U.S. Department of the Interior
Geological Survey

Calculation of transient soundings for a coincident loop system (Program TCOLOOP)

by

Walter L. Anderson

Open-File Report 82-378

1982

DISCLAIMER

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

----

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

CONTENTS

INTRODUCTION

Program TCOLOOP is designed to compute transient (time-domain) decay sounding curves over layered earth models for a coincident (CO) loop system, assuming the quasi-static case (i.e., neglecting displacement currents). A transient derivative response (TDR) is defined as the time-derivative of the tangential electric field E(r), which is measured with a receiving loop coincident with the transmitting loop and of equal radius a>>0. Several geometric variations may exist; e.g., two-distinct but slightly offset loops, a single-loop (used both as transmitter and receiver), or a two-wire single-cable system, etc. In any case, the coincident loop system is assumed to be placed on the earth's surface in this program. The TDR-sounding is evaluated rapidly and accurately using Fourier and squared-Hankel transform digital filters developed by Anderson (1975, 1982). We assume the measurement system is driven by an "on-off" step current source of arbitrary magnitude. The CO transient voltage is computed during the off-time over any defined time range (t>0 sec.).

Background material on computing transient soundings for finite-wire sources using digital filters may be found in Kauahikaua and Anderson (1977). Program TCOLOOP is similar in design to the methods used by Anderson (1979) to compute transient soundings for horizontal coplanar loops or wire-loop systems over a stratified earth. However, some notable differences have been programmed in the CO program (e.g., improved frequency-domain approximations and late-time asymptote approximations). But for practical fields situations, the earlier techniques used by Anderson (1979) are usually satisfactory, when considering the currently available time-domain electromagnetic (TDEM) hardware accuracy and dynamic range.

Program TCOLOOP also follows quite closely the method used by Anderson (1981) for computing transient soundings for a central-induction loop system (program TCILOOP). However, in the frequency-domain, program TCILOOP used the vertical magnetic field (Hz) and a J1-filter, whereas program TCOLOOP uses the tangential electric field E(r) and a J1**2-filter (Anderson, 1982).

A summary of the general computations is given, followed by a detailed description of the program parameters and 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 transient soundings computed by varying certain model parameters; and Appendix 4 gives a source listing.

SUMMARY OF CALCULATIONS

The transient decay voltage V(t) induced in a receiving circular loop of radius a>0 (coincident with a transmitting circular loop of the same radius), placed on a horizontally stratified earth, and driven by a step (on-off) current source, can be expressed (see Anderson, 1974, p.18) as the real (Re) cosine integral,

$$V(t) = \frac{4}{\pi} C \int_0^\infty B \, \mathrm{Re}[E(B)/E_0] \cos(B^2 T) \, dB, \qquad (1)$$

where

C= any constant (usually $C = 4C_2\pi/\sigma_1 a$; see C_2 below),

a= transmitting or receiving loop radius (a>0 m.; note that a square loop of area L^2 can be used, where $L^2 = \pi a^2$),

B= induction number = a/δ,

δ= $[2/(\sigma_1 \mu_0 w)]^{1/2}$ = skin depth in layer 1,

μ_0= $4\pi*10^{-7}$ = permeability of free-space,

σ_1 = conductivity of layer 1,

w= $2\pi f$ = angular frequency (f>0 Hertz),

T= $2t/(\sigma_1 \mu_0 a^2)$ = normalized time (T>0),

t= real time (t>0 seconds),

E(w)= frequency-domain response function for a CO-loop on a layered halfspace [see eq. (2) below, and also Morrison and others (1969), p.87, eq. (22) with r=a and h->0],

E_0= limit of E(w) in free-space (i.e. as σ_1 ->0):

E_0= $-iw\mu_0 IC_2$, where $C_2 = (a/2)\int_0^\infty \exp(-hx)\, J_1^2(ax)\, dx$, which is constant for a sufficiently small h>0, and for any radius a>0 [C_2=0.73092 has been evaluated for $h \leq 10^{-5}$ using subprogram SQJ1 (Anderson, 1982) in Appendix 4],

i= $(-1)^{1/2}$, and

I= driving current given as I_0*exp(iwt), $I_0 \geq 0$ amps.

For computational ease, equation (1) may be transformed to a conventional Fourier cosine integral using the substitution B^2=b,

$$V(t) = \frac{2}{\pi} C \int_0^\infty \mathrm{Re}[E(\sqrt{b})/E_0] \cos(bT)\, db. \qquad (1.1)$$

Without loss of generality, we may consider V(t)/C in eq. (1.1) to be a scaled (or amplitude shifted) TDR-function, since any constant (C) normalization will not change the transient shape. (For simplicity, the program always uses C=1; however, one may use the input parameter XNORM to apply any desired shift factor.)

Because this forward solution may be used as the basis for future inverse solutions, we seek a rapid and accurate method of evaluating the theoretical transient sounding for any desired layered model.

The method of evaluating the transient V(t) in eq. (1.1) for any time-range is by fast "lagged convolution" (Anderson, 1975) using a Fourier cosine digital filter. The squared-Hankel transform of order-1 giving E(w) is of the form (see Morrison and others, 1969, p.87, eq. (22) with r=a and h->0),

$$E(w) = -iw\mu_0 aI \int_0^\infty e(x)\ J_1^2(ax)\ dx, \qquad (2)$$

where $w = 2B^2/(\sigma_1 \mu_0 a^2)$, $i=(-1)^{1/2}$, and

the complex kernel e(x) depends on the layer parameters (conductivities and thicknesses for M-layers and M>0), and the angular frequency w>0. To insure convergence of the integral in eq. (2), it is convenient to subtract the limit 1/2 from e(x), and add an equivalent integral outside, which becomes,

$$E(w) = -iw\mu_0 aI \ \{ \int_0^\infty [e(x)-1/2]\ J_1^2(ax)\ dx + (1/2) \int_0^\infty J_1^2(ax)\ dx \}. \qquad (2.1)$$

The last term in eq. (2.1) has the limiting value C_2/a, as determined in E_0. Thus, the final normalized frequency function can be expressed as,

$$E(w)/E_0 = (a/C_2)\ \{ \int_0^\infty [e(x)-1/2]\ J_1^2(ax)\ dx \} + 1. \qquad (2.2)$$

Instead of evaluating eq. (2.2) directly during the lagged convolution in eq. (1.1), we can easily replace the normalized real function $Re[E/E_0]$ by a suitable cubic spline function with sufficient knots per decade in w (or equivalently B) to adequately define E/E_0 from some initial induction number $B0=a/\delta max$ to $Bm=a/\delta min$, where δ is the skin depth in layer 1. In fact, the asymptotic values as w->0 and $w\to\infty$ can be easily incorporated into the splined-frequency response by observing the limits,

$$\lim_{w\to 0} Re[E(w)/E_0]=1, \text{ and } \lim_{w\to\infty} Re[E(w)/E_0]=0.$$

In practice, it has been observed that most frequency-domain curve calculations are nearly identical after B>1 (e.g., see the normalized frequency response plots in Appendix 3 for models A-D, and I-L). This suggests that a recomputation savings of approximately one-half or more is possible following an initial frequency curve. Therefore, in program TCOLOOP, we have included a "threshold test" to use the rest of a previous (OLD) curve whenever a new point NEW(B) is such that

$$|[NEW(B)-OLD(B)]/OLD(B)| < THRESH,$$

where $B\geq 1$ and THRESH=1.E-7.

The above procedure for computing eq. (2.2) works very fast, since we are convolving a J1**2-filter (Anderson, 1982) with a cubic spline function, which is sufficiently accurate, provided B0 and Bm (and NB=number of B-points per decade) are adequately chosen. In practice, the input parameter default values ($B0=10^{-3}$, NB=6, $Bm=10^{5}$) are usually quite satisfactory for most field situations. A choice is generally not necessary, mainly because a dimensionless induction number range (B0,Bm) is used instead of frequency. However, one can change several program control parameters (see B0,NB,BM,EPS below) to vary the accuracy--and of course the execution speed. For example, if only moderate accuracy (but fast execution) is desired, then one may set NB<6 (it is not recommended that NB<4 be generally used); if greater accuracy (but slower execution) is desired, then one may set NB=0 (or 12) to select a "direct convolution" mode to evaluate the entire frequency function in eq. (2.2), but as controlled by the "lagged convolution" procedure for eq. (1.1). [It should be observed that a normalized transform parameter, a/Hmax, a=loop radius, Hmax=maximum layer thickness, is used; this transformation results in using moderate Hankel transform parameters, instead of using a>>0 directly as given in eq. (2.2); for a halfspace model, Hmax=a is used.]

The relationship between normalized time T and real time t (sec.) in this program are given by the formulas,

$$T=2t/\sigma_1 \mu_0 a^2, \text{ and } \quad t=\sigma_1 \mu_0 a^2 T/2.$$

The solution is now complete, except for discussing the CO asymptotic limits of V(t). It can be shown that

$$\lim_{t\to 0} [V(t)/C]=\infty, \quad \text{and} \quad \lim_{t\to\infty} [V(t)/C]=0,$$

for any horizontally layered earth model. It turns out that the non-existence of the limit at t=0 is not important for coincident loops, inasmuch as it is impossible to electronically switch at t=0 to measure the transient decay. For large (finite) times, the decay represents the transient in the semi-infinite basement (i.e., bottom layer of constant conductivity and with infinite thickness). For a one-dimensional model, the transient V(t)/C will be perturbed from a half-space response only by introduction of conductive or resistive layers over the half-space layer. However, in this case, the curves will change shape and be shifted in time, depending on the assigned layer conductivities and thicknesses (see Appendix 3 for several album-type curves for 1,2, and 3-layer models). It should also be observed that the probing depth is directly related to the loop radius a>0. Simply stated, to achieve a large probing depth, the dynamic range of the transient data must be increased for a small radius, but this range can be reduced if the radius is also increased proportional to the maximum probing depth. Of course, field logistics may prohibit very large-sized loops. Also, instrumental signal-to-noise ratios may further constrain the effective loop radius.

After considering these practical field and instrumental problems, it would probably not be worth the computational expense to try to rigorously evaluate the very late-time transient exactly whenever the dynamic range is many orders of magnitude lower than $V(t_o)/C$ for an initial switch time $t=t_o>0$. Heuristically, we determined that it is usually safe to spline-interpolate the very late-time asymptote after $V(t)/C<10^{-7}$ to the true transient limit $V(\infty)/C=0$, as long as a log-log transformation is performed first. This approach is of course very fast, and avoids "noisy" pertubations in the very late-time approximation. Generally, the transient at very late times cannot be observed accurately with present-day TDEM equipment, and therefore, do not warrant additional computational expense for practical solutions. In addition, the 10^{-7} cut-off is appropriate, since this is about the best relative error possible in the Fourier and Hankel transform digital filters (see Anderson, 1975, 1982) while using single-precision arithmetic with 32 or 36-bit floating-point words. No approximation is needed for the early-time transient, as long as parameters NB and BM are sufficiently large.

Test results using the current algorithm in program TCOLOOP have been compared with a completely different coincident (CO) program (Raiche, 1981, written communication; details are given in Raiche and Spies, 1981), and has produced stable transients that agreed to about 3-significant figures (except for the very late asymptote, which agreed to about 1-figure, but had the correct order of magnitude). We observed the new TCOLOOP algorithm ran about 10-to-30 (or more) times faster than Raiche's CO-algorithm. Raiche and Spies (1981, p.54-55) also gave a useful transformation for converting V/I transient curves into apparent conductivity (or reciprocal apparent resistivity) curves by assuming the earth to be a homogeneous halfspace. This same inverse transformation has been included in program TCOLOOP as additional output (see Appendix 2 and 3 for several examples).

PARAMETERS REQUIRED

Parameters required by program TCOLOOP are read using a FORTRAN NAMELIST simulator on the VAX (currently, VAX FORTRAN-77 Version 2.4 does not contain NAMELIST I/O; see subroutine NAMELIST in Appendix 4 for more details). The namelist name used is $PARMS. Default values are assumed whenever any parameter is omitted, except as noted otherwise. Preceding the $PARMS statement is an 80-character title.

The general input order read by program TCOLOOP is as follows:

1. Title record (always required, maximum of 80-characters).
2. $PARMS --nondefault parameters--$END. Note that $PARMS may begin in column 1 but cannot exceed column 72; records may be continued to succeeding records until the final $ or $END is encountered, where the "END" is optional.
3. Optionally, subsequent runs using changed $PARMS may be given by repeating steps 1-2, provided parameter ISTOP=0 was previously specified.

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

PROGRAM FILES

FOR005-- Title and $PARMS input parameters.

FOR006-- Output on-line terminal file (if default IOUT=6 is assumed).

FOR010-- Output solution disk file (only written if $IPCH \geq 1$).

FOR011-- Output frequency-domain solution disk plot file (only written if IPCH>1).

FOR012-- Output apparent resistivity solution disk plot file (only written if IPCH>1).

FOR013-- Output time-domain solution disk plot file (only written if IPCH>1).

FOR016-- Output disk print-file (if default IOUTS=16 is assumed).

DETAILED PARAMETER DEFINITIONS

<u>$PARMS</u> <u>parameters</u> (nondefault parameters must always be given):

M= Number of layers in the model ($1 \leq M \leq 10$; default M=1 for a homogeneous half-space).

SIG()= Array of M-layer conductivities (in mhos/m.), where SIG(1)>0 and $SIG(I) \geq 0$, for I=2,3,...,M.

H()= Array of M-1 layer thicknesses (in m.), where H(I)>0, for I=1,2,...,M-1. Array H is ignored if M=1.

A= Radius (in m.) of circular loop, where A>0 must be given. [For a square loop of side L (m.), use A=L/1.77245.]

EPS= Requested convolution integration tolerance used to compute all Fourier and Hankel transforms by digital filtering (default EPS=0.1E-9).

B0=1E-3 (default) is the lower induction number for which the E/E0 frequency response approaches 1.0 for B<B0. B0 must be given (or assumed 1E-3 by default) as a power of 10**-n. The default value is usually adequate for most models; for more accuracy in the late-time transient, B0<1E-3 can be used.

BM=1E5 (default) is the upper induction number for which the E/E0 frequency response approaches 0.0 for B>BM. BM must be given (or assumed 1E5 by default) as a power of 10**n. The default value is usually quite adequate for most models; for more accuracy in the early-time transient, BM>1E5 can be used.

NB=6 (default) represents the number of induction number points per decade (log-cycle) to evaluate the pre-splined frequency response function E(B)/E0. In general, 4≤NB≤11 is usually adequate for most applications (NB<4 is not recommended for accuracy reasons). If NB=0 (or NB>11) is specified, then a direct mode of evaluating the frequency function is used but as controlled by the outer time-integral via lagged convolution (i.e., the cosine filter using subroutine RLAG0. Note that NB=0 (or NB>11) is more accurate, but much more time-consuming than using NB<12.

TO= Initial normalized time to compute the transient, where TO>0 must be specified as a power of 10**±n. The normalized time T (called TAU in output files) and actual time (in sec.) are related by the formula: T=(2*time)/(SIG(1)*4*pi*10**-7*a*a).

TM= Maximum normalized time to compute the transient, where TM>TO must be specified as a power of 10**±n.

NT= Number of normalized time points to compute per time decade (log-cycle) between TO and TM, where NT>0 must be specified.

XNORM= Normalization factor (default 10.0) to use to shift the transient at TO. Note: both the normalized and unnormalized transient response will be printed along with a normalization of 1.0 at TO (see Appendix 2 for an example output listing).

IOUT=6 (default) is the primary print file unit number, which defaults to the users terminal (if on-line). To suppress the IOUT file output, set IOUT=0.

IOUTS=16 (default) is the secondary print-type disk file unit number. To suppress the IOUTS file output, set IOUTS=0.

IPCH= 0 (default) to ignore this output option.

IPCH= 1 to write FOR010 with the unnormalized transient response (TRANS), time (in sec.), and the apparent resistivity (APPRES) in the format (3E16.8). This option may be used to produce input data for other programs (e.g., test data for inversion routines, etc.).

IPCH=2 to write FOR010 (as in IPCH=1 above), and in addition, write files FOR011, FOR012, and FOR013 for possible plotting purposes--see the formats as used in Appendix 4 source listing, if interested.

ISTOP=1 (default) to end the run after the current problem.

ISTOP=0 to continue the run with a new title line and changed $PARMS on FOR005. The program will continue until ISTOP=1 is set on the last $PARMS or an end-of-file is encountered on FOR005.

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

EXAMPLES OF INPUT PARAMETERS

```
EXAMPLE TITLE
$PARMS M=2,SIG=.02,2,H=200; A=200,
TO=.1,NT=6,TM=100,NB=5,ISTOP=0$
MODIFIED EXAMPLE
$PARMS NB=11,A=1000,ISTOP=1$END
```

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

VAX OPERATING INSTRUCTIONS

Assuming program TCOLOOP (and all associated subprograms) was previously 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 the section PARAMETERS REQUIRED above. To assign FOR005, use the DCL command:

   ```
   $ASSIGN parameterfilename FOR005
   ```

2. If IPCH≥1 is selected, then a specific file name may be assigned to FOR010 (as in step 1); otherwise, the system will assume FOR010.DAT as a file name for FOR010 (similarly, if IPCH>1, FOR011.DAT, FOR012.DAT, and FOR013.DAT will be assumed for FOR011, FOR012, and FOR013, respectively). When IPCH=0 (default), this step may be ignored.

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

   ```
   $RUN TCOLOOP
   ```

   On the USGS system, use the command:

   ```
   $RUN [WANDERSON]TCOLOOP
   ```

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

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

ERROR MESSAGES

Most $PARMS syntactical errors are flagged and printed on files FOR006 and FOR016 by the VAX-NAMELIST simulator subroutine (see Appendix 4), and the job is aborted. If FOR005 was assigned to a disk parameter file, then correct the parameter file using any VAX editor and rerun the job (e.g., use $RUN or $SUBMIT). Other parameter errors (or omissions) are also flagged by program TCOLOOP, and the job is terminated.

REFERENCES

Anderson, W.L., 1974, Electromagnetic fields about a finite electric wire source: USGS Rept. GD-74-041, 205 p. (also available as NTIS Rept. PB-238-199).

----------, 1975, Improved digital filters for evaluating Fourier and Hankel transform integrals: USGS Rept. GD-75-012, 223 p. (also available as NTIS Rept. PB-242-800.)

Anderson, W.L., 1979, Programs TRANS_HCLOOP and TRANS_HZWIRE-- Calculation of transient horizontal coplanar loop soundings and transient wire-loop soundings: USGS Open-File Rept. 79-590, 46 p.

----------, 1981, Calculation of transient soundings for a central induction loop system (Program TCILOOP): USGS Open-File Rept. 81-1309, 80 p.

----------, 1982, Fast evaluation of squared-Hankel transforms of order-1 by linear digital filtering (Subprogram SQJ1): USGS Open-File Rept. 82-224, 13 p.

Kauahikaua, J., and Anderson, W.L., 1977, Calculation of standard transient and frequency sounding curves for a horizontal wire source of arbitrary length: USGS Rept. GD-77-007, 61 p. (also available as NTIS Rept. PB-274-119).

Morrison, H.F., Phillips, R.J., and O'Brien, D.P., 1969, Quantitative interpretation of transient electromagnetic fields over a layered halfspace: Geophys. Prosp., v. 17, p. 82-101.

Raiche, A.P., and Spies, B.R., 1981, Coincident loop transient electromagnetic master curves for interpretation of two-layer earths: Geophysics, v. 46, n. 1, p. 53-64.

Appendix 1.-- Conversion to other systems

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

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

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

(1) Variables with more than 6-characters.

(2) Use of the underscore character or dollar character in some variables and/or COMMON names.

(3) Character strings delimited by single-quote characters (e.g., 'STRING'); also, character string concatination (e.g., 'STRING1'//'STRING2').

(4) Passing variable-length character strings in subroutine calls; e.g., CHARACTER*(*) passed length character arguments.

(5) Need to suppress arithmetic or exponential underflow messages (note that a VAX-11 result is automatically set to 0.0 after any underflow--which is assumed for this program package); if the target system does not set underflows to 0.0 (and suppress warning messages), then a suitable conversion procedure must be used for proper operation of this program package.

