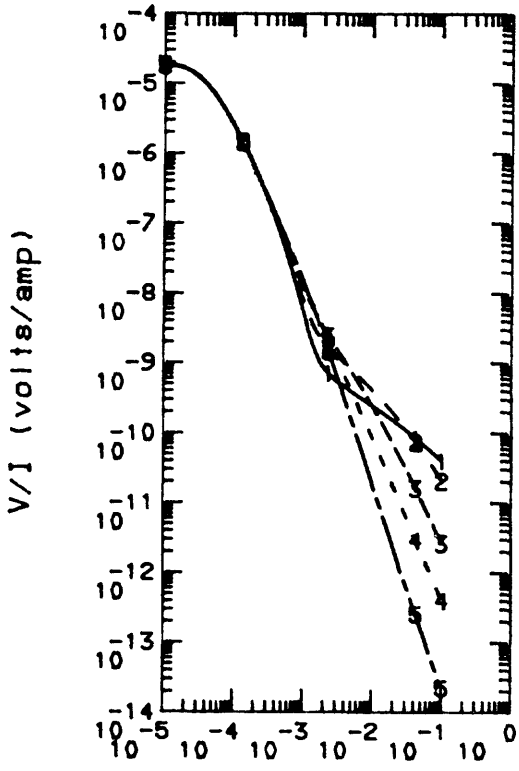
28	0.23713759E-02	0.30568671E+03
29	0.31622807E-02	0.32941959E+03
30	0.42169690E-02	0.35173892E+03
31	0.56234188E-02	0.37245239E+03
32	0.74989498E-02	0.39145465E+03
33	0.10000011E-01	0.40871466E+03
34	0.13335230E-01	0.42426120E+03
35	0.17782815E-01	0.43816537E+03
36	0.23713766E-01	0.45052805E+03
37	0.31622816E-01	0.46146698E+03
38	0.42169705E-01	0.47110724E+03
39	0.56234207E-01	0.47957755E+03
40	0.74989520E-01	0.48699860E+03
41	0.10000014E+00	0.49348764E+03

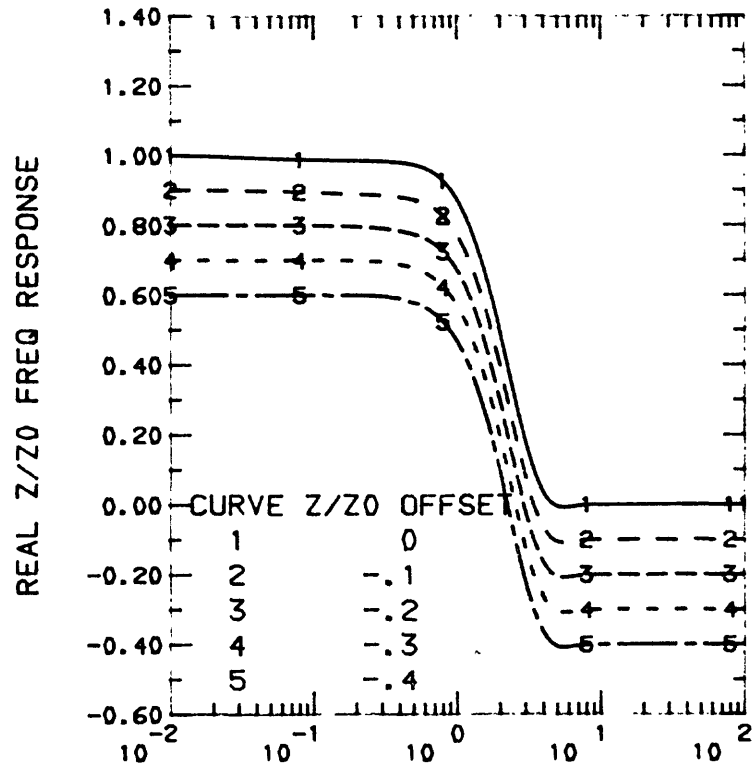
Appendix 3.-- Some sounding curve example plots

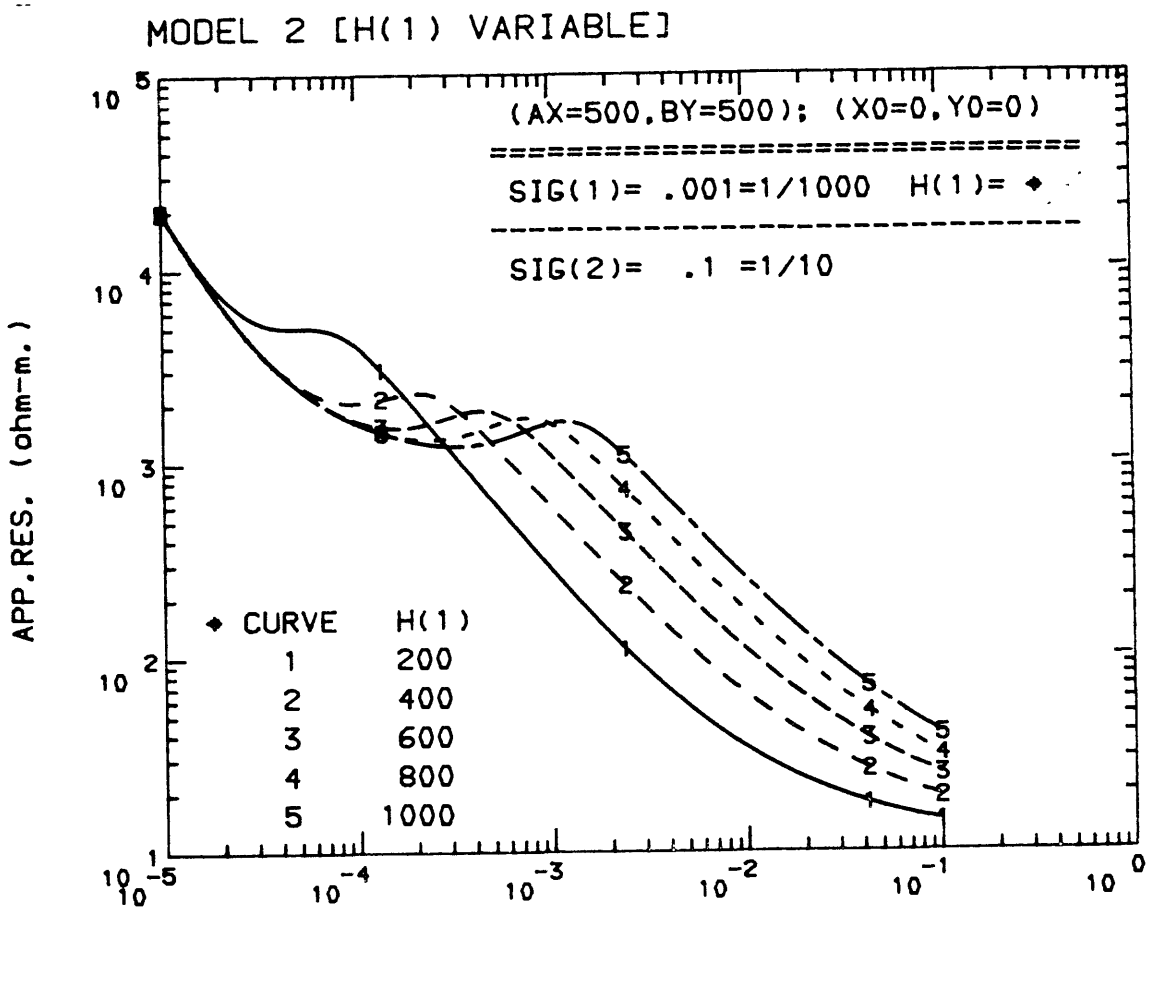
The attached plots were produced (after using $INIT IALL=1) for several layered models, and for several curve families. Only a single parameter is varied for each model defined. The legends in each plot gives the attributes of each model and set of curves. All intermediate curves are given mainly to illustate graphically the functions involved in determining the final apparent resistivity curves. In particular, the frequency-domain curves have the y-axis denoted as "REAL Z/Z0 FREQ RESPONSE"; voltage curves using IOPT=-1 are denoted as "V/I (volts/amp)"; the curves are marked with numbers as indicated in the legend in the apparent resisitvity curve plots. All apparent resistivity curves were computed using IOPT=2, with the exception of MODEL 6.

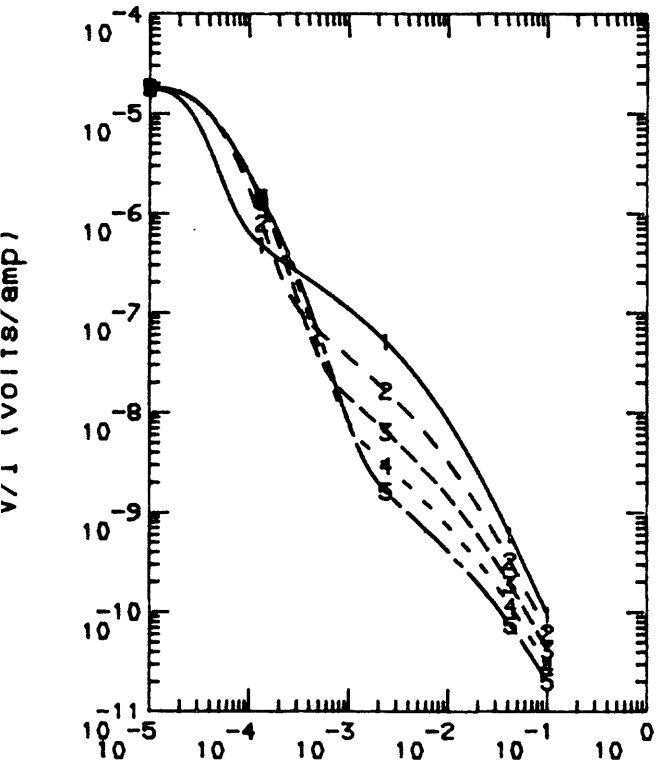
MODEL 1 [SIG(2) VARIABLE]