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

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

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

Appendix 2.-- Test problem input/output listing

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

FOR005

```
TEST MODEL
$PARMS M=2,A=200,TO=.1,NT=4,TM=.1E5,
SIG=.001,.1,H=200$
```

FOR016

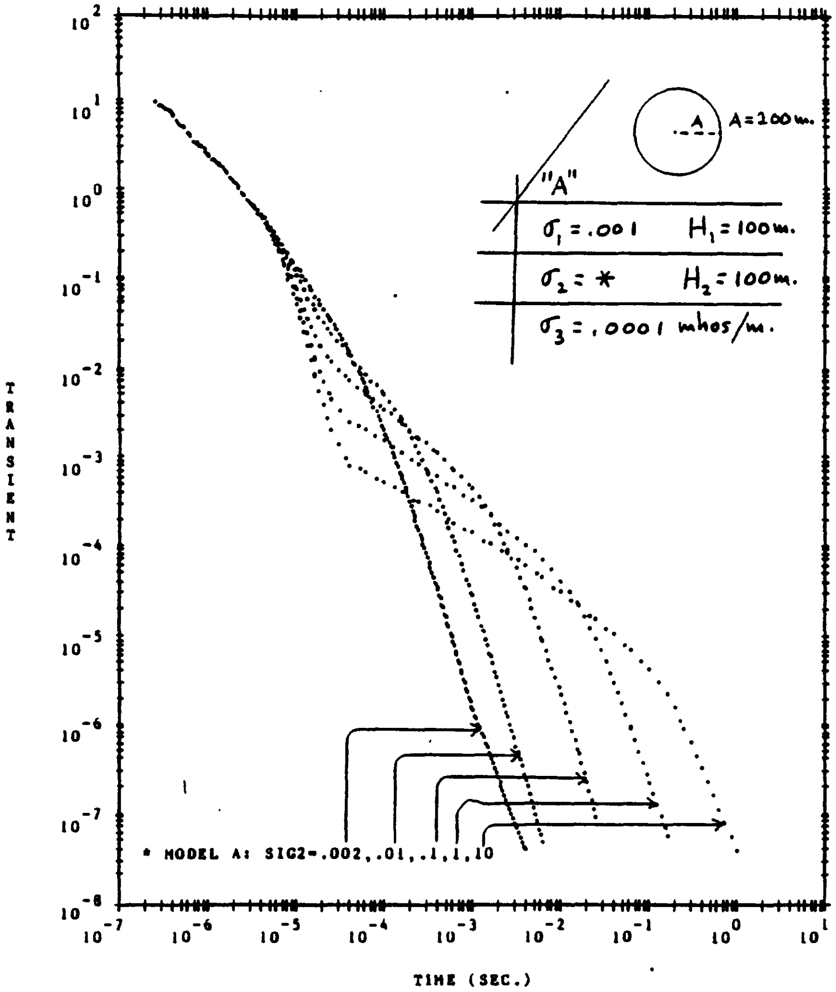
```
{TCOLOOP}:       TEST MODEL

M =  2           XNORM=0.10E+02  IPCH=    0        A= 0.2000E+03
IOUTS =  16      TO= 0.1000E+00  NT =     4        TM= 0.1000E+05  ISTOP = 1
IOUT =    6      BO= 0.1000E-02  NB =     6        BM= 0.1000E+06  EPS= 0.10E-09

SIG =  0.1000E-02  0.1000E+00  0.0000E+00  0.0000E+00  0.0000E+00
       0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00

  H =  0.2000E+03  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00
       0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00  0.0000E+00

   TAU(TO:TM)   TIME(SEC)    TRANS         TRANS(NORM)  NORM*XNORM   APP.RES.

  0.10000E+00  0.25133E-05  0.89830E+00  0.10000E+01  0.10000E+02  0.10007E+04
  0.17783E+00  0.44693E-05  0.44749E+00  0.49815E+00  0.49815E+01  0.99759E+03
  0.31623E+00  0.79477E-05  0.20415E+00  0.22726E+00  0.22726E+01  0.10096E+04
  0.56234E+00  0.14133E-04  0.76907E-01  0.85614E-01  0.85614E+00  0.11148E+04
  0.10000E+01  0.25133E-04  0.19823E-01  0.22067E-01  0.22067E+00  0.14561E+04
  0.17783E+01  0.44693E-04  0.42820E-02  0.47668E-02  0.47668E-01  0.18141E+04
  0.31623E+01  0.79477E-04  0.11996E-02  0.13354E-02  0.13354E-01  0.17172E+04
  0.56234E+01  0.14133E-03  0.70085E-03  0.78019E-03  0.78019E-02  0.93913E+03
  0.10000E+02  0.25133E-03  0.41705E-03  0.46426E-03  0.46426E-02  0.50674E+03
  0.17783E+02  0.44693E-03  0.24966E-03  0.27792E-03  0.27792E-02  0.27218E+03
  0.31623E+02  0.79477E-03  0.14162E-03  0.15765E-03  0.15765E-02  0.15208E+03
  0.56234E+02  0.14133E-02  0.74393E-04  0.82815E-04  0.82815E-03  0.89907E+02
  0.10000E+03  0.25133E-02  0.35421E-04  0.39431E-04  0.39431E-03  0.57072E+02
  0.17783E+03  0.44693E-02  0.15208E-04  0.16930E-04  0.16930E-03  0.38966E+02
  0.31623E+03  0.79477E-02  0.58785E-05  0.65440E-05  0.65440E-04  0.28573E+02
  0.56234E+03  0.14133E-01  0.20510E-05  0.22832E-05  0.22832E-04  0.22412E+02
  0.10000E+04  0.25133E-01  0.66162E-06  0.73652E-06  0.73652E-05  0.18472E+02
  0.17783E+04  0.44693E-01  0.19861E-06  0.22110E-06  0.22110E-05  0.15925E+02
  0.31623E+04  0.79477E-01  0.59422E-07  0.66149E-07  0.66149E-06  0.13719E+02
  0.56234E+04  0.14133E+00  0.18045E-07  0.20088E-07  0.20088E-06  0.11676E+02
  0.10000E+05  0.25133E+00  0.55616E-08  0.61912E-08  0.61912E-07  0.98278E+01


$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
TOTAL "ELAPSED" TIME=            7.59 SEC. (   0 MIN.  7.59 SEC.)
CPU_TIME=           7.26 SEC. (   0 M.  7.26 S.)   CPU % = 95.65%
BUF.I/O_COUNT=         2
DIR.I/O_COUNT=         4
PAGE_FAULTS=          58
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
```

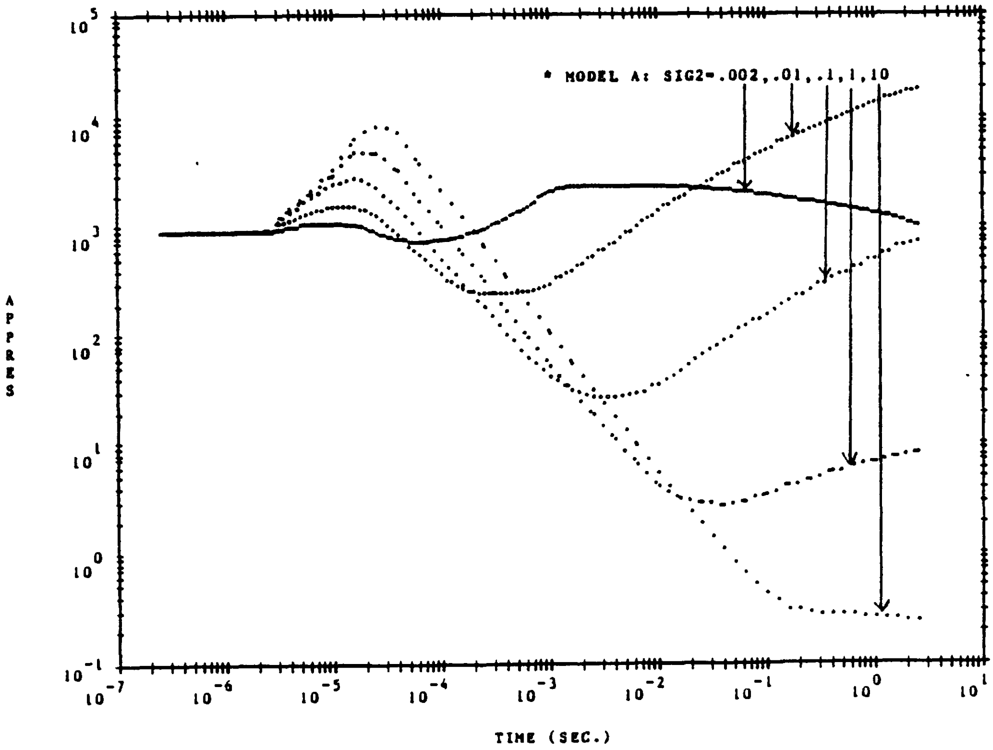
Appendix 3.-- Some sounding curve example plots

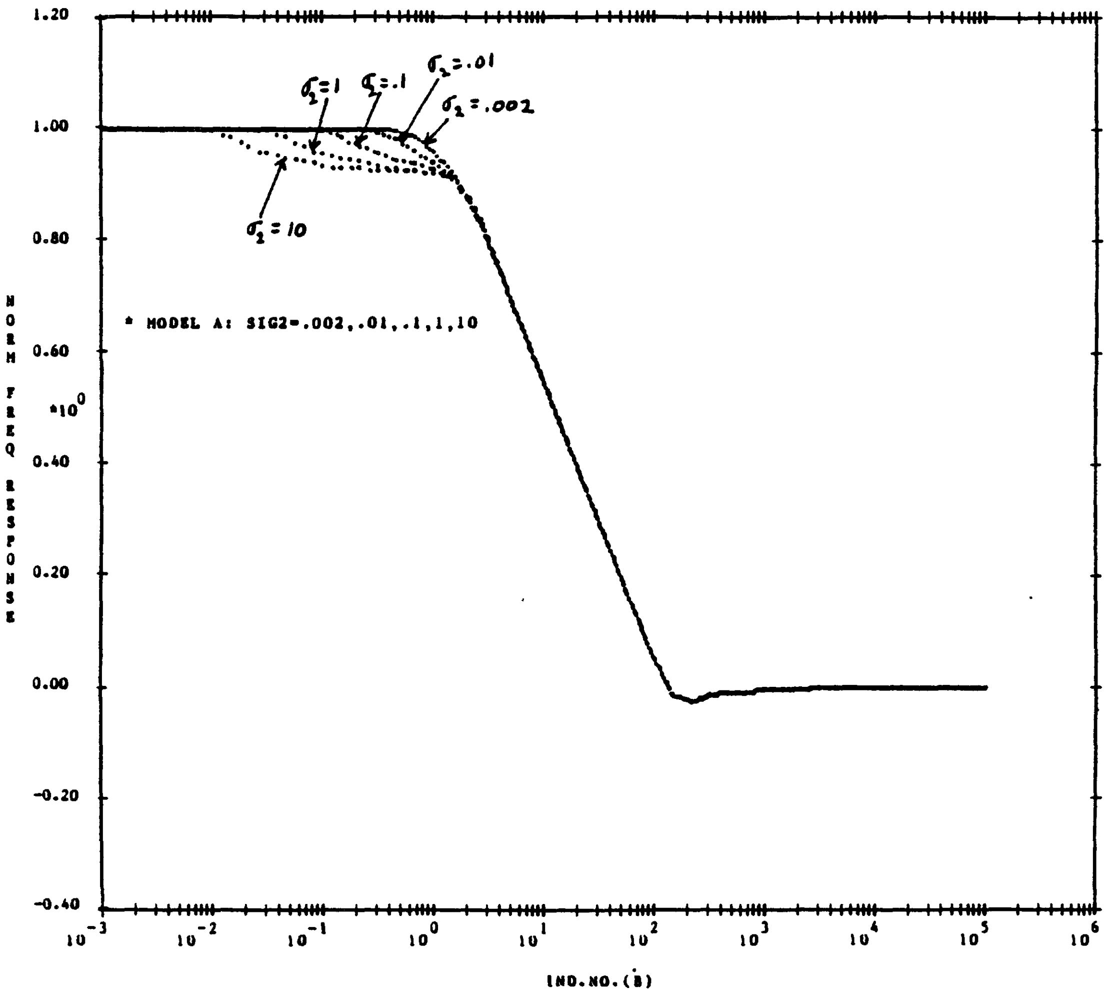
The attached plots were produced (after using IPCH>1) on an AJ-832 terminal for several layered models, and curve families, by varying certain model parameters. The beginning of each model (denoted "A","B",...) is indicated by a "model-figure insert" drawn on the unnormalized TRANSIENT* and apparent resistivity (APPRES in ohm-m.) plots, followed on a successive page with the corresponding normalized FREQUENCY response for the given model. The notation used is, hopefully, self-explainatory. [Note that the TCOLOOP models A-D and I-L are the same models as used in the TCILOOP program examples (see Anderson, 1981, p.24-39).]

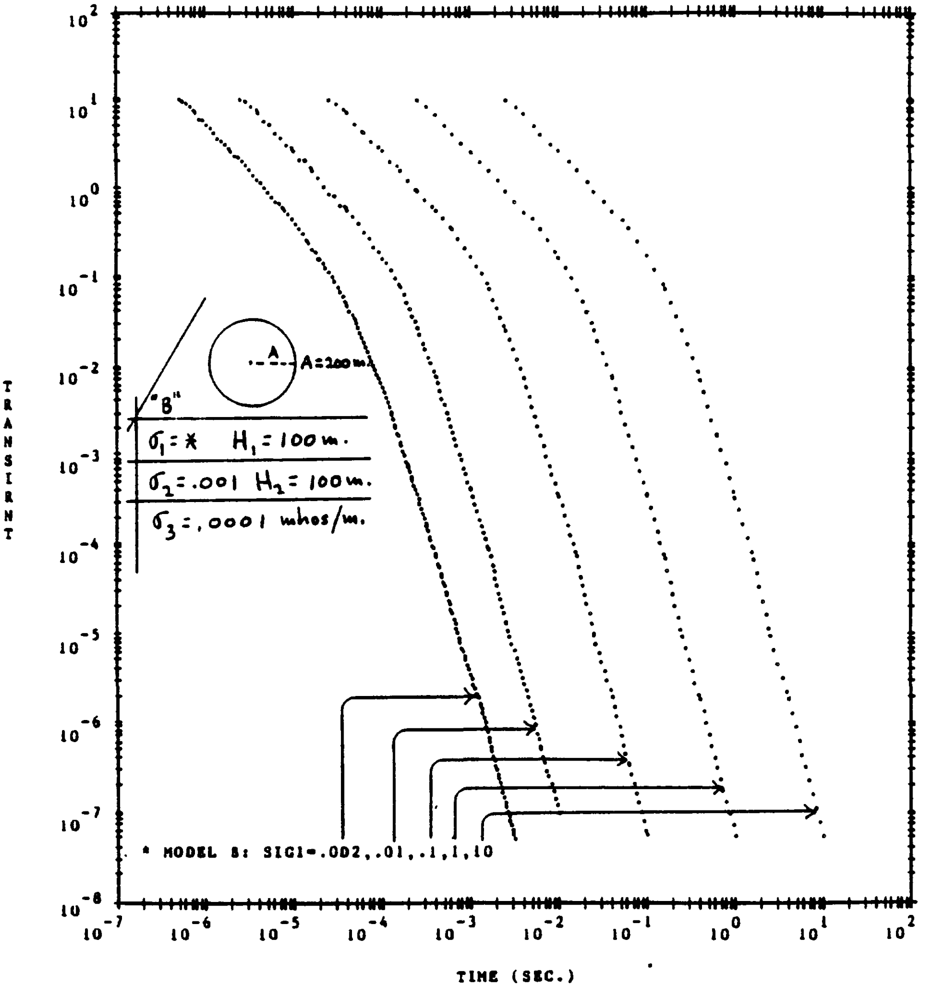
Much of the discussion in the INTRODUCTION and SUMMARY OF CALCULATIONS sections are illustrated in these plots. For example, referring to model "K", we observe that as the radius A decreases, the first deflection in each transient curve becomes progressively lower in magnitude at about 0.1 seconds. In fact, for A=100, the transient mostly "sees" the 1000m upper layer. This shows the relative importance of dynamic range versus loop radius in detecting the deeper layer interfaces.

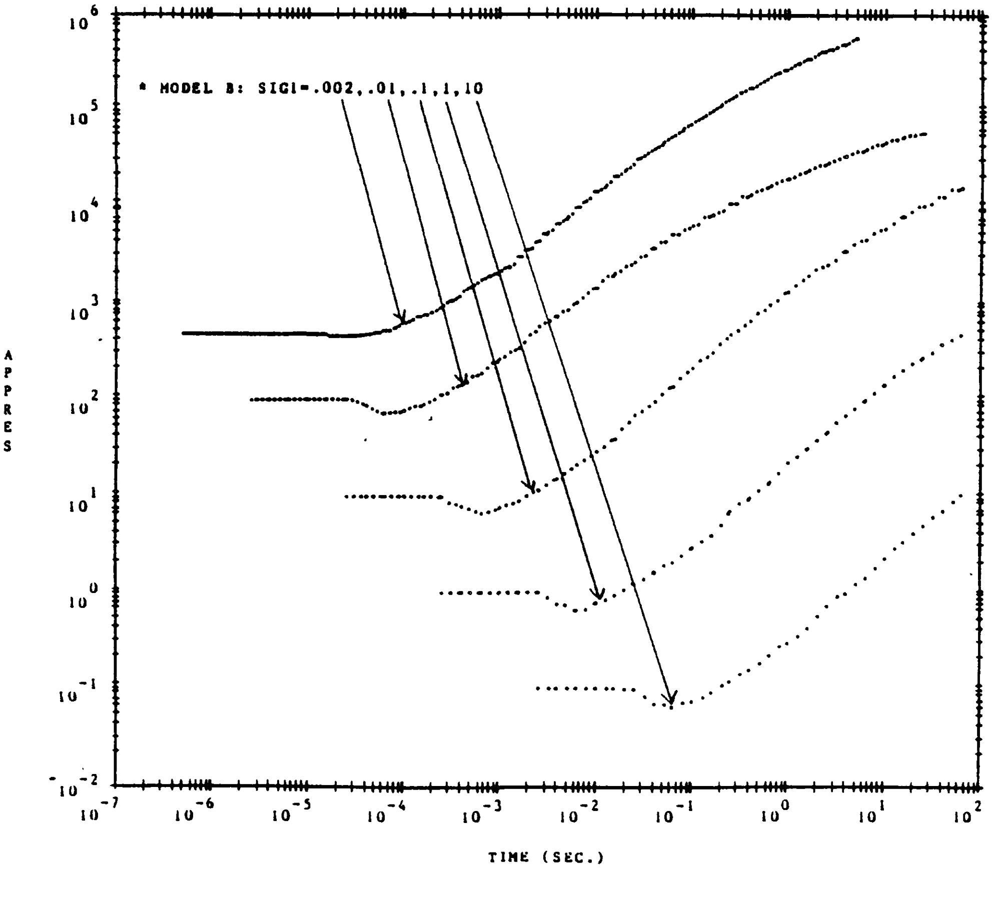
* The term "TRANSIENT" used in these plots refer to a TDR-sounding.

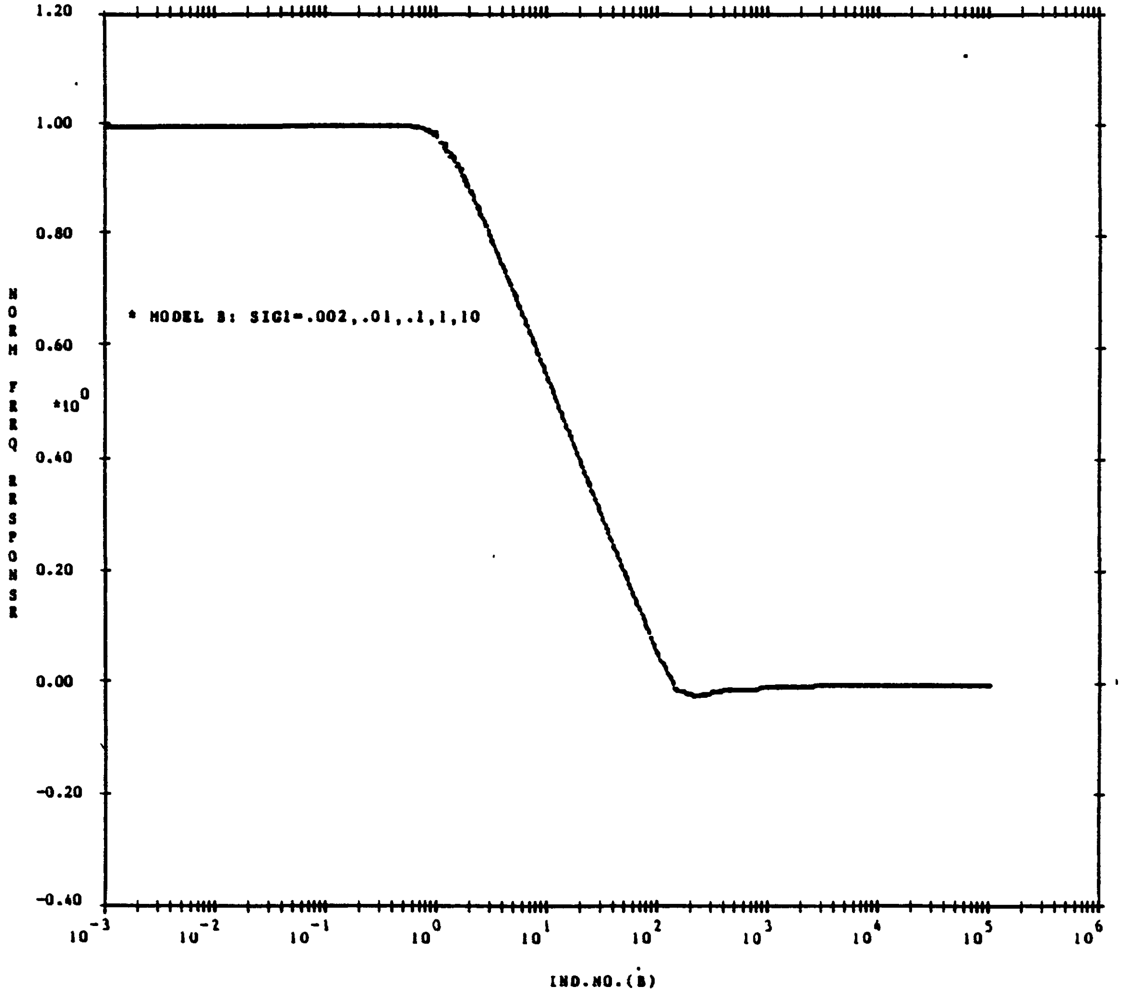

A
A=200m.
"A"
σ_1 = .001
H_1 = 100m.
σ_2 = *
H_2 = 100m.
σ_3 = .0001 mhos/m.
* MODEL A: SIG2=.002,.01,.1,1,10
TRANSIENT
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
TIME (SEC.)

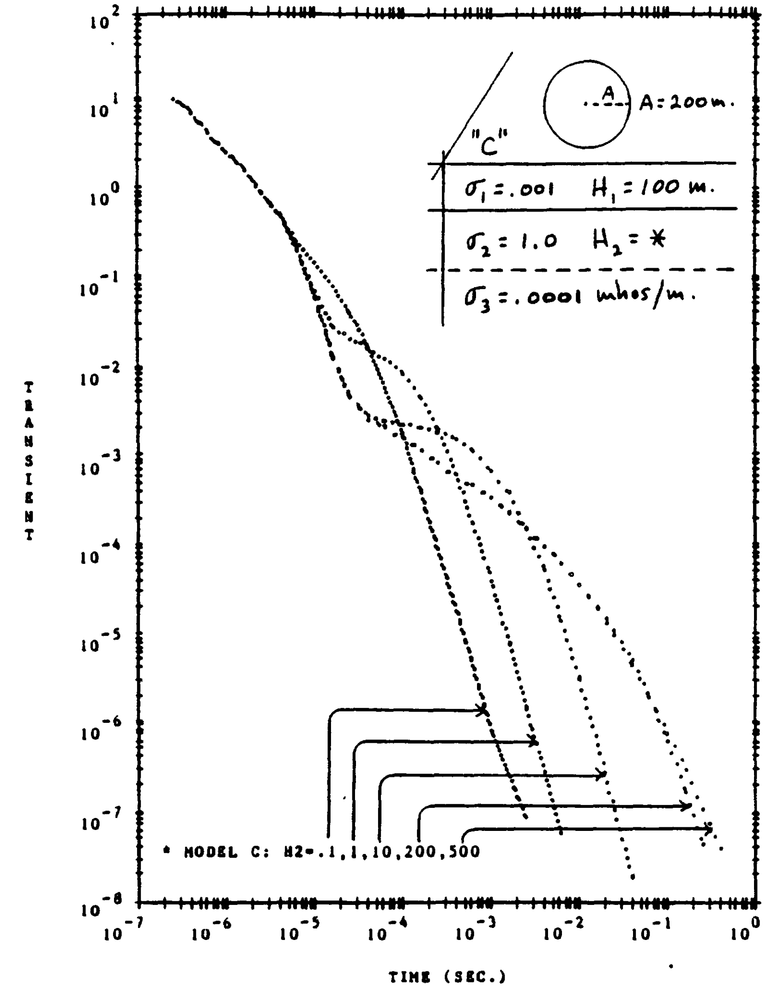

* MODEL A: SIG2=.002,.01,.1,1,10
APPRES
10^{5}
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
TIME (SEC.)