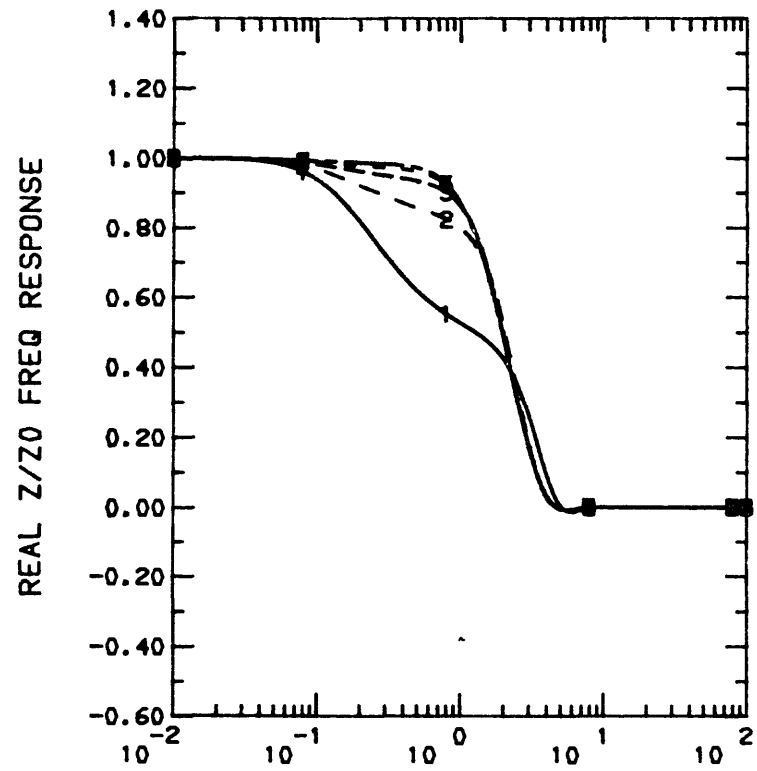


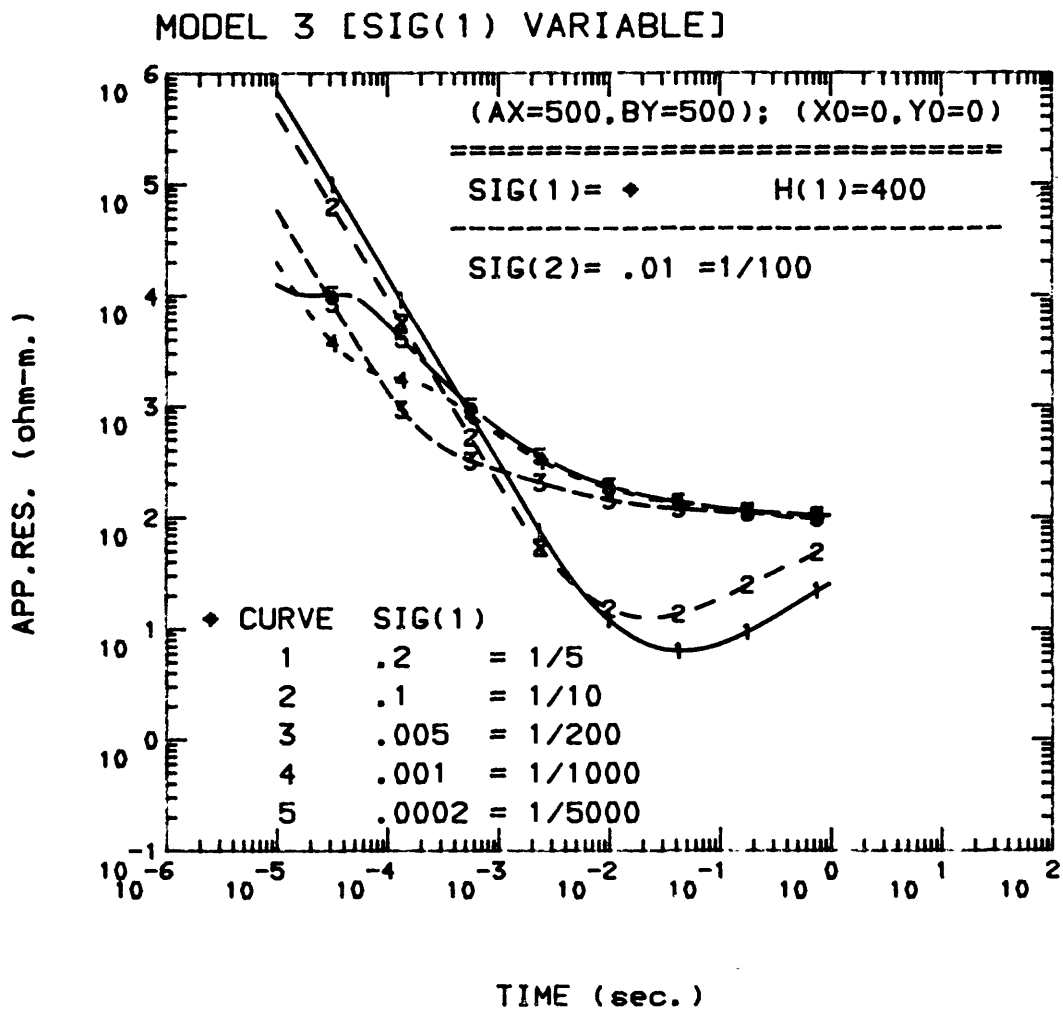


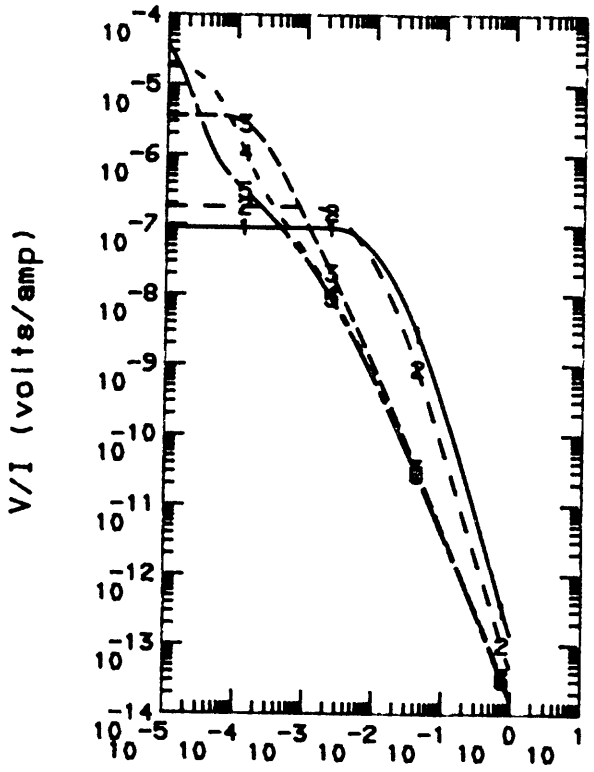


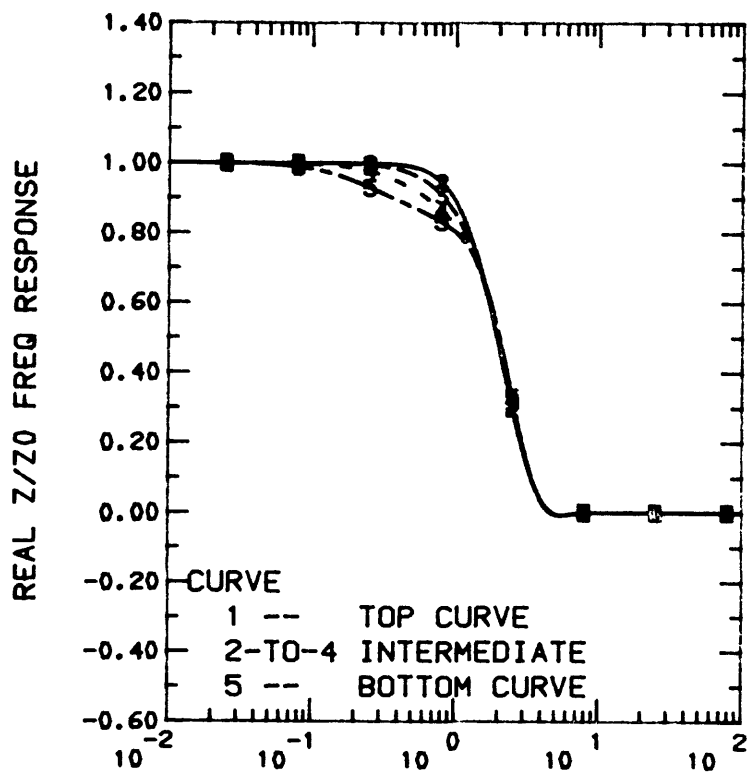

MODEL 2 [H(1) VARIABLE]
(AX=500,BY=500); (X0=0,Y0=0)
SIG(1)= .001=1/1000 H(1)= ◆
SIG(2)= .1 =1/10
◆ CURVE H(1)
1 200
2 400
3 600
4 800
5 1000
APP.RES. (ohm-m.)
TIME (sec.)


V/I (volts/amp)
TIME (sec.)

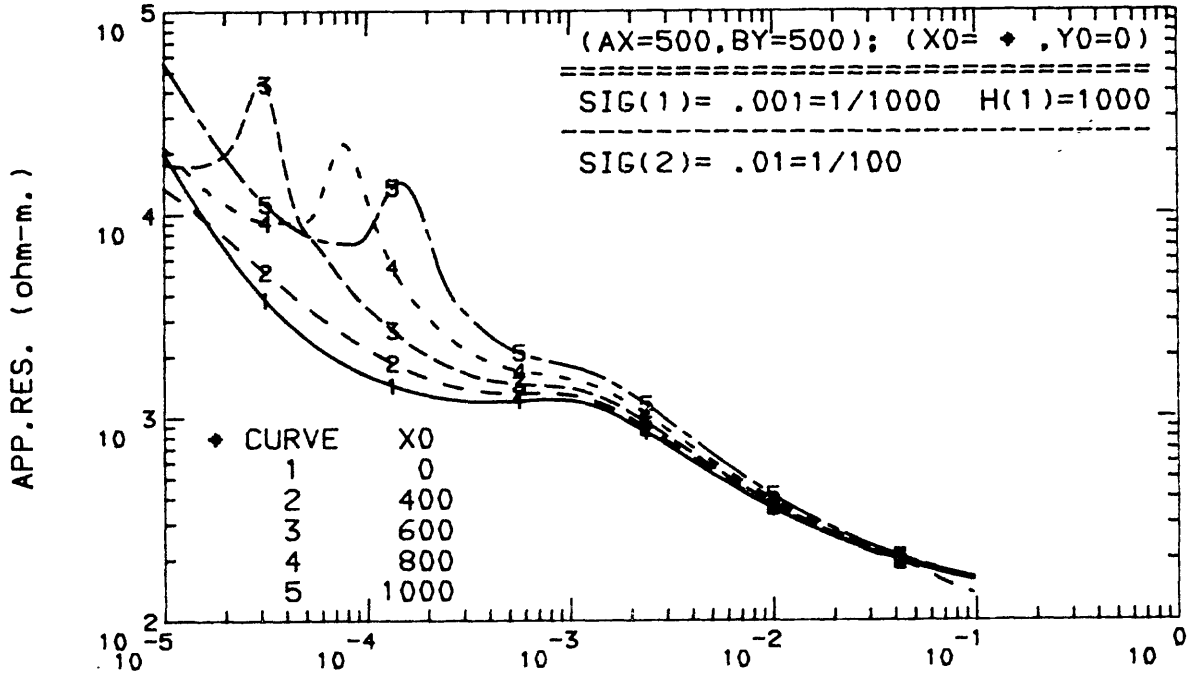

REAL Z/Z0 FREQ RESPONSE
IND.NO.(B)

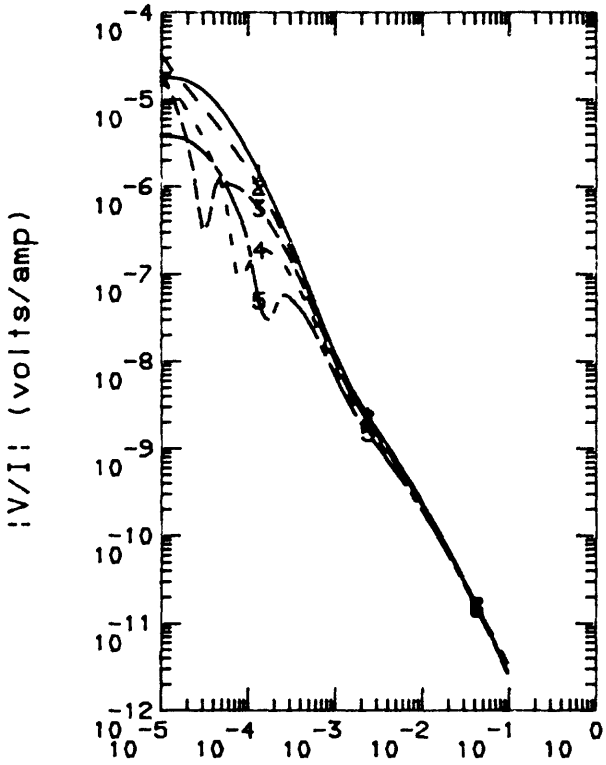

MODEL 3 [SIG(1) VARIABLE]
(AX=500,BY=500); (X0=0,Y0=0)
SIG(1)= ◆ H(1)=400
SIG(2)= .01 =1/100
◆ CURVE SIG(1)
1 .2 = 1/5
2 .1 = 1/10
3 .005 = 1/200
4 .001 = 1/1000
5 .0002 = 1/5000
APP.RES. (ohm-m.)
TIME (sec.)

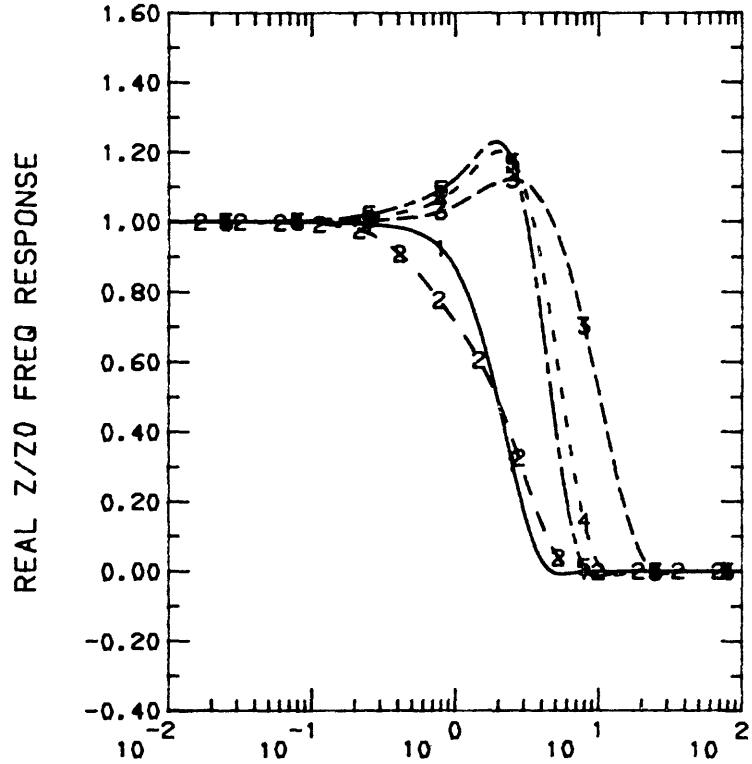

V/I (volts/amp)
TIME (sec.)