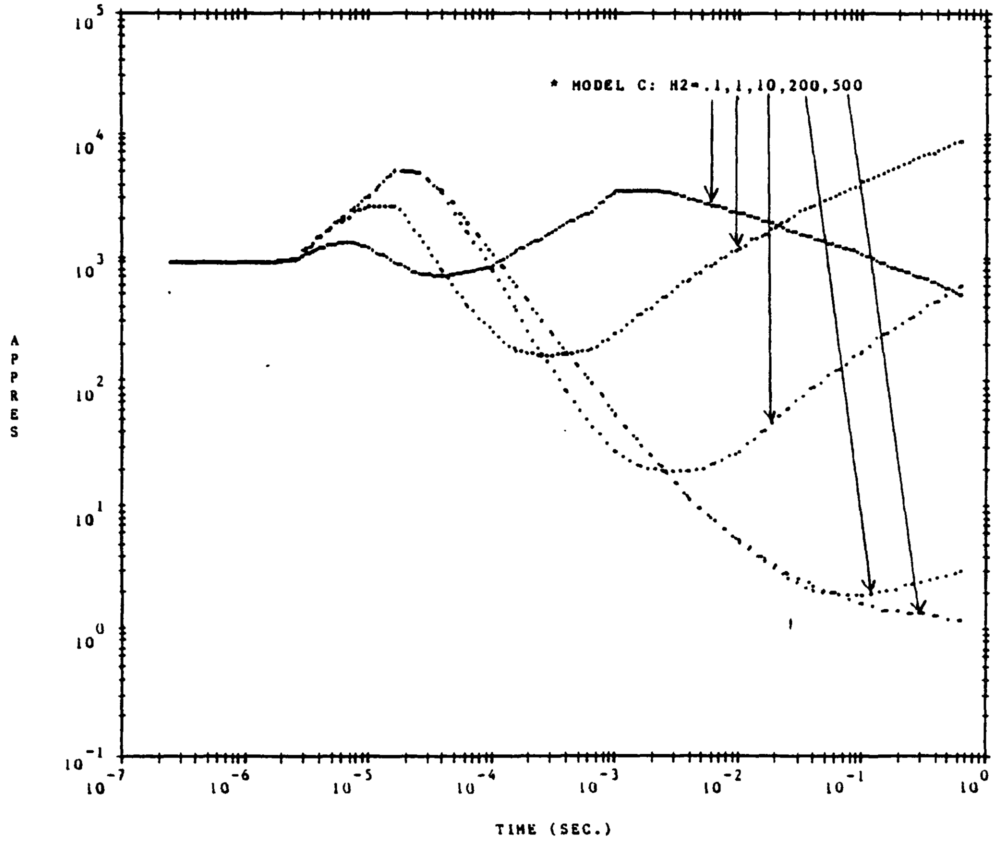

1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
*10^0
NORM FREQ RESPONSE
$\sigma_2 = 1$
$\sigma_2 = .1$
$\sigma_2 = .01$
$\sigma_2 = .002$
$\sigma_2 = 10$
* MODEL A: SIG2=.002,.01,.1,1,10
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
10^{6}
IND.NO.(B)

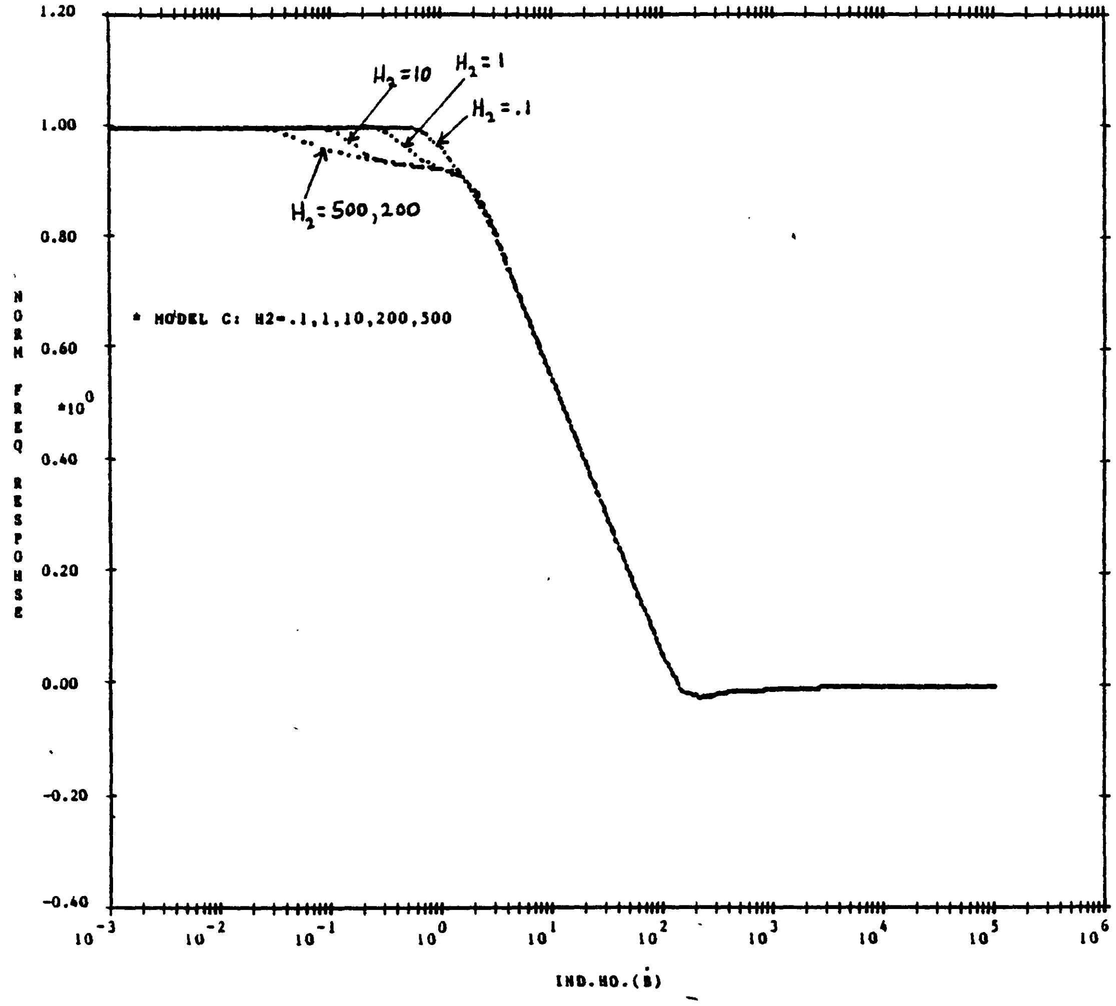

TRANSIENT
TIME (SEC.)
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
10^2
A
A=200m
"B"
σ1=* H1=100m.
σ2=.001 H2=100m.
σ3=.0001 mhos/m.
* MODEL B: SIGI=.002,.01,.1,1,10

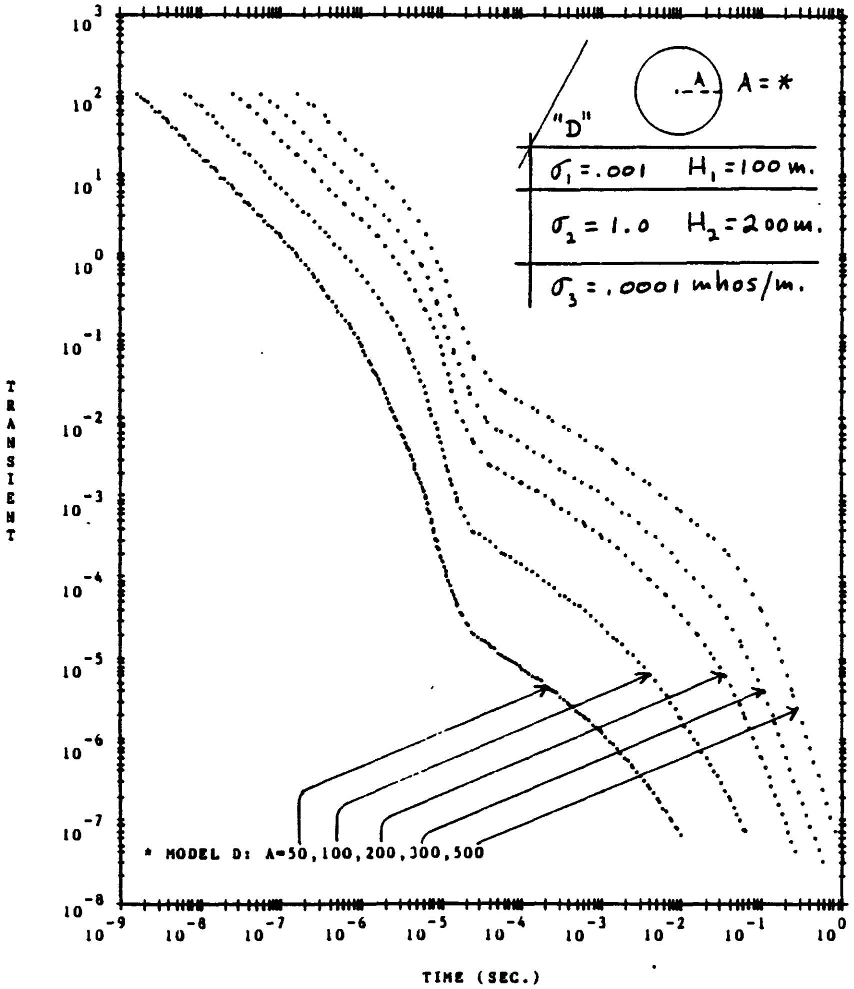

APPRES
TIME (SEC.)
10^6
10^5
10^4
10^3
10^2
10^1
10^0
10^-1
10^-2
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
10^2
* MODEL B: SIGI=.002,.01,.1,1,10

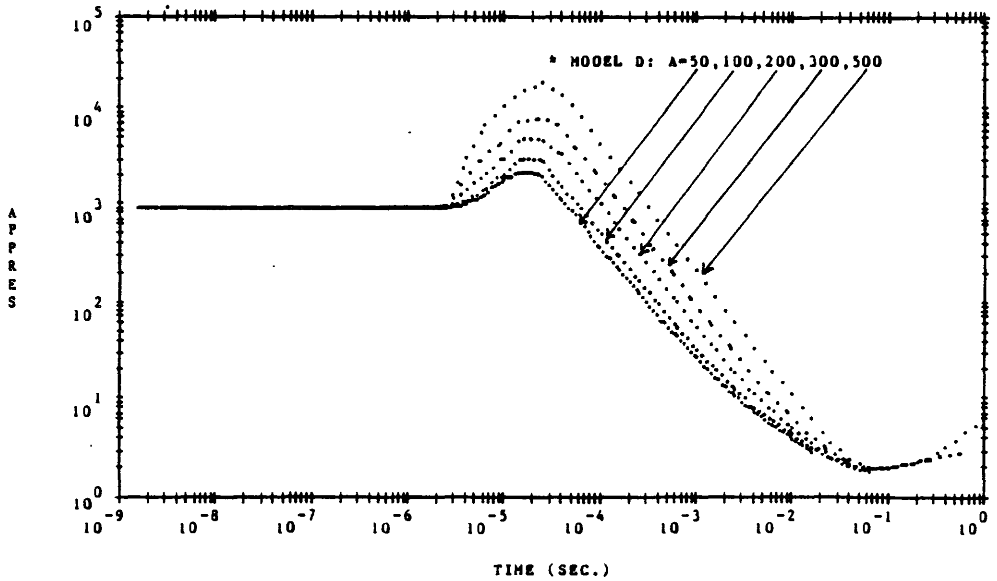

1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
* MODEL B: SIG1=.002,.01,.1,1,10
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3
10^4
10^5
10^6
IND.NO.(B)

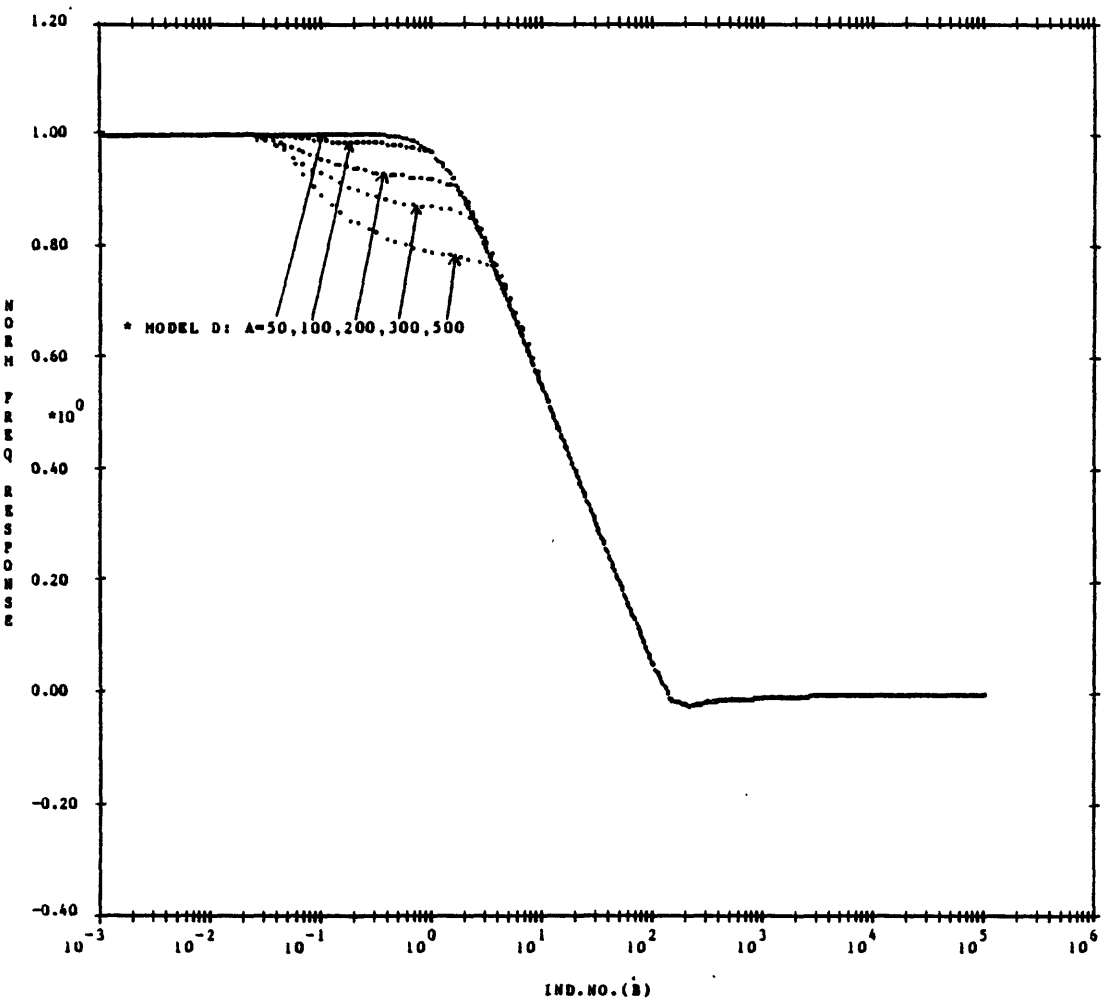

A
A: 200m.
"C"
σ₁ = .001 H₁ = 100 m.
σ₂ = 1.0 H₂ = *
σ₃ = .0001 mhos/m.
* MODEL C: H2=.1,1,10,200,500
TRANSIENT
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
TIME (SEC.)

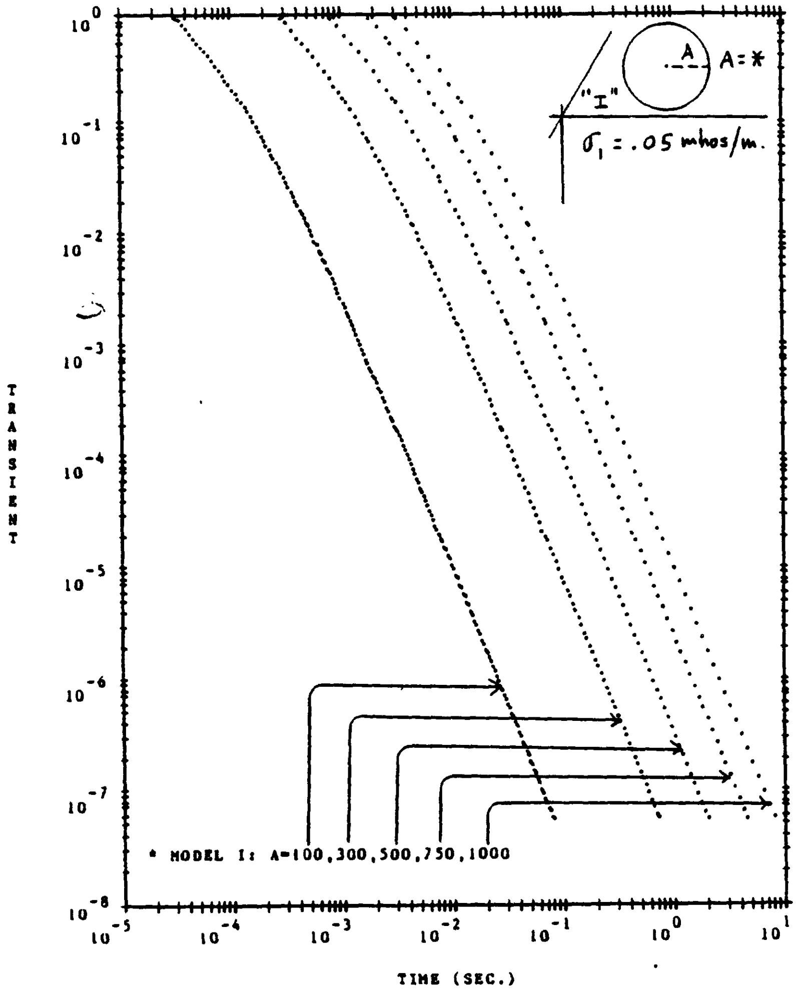

* MODEL C: H2=.1,1,10,200,500
APPRES
10^5
10^4
10^3
10^2
10^1
10^0
10^-1
10^-7
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
TIME (SEC.)

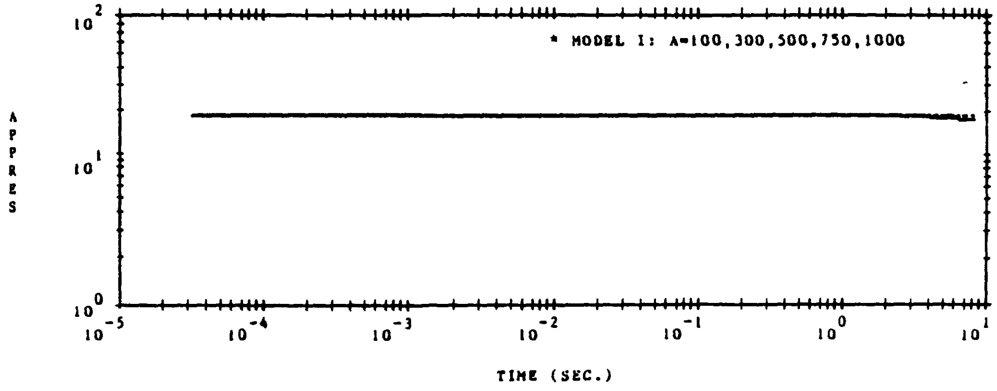

1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
$H_2=10$
$H_2=1$
$H_2=.1$
$H_2=500,200$
* MODEL C: H2=.1,1,10,200,500
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
10^{6}
IND.NO.(B)

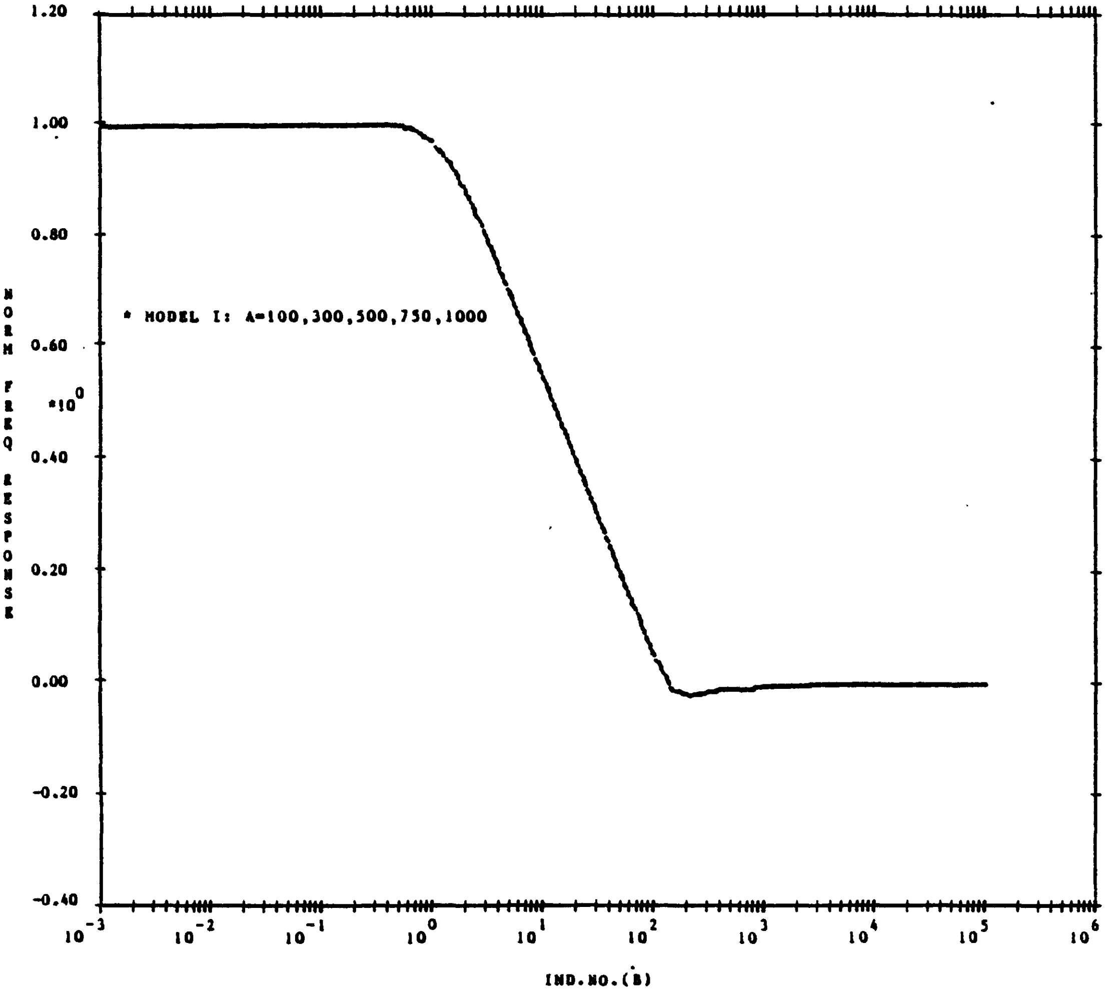

"D"
A = *
A
σ_1 = .001 H_1 = 100 m.
σ_2 = 1.0 H_2 = 200 m.
σ_3 = .0001 mhos/m.
* MODEL D: A=50,100,200,300,500
TRANSIENT
TIME (SEC.)
10^{3}
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{0}

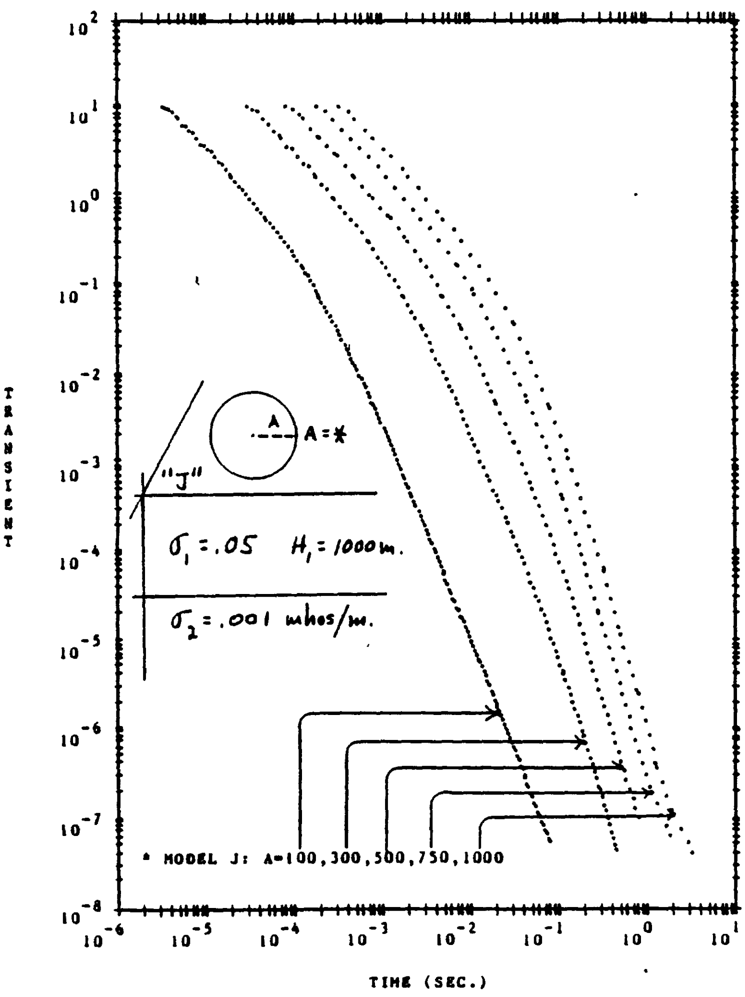

* MODEL D: A=50,100,200,300,500
APPRES
TIME (SEC.)
10^{5}
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-9}
10^{-8}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}

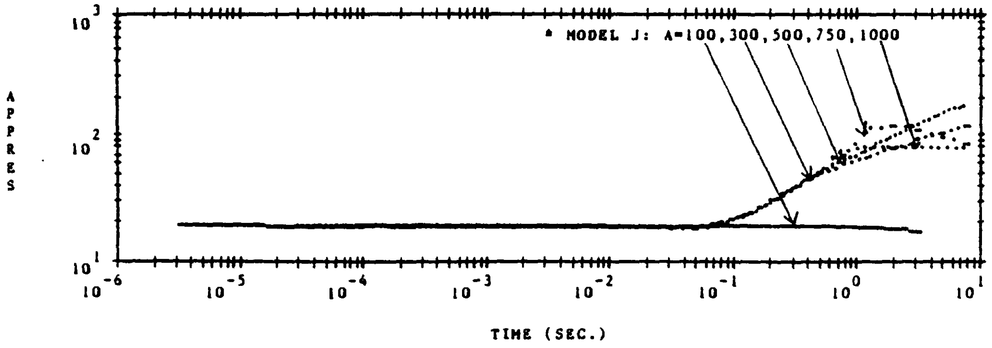

1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
* MODEL D: A=50,100,200,300,500
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
10^{2}
10^{3}
10^{4}
10^{5}
10^{6}
IND.NO.($\dot{B}$)

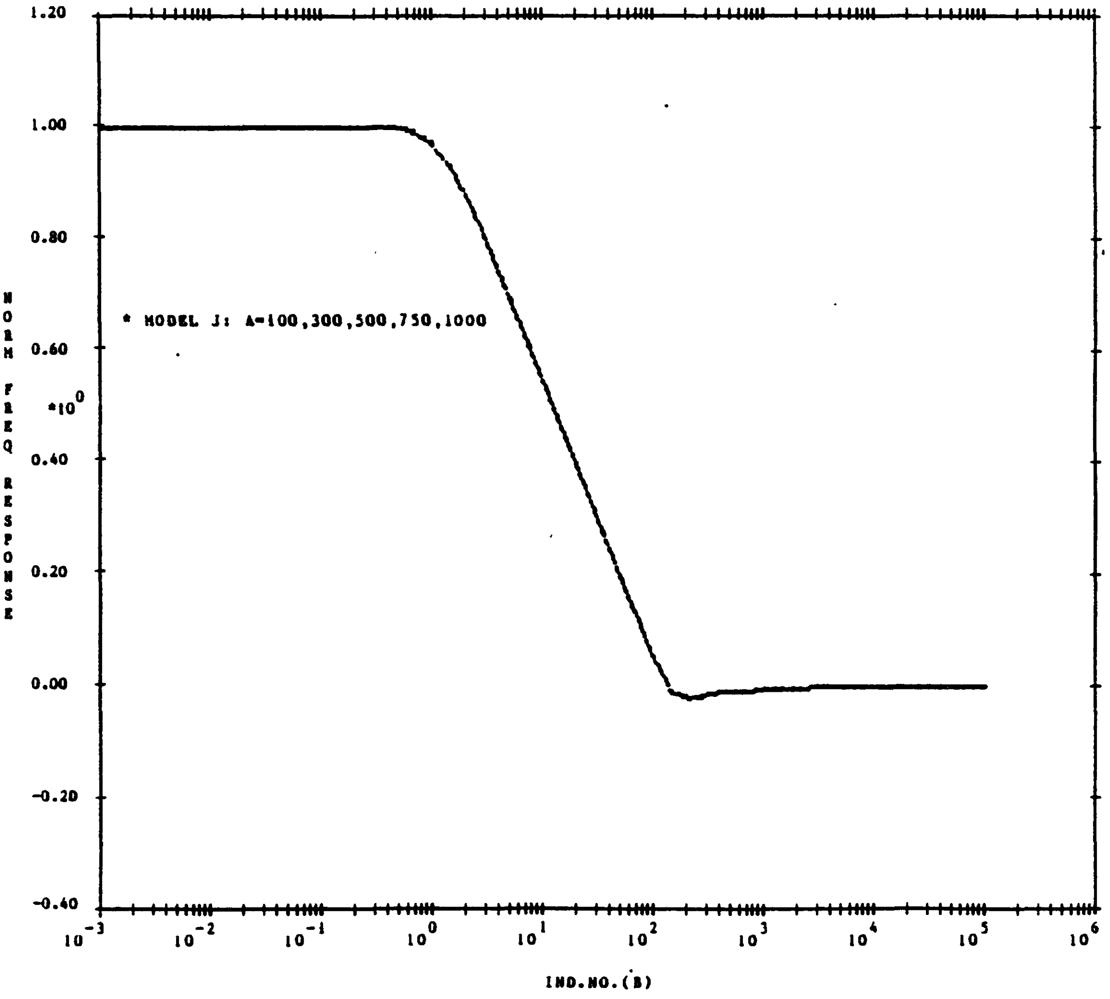

A
A=*
"I"
σ₁ = .05 mhos/m.
* MODEL I: A=100,300,500,750,1000
TRANSIENT
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
TIME (SEC.)

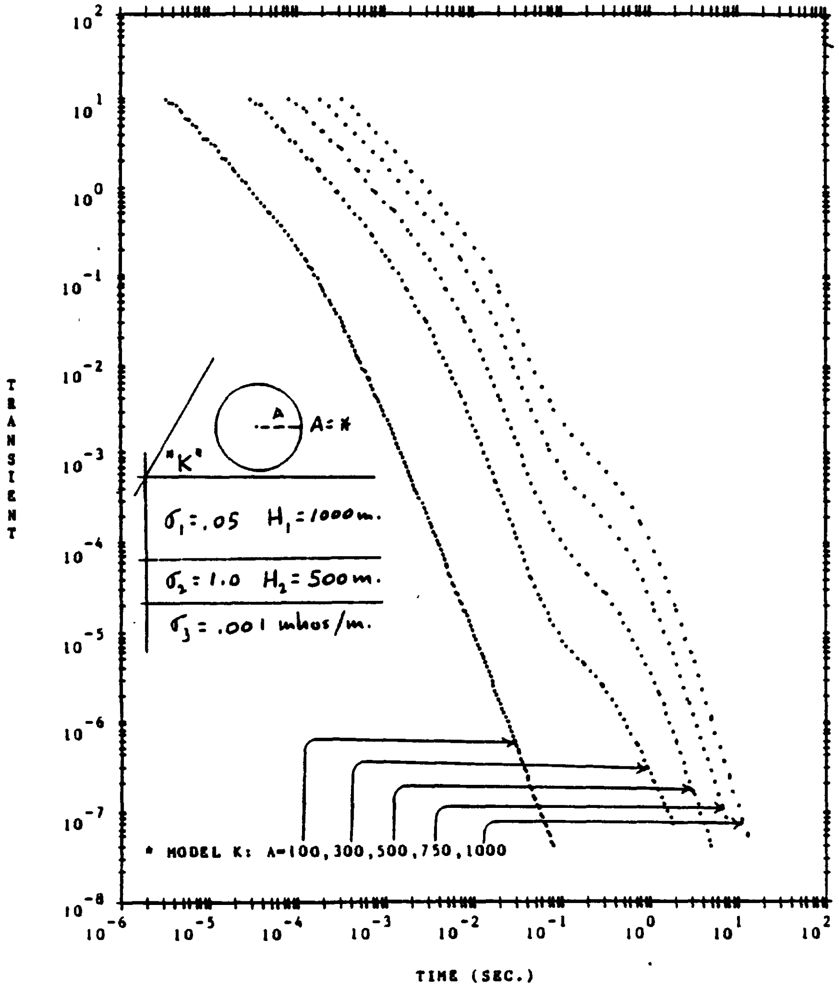

* MODEL I: A=100,300,500,750,1000
APPRES
10^2
10^1
10^0
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
TIME (SEC.)

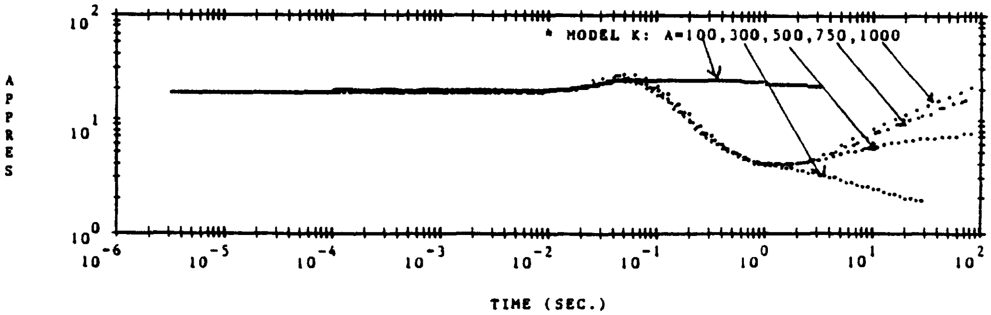

1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
* MODEL I: A=100,300,500,750,1000
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3
10^4
10^5
10^6
IND.NO.(B)

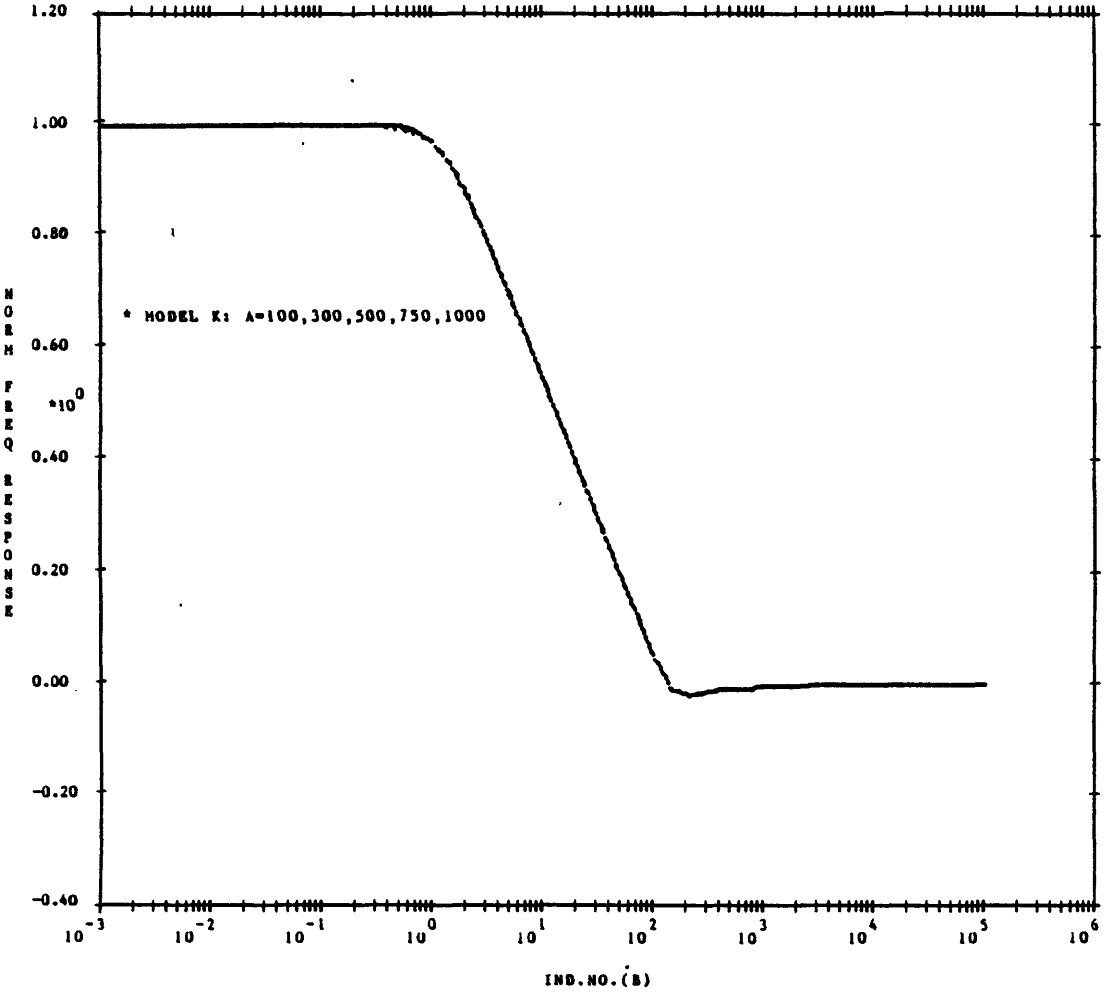

10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
TRANSIENT
A
A = *
"J"
σ1 = .05 H1 = 1000m.
σ2 = .001 mhos/m.
* MODEL J: A=100,300,500,750,1000
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
TIME (SEC.)

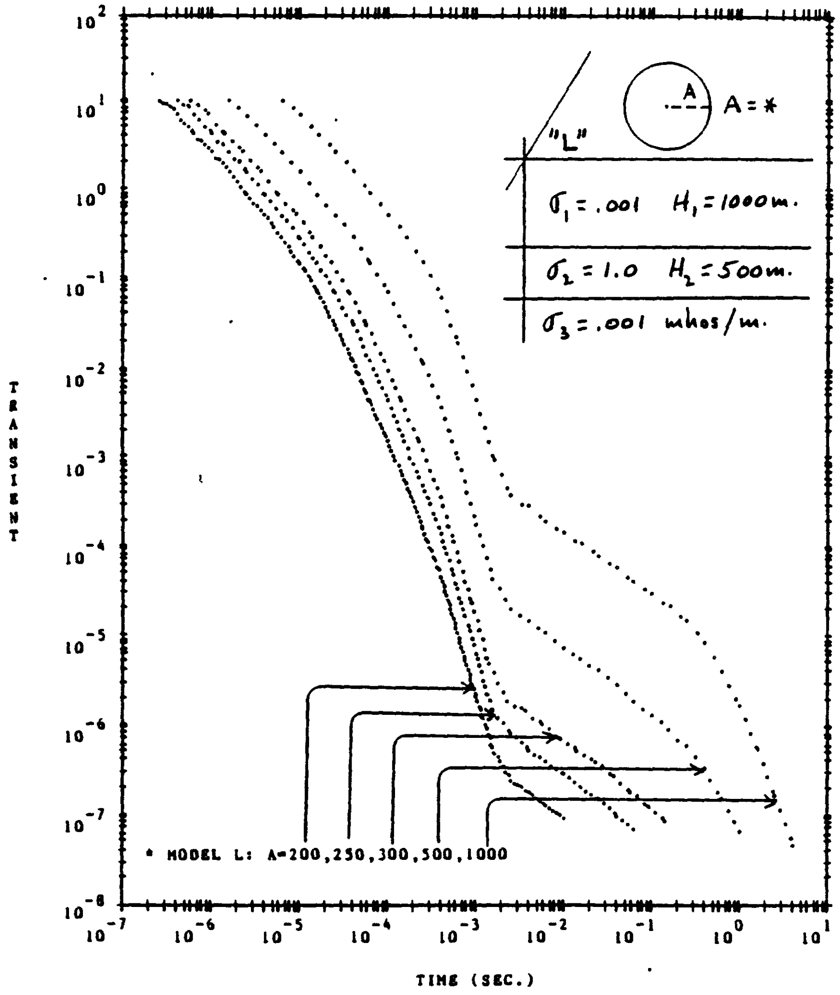

10^3
10^2
10^1
APPRES
* MODEL J: A=100,300,500,750,1000
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
TIME (SEC.)

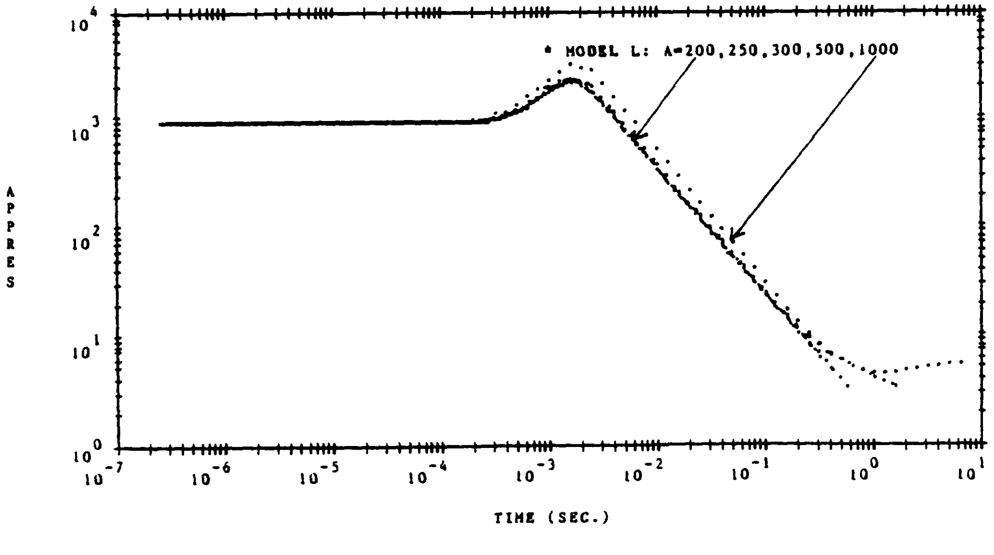

1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
* MODEL J: A=100,300,500,750,1000
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3
10^4
10^5
10^6
IND.NO.(B)

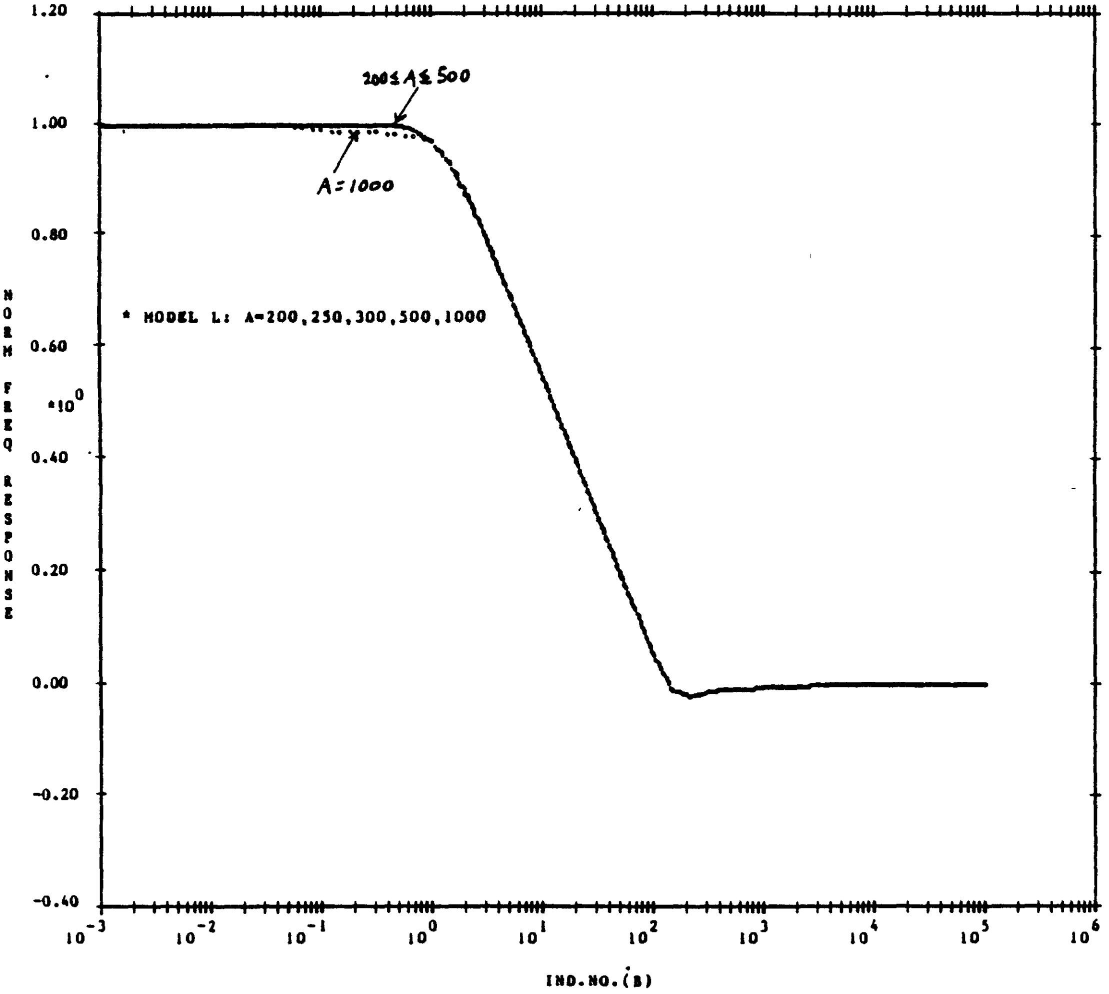

TRANSIENT
TIME (SEC.)
10^2
10^1
10^0
10^-1
10^-2
10^-3
10^-4
10^-5
10^-6
10^-7
10^-8
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
10^2
"K"
A
A=*
σ1=.05 H1=1000m.
σ2=1.0 H2=500m.
σ3=.001 mhos/m.
* MODEL K: A=100,300,500,750,1000


APPRES
TIME (SEC.)
10^2
10^1
10^0
10^-6
10^-5
10^-4
10^-3
10^-2
10^-1
10^0
10^1
10^2
* MODEL K: A=100,300,500,750,1000


1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
* MODEL K: A=100,300,500,750,1000
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3
10^4
10^5
10^6
IND.NO.(B)


"L"
A
A = *
$\sigma_1 = .001$ $H_1 = 1000$m.
$\sigma_2 = 1.0$ $H_2 = 500$m.
$\sigma_3 = .001$ mhos/m.
* MODEL L: A=200,250,300,500,1000
TRANSIENT
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
TIME (SEC.)