REAL Z/Z0 FREQ RESPONSE
CURVE
1 -- TOP CURVE
2-TO-4 INTERMEDIATE
5 -- BOTTOM CURVE
IND.NO.(B)

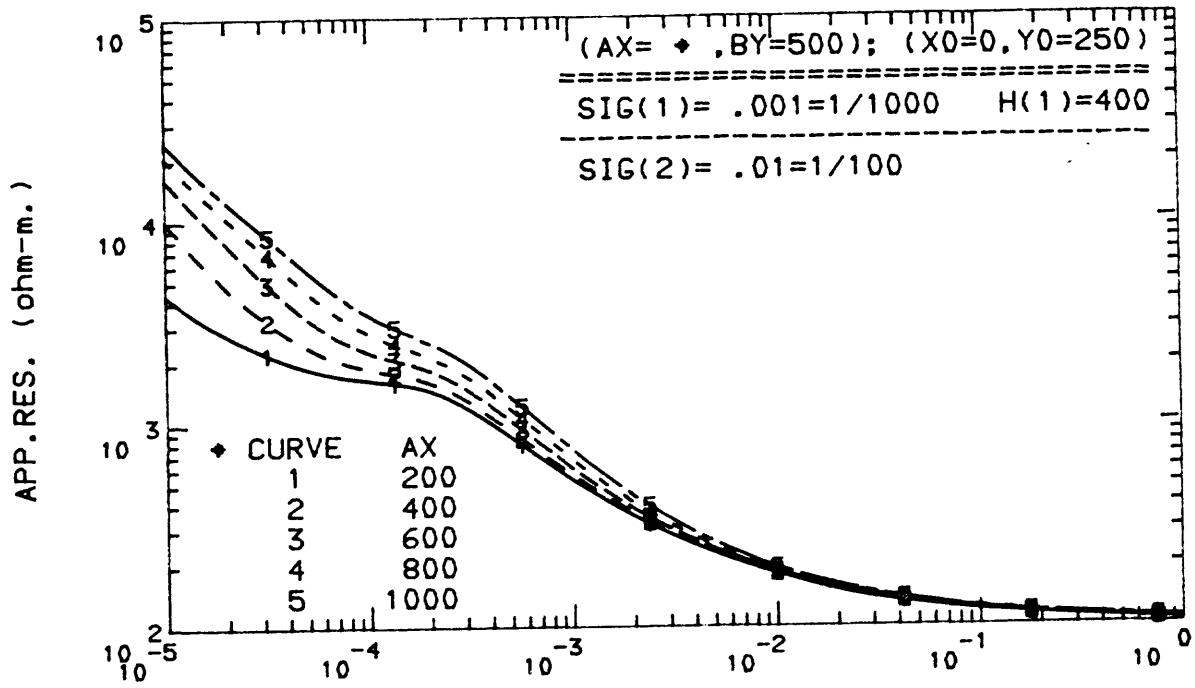
MODEL 4 (X0 VARIABLE)

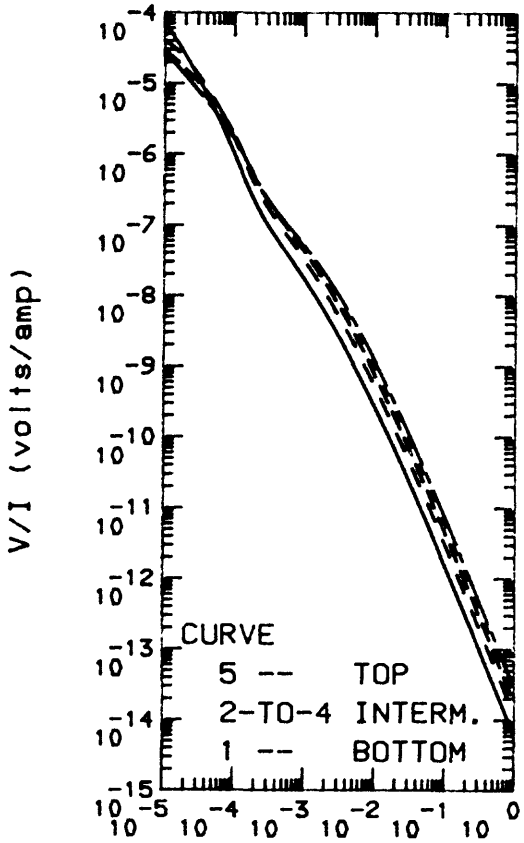


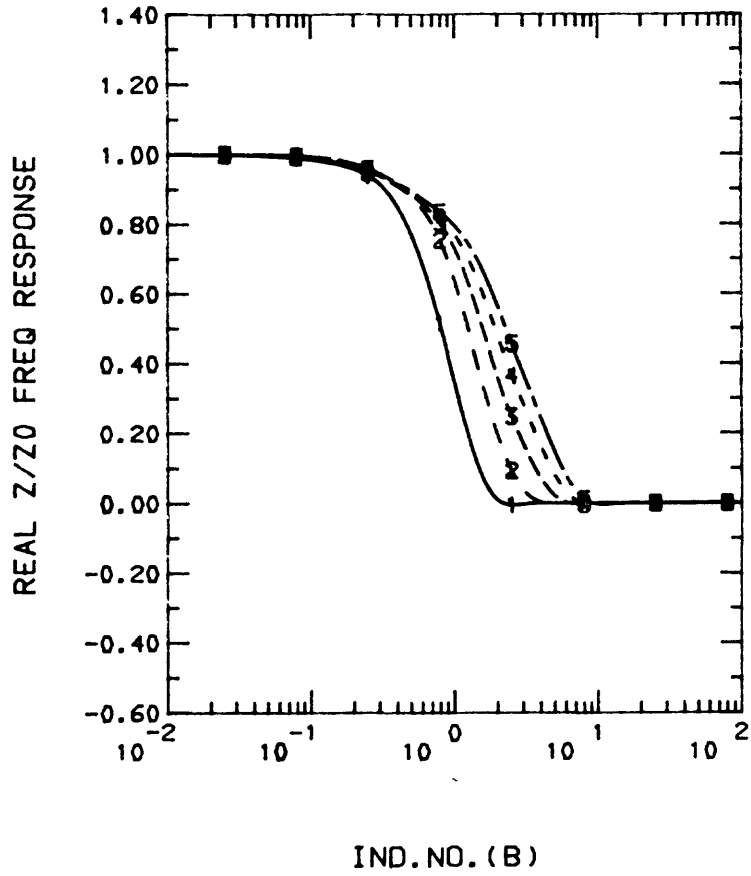


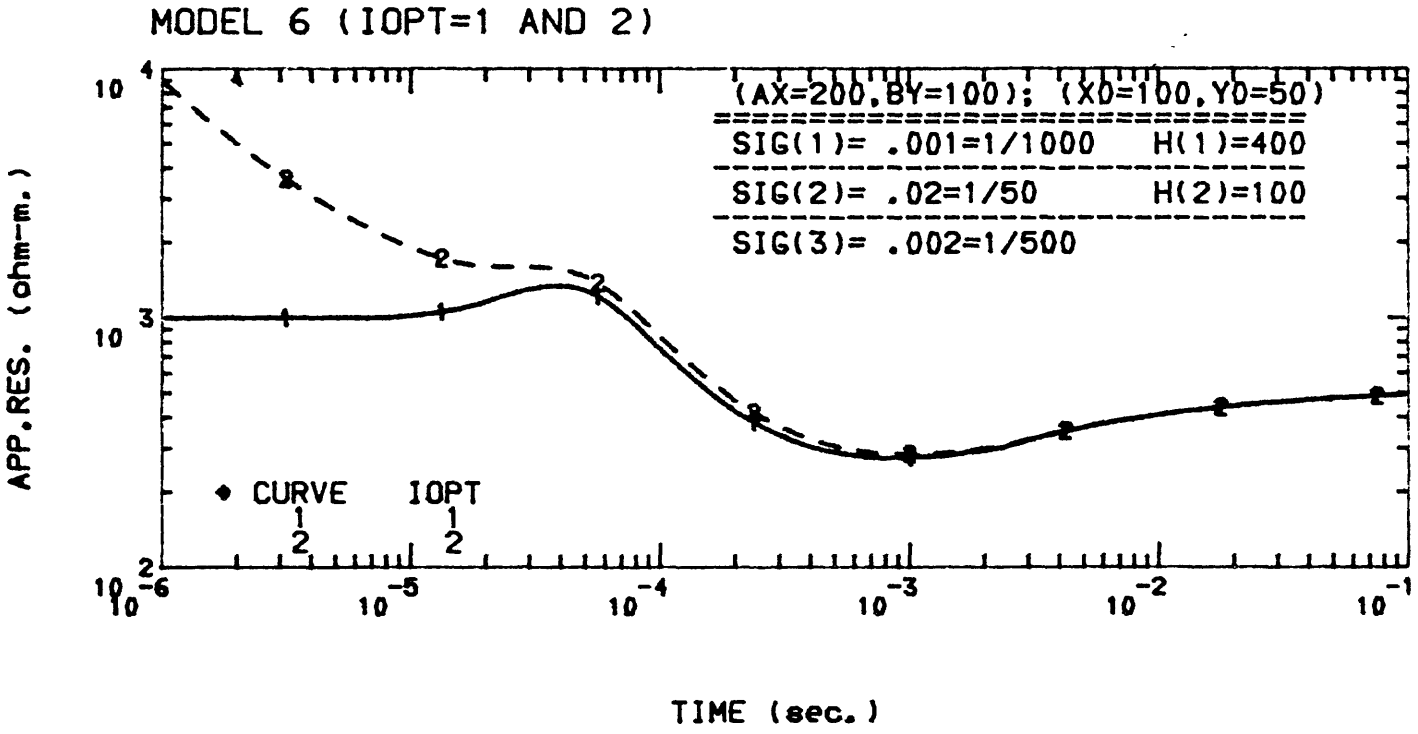


MODEL 5 (AX VARIABLE)








MODEL 6 (IOPT=1 AND 2)
(AX=200.BY=100); (X0=100.Y0=50)
SIG(1)= .001=1/1000 H(1)=400
SIG(2)= .02=1/50 H(2)=100
SIG(3)= .002=1/500
CURVE IOPT
1 1
2 2
APP.RES. (ohm-m.)
TIME (sec.)

Appendix 4.-- 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. 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 one-file named "FWDTHZ.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  OF HZ FIELD NEAR A RECTANGULAR LOOP SOURCE ON A LAYERED EARTH.        00000020
C                                                                        00000030
C** VAX-11/780 VERSION                                                   00000040
C                                                                        00000050
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.            00000060
C                                                                        00000070
C***********************************************************************00000080
C REFERENCES:                                                            00000090
C                                                                        00000100
C   ANDERSON, W.L., 1984, FAST EVALUATION OF HR AND HZ FIELD             00000110
C     SOUNDINGS NEAR A RECTANGULAR LOOP SOURCE ON A LAYERED EARTH:       00000120
C     USGS OPEN-FILE REPT. 84-257, 80 P.                                 00000130
C                                                                        00000140
C   ---------, 1984, A GENERAL INTERFACE FOR PRODUCING FORWARD           00000150
C     SOLUTION PROGRAMS: USGS OPEN-FILE REPT. 84-348, 43 P.              00000160
C                                                                        00000170
C***********************************************************************00000180
C                                                                        00000190
      EXTERNAL FCODE,SUBZ                                                00000200
      COMMON/CLOSE/ICL                                                   00000210
      CALL SETTIME                                                       00000220
      CALL FWDSOL(FCODE,SUBZ)                                            00000230
      CALL CPUTIME(0,16)                                                 00000240
      IF(ICL.NE.0) THEN                                                  00000250
        WRITE(21,10)                                                     00000260
10      FORMAT('0'/'1'/' ')                                              00000270
        WRITE(22,10)                                                     00000280
        IF(ICL.GT.0) WRITE(23,10)                                        00000290
      ENDIF                                                              00000300
      CALL EXIT                                                          00000310
      END                                                                00000320
      SUBROUTINE FCODE(Y,X,B,PRNT,F,IN,IDER)                             00000330
C--FUNCT. EVAL. FOR 'NLSTHZ' OR 'FWDTHZ'                                 00000340
C                                                                        00000350
C--PARAMETERS--                                                          00000360
C       Y=       OBSERVED DEPENDENT VARIABLE ARRAY (DIM. N)              00000370
C       X=       OBSERVED INDEPENDENT VARIABLE ARRAY (DIM. N,5)          00000380
C       B=       CURRENT PARAMETER ARRAY ESTIMATES (DIM. K)              00000390
C       PRNT=    WORK AND PRINT ARRAY (DIM. 5)                           00000400
C       F=       OUTPUT FUNCTION VALUE EVAL. FOR GIVEN Y,X,B AT OBS. IN  00000410
C       IN=      OBSERVATION NO. TO EVAL. F (1<=IN<=N)                   00000420
C       IDER=    0 IF ANALYTIC DERIVATIVES ARE USED LATER (PCODE CALLED) 00000430
C                1 IF ESTIMATED DERIVATIVES USED ONLY (PCODE NOT CALLED) 00000440
C [NOTE: CURRENTLY ONLY IDER=1 CAN BE USED; IDER=0 MAY BE ADDED LATER]    00000450
C                                                                        00000460
      PARAMETER (XMU0=1.256637061435917E-6,TWOPI7=6.28318531E-7,         00000470
     1           CON1=5.6418958E-8)                                      00000480
      COMPLEX ZANS(200),ZROOT(61)                                        00000490
      CHARACTER*40 TITLE3                                                00000500
      REAL Y(1),X(500,5),B(1),PRNT(5),SIG(10),H(9),DER(2),DD(9),         00000510
     1 BSAVE(20),W1(200),W2(200), AR(500),TAR(500), RARG(200),           00000520
     2 T(500),VSAVE(500),ACOEF(61),BCOEF(122),RK(10)                     00000530
      EXTERNAL HZREC                                                     00000540
      COMMON/PASS/H,EPS,EPS2,Z0,SIG1,RHOXY2,W0,WM,DELCON,                00000550
     1 MXEVAL,M1,IOUT,IOUTS                                              00000560
```