* MODEL L: A=200,250,300,500,1000
APPRES
10^{4}
10^{3}
10^{2}
10^{1}
10^{0}
10^{-7}
10^{-6}
10^{-5}
10^{-4}
10^{-3}
10^{-2}
10^{-1}
10^{0}
10^{1}
TIME (SEC.)


1.20
1.00
0.80
0.60
0.40
0.20
0.00
-0.20
-0.40
NORM FREQ RESPONSE
*10^0
10^-3
10^-2
10^-1
10^0
10^1
10^2
10^3
10^4
10^5
10^6
IND.NO.(B)
200≤A≤500
A=1000
* MODEL L: A=200,250,300,500,1000

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*. A magnetic tape copy can be sent to requestors to be copied and returned. This method of releasing the source code was selected in order to satisfy requests for the latest (e.g., possibly updated) version. The magnetic tape is usually recorded in the following mode (unless requested otherwise):

> Industry compatible: 9-track, standard ANSI-labeled, ASCII-mode, odd-parity, 800-bpi density, 80-character card-image records (blocked 50-card images, or 4000-characters, per physical block), and contained on one file named "TCOLOOP.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 {TCOLOOP}: TRANSIENT SOUNDING FOR COINCIDENT LOOP  {11/18/81}         00000010
C  FORWARD SOLUTIONS, WHERE CIRCULAR LOOP HAS RADIUS A>0.0 AND          00000020
C  THE LOOP IS PLACED ON THE EARTH'S SURFACE (I.E., ONLY                00000030
C  THE GROUND CASE Z=0.0 IS CONSIDERED HERE).                           00000040
C  THE TRANSIENT FIELD (TX=RX SAME LOOP) IS ASSUMED MEASURED AT THE     00000050
C  LOOP CENTER, BUT AT THE SURFACE OF THE EARTH.                        00000060
C                                                                       00000070
C  BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.           00000080
C                                                                       00000090
C--REFERENCES:                                                          00000100
C                                                                       00000110
C  ANDERSON,W.L., 1975, NTIS REPORT PB-242-800.                         00000120
C  ANDERSON,W.L., 1979, USGS OPEN-FILE REPT. 79-590.                    00000130
C  MORRISON, ET AL, 1969, GEOPHYS. PROSP. V.17, P.82-101.               00000140
C  RAICHE AND SPIES, 1981, GEOPHYSICS, V.46, NO.1, P.53-64.             00000150
C                                                                       00000160
C  NOTE THAT NORMALIZED TIME (TAU) IS USED FROM TO TO TM, WHERE         00000170
C  TIME=0.5*TAU*SIG1*(FOURPI*E-7)*A**2  (TIME IN SEC.)                  00000180
C  I.E., TAU=(2.0*TIME)/(SIG1*FOURPI*E-7*A**2).                         00000190
C  IPCH=1 OPTION (DEFAULT 0) WILL WRITE FILE10 WITH                     00000200
C  (TRANS,TIME,APPRES) IN FORMAT (3E16.8).                              00000210
C  IPCH>1 WILL WRITE FILE10 (AS ABOVE), AND ALSO FOR011,                00000220
C  FOR012, AND FOR013 FOR POSSIBLE PLOTTING PURPOSES (LATER).           00000230
C                                                                       00000240
C--SUBPROGRAMS RFLAGS AND ELOOP ARE CALLED TO COMPUTE THE TRANIENT      00000250
C  USING LAGGED-CONVOLUTION IN TIME (DEPENDING ON NB OPTION--SEE DOC.)  00000260
C  AND DIRECT OR SPLINED FREQ FUNCTION IN (B0,BM)--MIN,MAX IND.NUMBER.  00000270
C  NOTE: FREQ.FUNCT E/E0=1.0 IS ASSUMED IF B<B0 AND =0.0 IF B>BM, WHERE 00000280
```

```
C  DEFAULT BO=.001, BM=.1E6 ARE USUALLY ADEQUATE FOR MOST MODELS.
C
      CHARACTER*80 TITLE
      REAL SIG(10),H(10),DER(2), T(200),V(200), AR(200)
      COMPLEX K2(10),KS1,C4,ZA,ZAC2
      EXTERNAL ELOOP
      COMMON/PASS/ZAC2,ANORM,SIG,BO,BM,SIG1,EPS
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M
C**
C** SEE CALL NAMELIST SIMULATOR FOR THE VAX
C**
C**         NAMELIST/PARMS/M,SIG,H,A,Z,EPS,IFILL,
C**        1 BO,BM,NB,TO,TM,NT,XNORM,IOUT,IOUTS,IPCH,ISTOP
      COMMON/NAME_LIST/M_,SIG_(10),H_(9),A_,IFILL1,EPS_,IFILL2,
     1 BO_,BM_,NB,TO,TM,NT,XNORM,IOUT,IOUTS,IPCH,ISTOP
      DATA DER/2*0.0/,C2/.730921017/,THRESH/.1E-6/
C--PRESET
      DO I=1,200
        XS(I)=0.0
        YS(I)=0.0
      ENDDO
      R=0.0
      Z=0.0
      IPCH=0
      BO_=.001
      NB=6
      BM_=.1E6
      M_=1
      XNORM=10.
      DO 10 I=1,9
      SIG(I)=0.0
10    H_(I)=0.0
      SIG(10)=0.0
      A_=0.0
      EPS_=.1E-9
      ISTOP=1
      IOUTS=16
      TO=0.0
      NT=0
      TM=0.0
      IN=5
      IOUT=6
20    READ(IN,30,END=999) TITLE
30    FORMAT(A)
      CALL SETTIME
C**         READ(IN,PARMS,END=999)
      CALL NAMELIST(IN,'$PARMS',*999)
      M=M_
      DO 35 I=1,9
      SIG(I)=SIG_(I)
35    H(I)=H_(I)
      SIG(10)=SIG_(10)
      EPS=EPS_
      BO=BO_
      BM=BM_
      A=A_
      CALL NONBLANK(TITLE,NONBLK)
      IF(IOUT.GT.0)
     2  WRITE(6,40)
     3 TITLE,M,XNORM,IPCH,A,
     1 IOUTS,T0,NT,TM,ISTOP,IOUT,B0,NB,BM,EPS,SIG,H
40    FORMAT('1{TCOLOOP}:',6X,A<NONBLK>//
     2 5H M = ,I2,10X,6HXNORM=,E8.2,2X,5HIPCH=,I4,7X,
     3 2HA=,E11.4/
     4 9H IOUTS = ,I3,5X,3HT0=,E11.4,2X,5HNT = ,I4,7X,3HTM=,E11.4,2X,
     5 8HISTOP = ,I1/7H IOUT =,I5,
     6 5X,3HB0=,E11.4,2X,5HNB = ,I4,7X,3HBM=,E11.4,2X,4HEPS=,E9.2//
     7 6H SIG =,5E12.4/6X,5E12.4//
     8 6H   H =,5E12.4/6X,5E12.4)
      IF(IOUTS.GT.0) WRITE(IOUTS,40)
     & TITLE,M,XNORM,IPCH,A,
     1 IOUTS,T0,NT,TM,ISTOP,IOUT,B0,NB,BM,EPS,SIG,H
      IF(NT.LE.0) CALL ERRMSG('NT<=0',1,IOUT,IOUTS)
      IF(M.LT.1.OR.M.GT.10) CALL ERRMSG('M<1 OR M>10',4,IOUT,IOUTS)
      IF(A.LE.0.0) CALL ERRMSG('A<=0',2,IOUT,IOUTS)
      IF(B0.LE.0.0.OR.BM.LE.B0)
     & CALL ERRMSG('B0<=0 OR BM<=B0',3,IOUT,IOUTS)
      IF(T0.LE.0.0.OR.TM.LE.T0)
     & CALL ERRMSG('T0<=0 OR TM<=T0',3,IOUT,IOUTS)
      IF(SIG(1).LE.0.0)CALL ERRMSG('SIG(1)<=0',3,IOUT,IOUTS)
C--PRESET SOME CONSTANTS
      AA=A*A
      SIG1=SIG(1)
      TCON=6.28318531E-7*SIG1*AA
      ZA=CMPLX(A,0.0)
      IF(M.EQ.1) THEN
        HMAX=A
      ELSE
        CALL MINMAX(H,M-1,TEM,HMAX)
        IF(TEM.LE.0.0.OR.HMAX.LE.0.0)
     1    CALL ERRMSG('SOME H(I)<=0 FOR I<M',0,IOUT,IOUTS)
      ENDIF
      ANORM=A/HMAX
      ZAC2=ANORM/C2
      ISPLN=0
      IF(NB.GT.0.AND.NB.LT.12) ISPLN=1
      IF(ISPLN.EQ.0) GO TO 49
C--GET PRE-SPLINED FREQ. FUNCTION (0<NB<12 OPTION)
      DB=EXP(2.30258509/FLOAT(NB))
      BMTEST=0.5*(BM+BM*DB)
      MS=0
      TEM=B0/DB
      ISPLN=0
46    TEM=TEM*DB
      IF(TEM.GE.BMTEST) GO TO 47
      MS=MS+1
      IF(MS.GT.200)CALL ERRMSG('SPLINED MS>200',1,IOUT,IOUTS)
      OLDX=XS(MS)
      XS(MS)=TEM
      OLDY=YS(MS)
      YS(MS)=ELOOP(TEM*TEM)
C
C--APPLY THE 'THRESH TEST' TO SEE IF REST OF PREVIOUS CURVE CAN BE
```

```
C  USED TO SAVE RECOMPUTING REST OF FREQ RESPONSE. (NOTE THAT THE
C  FIRST CURVE, OR A CHANGE IN BO,NB,OR BM, WILL FALL-THRU ALL IF
C  TESTS AND ESTABLISH A NEW 'PREV CURVE' FOR SUBSEQUENT TESTS.)
C--BEGIN 'THRESH TEST':
      IF(TEM.GE.1.0) THEN
        IF(TEM.EQ.OLDX) THEN
          IF(OLDY.NE.0.0) THEN
            IF(ABS((YS(MS)-OLDY)/OLDY).LT.THRESH) THEN
              MS=NS
              GO TO 47
            ENDIF
          ENDIF
        ENDIF
      ENDIF
C--END OF 'THRESH TEST'
C
      GO TO 46
47    NS=MS
        CALL SPLIN1(NS,0.0,XS,YS,AS,BS,CS,0,DER,T,V)
C  WRITE FILE11 IF IPCH>1 (FOR LATER PLOTTING--IF DESIRED)
      IF(IPCH.GT.1) WRITE(11,1000) TITLE(1:40),NS,(XS(I),YS(I),I=1,NS)
1000  FORMAT('3'/'IND.NO.(B)'/'NORM FREQ RESPONSE'/A/I/(2G16.8))
      ISPLN=1
49    NEW=1
      DT=EXP(2.30258509/FLOAT(NT))
      TMTEST=0.5*(TM+TM*DT)
      IT=0
      TEM=TO/DT
      IF(IOUT.GT.0) WRITE(IOUT,50)
50    FORMAT('0',4X,'TAU(TO:TM)',3X,'TIME(SEC)',4X,'TRANS',8X,
     &'TRANS(NORM)',2X,'NORM*XNORM',3X,'APP.RES.'/)
      IF(IOUTS.GT.0) WRITE(IOUTS,50)
      LATE=0
60    TEM=TEM*DT
      IF(TEM.GE.TMTEST) GO TO 82
      TIME=TCON*TEM
      IF(LATE.EQ.1) THEN
        CALL APROX1(TEM,TRANS)
      ELSE
C--GET TRANSIENT IMPULSE RESPONSE VIA LAGGED CONVOLUTION IN TIME.
          TRANS=.63661977*RFLAGS(0,ELOOP,EPS,0.5*TO,TMTEST,TEM,NEW)
          NEW=0
        IF(TRANS.LT.1.E-7) THEN
        IF(IT.LT.3)
     1      CALL ERRMSG('IT<3--TO TOO BIG',1,IOUT,IOUTS)
        CALL APROXO(IT,T,V)
        CALL APROX1(TEM,TRANS)
        LATE=1
        ENDIF
      ENDIF
      IT=IT+1
      IF(IT.GT.200)CALL ERRMSG('IT>200--NT,TM TOO BIG',1,IOUT,IOUTS)
      T(IT)=TEM
      V(IT)=TRANS
      IF(IT.EQ.1) TRANS1=TRANS
      TNORM=TRANS/TRANS1
      TXNORM=TNORM*XNORM
```

```
C--GET APP.RES.
      S0=1.29552377*TEM*TRANS
      Y0=S0**.66666667
      X1=(((((((((110000.*Y0+12360.90299)*Y0+
     1  3379.08752)*Y0+955.90217)*Y0+
     2  255.84635)*Y0+71.89746)*Y0+
     3  20.88351)*Y0+6.49229)*Y0+
     4  2.38095)*Y0+1.70998)**2
      X1=Y0*X1
      IF(X1.LE.1.4) THEN
        X2=X1
      ELSE IF(X1.GT.1.4.AND.X1.LE.2.8) THEN
        X2=X1+0.001635*X1**4.892
      ELSE IF(X1.GT.2.8.AND.X1.LE.5.69) THEN
        X2=X1+0.004018*X1**4.01364
      ELSE
        CALL WARN('X1>5.69; APP.RES.=1./SIG1 USED.',0,IOUT,IOUTS,*66)
66      APPRES=1./SIG1
        GO TO 68
      ENDIF
      APPRES=0.5/(SIG1*TEM*X2)
68    AR(IT)=APPRES
      IF(IOUT.GT.0) WRITE(IOUT,70) TEM,TIME,TRANS,TNORM,TXNORM,
     1  APPRES
70    FORMAT(1X,6E13.5)
      IF(IOUTS.GT.0) WRITE(IOUTS,70) TEM,TIME,TRANS,TNORM,TXNORM,
     1  APPRES
      IF(IPCH.NE.0) WRITE(10,100) TRANS,TIME,APPRES
100   FORMAT(3E16.8)
      GO TO 60
82    IF(IOUTS.GT.0) WRITE(IOUTS,90)
90    FORMAT(129X)
      CALL CPUTIME(IOUT,IOUTS)
C  WRITE FILE13 IF IPCH>1 (FOR LATER PLOTTING--IF DESIRED)
      IF(IPCH.GT.1) THEN
        WRITE(12,1900) TITLE(1:40)
1900    FORMAT('3'/'TIME (SEC.)'/'APPRES'/A)
        WRITE(12,2002) IT,(TCON*T(J),AR(J),J=1,IT)
        WRITE(13,2000) TITLE(1:40)
2000  FORMAT('3'/'TIME (SEC.)'/'TRANSIENT'/A)
      DO I=1,IT
        II=I
        IF(V(I).LT.1.E-7) GO TO 2001
      ENDDO
2001  WRITE(13,2002) II,(TCON*T(J),V(J),J=1,II)
2002  FORMAT(I/(2G16.8))
      ENDIF
      IF(ISTOP.NE.1) GO TO 20
999   CALL EXIT
      END
      SUBROUTINE APROXO(IT,T,V)
C--LATE TIME APPROXIMATION INITIALIZATION WHEN 1ST COMPUTED TRANS<1E-7
C  AND 2<IT<201 (REQUIRED).
C
      SAVE
      DIMENSION A(201),B(201),C(201),D(2),T(1),V(1),W1(201),W2(201),
     1 TLOG(201),VLOG(201)
```

```
      DATA D/2*0.0/
      DO 10 I=1,IT
        TLOG(I)=ALOG(T(I))
10      VLOG(I)=ALOG(V(I))
      NT=IT+1
      TLOG(NT)=87.498234
      VLOG(NT)=-87.498234
      CALL SPLIN1(NT,0.0,TLOG,VLOG,A,B,C,0,D,W1,W2)
      RETURN
C** ENTRY APROX1(TEM,TRANS)
      ENTRY APROX1(TEM,TRANS)
      AT=ALOG(TEM)
      IF(AT.GT.87.498234) THEN
        TRANS=0.0
        RETURN
      ENDIF
      CALL SPOINT(NT,TLOG,VLOG,A,B,C,AT,TT)
      TRANS=EXP(TT)
      RETURN
      END
      REAL FUNCTION ELOOP(B2)
C--COSINE-TRANSFORM KERNEL FOR COINCIDENT LOOP WITH
C  A>0,R=0, AND Z=0.0.
C
      REAL SIG(10),H(10),Z
      COMPLEX ZAC2,K2(10),KS1,ZFLD
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M
      COMMON/PASS/ZAC2,ANORM,SIG,B0,BM,SIG1,EPS
      COMMON/SPLN/XS(200),YS(200),AS(200),BS(200),CS(200),NS,ISPLN
      EXTERNAL F3ZH
      B=SQRT(B2)
      IF(B.LT.B0) GO TO 3
      IF(B.GT.BM) GO TO 4
      IF(ISPLN.EQ.0) GO TO 10
C--ISPLN=1 (0<NB<12 OPTION) INTERPOLATE PRE-SPLINED FREQ. FUNCTION
      CALL SPOINT(NS,XS,YS,AS,BS,CS,B,ELOOP)
      RETURN
10    F=(B/A)**2/(39.47841762E-7*SIG1)
      KS1=CMPLX(0.0,-7.895683523E-6*F)
      DO 1 I=1,M
1     K2(I)=KS1*CMPLX(SIG(I),0.0)
      ZFLD=ZAC2*SQJ1(ANORM,F3ZH,EPS,LL) + 1.0
      ELOOP=REAL(ZFLD)
      RETURN
3     ELOOP=1.0
      RETURN
4     ELOOP=0.0
      RETURN
      END
      COMPLEX FUNCTION F3ZH(X)
C--KERNEL FOR HANKEL TRANSFORM IN CURLOOP WHEN R=0.0 AND Z=0.0
C  SCALED BY HMAX STORED IN COMMON/MODEL/
C
      COMPLEX Z1,Z0,K2(10),KS1,HALF
      REAL H(10),Z
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M
      DATA HALF/(0.5,0.0)/
      Y=X/HMAX
      CALL RECUR(Y,Z1,Z0)
      F3ZH=Z1/(Z0+Z1)-HALF
      RETURN
      END
      SUBROUTINE RECUR(Y,Z1,Z0)
C--BACKWARD RECURRENCE FOR COMPLEX IMPEDANCES Z1,Z0 GIVEN ARGUMENT
C     Y(=X/HMAX) AND MODEL PARAMETERS IN COMMON/MODEL/
C
      REAL H(10),Z
      COMPLEX Z1,Z0,K2(10),KS1,ONE,ZZ,X2,U
      COMMON/MODEL/K2,KS1,H,Z,A,R,HMAX,M
      DATA ONE/(1.0,0.0)/
      X2=CMPLX(Y*Y,0.0)
      Z0=KS1/CMPLX(Y,0.0)
      Z1=KS1/CSQRT(X2-K2(M))
      IF(M.EQ.1) GO TO 20
      J=M-1
10    U=CSQRT(X2-K2(J))
      ZZ=KS1/U
      U=CEXP(CMPLX(-2.0*H(J),0.0)*U)
      U=(ONE-U)/(ONE+U)
      Z1=ZZ*((Z1+ZZ*U)/(ZZ+Z1*U))
      IF(J.EQ.1) GO TO 20
      J=J-1
      GO TO 10
20    RETURN
      END
      SUBROUTINE NAMELIST(IUNIT,NAME,*)
C
C {NAMELIST INPUT ON VAX-11/780} VIA "CALL NAMELIST" {VERSION: 12/10/80}
C
C--A SIMULATED 'NAMELIST/NAME/' PROCESSOR FOR VAX-11 FORTRAN-77 TO
C  IMPLEMENT "CALL NAMELIST(IUNIT,'$NAME',*EOF)" ON VAX, WHICH
C  IS SIMILAR TO "READ(IUNIT,NAME,END=EOF)" ON MOST LARGE SYSTEMS.
C
C--BY W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO.
C
C--THIS IS A SUBSET OF THE ACTUAL NAMELIST/NAME/ AVAILABLE ON
C  MOST LARGE MAIN-FRAME SYSTEMS. CURRENT OPTIONS ARE:
C
C  (1) ALL VARNAM'S ARE RESTRICTED TO 1 TO 6 CHAR'S (ALP,NUM, AND '_')
C      BUT MUST BEGIN WITH AN ALP CHAR (E.G., A3_, BVAR, C_2, ETC.)
C  (2) ONLY VARIABLE TYPES REAL*4 *8 (NAMTYP=1) AND INTEGER*2 *4
C      (NAMTYP=0). SEE C==== EXAMPLE STATEMENTS FOR NAMTYP BELOW =====.
C      {NOTE: COMPLEX,LOGICAL, OR CHARACTER VARIABLE TYPES ARE "NOT"
C      CODED IN THIS VERSION.}
C  (3) MAX. 60 VARNAM'S ALLOWED IN NAMELIST (FOR ALL '$NAMES' USED).
C  (4) MAX. NUMBER FIELD (FLOAT OR FIXED) IS 20 CHAR WIDE, WHERE
C      BLANK CHAR'S ARE IGNORED, AND TYPE CONVERSION IS AUTOMATIC.
C      FLOAT NUMBERS WITH OPTIONAL E+XX OR D-XX AND WITH OR WITHOUT '.'
C      IN THE MANTISSA IS ALLOWED (E.G., 123E-3, .123D+02, -3.14, ETC.).
C  (5) PARTIAL ARRAY'S ALLOWED; E.G., A(10)=25.1,
C      AND B=1,3.2,...
C  (6) REPEAT FACTORS ALLOWED; E.G., C=2*1,3,..
C  (7) ONLY 1-DIM ARRAYS ALLOWED WITH MAX SIZE 99999.
C  (8) THE NAMELIST '$NAME' MUST BE 2 TO 7 CHAR'S, AND MUST BEGIN WITH
```

```
C      A "$" CHAR (E.G., '$P', '$PARMS', ETC.);  ALSO, THE FIRST CHAR IN00003710
C      IFILE MAY BEGIN IN COL. 1 BUT LESS THAN COL. 72 (BUFFER IS 80).  00003720
C      LINES IN IFILE MAY BE CONTINUED TO COL. 1 ON NEXT LINE, AND      00003730
C      TERMINATE THE NAMELIST BY "$[END]"--THE "END" IS OPTIONAL. E.G., 00003740
C                                                                       00003750
C      $PARMS A=1,B=2.3,7*1,C(3)=-.123E-10,                             00003760
C      D=1800, E=5*20$END                                               00003770
C      $NEXNAM F=123, G=-10,C(2)=15.02 $                                00003780
C      ...END-OF-IFILE...                                               00003790
C  (9) ABOUT 98% OF ALL THE POSSIBLE ERRORS ARE DETECTED AND AN         00003800
C      ERROR MESSAGE IS PRINTED ON UNIT 06, FOLLOWED BY CALL EXIT.      00003810
C      {NOTE: WATCH OUT FOR THE REMAINING 2% UNDETECTED ERRORS!}        00003820
C                                                                       00003830
C--SUBROUTINES CALLED:                                                  00003840
C                                                                       00003850
C  DECODEIX, DECODEX, AND NONBLANK.                                     00003860
C                                                                       00003870
C--USAGE:                                                               00003880
C                                                                       00003890
C  1. MODIFY FILE 'INCLNAMES.FOR' AS REQUIRED (USE ANY EDITOR).         00003900
C     (SEE C==== EXAMPLE STATEMENTS BELOW =====.)                       00003910
C  2. RECOMPILE SUBROUTINE 'NAMELIST' WITH THE DESIRED INCLNAMES.FOR.   00003920
C  3. IN USERS CALLING PROGRAM, USE:                                    00003930
C     CALL NAMELIST(IUNIT,'$NAME',*N)  --ON VAX, WHERE N=E.O.F RETURN   00003940
C     STATEMENT LABEL.  THIS SIMULATES ON VAX:                          00003950
C      'READ(IUNIT,NAME,END=N)' ON SYSTEMS WITH NAMELIST/NAME/...       00003960
C                                                                       00003970
C***********************************************************************00003980
C                                                                       00003990
      CHARACTER*(*) NAME                                                00004000
      CHARACTER*1 C(47),BUFI                                            00004010
      CHARACTER*6 VARNAM                                                00004020
      CHARACTER*20 NUMFLD                                               00004030
      CHARACTER*80 BUF                                                  00004040
C                                                                       00004050
C=======================================================================00004060
C====== THE USER MUST CHANGE THE FOLLOWING STATEMENTS FOR THE SPECIFIC  00004070
C====== NAMELIST VARIABLES DESIRED (E.G., USE TECO OR EDT, ETC.)========00004080
C====== DIMENSION NO_NAM VARIABLES TO AGREE WITH CHANGED DATA STATEMENTS00004090
C==                                                                     00004100
C==ON VAX USE THE FOLLOWING INCLUDE STATEMENT (OPTIONALLY, USE /LIST):  00004110
C==                                                                     00004120
C>>   INCLUDE 'INCLNAMES.FOR/NOLIST'                                    00004130
C                                                                       00004140
C======================== INCLNAM12.FT =================================00004150
C==================== FOR USE IN CALL NAMELIST =========================00004160
C NORMALLY, ONE SHOULD COPY 'INCLNAM12.FT' TO 'INCLNAMES.FT'; THEN      00004170
C   EDIT 'INCLNAMES.FT' AS DESIRED FOR USERS CALL NAMELIST. NOTE THAT   00004180
C   ONE MUST RECOMPILE 'NAMELIST.FT' WITH USERS CALLING PROGRAM,        00004190
C   WHERE 'NAMELIST.FT' CONTAINS THE FOLLOWING STATEMENT:               00004200
C                                                                       00004210
C     INCLUDE 'INCLNAMES.FT/LIST'                                       00004220
C=======================================================================00004230
C                                                                       00004240
C***********************************************************************00004250
C  THIS IS "$PARMS INPUT" FOR PROGRAMS "TCILOOP" AND "TCOLOOP"          00004260
C***********************************************************************00004270
```

```
C
      COMMON/NAME_LIST/V1,V2,V3,V4,V5,V6,V7,V8,V9,V10,
     * V11,V12,V13,V14,V15,V16,V17,V18
      INTEGER V1,V7,V10,V13,V15,V16,V17,V18
      DIMENSION V1(1),V2(10),V3(9),V4(1),
     * V5(1),V6(1),V7(1),V8(1),V9(1),V10(1),
     * V11(1),V12(1),V13(1),V14(1),V15(1),
     * V16(1),V17(1),V18(1),V19(1),V20(1),
     * V21(1),V22(1),V23(1),V24(1),V25(1),
     * V26(1),V27(1),V28(1),V29(1),V30(1),
     * V31(1),V32(1),V33(1),V34(1),V35(1),
     * V36(1),V37(1),V38(1),V39(1),V40(1),
     * V41(1),V42(1),V43(1),V44(1),V45(1),
     * V46(1),V47(1),V48(1),V49(1),V50(1),
     * V51(1),V52(1),V53(1),V54(1),V55(1),
     * V56(1),V57(1),V58(1),V59(1),V60(1)
      DIMENSION NAMDIM(60),NAMLEN(60),NAMTYP(60)
      CHARACTER*6 NAM(60)
      DATA NAM/'M','SIG','H','A','Z','EPS','ISTEP',
     1 'BO','BM','NB','TO','TM','NT','XNORM','IOUT',
     2 'IOUTS','IPCH','ISTOP',42*' '/
      DATA NAMDIM/1,10,9,15*1,42*0/
      DATA NAMLEN/1,3,3*1,3,5,6*2,5,4,5,4,5,42*0/
      DATA NAMTYP/0,5*1,0,2*1,0,2*1,0,1,4*0,42*0/
      DATA NO_NAM/18/
C====== END OF INCLUDE STATEMENTS ======================================
C
C==
C== FOR EXAMPLE, FILE 'INCLNAMES.FOR' MAY CONTAIN (WITHOUT "C=="):
C==
C==      COMMON/NAME_LIST/V1,V2,V3,V4
C==      REAL*8 V1
C==      INTEGER V3
C==      DIMENSION V1(1),V2(2),V3(3),V4(4),
C==     * V5(1),V6(1),V7(1),V8(1),V9(1),V10(1),
C==     * V11(1),V12(1),V13(1),V14(1),V15(1),
C==     * V16(1),V17(1),V18(1),V19(1),V20(1),
C==     * V21(1),V22(1),V23(1),V24(1),V25(1),
C==     * V26(1),V27(1),V28(1),V29(1),V30(1),
C==     * V31(1),V32(1),V33(1),V34(1),V35(1),
C==     * V36(1),V37(1),V38(1),V39(1),V40(1),
C==     * V41(1),V42(1),V43(1),V44(1),V45(1),
C==     * V46(1),V47(1),V48(1),V49(1),V50(1),
C==     * V51(1),V52(1),V53(1),V54(1),V55(1),
C==     * V56(1),V57(1),V58(1),V59(1),V60(1)
C==      DIMENSION NAMDIM(60),NAMLEN(60),NAMTYP(60)
C==      CHARACTER*6 NAM(60)
C==      DATA NAM/'A','BB','ICC','DDD_4',56*' '/
C==      DATA NAMDIM/1,2,3,4,56*0/
C==      DATA NAMLEN/1,2,3,5,56*0/
C==      DATA NAMTYP/2*1,0,1,56*0/
C==      DATA NO_NAM/4/
C====== END OF EXAMPLE INCLUDE STATEMENTS ==============================
C
C***********************************************************************
C NOTE: THE ABOVE EXAMPLE SIMULATES
C       'NAMELIST/NAME/A,BB,ICC,DDD_4'
```

```
C       'READ(IUNIT,NAME,END=EOF)'
C       'READ(IUNIT,ANYNAM,END=EOF)'
C       IN THE CALLING PROGRAM USING:
C       ...
C       REAL*8 A
C       ...
C       COMMON/NAME_LIST/A,BB(2),ICC(3),DDD_4(4)
C       ...
C       CALL NAMELIST(IUNIT,'$NAME',*EOF)
C       ...
C       CALL NAMELIST(IUNIT,'$ANYNAM',*EOF)
C       ...
C***********************************************************************
C
      DATA C/'A','B','C','D','E','F','G','H','I','J','K','L','M','N',
     * 'O','P','Q','R','S','T','U','V','W','X','Y','Z','_',
     * '1','2','3','4','5','6','7','8','9','0',
     * ' ','$','=',',','(','*',')','.','+','-'/
      J=LEN(NAME)
      IF(J.LT.2.OR.J.GT.7) THEN
        CALL ERRMSG('CALL NAMELIST ILLEGAL WITH NAME= '//
     1 NAME//'   (LENGTH<2 OR >7 CHAR''S)',1,6,0)
      ENDIF
      IF(NAME(1:1).NE.'$')
     1 CALL ERRMSG('CALL NAMELIST ILLEGAL WITH NAME= '//
     2 NAME//'   (1ST CHAR MUST BE "$" CHAR)',1,6,0)
C--INITIALIZE
      INAME=0
10    READ(IUNIT,11,END=99991,ERR=99992) BUF
11    FORMAT(A80)
      IF(INAME.EQ.1) GO TO 20
C--LOOK FOR "$NAME"
        I=INDEX(BUF,NAME)
      IF(I.EQ.0) GO TO 10
      INAME=1
      ICOL=I+J
      JNAM=0
      ILEN=0
      VARNAM=' '
      NUMLEN=0
      IELE=1
      GO TO 30
20    ICOL=1
30    CALL NONBLANK(BUF,LENBUF)
C--BEGIN PARSER LOOP (THE BIG 20000 LOOP)
      IEND=0
      DO 20000 I=ICOL,LENBUF
         BUFI=BUF(I:I)
      DO 40 IC=1,27
         IF(BUFI.EQ.C(IC)) GO TO 100
40    CONTINUE
      DO 50 IC=28,37
      IF(BUFI.EQ.C(IC)) GO TO 200
50    CONTINUE
      DO 60 IC=38,47
      IC_=IC-37
      IF(BUFI.EQ.C(IC)) GO TO 70
```

```
60    CONTINUE
61    WRITE(6,66) I,BUF
66    FORMAT(/' {NAMELIST}: ERROR IN FOLLOWING RECORD AT COL(',I2,'):'/
     1 1X,A80/<I>X,'^')
      CALL ERRMSG('ILLEGAL CHAR="'//BUFI//'" FOUND',0,6,0)
67    WRITE(6,66) I,BUF
      CALL ERRMSG('NUMLEN<1 IN DECODEIX    ',0,6,0)
68    WRITE(6,66) I,BUF
      CALL ERRMSG('NUMLEN<1 IN DECODEX',0,6,0)
70    GO TO (20000,72,73,74,75,76,77,78,79,79),IC_
C--'$' CHAR
72    IEND=1
      IF(NUMLEN.GT.0) GO TO 798
      IF(JNAM.EQ.0) GO TO 99990
      WRITE(6,66) I,BUF
      CALL ERRMSG('MISPLACED "$" CHAR',0,6,0)
C--'=' CHAR
73    IEQ=1
C--CHECK FOR VALID VARNAM, LENGTH ILEN, ETC.
      IF(ILEN.LT.1) GO TO 733
      DO 732 J=1,NO_NAM
      JNAM=J
      JLEN=NAMLEN(J)
      IF(JLEN.NE.ILEN) GO TO 732
      DO 731 K=1,JLEN
      IF(VARNAM(K:K).NE.NAM(JNAM)(K:K)) GO TO 732
731   CONTINUE
C--VARNAM VERIFIED OK TO PROCEED TO NUMFLD(S)
C
      IDIM=NAMDIM(JNAM)
      NUMLEN=0
      NDEC=0
      NREP=1
      NEXP=0
      GO TO 20000
732   CONTINUE
      WRITE(6,66) I,BUF
      CALL ERRMSG('ILLEGAL VARNAM='//VARNAM//' FOUND',0,6,0)
733   WRITE(6,66) I,BUF
      CALL ERRMSG('MISPLACED "=" CHAR  ',0,6,0)
C--',' CHAR
74    IF(NUMLEN.GT.0) GO TO 799
      WRITE(6,66) I,BUF
      CALL ERRMSG('MISPLACED "," CHAR',0,6,0)
C--'(' CHAR
75    IELE=0
      GO TO 20000
C--'*' CHAR
76    IF(JNAM.EQ.0.OR.NUMLEN.LT.1.OR.NUMLEN.GT.5) GO TO 767
760   CALL DECODEIX(NUMFLD,NUMLEN,NREP,*67)
      NUMLEN=0
      IF(NREP.GT.0.AND.NREP.LE.NAMDIM(JNAM)) GO TO 20000
      WRITE(6,66) I,BUF
      CALL ERRMSG('REPEAT FACTOR <1 OR >NAMDIM   ',0,6,0)
767   WRITE(6,66) I,BUF
      CALL ERRMSG('REPEAT WIDTH > 5 OR MISPLACED "*" CHAR',0,6,0)
C--')' CHAR
```

```
77     IF(IELE.NE.0) GO TO 772
       CALL DECODEIX(NUMFLD,NUMLEN,IELE,*67)
       IF(IELE.LT.1) GO TO 773
       NREP=1
       GO TO 20000
772    WRITE(6,66) I,BUF
       CALL ERRMSG('MISPLACED ")" CHAR',0,6,0)
773    WRITE(6,66) I,BUF
       CALL ERRMSG('ARRAY IELE<1 OR >NAMDIM  ',0,6,0)
C--'.' CHAR
78     IF(JNAM.EQ.0.OR.NEXP.GT.0.OR.NDEC.GT.0) GO TO 781
       NDEC=NUMLEN+1
       IF(NAMTYP(JNAM).EQ.1) GO TO 200
781    WRITE(6,66) I,BUF
       CALL ERRMSG('MISPLACED "." CHAR',0,6,0)
C--'-' OR '+' CHAR
79     IF(IELE.GT.0.OR.NEXP.GT.0) GO TO 210
       WRITE(6,66) I,BUF
       CALL ERRMSG('MISPLACED "-" OR "+" CHAR',0,6,0)
C--<ALP> CHAR
100    IF(NUMLEN.GT.0) GO TO 209
       IF(ILEN.GT.0) GO TO 102
       IEQ=0
       IELE=1
102    ILEN=ILEN+1
       IF(ILEN.GT.6) GO TO 101
       VARNAM(ILEN:ILEN)=BUFI
       GO TO 20000
101    WRITE(6,66) I,BUF
       CALL ERRMSG('VARNAM>6 CHAR''S',0,6,0)
C--<+-NUM> CHAR
200    IF(IELE.EQ.0) GO TO 210
       IF(IEQ.EQ.0) GO TO 102
       GO TO 210
209    IF(BUFI.EQ.'E'.OR.BUFI.EQ.'D') THEN
         NEXP=NUMLEN+1
       ELSE
         GO TO 61
       ENDIF
210    NUMLEN=NUMLEN+1
       IF(NUMLEN.GT.20) GO TO 211
       NUMFLD(NUMLEN:NUMLEN)=BUFI
       GO TO 20000
211    WRITE(6,66) I,BUF
       CALL ERRMSG('NUM FIELD>20 CHAR''S',0,6,0)
C--PROCESS NUMBER FIELD
799    IDIM=IDIM-1
       IF(IDIM.LT.0) GO TO 10004
798    IF(NEXP.GT.0) GO TO 1000
C--[NEXP=0]
       IF(NDEC.GT.0) GO TO 899
C--[NEXP=0, NDEC=0]
       CALL DECODEIX(NUMFLD,NUMLEN,IX,*67)
C--CONVERT IX AND STORE IN COMMON
800    X=IX
       IF(IELE.GT.NAMDIM(JNAM)) GO TO 773
8000   GO TO (801,802,803,804,805,806,807,808,809,810,
     * 811,812,813,814,815,816,817,818,819,820,
     * 821,822,823,824,825,826,827,828,829,830,
     * 831,832,833,834,835,836,837,838,839,840,
     * 841,842,843,844,845,846,847,848,849,850,
     * 851,852,853,854,855,856,857,858,859,860),JNAM
801   V1(IELE)=X
      GO TO 10000
802   V2(IELE)=X
      GO TO 10000
803   V3(IELE)=X
      GO TO 10000
804   V4(IELE)=X
      GO TO 10000
805   V5(IELE)=X
      GO TO 10000
806   V6(IELE)=X
      GO TO 10000
807   V7(IELE)=X
      GO TO 10000
808   V8(IELE)=X
      GO TO 10000
809   V9(IELE)=X
      GO TO 10000
810   V10(IELE)=X
      GO TO 10000
811   V11(IELE)=X
      GO TO 10000
812   V12(IELE)=X
      GO TO 10000
813   V13(IELE)=X
      GO TO 10000
814   V14(IELE)=X
      GO TO 10000
815   V15(IELE)=X
      GO TO 10000
816   V16(IELE)=X
      GO TO 10000
817   V17(IELE)=X
      GO TO 10000
818   V18(IELE)=X
      GO TO 10000
819   V19(IELE)=X
      GO TO 10000
820   V20(IELE)=X
      GO TO 10000
821   V21(IELE)=X
      GO TO 10000
822   V22(IELE)=X
      GO TO 10000
823   V23(IELE)=X
      GO TO 10000
824   V24(IELE)=X
      GO TO 10000
825   V25(IELE)=X
      GO TO 10000
826   V26(IELE)=X
      GO TO 10000
```

```
827     V27(IELE)=X                                                        00007130
        GO TO 10000                                                        00007140
828     V28(IELE)=X                                                        00007150
        GO TO 10000                                                        00007160
829     V29(IELE)=X                                                        00007170
        GO TO 10000                                                        00007180
830     V30(IELE)=X                                                        00007190
        GO TO 10000                                                        00007200
831     V31(IELE)=X                                                        00007210
        GO TO 10000                                                        00007220
832     V32(IELE)=X                                                        00007230
        GO TO 10000                                                        00007240
833     V33(IELE)=X                                                        00007250
        GO TO 10000                                                        00007260
834     V34(IELE)=X                                                        00007270
        GO TO 10000                                                        00007280
835     V35(IELE)=X                                                        00007290
        GO TO 10000                                                        00007300
836     V36(IELE)=X                                                        00007310
        GO TO 10000                                                        00007320
837     V37(IELE)=X                                                        00007330
        GO TO 10000                                                        00007340
838     V38(IELE)=X                                                        00007350
        GO TO 10000                                                        00007360
839     V39(IELE)=X                                                        00007370
        GO TO 10000                                                        00007380
840     V40(IELE)=X                                                        00007390
        GO TO 10000                                                        00007400
841     V41(IELE)=X                                                        00007410
        GO TO 10000                                                        00007420
842     V42(IELE)=X                                                        00007430
        GO TO 10000                                                        00007440
843     V43(IELE)=X                                                        00007450
        GO TO 10000                                                        00007460
844     V44(IELE)=X                                                        00007470
        GO TO 10000                                                        00007480
845     V45(IELE)=X                                                        00007490
        GO TO 10000                                                        00007500
846     V46(IELE)=X                                                        00007510
        GO TO 10000                                                        00007520
847     V47(IELE)=X                                                        00007530
        GO TO 10000                                                        00007540
848     V48(IELE)=X                                                        00007550
        GO TO 10000                                                        00007560
849     V49(IELE)=X                                                        00007570
        GO TO 10000                                                        00007580
850     V50(IELE)=X                                                        00007590
        GO TO 10000                                                        00007600
851     V51(IELE)=X                                                        00007610
        GO TO 10000                                                        00007620
852     V52(IELE)=X                                                        00007630
        GO TO 10000                                                        00007640
853     V53(IELE)=X                                                        00007650
        GO TO 10000                                                        00007660
854     V54(IELE)=X                                                        00007670
        GO TO 10000                                                        00007680
855     V55(IELE)=X                                                        00007690
```