```
      COMMON/FPASS/TO,TM,BO,BM,DB,BMTEST,TASY,SCALE2,SCAL,CON,TOFF,     00000570
     1 BOSKIP,BMSKIP,NBSKIP,ISKIP, M21,M2,JSPLN,NN,IFIRST,IOPT,        00000580
     2 IALL,TITLE3                                                     00000590
      COMMON/PASS2/DK(30),NK                                           00000600
      COMMON/MODEL/RK,DD,M                                             00000610
      COMMON/HZSET/ZANS,RARG,DEL2,DEL,RMAX,RMIN,NR,ISET                00000620
      COMMON/DAT/AX,BY,XO,YO                                           00000630
      COMMON/COEF/SAX,SBY,SX,SY                                        00000640
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN     00000650
      DATA DER/2*0.0/                                                  00000660
      IF(IN.GT.1.OR.M.EQ.1) GO TO 20                                   00000670
      DO 10 J=2,M                                                      00000680
      IF(B(J).EQ.B(J-1)) CALL ERRMSG('SOME SIG(J)=SIG(J-1)',4,6,16)    00000690
10    CONTINUE                                                         00000700
20    DO 30 J=1,5                                                      00000710
30    PRNT(J)=X(IN,J)                                                  00000720
      IF(IN.GT.1) GO TO 800                                            00000730
      IF(IDER.EQ.1) GO TO 8001                                         00000740
35    SIG1=B(1)                                                        00000750
      RSIG1=1./SIG1                                                    00000760
      IF(M.EQ.1) GO TO 45                                              00000770
      DO 40 J=1,M1                                                     00000780
      H(J)=B(M+J)                                                      00000790
      SIG(J)=B(J)                                                      00000800
40    RK(J)=SIG(J)/SIG1                                                00000810
45    SIG(M)=B(M)                                                      00000820
      RK(M)=SIG(M)/SIG1                                                00000830
      TCON=TWOPI7*SIG1*RHOXY2                                          00000840
C--GET WO,WM FROM BO,BM                                                00000850
      WO=BO**2/TCON                                                    00000860
      WM=BM**2/TCON                                                    00000870
      DELCON=1./(SIG1*TWOPI7)                                          00000880
      IF(JSPLN.EQ.0) GO TO 49                                          00000890
C--GET PRE-SPLINED FREQ FUNCTION (NB>0 OPTION)                         00000900
      NS=0                                                             00000910
      TEM=BO/DB                                                        00000920
      ISPLN=0                                                          00000930
46    TEM=TEM*DB                                                       00000940
      IF(TEM.GE.BMTEST) GO TO 47                                       00000950
      IF(NBSKIP.GT.1) THEN                                             00000960
        IF(TEM.LT.BOSKIP.OR.TEM.GT.BMSKIP) THEN                        00000970
          ISKIP=ISKIP+1                                                00000980
          IF(MOD(ISKIP,NBSKIP).NE.0.AND.TEM*DB.LT.BMTEST) GO TO 46     00000990
        ELSE                                                           00001000
          ISKIP=-1                                                     00001010
        ENDIF                                                          00001020
      ENDIF                                                            00001030
      W=TEM**2/TCON                                                    00001040
      NS=NS+1                                                          00001050
      IF(NS.GT.200) CALL ERRMSG('SPLINED NS>200 IN FCODE',3,6,16)      00001060
      XS(NS)=ALOG(W)                                                   00001070
      YS(NS)=HZREC(W)                                                  00001080
      GO TO 46                                                         00001090
47    CALL SPLIN1(NS,0.0,XS,YS,AS,BS,CS,0,DER,W1,W2)                   00001100
      ISPLN=1                                                          00001110
```

```
C** PRESET FOR POSSIBLE ASYMPTOTIC APPROXIMATION (IF TASY.NE.0.0)        00001120
49    IF(TASY.NE.0.0) CALL SETASY(B,RHOXY2,M)                           00001130
      NEW=1                                                             00001140
C*****************************************************************     00001150
C GET ENTIRE TRANSIENT FOR ALL REAL-TIME IN X(I,1), I=1,NN,             00001160
C BEFORE APP.RES. CALC AND POSSIBLE ASYMPTOTIC APPROXIMATIONS.          00001170
C*****************************************************************     00001180
      DO 70 I=1,NN                                                      00001190
      T(I)=X(I,1)                                                       00001200
      IF(TASY.GT.0.0.AND.X(I,1).GE.TASY) GO TO 70                       00001210
C--GET TRANSIENT IMPULSE RESPONSE VIA LAGGED CONVOLUTION IN TIME.       00001220
          TRANS=.63661977*RFLAGS(0,HZREC,EPS,T0,TM,T(I),NEW)            00001230
          NEW=0                                                         00001240
        VSAVE(I)=TRANS                                                  00001250
70    CONTINUE                                                          00001260
C                                                                       00001270
C-- CHECK IF ASYMPTOTIC APPROXIMATION TO BE DONE (TASY.NE.0.0)          00001280
      IF(NN.LT.3) GO TO 77                                              00001290
      IF(TASY.LT.0.0) THEN   !<<< DETERMINE I TO USE ASYMPTOTICS        00001300
        DO I=3,NN                                                       00001310
          IF(ABS(TCON*VSAVE(I)).LE..1E-6) THEN                          00001320
            CALL GETASY(X,I-1,NN,M,VSAVE)                               00001330
            GO TO 77                                                    00001340
          ENDIF                                                         00001350
        ENDDO                                                           00001360
      ELSE IF(TASY.GT.0.0) THEN   !<<<                                  00001370
        DO I=3,NN                                                       00001380
          IF(X(I,1).GE.TASY) THEN                                       00001390
            CALL GETASY(X,I-1,NN,M,VSAVE)                               00001400
            GO TO 77                                                    00001410
          ENDIF                                                         00001420
        ENDDO                                                           00001430
      ENDIF             !<<<                                          00001440
C                                                                       00001450
C--CHECK IF RAMP ADJUSTMENT TO BE MADE (TOFF.NE.0.0)                    00001460
77    IF(TOFF.NE.0.0) CALL RAMPADJ(X,VSAVE,TOFF,NN,1.,1.)               00001470
C--IF IOPT>0, THEN CONVERT COMPUTED VSAVE(I) TO APP.RES.                00001480
      IF(IOPT.EQ.1) THEN                                                00001490
C**GET APP.RES. (SEE C**END OF "IF(IOPT.EQ.1) THEN" BELOW)              00001500
      CON2=3.1415927E-7*SIG1                                            00001510
C--GET DK(K),K=1,NK AND STORE IN COMMON/PASS2/                          00001520
      CALL DCOEF()                                                      00001530
      DO 68 I=1,NN                                                      00001540
        TI=T(I)/SCALE2                                                  00001550
        TSEC=T(I)                                                       00001560
        TRANS=VSAVE(I)                                                  00001570
        VI=CON*TRANS                                                    00001580
        CON3=-(CON1/TI)/SCAL                                            00001590
        ALP=SQRT(CON2/TI)                                               00001600
        SGN=-1.                                                         00001610
        FAC=1.                                                          00001620
        SUMK=0.0                                                        00001630
        DO K=1,NK                                                       00001640
          SGN=-SGN                                                      00001650
          FAC=FAC*K                                                     00001660
```

```
          CK=SGN*(2.-6./(2.*K+3.))/FAC                                    00001670
          TERM=CK*DK(K)*ALP**(2*K+1)                                      00001680
          CALL ERRTST(88,II)                                              00001690
          IF(II.EQ.1) GO TO 150                                           00001700
         BCOEF(K)=CON3*CK*DK(K)                                           00001710
          MK=K                                                            00001720
          SUMK=SUMK+TERM                                                  00001730
          ATERM=ABS(TERM)                                                 00001740
          IF(ATERM.LT.EPS2) THEN                                          00001750
            GO TO 160                                                     00001760
          ELSE IF(ATERM.GT.1.E9) THEN                                     00001770
            GO TO 150                                                     00001780
          ENDIF                                                           00001790
        ENDDO                                                             00001800
150         APPRES=RSIG1                                                  00001810
            GO TO 170                                                     00001820
160   NDEG=2*MK+1                                                         00001830
      NDEG1=NDEG+1                                                        00001840
      K=MK                                                                00001850
      DO II=1,NDEG                                                        00001860
        ACOEF(II)=0.0                                                     00001870
      ENDDO                                                               00001880
      DO II=1,NDEG-2,2                                                    00001890
        ACOEF(II)=BCOEF(K)                                                00001900
        K=K-1                                                             00001910
      ENDDO                                                               00001920
      ACOEF(NDEG1)=-VI                                                    00001930
C*********************************************************************  00001940
C CALL I.M.S.L. ROUTINE ZPOLR (OR EQUIV SUBSTITUTION) TO COMPUTE ALL      00001950
C COMPLEX ROOTS OF A POLYNOMIAL OF DEG=NDEG WITH REAL COEF'S, WHERE       00001960
C ACOEF(I),I=1,NDEG ARE THE REAL*4 COEF'S, AND ZROOT(I),I=1,NDEG ARE      00001970
C THE COMPLEX ROOTS ON RETURN FROM ZPOLR (SEE IMSL REF IN THZRECT DOC)    00001980
C*********************************************************************  00001990
      CALL ZPOLR(ACOEF,NDEG,ZROOT,IER)                                    00002000
C*********************************************************************  00002010
      IF(IER.GT.0) THEN                                                   00002020
        APPRES=RSIG1                                                      00002030
        GO TO 170                                                         00002040
      ENDIF                                                               00002050
      KK=0                                                                00002060
      DO K=1,NDEG                                                         00002070
        IF(AIMAG(ZROOT(K)).EQ.0.0) THEN                                   00002080
          KK=KK+1                                                         00002090
          TAR(KK)=XMU0/(4.*TI*REAL(ZROOT(K))**2)                          00002100
        ENDIF                                                             00002110
      ENDDO                                                               00002120
      CALL MINMAX(TAR,KK,TEM,APPRES)                                      00002130
170   AR(I)=APPRES                                                        00002140
68      CONTINUE                                                          00002150
C--IOPT=2 IS LATE-TIME APPRES METHOD                                      00002160
      ELSE IF(IOPT.EQ.2) THEN                                             00002170
        CON2=1.6*AX*BY*XMU0                                               00002180
        DO I=1,NN                                                         00002190
          VI=ABS(CON*VSAVE(I))                                            00002200
          TSEC=T(I)                                                       00002210
```