```
      GO TO 10000
856   V56(IELE)=X
      GO TO 10000
857   V57(IELE)=X
      GO TO 10000
858   V58(IELE)=X
      GO TO 10000
859   V59(IELE)=X
      GO TO 10000
860   V60(IELE)=X
      GO TO 10000
C--[NEXP=0, NDEC>0]
899   CALL DECODEX(NUMFLD,NUMLEN,NDEC,X,*68)
C--CONVERT X AND STORE IN COMMON
900   IF(IELE.GT.NAMDIM(JNAM)) GO TO 773
      GO TO 8000
C--[NEXP>0]
1000  IF(NDEC.GT.0) GO TO 2000
C--[NEXP>0, NDEC=0]
      CALL DECODEIX(NUMFLD,NEXP-1,IX,*67)
      X=IX
1002  J=1
      DO 1001 K=NEXP+1,NUMLEN
      NUMFLD(J:J)=NUMFLD(K:K)
1001  J=J+1
      CALL DECODEIX(NUMFLD,NUMLEN-NEXP,IE,*67)
      X=X*10.**IE
C** {LATER INSERT A CALL TO A OVERFLOW HANDLER, ETC.}
      GO TO 900
C--[NEXP>0, NDEC>0]
2000  CALL DECODEX(NUMFLD,NEXP-1,NDEC,X,*68)
      GO TO 1002
C--NEXT IELE?
10000 IELE=IELE+1
      IF(IELE.GT.NAMDIM(JNAM)) GO TO 10002
      IF(NREP.GT.1) GO TO 10003
10001 IF(IEND.EQ.1) GO TO 99990
      NUMLEN=0
      NDEC=0
      NEXP=0
      NREP=1
      ILEN=0
      VARNAM=' '
      GO TO 20000
10002 IELE=1
      GO TO 10001
10003 NREP=NREP-1
      IDIM=IDIM-1
      IF(IDIM.GE.0) GO TO 8000
10004 WRITE(6,66) I,BUF
      CALL ERRMSG('TOO MANY ELEMENTS FOR GIVEN NAMDIM.',0,6,0)
C==END OF DO 20000   CONTINUE PARSER -OR- READ IN NEXT BUF, ETC.
20000 CONTINUE
               GO TO 10
C--'$' CHAR (DELIMITER $[END] FOR THIS $NAME --$)
99990 RETURN
C--E.O.F. ON FILE IUNIT ENCOUNTERED.
```

```
99991 RETURN 1
99992 CALL ERRMSG('CANNOT OPEN/READ CALL NAMELIST(IFILE,...)',1,6,0)
      END
      SUBROUTINE CPUTIME(I1,I2)
C
C  CPUTIME WRITES "ELAPSED & CPU" TIME FROM PREVIOUS "CALL SETTIME" ON
C  FORTRAN UNITS I1 (IF NOT 0) AND I2 (IF NOT 0).
C
C  WILL EJECT FIRST IF I1>0 (OR I2>0).
C  DOUBLE SPACE FIRST IF I1<0 (OR I2<0).
C
C  E.G., USE TO TIME ELAPSED & CPU TIME FOR PROGRAM OR CODE SEGMENTS AS:
C
C     CALL SETTIME    ! DON'T FORGET TO DO THIS!
C     >>>>> THE CODE TO TIME IS HERE <<<<<  ! USUALLY A COMPLETE PROGRAM
C     CALL CPUTIME(-6,16) ! OR USE I1 OR I2=0 TO OMIT WRITE.
C
      SAVE
      INTEGER*4 ABSVAL(4),INCRVAL(4)
      CALL PROCINFO(ABSVAL,INCRVAL)
      TIMES=SECNDS(TIME0)
      MIN=TIMES/60.0
      SEC=AMOD(TIMES,60.0)
      CPUSEC=INCRVAL(1)*.01
      IMIN=CPUSEC/60.0
      CSEC=AMOD(CPUSEC,60.0)
      PCPU=100.*(CPUSEC/TIMES)
      IF(I1.NE.0) THEN
        IF(I1.GT.0) THEN
          J=1
        ELSE
          J=0
        ENDIF
      WRITE(IABS(I1),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,
     1 (INCRVAL(I),I=2,4)
60    FORMAT(I1,65('$')/' TOTAL "ELAPSED" TIME=',F16.2,' SEC. (',
     1 I4,' MIN.',F6.2,' SEC.)'/
     2 ' CPU_TIME=',F15.2,' SEC. (',I4,' M. ',F5.2,
     1 ' S.)    CPU % =',F6.2,'%'/
     3 ' BUF.I/O_COUNT=',I10/
     4 ' DIR.I/O_COUNT=',I10/
     5 ' PAGE_FAULTS=',2X,I10/
     6 ' ',65('$')//)
      ENDIF
      IF(I2.NE.0) THEN
        IF(I2.GT.0) THEN
          J=1
        ELSE
          J=0
        ENDIF
      WRITE(IABS(I2),60) J,TIMES,MIN,SEC,CPUSEC,IMIN,CSEC,PCPU,
     1 (INCRVAL(I),I=2,4)
      ENDIF
      RETURN
C** ENTRY 'CALL SETTIME'--MUST BE DONE BEFORE 'CALL CPUTIME(I1,I2)'
      ENTRY SETTIME()
      TIME0=SECNDS(0.0)
      CALL PROCINFO(ABSVAL,INCRVAL)
      RETURN
      END
      SUBROUTINE DECODEIX(NUMFLD,NUMLEN,IX,*)
C--USED IN CALL NAMELIST(IUNIT,'$NAME',*)
      CHARACTER*9 FMT
      CHARACTER*20 NUMFLD
      IF(NUMLEN.LT.1) RETURN 1
      IDIFF=20-NUMLEN
      IF(IDIFF.EQ.0) THEN
        ENCODE(9,991,FMT) NUMLEN
      ELSE
        ENCODE(9,992,FMT) NUMLEN,IDIFF
      ENDIF
991   FORMAT('(I',I2,'    )')
992   FORMAT('(I',I2,',',I2,'X)')
      DECODE(9,FMT,NUMFLD) IX
      RETURN
      END
      SUBROUTINE DECODEX(NUMFLD,NUMLEN,NDEC,X,*)
C--USED IN CALL NAMELIST(IUNIT,'$NAME',*)
      CHARACTER*12 FMT
      CHARACTER*20 NUMFLD
      IF(NUMLEN.LT.1) RETURN 1
      LENDEC=NUMLEN-NDEC
      IDIFF=20-NUMLEN
      IF(IDIFF.EQ.0) THEN
       ENCODE(12,991,FMT) NUMLEN,LENDEC
      ELSE
       ENCODE(12,992,FMT) NUMLEN,LENDEC,IDIFF
      ENDIF
991   FORMAT('(F',I2,'.',I2,'    )')
992   FORMAT('(F',I2,'.',I2,',',I2,'X)')
      DECODE(12,FMT,NUMFLD) X
      RETURN
      END
      SUBROUTINE ERRMSG(MSG,ISKIP,IUNIT1,IUNIT2)
C
C  GENERAL ERROR MESSAGE OUTPUT AND EXIT ON VAX-11/780
C
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2
C         > 0 FOR ONE BLANK LINE BEFORE.
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).
C
C  MESSAGES ARE WRITTEN IN THE FORM:
C
C  {ERRMSG}: _MSG_HERE_
C
      CHARACTER*(*) MSG
      I=LEN(MSG)
      DO 1 J=1,2
        IF(J.EQ.1) THEN
       JUNIT=IUNIT1
          ELSE
       JUNIT=IUNIT2
```

```
        ENDIF
        IF(JUNIT.GT.0) THEN
          IF(ISKIP.EQ.0) THEN
            WRITE(JUNIT,2) MSG
            ELSE
       WRITE(JUNIT,3) MSG
          ENDIF
      ENDIF
1     CONTINUE
      CALL EXIT
2     FORMAT(1X,'{ERRMSG}: ',A<I>)
3     FORMAT(/1X,'{ERRMSG}: ',A<I>)
      END
      SUBROUTINE MINMAX(A,N,AMIN,AMAX)
      DIMENSION A(1)
      AMIN=A(1)
      AMAX=AMIN
      DO 1 I=2,N
      AMIN=AMIN1(AMIN,A(I))
      AMAX=AMAX1(AMAX,A(I))
    1 CONTINUE
      RETURN
      END
      SUBROUTINE NONBLANK(C,NB)
C--DETERMINE NON-BLANK CHAR LENGTH (=NB ON EXIT) OF C*(*)
C  NOTE THAT NB WILL BE IN [0,LEN(C)].
C
      CHARACTER*(*) C
      L=LEN(C)
      DO 10 I=L,1,-1
        NB=I
        IF(C(I:I).NE.' ') RETURN
10    CONTINUE
      NB=0
      RETURN
      END
      SUBROUTINE PROCINFO(ABS_VALUES,INCR_VALUES)
C
C** SUBROUTINE TO OBTAIN ABSOLUTE AND INCREMENTAL VALUES OF PROCESS
C   PARAMETERS: CPU TIME, BUFFERED I/O COUNT, DIRECT I/O COUNT, AND
C   PAGE FAULTS.
C
      IMPLICIT INTEGER*2(W),INTEGER*4(L)
      PARAMETER (JPI$_CPUTIM = '00000407'X,
     1 JPI$_BUFIO = '0000040C'X,JPI$_DIRIO = '0000040B'X,
     2 JPI$_PAGEFLTS= '0000040A'X)
      INTEGER*4 ABS_VALUES(4),INCR_VALUES(4),LCL_VALUES(4)
      COMMON/ITEMLIST/
     1 W_LEN1,W_CODE1,L_ADDR1,L_LENADDR1,
     2 W_LEN2,W_CODE2,L_ADDR2,L_LENADDR2,
     3 W_LEN3,W_CODE3,L_ADDR3,L_LENADDR3,
     4 W_LEN4,W_CODE4,L_ADDR4,L_LENADDR4,
     5 W_LEN5,W_CODE5
      DATA W_LEN1,W_LEN2,W_LEN3,W_LEN4,W_LEN5/5*4/
      DATA W_CODE1/JPI$_CPUTIM/,
     1 W_CODE2/JPI$_BUFIO/,
     2 W_CODE3/JPI$_DIRIO/,
     3 W_CODE4/JPI$_PAGEFLTS/,
     4 W_CODE5/0/
      DATA L_LENADDR1,L_LENADDR2,L_LENADDR3,L_LENADDR4/4*0/
      L_ADDR1=%LOC(LCL_VALUES(1))
      L_ADDR2=%LOC(LCL_VALUES(2))
      L_ADDR3=%LOC(LCL_VALUES(3))
      L_ADDR4=%LOC(LCL_VALUES(4))
C** PERFORM THE SYSTEM SERVICE CALL
      CALL SYS$GETJPI(,,,W_LEN1,,,)
C** ASSIGN THE NEW VALUES TO THE ARGUMENTS
      DO I=1,4
        INCR_VALUES(I)=LCL_VALUES(I)-ABS_VALUES(I)
        ABS_VALUES(I)=LCL_VALUES(I)
      END DO
      RETURN
      END
      REAL FUNCTION RFLAGS(N,FUN,TOL,TO,TM,T,NEW)
C--FOURIER TRANSFORM LAG CONVOLUTION & SPLINE INTERPOLATION
C  GIVES FOURIER COSINE OR SINE TRANSFORMS VIA RLAGFO,RLAGF1
C  REF: ANDERSON,1975,NTIS REPT. PB-242-800,P.76-87.
C
C       N = 0 FOR COSINE TRANSFORM (VIA RLAGFO)
C       N = 1 FOR SINE TRANSFORM (VIA RLAGF1)
C     FUN = EXTERNAL REAL KERNEL FUNCTION.
C     TOL = TOLERANCE REQUESTED FOR RLAGFO OR RLAGF1
C      TO = TMIN TO USE (E.G., LET TO=.5*TMIN, TMIN=TRUE)
C      TM = TMAX TO USE (TM>TO)
C       T = TRANSFORM PARAMETER (TO<=T<=TM) FOR THIS CALL (NEW=1 OR 0)
C     NEW = 1 REQUIRED FOR 1ST CALL OR TO RESET SPLINE COEFFICIENTS.
C     NEW = 0 FOR ALL CALLS AFTER 1ST--USES SPLINE INTERPOLATION ONLY.
C
      REAL ARG(200),Y(200),AR(200),BR(200),CR(200),
     & D(2),W1(200),W2(200)
      EXTERNAL FUN
      DATA D/2*0.0/
      IF(NEW.EQ.0) GO TO 3
      NT=AINT(5.*ALOG(TM/TO))+5
      IF(NT.GT.200)CALL ERRMSG('IN RFLAGS: NT>200   ',4,6,16)
      NT1=NT+1
      XO=ALOG(TO)+.2*NT
      NU=1
      DO 1 J=1,NT
      I=NT1-J
      X=XO-.2*J
      EX=EXP(X)
      ARG(I)=EX
      IF(N.EQ.0) Y(I)=RLAGFO(X,FUN,TOL,L,NU)/EX
      IF(N.NE.0) Y(I)=RLAGF1(X,FUN,TOL,L,NU)/EX
1     NU=0
      CALL SPLIN1(NT,0.0,ARG,Y,AR,BR,CR,0,D,W1,W2)
2     IF(NT.LT.0) CALL ERRMSG('IN RFLAGS: NT<0 AFTER SPLIN1  ',6,6,16)
3     IF(T.LT.TO) CALL ERRMSG('IN RFLAGS: T<TO',3,6,16)
      IF(T.GT.TM) CALL ERRMSG('IN RFLAGS: T>TM',3,6,16)
      CALL SPOINT(NT,ARG,Y,AR,BR,CR,T,X)
      RFLAGS=X
      RETURN
      END
      SUBROUTINE SPLIN1(M,H,X,Y,A,B,C,IT,D,P,S)
C--ONE DIMENSIONAL CUBIC SPLINE COEFFICIENT DETERMINATION.
C
C         BY  W.L.ANDERSON, U.S. GEOLOGICAL SURVEY, DENVER, COLORADO
C
C  PARMS--- M= NUMBER OF DATA POINTS .GT. 2
C           H= EQUAL INTERVAL OPTION WHEN H.GT.0. (USE DUMMY X HERE),
C              UNEQUAL INTERVALS IF H=0. (X REQUIRED STORAGE)
C           X= INDEP.VAR WHEN H=0. (DIM .GE. M).
C           Y= DEPENDENT VARIABLE  (DIM .GE. M).
C           A,B,C=COEFF.ARRAYS (EACH DIM .GE. M)
C                 RESULTS ARE RETURNED IN 1ST(M-1) ELEMENTS OF A,B,&C.
C                 ALSO USED AS WORK ARRAYS DURING EXECUTION.
C           IT= TYPE OF BOUNDARY CONDITION SUPPLIED IN D ARRAY. USE
C              IT=1 IF 1ST DERIVATIVES GIVEN AT END POINTS, OR
C              IT=0 IF 2ND DERIVATIVES GIVEN AT END POINTS.
C           D= BOUNDARY ARRAY (DIM 2) AT POINT 1 AND M RESPECTIVELY.
C           P,S= WORK ARRAYS (EACH DIM=M).
C--ERROR RETURN WITH M=-(ABS(M)) IF ANY PARM OUT OF RANGE.
C  THE RESULTING CUBIC SPLINE IS OF THE FORM:
C     Y=Y(I)+A(I)*(X-X(I))+B(I)*(X-X(I))**2+C(I)*(X-X(I))**3
C       FOR I=1,2,...,M-1
C
C
      REAL*4  X(1),Y(1),A(1),B(1),C(1),D(2),P(1),S(1),MUL
      IF(IT.LT.0.OR.IT.GT.1.OR.H.LT.0..OR.M.LT.3) GO TO 999
      N=M-1
      IF(IT.EQ.0) GO TO 20
C--1ST DERIVATIVE BOUNDARIES GIVEN
      NE=N-1
      IF(H) 999,11,1
C--EQUAL SPACING H .GT. 0. AND IT=1
    1 HH=3.0/H
      DO 2 I=1,NE
      B(I)=4.0
      C(I)=1.0
      A(I)=1.0
    2 P(I)=HH*(Y(I+2)-Y(I))
      P(1)=P(1)-D(1)
      P(NE)=P(NE)-D(2)
C--SOLUTION OF TRIDIAGONAL MATRIX EQ. OF ORDER NE
    3 C(1)=C(1)/B(1)
      P(1)=P(1)/B(1)
      DO 4 I=2,NE
      MUL=1.0/(B(I)-A(I)*C(I-1))
      C(I)=MUL*C(I)
    4 P(I)=MUL*(P(I)-A(I)*P(I-1))
C--OBTAIN SPLINE COEFFICIENTS
      A(NE+IT)=P(NE)
      I=NE-1
    5 A(I+IT)=P(I)-C(I)*A(I+IT+1)
      I=I-1
       IF(I.GE.1) GO TO 5
      IF(IT.EQ.0) GO TO 6
      A(1)=D(1)
      A(M)=D(2)
    6 IF(H.EQ.0.) GO TO 14
```

```
      HH=1.0/H
      DO 7 I=1,N
      MUL=HH*(Y(I+1)-Y(I))
      B(I)=HH*(3.0*MUL-(A(I+1)+2.0*A(I)))
    7 C(I)=HH*HH*(-2.0*MUL+A(I+1)+A(I))
      RETURN
C--UNEQUAL SPACING H=0.. AND IT=1
   11 DO 12 I=1,N
   12 S(I+1)=X(I+1)-X(I)
      DO 13 I=1,NE
      B(I)=2.0*(S(I+1)+S(I+2))
      C(I)=S(I+1)
      A(I)=S(I+2)
   13 P(I)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/
     $ (S(I+1)*S(I+2))
      P(1)=P(1)-S(3)*D(1)
      P(NE)=P(NE)-S(N)*D(2)
      GO TO 3
   14 DO 15 I=1,N
      HH=1.0/S(I+1)
      MUL=(Y(I+1)-Y(I))*HH**2
      B(I)=3.0*MUL-(A(I+1)+2.0*A(I))*HH
   15 C(I)=-2.0*MUL*HH+(A(I+1)+A(I))*HH**2
      RETURN
C--2ND DERIVATIVE BOUNDARIES GIVEN
   20 NE=N+1
      IF(H) 999,31,21
C--EQUAL SPACING H .GT. 0 AND IT=0
   21 HH=3.0/H
      DO 22 I=2,N
      B(I)=4.0
      C(I)=1.0
      A(I)=1.0
   22 P(I)=HH*(Y(I+1)-Y(I-1))
      B(1)=2.0
      B(NE)=2.0
      C(1)=1.0
      C(NE)=1.0
      A(NE)=1.0
      P(1)=HH*(Y(2)-Y(1))-0.5*H*D(1)
      P(NE)=HH*(Y(M)-Y(N))+0.5*H*D(2)
      GO TO 3
C--UNEQUAL SPACING H=0 AND IT=0
   31 DO 32 I=1,N
   32 S(I+1)=X(I+1)-X(I)
      N1=N-1
      DO 33 I=1,N1
      B(I+1)=2.0*(S(I+1)+S(I+2))
      C(I+1)=S(I+1)
      A(I+1)=S(I+2)
   33 P(I+1)=3.0*(S(I+1)**2*(Y(I+2)-Y(I+1))+S(I+2)**2*(Y(I+1)-Y(I)))/
     *      (S(I+1)*S(I+2))
      B(1)=2.0
      B(NE)=2.0
      C(1)=1.0
      C(NE)=1.0
      A(NE)=1.0
      P(1)=3.0*(Y(2)-Y(1))/S(2)-0.5*S(2)*D(1)
      P(NE)=3.0*(Y(M)-Y(N))/S(M)+0.5*S(M)*D(2)
      GO TO 3
  999 M=-IABS(M)
      RETURN
      END
      SUBROUTINE SPOINT(M,X,Y,A,B,C,XX,YY)
C--GIVEN CUBIC SPLINE COEFF'S A,B,C,AND M OBS.DATA ARRAYS X,Y
C  SPOINT EVALUATES THE PIECEWISE CUBIC SPLINE ORDINATE YY AT THE
C  ABSCISSA XX, WHERE XX IS IN THE CLOSED INTERVAL (X(1),X(M)).
C  NOTE: IF COMPUTING OVER EQUAL INTERVALS, USE THE SUBR 'CUBIC'
C  WHICH REQUIRES ONLY ONE CALL.
C
      DIMENSION X(1),Y(1),A(1),B(1),C(1)
      IF(XX.LT.X(1).OR.XX.GT.X(M)) GO TO 9
      M1=M-1
      DO 1 I=1,M1
      J=I
      IF(XX.LE.X(I+1)) GO TO 2
    1 CONTINUE
    9 WRITE(6,60) XX,X(1),X(M)
   60 FORMAT('OERROR IN SPOINT CALL--XX=',E16.8,' NOT IN CLOSED INTERVAL
     * (',E16.8,',',E16.8,')')
      RETURN
    2 Z=XX-X(J)
      YY=Y(J)+((C(J)*Z+B(J))*Z+A(J))*Z
      RETURN
      END
      REAL*4 FUNCTION SQJ1(B,FUN,TOL,NF)
C=======================================================================
C** THIS IS A REAL*4 VERSION WRITTEN FOR THE VAX-11/780 BY
C  W.L.ANDERSON, U.S.GEOLOGICAL SURVEY, DENVER, COLORADO, USA.
C=======================================================================
C  SUBPROGRAM SQJ1 WILL COMPUTE THE FOLLOWING INFINITE INTEGRAL:
C  THE REAL*4 HANKEL TRANSFORM-SQUARE OF ORDER-1 FOR BOUNDED CONTINUOUS
C  KERNEL FUNCTIONS AND A FIXED TRANSFORM ARGUMENT B.GT.0.  THE
C  METHOD IS SIMILAR TO THE NEW=1 CASE FOR SINGLE-POWER J0,J1-FILTERS
C  DESIGNED AND PUBLISHED IN THE FOLLOWING REFERENCE:
C
C--REF: ANDERSON, W.L., 1979, GEOPHYSICS, VOL. 44, NO. 7, P. 1287-1305.
C
C--SPECIFICALLY,  SQJ1 EVALUATES THE INTEGRAL FROM 0 TO INFINITY OF
C  FUN(G)*[J1(G*B)]**2 *DG, DEFINED AS THE J1**2 HANKEL TRANSFORM OF
C  ORDER N=1 AND TRANSFORM ARGUMENT B.GT.0.  THE METHOD IS BY
C  ADAPTIVE DIGITAL FILTERING OF THE REAL*4 KERNEL FUNCTION FUN (SEE
C  THE ABOVE REFERENCE FOR ADDITIONAL INFORMATION).
C
C--PARAMETERS (ALL INPUT, EXCEPT NF)
C
C     B     = REAL*4 TRANSFORM ARGUMENT B>0.0 OF THE HANKEL TRANSFORM.
C     FUN(G)= EXTERNAL DECLARED REAL*4 FUNCTION NAME (USER SUPPLIED)
C             OF A REAL*4 ARGUMENT G>0. THIS REFERENCE MUST BE SUPPLIED.
C             IF PARAMETERS OTHER THAN G ARE REQUIRED IN FUN, USE COMMON
C             IN THE CALLING PROGRAM AND IN SUBPROGRAM FUN.  FUN(G)
C             MUST BE A CONTINUOUS BOUNDED FUNCTION FOR G.GT.0.
C             THE VALUE OF G IN FUN(G) MUST NOT BE CHANGED BY THE USER.
C             (G>0.0 WILL BE ASSIGNED AN ABSCISSA VALUE BY SQJ1.)
```