```
          AR(I)=1.E-7*(CON2/(TSEC*VI))**.66666667/TSEC                  00002220
        ENDDO                                                           00002230
      ENDIF                                                             00002240
C**END OF "IF(IOPT.EQ.1) THEN" ABOVE FOR APP.RES.                       00002250
C                                                                       00002260
C--IF IALL=1 (DEFAULT 0), THEN WRITE "ALL" INTERMEDIATE CURVES          00002270
C  (FREQ, V/I, AND APPRES) TO PLOT-TYPE FILES:                          00002280
C  'FWDTHZ.F', 'FWDTHZ.V', AND 'FWDTHZ.R', RESPECTIVELY, WHERE          00002290
C  FILE 'FWDTHZ.R' IS WRITTEN ONLY WHEN IOPT>0 HAS BEEN SELECTED.       00002300
C  (SEE W.L. ANDERSON ON USE OF PLOT FILES ON USGS VAX ONLY.)           00002310
C                                                                       00002320
C  NOTE: IALL=1 SHOULD ONLY BE USED WITH FWDTHZ--BUT NOT NLSTHZ!!       00002330
C                                                                       00002340
      IF(IALL.NE.0) THEN  !! SEE ENDIF FOR IALL BELOW                   00002350
        IF(IFIRST.EQ.1) THEN                                            00002360
          OPEN(21,FILE='FWDTHZ.F',STATUS='NEW',                         00002370
     1    CARRIAGECONTROL='LIST')                                       00002380
          OPEN(22,FILE='FWDTHZ.V',STATUS='NEW',                         00002390
     1    CARRIAGECONTROL='LIST')                                       00002400
          IF(IOPT.GT.0) OPEN(23,FILE='FWDTHZ.R',STATUS='NEW',           00002410
     1    CARRIAGECONTROL='LIST')                                       00002420
        ENDIF                                                           00002430
        WRITE(21,210) TITLE3,NS,                                        00002440
     1   (SQRT(TCON*EXP(XS(I))),YS(I),I=1,NS)                          00002450
210     FORMAT('3'/'IND.NO.(B)'/'REAL Z/Z0 FREQ RESPONSE'/              00002460
     1   A/I/(2G16.8))                                                 00002470
        WRITE(22,220) TITLE3,NN,(T(I),B(M2)*CON*VSAVE(I),I=1,NN)        00002480
220     FORMAT('3'/'TIME (SEC.)'/'V/I (VOLTS/AMP)'/A/I/(2G16.8))        00002490
        IF(IOPT.GT.0) WRITE(23,230)TITLE3,NN,(T(I),AR(I),I=1,NN)        00002500
230     FORMAT('3'/'TIME (SEC.)'/'APP.RES. (OHM-M.)'/A/I/(2G16.8))      00002510
      ENDIF  !! END OF "IF(IALL.NE.0) THEN"  ABOVE                      00002520
      IF(IDER.EQ.0) GO TO 600                                           00002530
      IFIRST=0                                                          00002540
      DO 80 J=1,M21                                                     00002550
80    BSAVE(J)=B(J)                                                     00002560
C--GET PRE-SPLINED TRANSIENT -OR- APPRES                                00002570
600   IF(IOPT.EQ.0) THEN                                                00002580
          F=B(M2)*TCON*VSAVE(IN)/SIG1                                   00002590
      ELSE IF(IOPT.EQ.-1) THEN                                          00002600
        F=B(M2)*CON*VSAVE(IN)                                           00002610
      ELSE                                                              00002620
        F=B(M2)*AR(IN)                                                  00002630
      ENDIF                                                             00002640
      RETURN                                                            00002650
800   IF(IDER.EQ.0) GO TO 600                                           00002660
C--IDER=1 EST.DER.OPTION                                                00002670
8001  IF(IFIRST.EQ.1) GO TO 35                                          00002680
      DO 802 J=1,M21                                                    00002690
        IF(B(J).NE.BSAVE(J)) GO TO 35                                   00002700
802   CONTINUE                                                          00002710
      GO TO 600                                                         00002720
      END                                                               00002730
      SUBROUTINE DCOEF()                                                00002740
C--GET DK(K),K=1,NK, WHERE ALL PARMS ARE IN COMMON/COEF/ AND /PASS2/    00002750
      COMMON/COEF/SAX,SBY,SX,SY                                         00002760
```

```
      COMMON/PASS2/DK(30),NK                                            00002770
      C=SY-SBY                                                          00002780
      D=SX-SAX                                                          00002790
      E=SY+SBY                                                          00002800
      F=SX+SAX                                                          00002810
      DK(1)=-8.*SAX*SBY                                                 00002820
      DO K=2,NK                                                         00002830
        FACJ=1.                                                         00002840
        FACK=1.                                                         00002850
        SUM1=0.0                                                        00002860
        SUM2=0.0                                                        00002870
        DO J=0,K-1                                                      00002880
          IF(J.GT.0) THEN                                               00002890
            FACJ=FACJ*J                                                 00002900
            FACK=FACK*(K-J)                                             00002910
          ENDIF                                                         00002920
          KJ=2*(K-J)-1                                                  00002930
          JJ=2*J+1                                                      00002940
          BIN=FACK/FACJ                                                 00002950
          SUM1=SUM1+BIN*(((-D)**KJ-((-F)**KJ))*(C**JJ-E**JJ))/KJ        00002960
          SUM2=SUM2+BIN*(((-C)**KJ-((-E)**KJ))*(D**JJ-F**JJ))/KJ        00002970
        ENDDO                                                           00002980
        DK(K)=SUM1+SUM2                                                 00002990
      ENDDO                                                             00003000
      RETURN                                                            00003010
      END                                                               00003020
      REAL FUNCTION HZREC(W)                                            00003030
C--COSINE-TRANSFORM KERNEL FOR RECTANGULAR LOOP FOR HZ                  00003040
C                                                                       00003050
      SAVE                                                              00003060
      CHARACTER*34 GENMSG                                               00003070
      COMPLEX AZ,BZ,CZ,DZ,HZ,ZANS,ZWORK,HZSPLN,ZERR(4),ZSUBA1,          00003080
     1 ZERO                                                             00003090
      DIMENSION SIG(10),H(9),RARG(200),ZANS(200),                       00003100
     1 RK(10),DD(9),IJREL(2,1),ZWORK(283),NEVAL(4)                      00003110
      EXTERNAL FF4,FA,FB,FC,FD                                          00003120
      COMMON/PASS/H,EPS,EPS2,Z0,SIG1,RHOXY2,W0,WM,DELCON,               00003130
     1 MXEVAL,M1,IOUT,IOUTS                                             00003140
      COMMON/MODEL/RK,DD,M                                              00003150
      COMMON/HZSET/ZANS,RARG,DEL2,DEL,RMAX,RMIN,NR,ISET                 00003160
      COMMON/DAT/AX,BY,X,Y                                              00003170
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN      00003180
      DATA PIMU0/3.947841762E-6/,NORD/1/,                               00003190
     1 ZERO/(0.,0.)/,                                                   00003200
     2 GENMSG/'; NB,MXEVAL TOO SMALL AT (X0,Y0) ?'/                     00003210
      IF(W.LT.W0) GO TO 30                                              00003220
      IF(W.GT.WM) GO TO 40                                              00003230
      IF(ISPLN.EQ.0) GO TO 10                                           00003240
C--ISPLN=1 (NB>0 OPTION) INTERPOLATE PRE-SPLINED FREQ. FUNCTION         00003250
      ALOGW=ALOG(W)                                                     00003260
      IF(ALOGW.LT.XS(1)) GO TO 30                                       00003270
      IF(ALOGW.GT.XS(NS)) GO TO 40                                      00003280
      CALL SPOINT(NS,XS,YS,AS,BS,CS,ALOGW,HZREC)                        00003290
      RETURN                                                            00003300
C--ISPLN=0 GET INTEGRATED FREQ. FUNCTION VIA CALL HANKEL, ETC.          00003310
```

```
10     DEL2=DELCON/W                                                        00003320
       DEL=SQRT(DEL2)                                                       00003330
C--IF M>1, CALL HANKEL FOR ALL LAGGED TRANSFORMS FOR R IN [RMIN,RMAX]       00003340
       IF(M.GT.1) THEN                                                      00003350
         DO I=1,M1                                                          00003360
           DD(I)=2.*H(I)/DEL                                                00003370
         ENDDO                                                              00003380
         CALL HANKEL(RMAX,NR,1,EPS,1,NORD,FF4,IJREL,ZWORK,                  00003390
      1  ZANS,RARG,NOFUN,IERR)                                              00003400
         IF(IERR.NE.0)CALL ERRMSG('IERR NOT 0 AFTER CALL HANKEL?',          00003410
      2 0,IOUT,IOUTS)                                                       00003420
       ELSE                                                                 00003430
C--SPECIAL CASE M=1 -- USE DIRECT QUADRATURE SINCE NO HANKEL TRANSFORMS      00003440
         DO I=1,NR                                                          00003450
           RARG(NR+1-I)=EXP(-.2*(I-1))*RMAX                                 00003460
           ZANS(I)=ZERO                                                     00003470
         ENDDO                                                              00003480
       ENDIF                                                                00003490
C--SETUP HZSPLN(R) FOR ALL SUBSEQUENT SPLINE INTEPOLATIONS, ETC.            00003500
       ISET=1                                                               00003510
       HZ=HZSPLN(0.0)                                                       00003520
C--GET THE 4-DEFINITE INTEGRALS BY FAST ADAPTIVE SPLINE QUADRATURE          00003530
20     IF(BY.EQ.Y) THEN                                                     00003540
         AZ=ZERO                                                            00003550
       ELSE                                                                 00003560
       AZ=-(BY-Y)*ZSUBA1(-AX,AX,EPS2,NEVAL(1),ICK,ZERR(1),FA,MXEVAL)        00003570
       IF(ICK.LT.0)                                                         00003580
      1 CALL ERRMSG('ICK<0 IN AZ AFTER ZSUBA1'//GENMSG,                     00003590
      2 0,IOUT,IOUTS)                                                       00003600
       IF(NEVAL(1).GT.MXEVAL)CALL ERRMSG(                                   00003610
      1 'NEVAL(1)>MXEVAL IN AZ AFTER ZSUBA1'//GENMSG,0,IOUT,IOUTS)          00003620
       ENDIF                                                                00003630
       IF(AX.EQ.X) THEN                                                     00003640
         BZ=ZERO                                                            00003650
       ELSE                                                                 00003660
       BZ=-(AX-X)*ZSUBA1(-BY,BY,EPS2,NEVAL(2),ICK,ZERR(2),FB,MXEVAL)        00003670
       IF(ICK.LT.0)                                                         00003680
      1 CALL ERRMSG('ICK<0 IN BZ AFTER ZSUBA1'//GENMSG,                     00003690
      2 0,IOUT,IOUTS)                                                       00003700
       IF(NEVAL(2).GT.MXEVAL)CALL ERRMSG(                                   00003710
      1 'NEVAL(2)>MXEVAL IN BZ AFTER ZSUBA1'//GENMSG,0,IOUT,IOUTS)          00003720
       ENDIF                                                                00003730
       IF(BY.EQ.-Y) THEN                                                    00003740
         CZ=ZERO                                                            00003750
       ELSE IF(Y.EQ.0.0) THEN                                               00003760
         CZ=AZ                                                              00003770
       ELSE                                                                 00003780
       CZ=-(BY+Y)*ZSUBA1(-AX,AX,EPS2,NEVAL(3),ICK,ZERR(3),FC,MXEVAL)        00003790
       IF(ICK.LT.0)                                                         00003800
      1 CALL ERRMSG('ICK<0 IN CZ AFTER ZSUBA1'//GENMSG,                     00003810
      2 0,IOUT,IOUTS)                                                       00003820
       IF(NEVAL(3).GT.MXEVAL)CALL ERRMSG(                                   00003830
      1 'NEVAL(3)>MXEVAL IN CZ AFTER ZSUBA1'//GENMSG,0,IOUT,IOUTS)          00003840
       ENDIF                                                                00003850
       IF(AX.EQ.-X) THEN                                                    00003860
```

```
        DZ=ZERO                                                          00003870
      ELSE IF(X.EQ.0.0) THEN                                             00003880
        DZ=BZ                                                            00003890
      ELSE                                                               00003900
      DZ=-(AX+X)*ZSUBA1(-BY,BY,EPS2,NEVAL(4),ICK,ZERR(4),FD,MXEVAL)      00003910
      IF(ICK.LT.0)                                                       00003920
     1 CALL ERRMSG('ICK<0 IN DZ AFTER ZSUBA1'//GENMSG,                   00003930
     2 0,IOUT,IOUTS)                                                     00003940
      IF(NEVAL(4).GT.MXEVAL)CALL ERRMSG(                                 00003950
     1 'NEVAL(4)>MXEVAL IN DZ AFTER ZSUBA1'//GENMSG,0,IOUT,IOUTS)        00003960
      ENDIF                                                              00003970
C--SUM TO GET THE FINAL HZ                                               00003980
      HZ=(AZ+BZ+CZ+DZ)/6.283185307                                       00003990
C--GET NORMALIZED RATIO REAL(HZ)/Z0                                      00004000
      HZREC=REAL(HZ)/Z0                                                  00004010
      RETURN                                                             00004020
30    HZREC=1.0                                                          00004030
      RETURN                                                             00004040
40    HZREC=0.0                                                          00004050
      RETURN                                                             00004060
      END                                                                00004070
      COMPLEX FUNCTION FF4(X)                                            00004080
C--KERNEL USED IN HZRECT IN LAMBDA(X)-DEL FORM (I.E., G=X*DEL)           00004090
      COMPLEX V1,F1,C,ONE,ZANS                                           00004100
      COMMON/HZSET/ZANS(200),RARG(200),DEL2,DEL,RMAX,RMIN,NR,ISET        00004110
      DATA ONE/(1.0,0.0)/                                                00004120
      G=X*DEL                                                            00004130
      CALL RECUR1(G,V1,F1)                                               00004140
      C=G                                                                00004150
      FF4=C*(V1*C*(ONE-F1))/((C+V1*F1)*(C+V1))                           00004160
      RETURN                                                             00004170
      END                                                                00004180
      COMPLEX FUNCTION FA(X)                                             00004190
C--INTEGRAND FOR AZ                                                      00004200
      COMPLEX HZSPLN                                                     00004210
      COMMON/DAT/AX,BY,XX,YY                                             00004220
      R=SQRT((X-XX)**2+(BY-YY)**2)                                       00004230
      FA=HZSPLN(R)/R                                                     00004240
      RETURN                                                             00004250
      END                                                                00004260
      COMPLEX FUNCTION FB(Y)                                             00004270
C--INTEGRAND FOR BZ                                                      00004280
      COMPLEX HZSPLN                                                     00004290
      COMMON/DAT/AX,BY,XX,YY                                             00004300
      R=SQRT((AX-XX)**2+(Y-YY)**2)                                       00004310
      FB=HZSPLN(R)/R                                                     00004320
      RETURN                                                             00004330
      END                                                                00004340
      COMPLEX FUNCTION FC(X)                                             00004350
C--INTEGRAND FOR CZ                                                      00004360
      COMPLEX HZSPLN                                                     00004370
      COMMON/DAT/AX,BY,XX,YY                                             00004380
      R=SQRT((X-XX)**2+(BY+YY)**2)                                       00004390
      FC=HZSPLN(R)/R                                                     00004400
      RETURN                                                             00004410
```

```
      END                                                               00004420
      COMPLEX FUNCTION FD(Y)                                            00004430
C--INTEGRAND FOR DZ                                                     00004440
      COMPLEX HZSPLN                                                    00004450
      COMMON/DAT/AX,BY,XX,YY                                            00004460
      R=SQRT((AX+XX)**2+(Y-YY)**2)                                      00004470
      FD=HZSPLN(R)/R                                                    00004480
      RETURN                                                            00004490
      END                                                               00004500
      COMPLEX FUNCTION HZSPLN(R)                                        00004510
C--SPLINE-DEFINED FUNCTION FOR ANY R IN [RMIN,RMAX], WHERE              00004520
C  COMMON/HZSET/ CONTAINS ALL PRECOMPUTED VALUES VIA HANKEL, ETC.       00004530
C                                                                       00004540
      SAVE                                                              00004550
      COMPLEX ZANS(200),CB,N33,N3,I2,Z                                  00004560
      REAL RARG(200),YR(200),YI(200),AR(200),BR(200),CR(200),D(2),      00004570
     1 AI(200),BI(200),CI(200),W1(200),W2(200),X(200)                   00004580
      COMMON/HZSET/ZANS,RARG,DEL2,DEL,RMAX,RMIN,NR,ISET                 00004590
      DATA N33/(3.,3.)/,N3/(3.,0.)/,I2/(0.,2.)/,D/2*0.0/                00004600
      IF(ISET.EQ.1) THEN                                                00004610
        DO I=1,NR                                                       00004620
          X(I)=ALOG(RARG(I))                                            00004630
          B=RARG(I)/DEL                                                 00004640
          CB=B                                                          00004650
          Z=ZANS(I)/DEL -CMPLX(0.,0.5*DEL2/RARG(I)**4)*                 00004660
     1    (N3-(N3+CB*(N33+CB*I2))*CEXP(CMPLX(-B,-B)))                   00004670
          YR(I)=REAL(Z)                                                 00004680
          YI(I)=AIMAG(Z)                                                00004690
        ENDDO                                                           00004700
        CALL SPLIN1(NR,0.0,X,YR,AR,BR,CR,0,D,W1,W2)                     00004710
        IF(NR.LT.0)CALL ERRMSG('NR<0 AFTER SPLIN1 IN HZSPLN?',0,IOUT,   00004720
     * IOUTS)                                                           00004730
        CALL SPLIN1(NR,0.0,X,YI,AI,BI,CI,0,D,W1,W2)                     00004740
        IF(NR.LT.0)CALL ERRMSG('NR<0 AFTER SPLIN1 IN HZSPLN?',0,IOUT,   00004750
     * IOUTS)                                                           00004760
        ISET=0                                                          00004770
        HZSPLN=(0.,0.)                                                  00004780
      ELSE                                                              00004790
        IF(R.LT.RMIN.OR.R.GT.RMAX)CALL ERRMSG(                          00004800
     1 'R<RMIN OR >RMAX IN HZSPLN?',0,IOUT,IOUTS)                       00004810
        RLOG=ALOG(R)                                                    00004820
        IF(RLOG.LT.X(1)) THEN                                           00004830
          ANSR=YR(1)                                                    00004840
          ANSI=YI(1)                                                    00004850
        ELSE IF(RLOG.GT.X(NR)) THEN                                     00004860
          ANSR=YR(NR)                                                   00004870
          ANSI=YI(NR)                                                   00004880
        ELSE                                                            00004890
          CALL SPOINT(NR,X,YR,AR,BR,CR,RLOG,ANSR)                       00004900
          CALL SPOINT(NR,X,YI,AI,BI,CI,RLOG,ANSI)                       00004910
        ENDIF                                                           00004920
        HZSPLN=CMPLX(ANSR,ANSI)                                         00004930
      ENDIF                                                             00004940
      RETURN                                                            00004950
      END                                                               00004960
```

```
      SUBROUTINE SUBZ(Y,X,B,PRNT,NPRNT,N,TITLE,IOUT)                    00004970
C-- INITIALIZATION ROUTINE (CALLED ONCE)                               00004980
C                                                                       00004990
C  SUBZ IS CALLED BY NLSOL AFTER THE DATA Y(I),X(I,5) ARE READ.         00005000
C  SUBZ CHECKS FOR DATA ERRORS, READS ADDITIONAL $INIT                   00005010
C  PARAMETERS, AND LOADS SOME CONSTANTS IN COMMON STORAGE...            00005020
C                                                                       00005030
C--PARAMETERS--                                                         00005040
C       Y,X,B,PRNT SAME AS IN SUBROUTINE FCODE.                         00005050
C       NPRNT=  CONTROL PARAMETERS TO USE PRNT(NPRNT) ARRAY             00005060
C               NPRNT REPRESENTS THE NO. X(I,NPRNT) VALUES              00005070
C       N=      NO. OBSERVATIONS GIVEN IN Y(N),X(N,5)                   00005080
C       TITLE=  ALPHA TITLE ARRAY (MAX. 80 CHAR'S)                      00005090
C       IOUT=   1 IF UNIT 6 AND 16 PRINT FILES USED                     00005100
C               0 IF ONLY UNIT 6 PRINT FILE USED.                       00005110
C                                                                       00005120
      CHARACTER*16 OPT(-1:2)                                            00005130
      CHARACTER*80 TITLE                                                00005140
      CHARACTER*40 TITLE3                                               00005150
      COMPLEX ZANS(200)                                                 00005160
      REAL Y(1),X(500,5),B(1),PRNT(1),SIG(10),H(9), RARG(200),          00005170
     1 RK(10),DD(9)                                                     00005180
      COMMON/PASS/H,EPS,EPS2,ZO,SIG1,RHOXY2,WO,WM,DELCON,               00005190
     1 MXEVAL,M1,IOUT6,IOUTS                                            00005200
      COMMON/FPASS/TO,TM,BO,BM,DB,BMTEST,TASY,SCALE2,SCAL,CON,TOFF,     00005210
     1 BOSKIP,BMSKIP,NBSKIP,ISKIP, M21,M2,JSPLN,NN,IFIRST,IOPT,         00005220
     2 IALL,TITLE3                                                      00005230
      COMMON/CLOSE/ICL                                                  00005240
      COMMON/PASS2/DK(30),NK                                            00005250
      COMMON/MODEL/RK,DD,M                                              00005260
      COMMON/HZSET/ZANS,RARG,DEL2,DEL,RMAX,RMIN,NR,ISET                 00005270
      COMMON/DAT/AX,BY,XO,YO                                            00005280
      COMMON/COEF/SAX,SBY,SX,SY                                         00005290
      COMMON/SPLN/FILL(1000),NS,ISPLN                                   00005300
      NAMELIST/INIT/MM,AX,BY,XO,YO,EPS,EPS2,MXEVAL,RFAC,BO,BM,NB,       00005310
     1 NBSKIP,TASY,TOFF,NK,IOPT,IALL                                    00005320
      DATA ISUBZ/0/,OPT/'Y(I) (VOLTS/AMP)','Y(I)/SIG1','APP.RES._1',    00005330
     1 'APP.RES._2'/                                                    00005340
C--PRESET (CAN OVERRIDE VIA $INIT INPUT):                               00005350
      DATA BO/.01/,NB/8/,BM/100./,MM/1/,IOPT/0/,IALL/0/,TOFF/0./,       00005360
     1 EPS/.1E-9/,EPS2/.1E-3/,RFAC/5./,MXEVAL/1000/,                    00005370
     2 IOUT6/6/,IOUTS/16/,NK/10/,IFIRST/1/,TASY/0./,NBSKIP/1/           00005380
      IF(ISUBZ.NE.0) GO TO 10                                           00005390
C--PRESET                                                               00005400
      WRITE(6,666)                                                      00005410
666   FORMAT('0** FWDTHZ IS RUNNING ... PRINT FILE IS FOR016.DAT')      00005420
      ISUBZ=1                                                           00005430
      CALL ERRSET(88,.TRUE.,.TRUE.,.FALSE.,.FALSE.,5)                   00005440
10    READ(99,INIT,END=11)                                              00005450
C   NOTE M=MM IS SET IN COMMON/MODEL/                                   00005460
      M=MM                                                              00005470
11    CALL NONBLANK(TITLE,NONBLK)                                       00005480
C//      WRITE(6,20) TITLE                                              00005490
20    FORMAT('1<FWDTHZ>:',5X,A<NONBLK>/)                                00005500
      IF(IOUT.EQ.1) WRITE(16,20) TITLE                                  00005510
```

```
C//       WRITE(6,INIT)                                                  00005520
       IF(IOUT.EQ.1) WRITE(16,INIT)                                     00005530
C--TEST $INIT PARMS                                                     00005540
       IF(MM.LT.1.OR.MM.GT.10)CALL ERRMSG('MM<1 OR >10',0,6,16)         00005550
       IF(AX.LE.0.)CALL ERRMSG('AX<=0',0,6,16)                          00005560
       IF(BY.LE.0.)CALL ERRMSG('BY<=0',0,6,16)                          00005570
       IF(NB.LT.0)CALL ERRMSG('NB<0',0,6,16)                            00005580
       IF(IOPT.LT.-1.OR.IOPT.GT.2)CALL ERRMSG(                          00005590
      1 'IOPT<-1 OR >2',0,6,16)                                         00005600
       IF(BM.LE.B0.OR.B0.LE.0.0)CALL ERRMSG(                            00005610
      1 'BM<=B0 OR B0<=0',0,6,16)                                       00005620
       IF(RFAC.LT.1.0)CALL ERRMSG('RFAC<1',0,6,16)                      00005630
       IF(NK.LT.1.OR.NK.GT.30)CALL ERRMSG('NK<1 OR >30',0,6,16)         00005640
C--TEST X(I, ) DATA BEFORE PROCEEDING                                   00005650
       DO 40 I=2,N                                                      00005660
       IF(X(I,1).LE.X(I-1,1).OR.X(I,1).LE.0.0)                          00005670
      & CALL ERRMSG('SOME X(I,1)<=0.0 OR NOT INCREASING.',7,6,16)       00005680
40     CONTINUE                                                         00005690
C--PRESET SOME GLOBAL CONSTANTS                                         00005700
       IFIRST=1                                                         00005710
       IF(IALL.NE.0) THEN                                               00005720
         TITLE3=TITLE(1:40)                                             00005730
         IF(IOPT.LE.0) THEN                                             00005740
           ICL=-1                                                       00005750
         ELSE                                                           00005760
           ICL=1                                                        00005770
         ENDIF                                                          00005780
       ELSE                                                             00005790
         ICL=0                                                          00005800
       ENDIF                                                            00005810
       NN=N                                                             00005820
       T0=0.5*X(1,1)                                                    00005830
       TM=0.5*(X(N,1)+X(N,1)*EXP(2.30258509/6.))                        00005840
       ISPLN=0                                                          00005850
       JSPLN=0                                                          00005860
       IF(NB.GT.0) JSPLN=1                                              00005870
       IF(JSPLN.EQ.1) THEN                                              00005880
        DB=EXP(2.30258509/FLOAT(NB))                                    00005890
        BMTEST=0.5*(BM+BM*DB)                                           00005900
       IF(NBSKIP.GT.1) THEN                                             00005910
         IF(NBSKIP.GE.NB) CALL ERRMSG('NBSKIP>=NB>0',0,6,16)            00005920
         B0SKIP=10.*B0                                                  00005930
         BMSKIP=BM/10.                                                  00005940
         IF(B0SKIP.GE.BMSKIP)CALL ERRMSG(                               00005950
      1 ' NBSKIP>1 AND 10.*B0>=BM/10.',0,6,16)                          00005960
         ISKIP=-1                                                       00005970
       ENDIF                                                            00005980
       ENDIF                                                            00005990
C--GET GLOBAL RMIN,RMAX                                                 00006000
       X1=X0+AX                                                         00006010
       X2=X0-AX                                                         00006020
       Y1=Y0+BY                                                         00006030
       Y2=Y0-BY                                                         00006040
       X12=X1*X1                                                        00006050
       X22=X2*X2                                                        00006060
```

```
      Y12=Y1*Y1                                                          00006070
      Y22=Y2*Y2                                                          00006080
      RARG(1)=SQRT(X12+Y22)                                              00006090
      RARG(2)=SQRT(X22+Y22)                                              00006100
      RARG(3)=SQRT(X12+Y12)                                              00006110
      RARG(4)=SQRT(X22+Y12)                                              00006120
        IF(X0.GE.-AX.AND.X0.LE.AX) THEN                                  00006130
          RARG(5)=ABS(ABS(Y0)-BY)                                        00006140
        ELSE                                                             00006150
          RARG(5)=RARG(2)                                                00006160
        ENDIF                                                            00006170
        IF(Y0.GE.-BY.AND.Y0.LE.BY) THEN                                  00006180
          RARG(6)=ABS(ABS(X0)-AX)                                        00006190
        ELSE                                                             00006200
          RARG(6)=RARG(1)                                                00006210
        ENDIF                                                            00006220
      CALL MINMAX(RARG,6,RMIN,RMAX)                                      00006230
      IF(RMIN.EQ.0.0)CALL ERRMSG('RMIN=0--(X0,Y0) ON WIRE?',0,6,16)      00006240
      RMIN=RMIN/RFAC                                                     00006250
      RMAX=RFAC*RMAX                                                     00006260
      NR=AINT(5.*ALOG(RMAX/RMIN))+3                                      00006270
      IF(NR.GT.200)CALL ERRMSG(                                          00006280
     1 'COMPUTED NR>200 -- CHECK X0,Y0 LOCATION OR RFAC',0,6,16)         00006290
      IF(X0.EQ.AX.OR.X0.EQ.-AX) THEN                                     00006300
        IF(Y0.GE.-BY.AND.Y0.LE.BY)CALL ERRMSG('X0,Y0 ON WIRE?',0,6,16)   00006310
      ELSE IF(Y0.EQ.BY.OR.Y0.EQ.-BY) THEN                                00006320
        IF(X0.GE.-AX.AND.X0.LE.AX)CALL ERRMSG('X0,Y0 ON WIRE?',0,6,16)   00006330
      ENDIF                                                              00006340
C--DETERMINE PRIMARY Z0 FIELD FOR THIS (X0,Y0)                           00006350
      XX2=X0*X0                                                          00006360
      YY2=Y0*Y0                                                          00006370
      RHOXY2=XX2+YY2                                                     00006380
      AA=1.2732395*AX*BY                                                 00006390
      IF(RHOXY2.EQ.0.0) THEN                                             00006400
        RHOXY2=AA                                                        00006410
      ELSE IF(RHOXY2.LT.AA) THEN                                         00006420
        RHOXY2=(SQRT(AA)-SQRT(RHOXY2))**2                                00006430
      ENDIF                                                              00006440
      Z0=0.0                                                             00006450
      IF(Y2.NE.0.0) Z0=-(X1/RARG(1)-X2/RARG(2))/Y2                       00006460
      IF(Y1.NE.0.0) Z0=Z0+(X1/RARG(3)-X2/RARG(4))/Y1                     00006470
      IF(X1.NE.0.0) Z0=Z0+(Y1/RARG(3)-Y2/RARG(1))/X1                     00006480
      IF(X2.NE.0.0) Z0=Z0-(Y1/RARG(4)-Y2/RARG(2))/X2                     00006490
      Z0=-Z0/12.566371                                                   00006500
        CON=-1.256637061435917E-6*Z0                                     00006510
      IF(IOPT.EQ.1) THEN                                                 00006520
C--USE SCALE MODEL APPROACH (BEFORE&AFTER) FOR LATER APPRES CALC.        00006530
        SCALE=MAX(AX,BY,X0,Y0)                                           00006540
        SCALE2=SCALE*SCALE                                               00006550
        SCAL=SCALE2*SCALE                                                00006560
        SAX=AX/SCALE                                                     00006570
        SBY=BY/SCALE                                                     00006580
        SX=X0/SCALE                                                      00006590
        SY=Y0/SCALE                                                      00006600
      ENDIF                                                              00006610
```

```
C//       WRITE(6,50)                                                    00006620
      IF(IOUT.EQ.1) WRITE(16,50)                                         00006630
50    FORMAT(//' PARAMETER ORDER--'/)                                    00006640
      M1=MM-1                                                            00006650
      M21=2*MM-1                                                         00006660
      M2=M21+1                                                           00006670
C//       WRITE(6,110) (I,I,I=1,MM)                                      00006680
      IF(IOUT.EQ.1) WRITE(16,110) (I,I,I=1,MM)                           00006690
110   FORMAT(5X,I3,6X,6HSIGMA(,I3,1H))                                   00006700
      IF(MM.EQ.1) GO TO 132                                              00006710
      DO 120 I=1,M1                                                      00006720
      J=MM+I                                                             00006730
      IF(IOUT.EQ.1) WRITE(16,130) J,I                                    00006740
C//120   WRITE(6,130) J,I                                                00006750
120   CONTINUE                                                           00006760
130   FORMAT(5X,I3,6X,6HTHICK(,I3,1H))                                   00006770
C//132   WRITE(6,131) M2,M2,OPT(IOPT)                                    00006780
132   CONTINUE                                                           00006790
131   FORMAT(5X,I3,10X,'B(',I3,') SHIFT PARAMETER IN: B(2*MM)*',A)       00006800
      IF(IOUT.EQ.1) WRITE(16,131) M2,M2,OPT(IOPT)                        00006810
      NPRNT=2                                                            00006820
      RETURN                                                             00006830
      END                                                                00006840
      SUBROUTINE GETASY(X,I,NN,M,VSAVE)                                  00006850
C** GET ASYMPTOTIC APPROXIMATION STARTING AT TIME X(I), 1<I<=NN.          00006860
C   THIS WILL REPLACE VSAVE(II),II=I,NN WITH ASYMPTOTIC SOLUTION;        00006870
C   ASSUMES THAT COMMON/ASY/ HAS BEEN PRESET BY A PREVIOUS               00006880
C   CALL TO SETASY(); ALSO, ASSUMES THE REAL TIMES X(I) AND              00006890
C   VSAVE() PROPERLY COMPUTED AND STORED VIA INTEGRATION RFLAGS().       00006900
C                                                                        00006910
      REAL X(NN),VSAVE(NN)                                               00006920
      COMMON/ASY/TMP(9),TAUCON,HS,S,S1,S2,S3Q                            00006930
C                                                                        00006940
      DO II=I,NN                                                         00006950
        TAU=TAUCON*SQRT(X(II))                                           00006960
        IF(M.EQ.1) THEN                                                  00006970
          VASY=-20.425845*(3.1415927/TAU)**5                             00006980
          GO TO 120                                                      00006990
        ENDIF                                                            00007000
        TMP(6)=3.1415927*HS/TAU                                          00007010
        TMP(1)=TMP(6)**5                                                 00007020
        DO J=2,4                                                         00007030
          TMP(J)=TMP(J-1)*TMP(6)                                         00007040
        ENDDO                                                            00007050
        VASY=-(20.425845*S3Q*TMP(1)-128.*S*S1*TMP(2)-                    00007060
     1  1531.9384*TMP(3)*(4.*TMP(5)*S3Q*S/105.-TMP(7))+                  00007070
     2  6144.*TMP(4)*(5.*TMP(5)*S2*S1/64.+TMP(8)))/TMP(9)                00007080
120     IF(II.EQ.I) VNORM=VSAVE(I)/ABS(VASY)                             00007090
        VSAVE(II)=ABS(VASY)*VNORM                                        00007100
      ENDDO                                                              00007110
      RETURN                                                             00007120
      END                                                                00007130
      SUBROUTINE RAMPADJ(T,VSAVE,TOFF,NN,A,SHIFT)                        00007140
C--REPLACE VSAVE() WITH RAMP-TYPE INTEGRATED VOLTAGE, WHERE              00007150
C  TOFF>0 TO INDICATE RAMP OFF-TIME FOR EM-37 TYPE SYSTEM;               00007160
```

```
C  TOFF<0 TO INDICATE RAMP OFF-TIME FOR SIROTEM TYPE SYSTEM.                00007170
C                                                                          00007180
C  NOTE THAT L.SQ. NN (TOTAL NO. POINTS GIVEN) CANNOT BE CHANGED;          00007190
C  THEREFORE AN EXTRAPOLATED POINT (NN+1) IS USED TO INTEGRATE,            00007200
C  AND THEN DISCARDED (SO THAT EXACTLY NN POINTS RETURN IN VSAVE).         00007210
C  ALSO, ARRAY T() IS ACTUAL TIME IN SECONDS (I.E., X() ARRAY IN           00007220
C  NLS-PGM); THUS, USE CALL RAMPADJ(X,VSAVE,...) IN CALLING PROGRAM.       00007230
C                                                                          00007240
      DIMENSION T(1),VSAVE(1),TT(501),VV(501),                             00007250
     1 AA(501),BB(501),CC(501),D(2),W1(501),W2(501)                       00007260
      COMMON/RAMPS/TT,VV,AA,BB,CC,N                                        00007270
      EXTERNAL RAMP                                                        00007280
      DATA MAXEV0/1000/,EPS0/.001/,D/2*0.0/                                00007290
      N=NN+1                                                               00007300
      CON=1./(SHIFT*A**3)                                                  00007310
      IF(TOFF.GT.0.0) THEN                                                 00007320
        DO I=1,NN                                                          00007330
          VV(I)=ALOG(ABS(CON*VSAVE(I)))                                    00007340
          TT(I)=ALOG(T(I))                                                 00007350
        ENDDO                                                              00007360
        N1=NN-1                                                            00007370
        N2=NN-2                                                            00007380
        TT(N)=ALOG(T(NN)+2.*TOFF)                                          00007390
        X1=TT(N)-TT(N2)                                                    00007400
        X2=TT(N)-TT(N1)                                                    00007410
        X3=TT(N)-TT(NN)                                                    00007420
        X12=TT(N2)-TT(N1)                                                  00007430
        X13=TT(N2)-TT(NN)                                                  00007440
        X23=TT(N1)-TT(NN)                                                  00007450
        VV(N)=X2*X3*VV(N2)/(X12*X13)+                                      00007460
     1        X1*X3*VV(N1)/(-X12*X23)+                                     00007470
     2        X1*X2*VV(NN)/(X13*X23)                                       00007480
      ELSE IF(TOFF.LT.0.0) THEN                                            00007490
        DO I=1,NN                                                          00007500
          VV(I+1)=ALOG(ABS(CON*VSAVE(I)))                                  00007510
          TT(I+1)=ALOG(T(I))                                               00007520
        ENDDO                                                              00007530
        TT(1)=ALOG(0.001*ABS(TOFF))                                        00007540
        X1=TT(1)-TT(2)                                                     00007550
        X2=TT(1)-TT(3)                                                     00007560
        X3=TT(1)-TT(4)                                                     00007570
        X12=TT(2)-TT(3)                                                    00007580
        X13=TT(2)-TT(4)                                                    00007590
        X23=TT(3)-TT(4)                                                    00007600
        VV(1)=X2*X3*VV(2)/(X12*X13)+                                       00007610
     1        X1*X3*VV(3)/(-X12*X23)+                                      00007620
     2        X1*X2*VV(4)/(X13*X23)                                        00007630
      ENDIF                                                                00007640
10    CALL SPLIN1(N,0.,TT,VV,AA,BB,CC,0,D,W1,W2)                           00007650
      DO I=1,NN                                                            00007660
        TLO=T(I)                                                           00007670
        THI=T(I)+TOFF                                                      00007680
        TEM=QSUBA(TLO,THI,EPS0,NPTS,ICK,REL,RAMP,MAXEV0)/TOFF              00007690
        IF(NPTS.GE.MAXEV0)CALL ERRMSG(                                     00007700
     1 'NPTS>MAXEV0 AFTER QSUBA IN RAMPADJ',0,6,16)                        00007710
```

```
        VSAVE(I)=SIGN(1.,VSAVE(I))*TEM/CON                              00007720
      ENDDO                                                             00007730
      RETURN                                                            00007740
      END                                                               00007750
      FUNCTION RAMP(T)                                                  00007760
C--SPLINED LOG-LOG INTEGRAND FUNCTION USED IN RAMPADJ WHEN TOFF.NE.0     00007770
      COMMON/RAMPS/TT(501),VV(501),AA(501),BB(501),CC(501),N            00007780
      ALOGT=ALOG(T)                                                     00007790
      IF(ALOGT.LE.TT(1)) THEN                                           00007800
        RAMP=VV(1)                                                      00007810
      ELSE IF(ALOGT.GE.TT(N)) THEN                                      00007820
        RAMP=VV(N)                                                      00007830
      ELSE                                                              00007840
        CALL SPOINT(N,TT,VV,AA,BB,CC,ALOGT,RAMP)                        00007850
      ENDIF                                                             00007860
      RAMP=EXP(RAMP)                                                    00007870
      RETURN                                                            00007880
      END                                                               00007890
      SUBROUTINE RECUR1(G,V1,F1)                                        00007900
C--BACKWARD RECURRENCE FOR COMPLEX V1,F1 GIVEN REAL*4 ARGUMENT G AND:   00007910
COMMON/MODEL/ PARAMETERS:                                               00007920
C     K(10) = NORMALIZED CONDUCTIVITY ARRAY (M VALUES,WHERE K(1)=1.0).  00007930
C     D(9)  = LAYER THICKNESS ARRAY (M-1 VALUES) D=2*THICKNESS/DEL.     00007940
C     M     = NUMBER LAYERS (M.GE.1.AND.M.LE.10)                        00007950
C             SPECIAL CASE WHEN M=1 (HOMOGENEOUS--D IGNORED)            00007960
C                                                                       00007970
C--NOTE: G,K,D ARE REAL*4                                               00007980
C                                                                       00007990
C                                                                       00008000
      COMMON/MODEL/K,D,M                                                00008010
      REAL*4 K(10),D(9)                                                 00008020
      COMPLEX C,VM,V1,F1,EVD,ONE                                        00008030
      DATA ONE/(1.0,0.0)/                                               00008040
      F1=ONE                                                            00008050
      G2=G*G                                                            00008060
      VM=CSQRT(CMPLX(G2,2.0*K(M)))                                      00008070
      IF(M.EQ.1) GO TO 2                                                00008080
      J=M-1                                                             00008090
    1 V1=CSQRT(CMPLX(G2,2.0*K(J)))                                      00008100
      EVD=CEXP(-V1*D(J))                                                00008110
      C=(ONE-EVD)/(ONE+EVD)                                             00008120
      F1=(VM*F1+V1*C)/(V1+VM*F1*C)                                      00008130
      IF(J.EQ.1) GO TO 3                                                00008140
      J=J-1                                                             00008150
      VM=V1                                                             00008160
      GO TO 1                                                           00008170
    2 V1=VM                                                             00008180
    3 RETURN                                                            00008190
      END                                                               00008200
      SUBROUTINE SETASY(B,AA,M)                                         00008210
C**PRESET COMMON/ASY/ FOR POSSIBLE ASYMPTOTIC APPROXIMATION BY          00008220
C  CALL GETASY() LATER -- CALLED ONLY WHEN TASY.NE.0 (SEE               00008230
C  $INIT PARAMETER "TASY" OPTIONS AND DEFINITIONS AVAILABLE.)           00008240
C                                                                       00008250
C  INPUT PARAMETERS:                                                    00008260
```

```
C                                                                       00008270
C        B(1):B(M) CONTAINS M LAYER CONDUCTIVITIES;                     00008280
C        B(M+1):B(2*M-1) CONTAINS M-1 LAYER THICKNESSES.                00008290
C        AA=A*A, THE LOOP RADIUS SQUARED (OR Y-SEP. SQUARED).           00008300
C        M= NUMBER OF LAYERS IN MODEL (1<=M<=10).                       00008310
C                                                                       00008320
C  OUTPUT PARAMETERS STORED IN COMMON/ASY/ ON RETURN.                   00008330
C                                                                       00008340
      REAL B(1)                                                         00008350
      COMMON/ASY/TMP(9),TAUCON,HS,S,S1,S2,S3Q                           00008360
        IF(M.EQ.1) THEN                                                 00008370
          TAUCON=SQRT(6.2831853E7/B(1))                                 00008380
          RETURN                                                        00008390
        ENDIF                                                           00008400
        HS=0.0                                                          00008410
        SIGS=0.0                                                        00008420
        DO I=1,M-1                                                      00008430
          HS=HS+B(M+I)                                                  00008440
          SIGS=SIGS+B(I)*B(M+I)                                         00008450
        ENDDO                                                           00008460
        SIGS=SIGS/HS                                                    00008470
        S=B(M)/SIGS                                                     00008480
        S1=S-1.                                                         00008490
        S2=S*S                                                          00008500
        S2Q=SQRT(S)                                                     00008510
        S3Q=S2Q**3                                                      00008520
        TAUCON=SQRT(6.2831853E7/SIGS)                                   00008530
        TMP(5)=AA/(HS*HS)                                               00008540
        TMP(7)=4.*S2Q*S1*(8.*S-9.)/105.                                 00008550
        TMP(8)=S1*(-5.*S2/24.+7.*S/12.-0.25)                            00008560
        TMP(9)=12.566371*SIGS*HS**5                                     00008570
      RETURN                                                            00008580
      END                                                               00008590
      SUBROUTINE CPUTIME(I1,I2)                                         00008600
C                                                                       00008610
C  CPUTIME WRITES "ELAPSED & CPU" TIME FROM PREVIOUS "CALL SETTIME" ON  00008620
C  FORTRAN UNITS I1 (IF NOT 0) AND I2 (IF NOT 0).                       00008630
C                                                                       00008640
C  WILL EJECT FIRST IF I1>0 (OR I2>0).                                  00008650
C  DOUBLE SPACE FIRST IF I1<0 (OR I2<0).                                00008660
C                                                                       00008670
C  E.G., USE TO TIME ELAPSED & CPU TIME FOR PROGRAM OR CODE SEGMENTS AS:00008680
C                                                                       00008690
C     CALL SETTIME    ! DON'T FORGET TO DO THIS!                        00008700
C     >>>>> THE CODE TO TIME IS HERE <<<<<  ! USUALLY A COMPLETE PROGRAM00008710
C     CALL CPUTIME(-6,16) ! OR USE I1 OR I2=0 TO OMIT WRITE.            00008720
C     >>>>> ALSO CAN USE CALL GETTIME(CPU) TO GET JUST THE CPU (SEC)    00008730
C           SINCE THE LAST CALL SETTIME WAS DONE.                       00008740
C                                                                       00008750
      SAVE                                                              00008760
      INTEGER*4 ABSVAL(4),INCRVAL(4)                                    00008770
      CALL PROCINFO(ABSVAL,INCRVAL)                                     00008780
      TIMES=SECNDS(TIME0)                                               00008790
      MIN=TIMES/60.0                                                    00008800
      SEC=AMOD(TIMES,60.0)                                              00008810
```

```
      CPUSEC=INCRVAL(1)*.01                                              00008820
      IMIN=CPUSEC/60.0                                                   00008830
      CSEC=AMOD(CPUSEC,60.0)                                             00008840
      PCPU=100.*(CPUSEC/TIMES)                                           00008850
      IF(I1.NE.0) THEN                                                   00008860
        IF(I1.GT.0) THEN                                                 00008870
          J=1                                                            00008880
        ELSE                                                             00008890
          J=0                                                            00008900
        ENDIF                                                            00008910
      WRITE(IABS(I1),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,          00008920
     1 (INCRVAL(I),I=2,4)                                                00008930
60    FORMAT(I1,65('$')/' TOTAL "ELAPSED" TIME=',F16.2,' SEC. (',        00008940
     1 I4,' MIN.',F6.2,' SEC.)'/                                         00008950
     2 ' CPU_TIME=',F15.2,' SEC. (',I4,' M. ',F5.2,                      00008960
     1 ' S.)   CPU % =',F6.2,'%'/                                        00008970
     3 ' BUF.I/O_COUNT=',I10/                                            00008980
     4 ' DIR.I/O_COUNT=',I10/                                            00008990
     5 ' PAGE_FAULTS=',2X,I10/                                           00009000
     6 ' ',65('$')//)                                                    00009010
      ENDIF                                                              00009020
      IF(I2.NE.0) THEN                                                   00009030
        IF(I2.GT.0) THEN                                                 00009040
          J=1                                                            00009050
        ELSE                                                             00009060
          J=0                                                            00009070
        ENDIF                                                            00009080
      WRITE(IABS(I2),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,          00009090
     1 (INCRVAL(I),I=2,4)                                                00009100
      ENDIF                                                              00009110
      RETURN                                                             00009120
C** ENTRY 'CALL SETTIME'--MUST BE DONE BEFORE 'CALL CPUTIME(I1,I2)'       00009130
      ENTRY SETTIME()                                                    00009140
      TIME0=SECNDS(0.0)                                                  00009150
      CALL PROCINFO(ABSVAL,INCRVAL)                                      00009160
      RETURN                                                             00009170
C** ENTRY 'CALL GETTIME(CPU)'--TO GET CPU(SEC) SINCE LAST CALL SETTIME    00009180
      ENTRY GETTIME(CPU)                                                 00009190
      CALL PROCINFO(ABSVAL,INCRVAL)                                      00009200
      CPU=INCRVAL(1)*.01                                                 00009210
      RETURN                                                             00009220
      END                                                                00009230
      SUBROUTINE ERRMSG(MSG,ISKIP,IUNIT1,IUNIT2)                         00009240
C                                                                        00009250
C  GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780                   00009260
C                                                                        00009270
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'                                   00009280
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2          00009290
C        > 0 FOR ONE BLANK LINE BEFORE.                                  00009300
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).      00009310
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).      00009320
C                                                                        00009330
C  MESSAGES ARE WRITTEN IN THE FORM:                                     00009340
C                                                                        00009350
C  {ERRMSG}: _MSG_HERE_                                                  00009360
```

```
C                                                                       00009370
      CHARACTER*(*) MSG                                                 00009380
      I=LEN(MSG)                                                        00009390
      DO 1 J=1,2                                                        00009400
        IF(J.EQ.1) THEN                                                 00009410
       JUNIT=IUNIT1                                                     00009420
          ELSE                                                          00009430
       JUNIT=IUNIT2                                                     00009440
        ENDIF                                                           00009450
        IF(JUNIT.GT.0) THEN                                             00009460
          IF(ISKIP.EQ.0) THEN                                           00009470
            WRITE(JUNIT,2) MSG                                          00009480
            ELSE                                                        00009490
       WRITE(JUNIT,3) MSG                                               00009500
          ENDIF                                                         00009510
      ENDIF                                                             00009520
1     CONTINUE                                                          00009530
      CALL EXIT                                                         00009540
2     FORMAT(1X,'{ERRMSG}: ',A<I>)                                      00009550
3     FORMAT(/1X,'{ERRMSG}: ',A<I>)                                     00009560
      END                                                               00009570
      SUBROUTINE FWDSOL(FCODE,SUBZ)                                     00009580
C                                                                       00009590
C <FWDSOL>: GENERAL FORWARD MODELING SOLUTION & PLOT FILE12  <12/13/83> 00009600
C           USING NLS (NONLINEAR LEAST SQUARES) INVERSION SUBPROGRAMS    00009610
C           FCODE AND SUBZ--EXACTLY AS REQUIRED AND CODED FOR THE       00009620
C           ADAPTIVE NONLINEAR LEAST-SQUARES ALGORITHM "NLSOL" (REF1&2)  00009630
C           BUT AS USED IN NLSOL2 (VAX NAMELIST VERSION).               00009640
C                                                                       00009650
C** THIS IS AN INTERFACE ROUTINE WRITTEN FOR THE VAX-11/780 BY           00009660
C   W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO.              00009670
C                                                                       00009680
C   THE USER ONLY NEEDS TO WRITE SUBROUTINES FCODE AND SUBZ             00009690
C   EXACTLY AS USED IN SUBROUTINE "NLSOL" FOR INVERSE SOLUTIONS         00009700
C   (SEE DETAILS BELOW OR SEE REF2 BELOW).                              00009710
C                                                                       00009720
C** REF1:  DENNIS, J.E., ET AL, 1979, AN ADAPTIVE NONLINEAR LEAST-      00009730
C           SQUARES ALGORITHM, NTIS REPORT AD-A079-716.                 00009740
C                                                                       00009750
C   REF2:  ANDERSON, W.L., 1982, ADAPTIVE NONLINEAR LEAST-SQUARES       00009760
C           FOR CONSTRAINED OR UNCONSTRAINED MINIMIZATION PROBLEMS:     00009770
C           USGS OPEN-FILE REPT. 82-68, 65 P.                           00009780
C                                                                       00009790
C***********************************************************************00009800
C                                                                       00009810
C****  THE USER MUST DECLARE THE CALLING PARAMETERS AS EXTERNAL IN THE  00009820
C      CALLING PROGRAM (ANY DESIRED NAMES MAY BE USED).                 00009830
```

The rest of the listing (mostly secondary subprograms) has been suppressed, but is available on the distributed tape.

$