```
C     TOL   = REQUESTED REAL*4 TRUNCATION TOLERANCE USED AT THE FILTER  00012260
C             TAILS FOR ADAPTIVE FILTERING.  A TRUNCATION CRITERION IS  00012270
C             DEFINED DURING CONVOLUTION IN A FIXED ABSCISSA RANGE AS   00012280
C             THE MAX. ABSOLUTE CONVOLVED PRODUCT TIMES TOL.  TYPICALLY,00012290
C             TOL.LE.0.00001E0 WOULD GIVE ABOUT .01 PER CENT ACCURACY   00012300
C             FOR WELL-BEHAVED KERNELS AND MODERATE VALUES OF B.  FOR   00012310
C             VERY LARGE OR SMALL B, A VERY SMALL TOL SHOULD BE USED.   00012320
C             IN GENERAL, DECREASING THE TOLERANCE WOULD PRODUCE HIGHER 00012330
C             ACCURACY IN THE CONVOLUTION SINCE MORE FILTER WEIGHTS ARE 00012340
C             USED (UNLESS EXPONENT UNDERFLOWS OCCUR IN THE KERNEL      00012350
C             EVALUATION -- SEE NOTE (1) BELOW).                        00012360
C             FOR MAXIMUM ACCURACY POSSIBLE, TOL=0.0E0 MAY BE USED.     00012370
C     NF    = TOTAL NUMBER OF FUNCTION CALLS USED DURING CONVOLUTION.   00012380
C             NF IS IN THE RANGE 39.LE.NF.LE.441.  USUALLY,             00012390
C             NF IS MUCH LESS THAN 441 FOR TOL>0.                       00012400
C                                                                       00012410
C=======================================================================00012420
C--SUBPROGRAM USAGE--                                                   00012430
C  FUNCTION SQJ1 IS CALLED AS FOLLOWS (ASSUMES B>0.0, TOL>=0.0):        00012440
C     ...                                                               00012450
C     EXTERNAL FUN                                                      00012460
C     ...                                                               00012470
C     ANS=SQJ1(B,FUN,TOL,NF1)                                           00012480
C     ...                                                               00012490
C     END                                                               00012500
C     REAL*4 FUNCTION FUN(G)                                            00012510
C     ...USER SUPPLIED CODE FOR EVALUATION OF FUN(G), G.GT.0.           00012520
C     END                                                               00012530
C=======================================================================00012540
C--NOTES                                                                00012550
C       (1).  EXP-UNDERFLOW MAY OCCUR IN EXECUTING THIS SUBPROGRAM.     00012560
C             THIS IS OK PROVIDED THE MACHINE SYSTEM CONDITIONALLY SETS 00012570
C             EXP-UNDERFLOW TO 0.0D0.                                   00012580
C       (2).  ANSI FORTRAN (AMERICAN STANDARD X3.9-1978) IS USED, EXCEPT00012590
C             DATA STATEMENTS MAY NEED TO BE CHANGED FOR SOME COMPILERS.00012600
C       (3).  THE FILTER ABSCISSA CORRESPONDING TO EACH FILTER WEIGHT   00012610
C             IS GENERATED IN DOUBLE-PRECISION (TO REDUCE ROUND-OFF),   00012620
C             BUT IS USED IN SINGLE-PRECISION IN FUNCTION FUN.          00012630
C       (4).  NO CHECKS ARE MADE ON CALLING PARAMETERS (TO SAVE TIME),  00012640
C             HENCE UNPREDICTABLE RESULTS COULD OCCUR IF SQJ1           00012650
C             IS CALLED INCORRECTLY (OR IF FUNCTION FUN IS IN ERROR).   00012660
C=======================================================================00012670
C                                                                       00012680
      DOUBLE PRECISION E,ER,Y1,Y                                        00012690
      DIMENSION WT(441)                                                 00012700
      EQUIVALENCE (C,T),(CMAX,TMAX)                                     00012710
C-----E=DEXP(.2D0), ER=1.0D0/E                                          00012720
      DATA E/1.221402758160169834 D0/,ER/.818730753077981859 D0/        00012730
C--J1**2 TRANSFORM FILTER WEIGHT ARRAY WT:                              00012740
      DATA                                                              00012750
     *WT(  1)/-1.347588263343194E-23/,WT(  2)/-1.004450143483504E-25/,  00012760
     *WT(  3)/ 1.683503939595247E-25/,WT(  4)/-2.282980314410168E-25/,  00012770
     *WT(  5)/ 2.923585715694195E-25/,WT(  6)/-3.675806489042468E-25/,  00012780
     *WT(  7)/ 4.593049803693359E-25/,WT(  8)/-5.727028100634057E-25/,  00012790
     *WT(  9)/ 7.135771935139985E-25/,WT( 10)/-8.888811014230883E-25/,  00012800
     *WT( 11)/ 1.107156069674981E-24/,WT( 12)/-1.378989609333908E-24/,  00012810
     *WT( 13)/ 1.717546562904475E-24/,WT( 14)/-2.139214301317015E-24/,  00012820
     *WT( 15)/ 2.664399191263380E-24/,WT( 16)/-3.318515490721890E-24/,  00012830
     *WT( 17)/ 4.133215669675101E-24/,WT( 18)/-5.147922382202117E-24/,  00012840
     *WT( 19)/ 6.411736449429337E-24/,WT( 20)/-7.985813510726213E-24/,  00012850
     *WT( 21)/ 9.946324264352379E-24/,WT( 22)/-1.238814130360626E-23/,  00012860
     *WT( 23)/ 1.542943057770673E-23/,WT( 24)/-1.921736946991326E-23/,  00012870
     *WT( 25)/ 2.393526835149547E-23/,WT( 26)/-2.981144052786852E-23/,  00012880
     *WT( 27)/ 3.713025019486333E-23/,WT( 28)/-4.624587308718179E-23/,  00012890
     *WT( 29)/ 5.759943567573909E-23/,WT( 30)/-7.174036219523911E-23/,  00012900
     *WT( 31)/ 8.935296256138077E-23/,WT( 32)/-1.112895477589067E-22/,  00012910
     *WT( 33)/ 1.386116753020867E-22/,WT( 34)/-1.726415206560751E-22/,  00012920
     *WT( 35)/ 2.150258605026952E-22/,WT( 36)/-2.678157641135945E-22/,  00012930
     *WT( 37)/ 3.335658490307941E-22/,WT( 38)/-4.154579043649808E-22/   00012940
      DATA                                                               00012950
     *WT( 39)/ 5.174548642169893E-22/,WT( 40)/-6.444925823915958E-22/,  00012960
     *WT( 41)/ 8.027186890075815E-22/,WT( 42)/-9.997900876036018E-22/,  00012970
     *WT( 43)/ 1.245243489344099E-21/,WT( 44)/-1.550956915200967E-21/,  00012980
     *WT( 45)/ 1.931724499103375E-21/,WT( 46)/-2.405972408035303E-21/,  00012990
     *WT( 47)/ 2.996650524068346E-21/,WT( 48)/-3.732343038761098E-21/,  00013000
     *WT( 49)/ 4.648651704078530E-21/,WT( 50)/-5.789918678280031E-21/,  00013010
     *WT( 51)/ 7.211372338845832E-21/,WT( 52)/-8.981799902919285E-21/,  00013020
     *WT( 53)/ 1.118687618908690E-20/,WT( 54)/-1.393330960673053E-20/,  00013030
     *WT( 55)/ 1.735400600656860E-20/,WT( 56)/-2.161450028230063E-20/,  00013040
     *WT( 57)/ 2.692096696238658E-20/,WT( 58)/-3.353019744156364E-20/,  00013050
     *WT( 59)/ 4.176202667191459E-20/,WT( 60)/-5.201481066783855E-20/,  00013060
     *WT( 61)/ 6.478470381490309E-20/,WT( 62)/-8.068966885313877E-20/,  00013070
     *WT( 63)/ 1.004993814299769E-19/,WT( 64)/-1.251724763687751E-19/,  00013080
     *WT( 65)/ 1.559029380806346E-19/,WT( 66)/-1.941778800516568E-19/,  00013090
     *WT( 67)/ 2.418495094800596E-19/,WT( 68)/-3.012247595873759E-19/,  00013100
     *WT( 69)/ 3.751769271050626E-19/,WT( 70)/-4.672847173158759E-19/,  00013110
     *WT( 71)/ 5.820054253400369E-19/,WT( 72)/-7.248906342833808E-19/,  00013120
     *WT( 73)/ 9.028548683470782E-19/,WT( 74)/-1.124510201604046E-18/,  00013130
     *WT( 75)/ 1.400583014884058E-18/,WT( 76)/-1.744433068534399E-18/   00013140
      DATA                                                               00013150
     *WT( 77)/ 2.172700010108969E-18/,WT( 78)/-2.706108602890933E-18/,  00013160
     *WT( 79)/ 3.370471641997983E-18/,WT( 80)/-4.197939091349257E-18/,  00013170
     *WT( 81)/ 5.228553889932608E-18/,WT( 82)/-6.512189716201318E-18/,  00013180
     *WT( 83)/ 8.110964483209388E-18/,WT( 84)/-1.010224635873323E-17/,  00013190
     *WT( 85)/ 1.258239777819303E-17/,WT( 86)/-1.567143863125380E-17/,  00013200
     *WT( 87)/ 1.951885428378836E-17/,WT( 88)/-2.431082949794263E-17/,  00013210
     *WT( 89)/ 3.027925831533634E-17/,WT( 90)/-3.771296591112811E-17/,  00013220
     *WT( 91)/ 4.697168546872664E-17/,WT( 92)/-5.850346644633538E-17/,  00013230
     *WT( 93)/ 7.286635665898844E-17/,WT( 94)/-9.075540741890711E-17/,  00013240
     *WT( 95)/ 1.130363085713088E-16/,WT( 96)/-1.407872810977746E-16/,  00013250
     *WT( 97)/ 1.753512545608324E-16/,WT( 98)/-2.184008543691115E-16/,  00013260
     *WT( 99)/ 2.720193437373482E-16/,WT(100)/-3.388014372976862E-16/,  00013270
     *WT(101)/ 4.219788649509144E-16/,WT(102)/-5.255767622638629E-16/,  00013280
     *WT(103)/ 6.546084554824806E-16/,WT(104)/-8.153180672284374E-16/,  00013290
     *WT(105)/ 1.015482683093636E-15/,WT(106)/-1.264788701627155E-15/,  00013300
     *WT(107)/ 1.575300580104799E-15/,WT(108)/-1.962044659701490E-15/,  00013310
     *WT(109)/ 2.443736322630846E-15/,WT(110)/-3.043685669954619E-15/,  00013320
     *WT(111)/ 3.790925547775887E-15/,WT(112)/-4.721616509430640E-15/,  00013330
     *WT(113)/ 5.880796702857484E-15/,WT(114)/-7.324561363939633E-15/   00013340
      DATA                                                               00013350
     *WT(115)/ 9.122777386285771E-15/,WT(116)/-1.136246433122082E-14/,  00013360
     *WT(117)/ 1.415200549260716E-14/,WT(118)/-1.762639279859651E-14/,  00013370
     *WT(119)/ 2.195375936323478E-14/,WT(120)/-2.734351581078372E-14/,  00013380
     *WT(121)/ 3.405648410969600E-14/,WT(122)/-4.241751930962383E-14/,  00013390
     *WT(123)/ 5.283122998327256E-14/,WT(124)/-6.580155810347774E-14/,  00013400
     *WT(125)/ 8.195616589686861E-14/,WT(126)/-1.020768097542610E-13/,  00013410
     *WT(127)/ 1.271371711525175E-13/,WT(128)/-1.583499751338939E-13/,  00013420
     *WT(129)/ 1.972256768251710E-13/,WT(130)/-2.456455553861335E-13/,  00013430
     *WT(131)/ 3.059527536147695E-13/,WT(132)/-3.810656668935341E-13/,  00013440
     *WT(133)/ 4.746191721095855E-13/,WT(134)/-5.911405245505210E-13/,  00013450
     *WT(135)/ 7.362684464159483E-13/,WT(136)/-9.170259879582664E-13/,  00013460
     *WT(137)/ 1.142160404137243E-12/,WT(138)/-1.422566423922795E-12/,  00013470
     *WT(139)/ 1.771813523133346E-12/,WT(140)/-2.206802483504752E-12/,  00013480
     *WT(141)/ 2.748583435180205E-12/,WT(142)/-3.423374195288688E-12/,  00013490
     *WT(143)/ 4.263829592860520E-12/,WT(144)/-5.310620635098167E-12/,  00013500
     *WT(145)/ 6.614404617771952E-12/,WT(146)/-8.238272983750129E-12/,  00013510
     *WT(147)/ 1.026081093659580E-11/,WT(148)/-1.277988853702784E-11/,  00013520
     *WT(149)/ 1.591741836864029E-11/,WT(150)/-1.982521593601139E-11/,  00013530
     *WT(151)/ 2.469241780034636E-11/,WT(152)/-3.075450410688651E-11/   00013540
      DATA                                                              00013550
     *WT(153)/ 3.830493121765962E-11/,WT(154)/-4.770890042100824E-11/,  00013560
     *WT(155)/ 5.942181847402575E-11/,WT(156)/-7.400990050559746E-11/,  00013570
     *WT(157)/ 9.218018552223107E-11/,WT(158)/-1.148099864740444E-10/,  00013580
     *WT(159)/ 1.429980278676499E-10/,WT(160)/-1.781017947970137E-10/,  00013590
     *WT(161)/ 2.218320516379996E-10/,WT(162)/-2.762831033152235E-10/,  00013600
     *WT(163)/ 3.441298524411762E-10/,WT(164)/-4.285828501125686E-10/,  00013610
     *WT(165)/ 5.338614271394212E-10/,WT(166)/-6.648188696939778E-10/,  00013620
     *WT(167)/ 8.282322996645696E-10/,WT(168)/-1.031208263660785E-09/,  00013630
     *WT(169)/ 1.285029421731887E-09/,WT(170)/-1.599318044627408E-09/,  00013640
     *WT(171)/ 1.994131320471997E-09/,WT(172)/-2.479741335832154E-09/,  00013650
     *WT(173)/ 3.095744008520559E-09/,WT(174)/-3.842619853134365E-09/,  00013660
     *WT(175)/ 4.809953067874334E-09/,WT(176)/-5.947178675707624E-09/,  00013670
     *WT(177)/ 7.486760702417851E-09/,WT(178)/-9.179902118256913E-09/,  00013680
     *WT(179)/ 1.169762997406174E-08/,WT(180)/-1.408838665765728E-08/,  00013690
     *WT(181)/ 1.842372041425834E-08/,WT(182)/-2.134962991708034E-08/,  00013700
     *WT(183)/ 2.950136491767512E-08/,WT(184)/-3.144119891760005E-08/,  00013710
     *WT(185)/ 4.882266264702902E-08/,WT(186)/-4.320270488285106E-08/,  00013720
     *WT(187)/ 8.588959358375533E-08/,WT(188)/-4.852194659790488E-08/,  00013730
     *WT(189)/ 1.669435673170516E-07/,WT(190)/-1.385910316986003E-08/   00013740
      DATA                                                              00013750
     *WT(191)/ 3.708738131945973E-07/,WT(192)/ 1.823797026421996E-07/,  00013760
     *WT(193)/ 9.467983934893969E-07/,WT(194)/ 9.597040523684165E-07/,  00013770
     *WT(195)/ 2.701748498631580E-06/,WT(196)/ 3.734876413066559E-06/,  00013780
     *WT(197)/ 8.282486621513228E-06/,WT(198)/ 1.324461998560607E-05/,  00013790
     *WT(199)/ 2.641448646264069E-05/,WT(200)/ 4.523524985049475E-05/,  00013800
     *WT(201)/ 8.588684980817776E-05/,WT(202)/ 1.517873902314913E-04/,  00013810
     *WT(203)/ 2.813826579228376E-04/,WT(204)/ 5.038194131853568E-04/,  00013820
     *WT(205)/ 9.213190246955773E-04/,WT(206)/ 1.653407812381240E-03/,  00013830
     *WT(207)/ 2.987499535105453E-03/,WT(208)/ 5.321388722355372E-03/,  00013840
     *WT(209)/ 9.440908759768161E-03/,WT(210)/ 1.643647423786764E-02/,  00013850
     *WT(211)/ 2.808662676906598E-02/,WT(212)/ 4.626203993578857E-02/,  00013860
     *WT(213)/ 7.253448218727225E-02/,WT(214)/ 1.043469122686500E-01/,  00013870
     *WT(215)/ 1.316642401108199E-01/,WT(216)/ 1.297016728924245E-01/,  00013880
     *WT(217)/ 7.958314538535249E-02/,WT(218)/ 5.959581319466263E-03/,  00013890
     *WT(219)/ 3.637761733766417E-02/,WT(220)/ 1.209369201455565E-01/,  00013900
     *WT(221)/ 3.996142046468204E-02/,WT(222)/ 6.204935352917316E-02/,  00013910
     *WT(223)/ 7.416825136755058E-02/,WT(224)/ 5.350601024506145E-02/,  00013920
     *WT(225)/ 7.257325296747682E-02/,WT(226)/ 5.551076116171530E-02/,  00013930
     *WT(227)/ 7.378792584731115E-02/,WT(228)/ 4.699105146655452E-02/   00013940
      DATA                                                              00013950
     *WT(229)/ 9.712542002484704E-02/,WT(230)/-8.710790311395239E-03/,  00013960
     *WT(231)/ 2.243194656925174E-01/,WT(232)/-2.943323232952508E-01/,  00013970
     *WT(233)/ 8.579845401683572E-01/,WT(234)/-1.682517936763538E+00/,  00013980
     *WT(235)/ 3.850205803583565E+00/,WT(236)/-7.990070788264109E+00/,  00013990
     *WT(237)/ 1.672259034426730E+01/,WT(238)/-3.296742939854201E+01/,  00014000
     *WT(239)/ 6.113461761572366E+01/,WT(240)/-9.893170516119710E+01/,  00014010
     *WT(241)/ 1.202968925957973E+02/,WT(242)/-1.055273972563466E+02/,  00014020
     *WT(243)/ 6.683237989195774E+01/,WT(244)/-2.793810874106434E+01/,  00014030
     *WT(245)/ 1.590990990202553E+00/,WT(246)/ 1.160669943373125E+01/,  00014040
     *WT(247)/-1.590041269652672E+01/,WT(248)/ 1.551471062170340E+01/,  00014050
     *WT(249)/-1.322642619402104E+01/,WT(250)/ 1.051035957858022E+01/,  00014060
     *WT(251)/-8.018476352610196E+00/,WT(252)/ 5.966975882060087E+00/,  00014070
     *WT(253)/-4.371574296080611E+00/,WT(254)/ 3.171154372941820E+00/,  00014080
     *WT(255)/-2.285863933761940E+00/,WT(256)/ 1.641078862938996E+00/,  00014090
     *WT(257)/-1.175142006825595E+00/,WT(258)/ 8.401183229994065E-01/,  00014100
     *WT(259)/-5.999848865403200E-01/,WT(260)/ 4.282092044767511E-01/,  00014110
     *WT(261)/-3.054871785604701E-01/,WT(262)/ 2.178803108055725E-01/,  00014120
     *WT(263)/-1.553720956115054E-01/,WT(264)/ 1.107858980261850E-01/,  00014130
     *WT(265)/-7.898940980104745E-02/,WT(266)/ 5.631660644909182E-02/   00014140
      DATA                                                              00014150
     *WT(267)/-4.015075436876059E-02/,WT(268)/ 2.862493689156548E-02/,  00014160
     *WT(269)/-2.040757592276983E-02/,WT(270)/ 1.454909156012523E-02/,  00014170
     *WT(271)/-1.037239010986780E-02/,WT(272)/ 7.394705684860287E-03/,  00014180
     *WT(273)/-5.271841981221853E-03/,WT(274)/ 3.758404609682646E-03/,  00014190
     *WT(275)/-2.679442677158320E-03/,WT(276)/ 1.910228325289486E-03/,  00014200
     *WT(277)/-1.361839769584138E-03/,WT(278)/ 9.708825536314996E-04/,  00014210
     *WT(279)/-6.921613621151932E-04/,WT(280)/ 4.934554956175334E-04/,  00014220
     *WT(281)/-3.517941529895520E-04/,WT(282)/ 2.508009847086945E-04/,  00014230
     *WT(283)/-1.788009630614640E-04/,WT(284)/ 1.274707284368542E-04/,  00014240
     *WT(285)/-9.087639280562825E-05/,WT(286)/ 6.478757008493505E-05/,  00014250
     *WT(287)/-4.618833453533555E-05/,WT(288)/ 3.292857322377349E-05/,  00014260
     *WT(289)/-2.347542826480734E-05/,WT(290)/ 1.673609507212648E-05/,  00014270
     *WT(291)/-1.193149173068922E-05/,WT(292)/ 8.506195399309457E-06/,  00014280
     *WT(293)/-6.064234196352591E-06/,WT(294)/ 4.323311969745620E-06/,  00014290
     *WT(295)/-3.082174233732943E-06/,WT(296)/ 2.197342702408592E-06/,  00014300
     *WT(297)/-1.566528880471568E-06/,WT(298)/ 1.116809285437889E-06/,  00014310
     *WT(299)/-7.961953306989225E-07/,WT(300)/ 5.676233291497361E-07/,  00014320
     *WT(301)/-4.046698484300778E-07/,WT(302)/ 2.884971033054787E-07/,  00014330
     *WT(303)/-2.056752657469096E-07/,WT(304)/ 1.466299469054255E-07/   00014340
      DATA                                                              00014350
     *WT(305)/-1.045353764411399E-07/,WT(306)/ 7.452532827238916E-08/,  00014360
     *WT(307)/-5.313057400462258E-08/,WT(308)/ 3.787783240273793E-08/,  00014370
     *WT(309)/-2.700385257281518E-08/,WT(310)/ 1.925157823238107E-08/,  00014380
     *WT(311)/-1.372482920494808E-08/,WT(312)/ 9.784700996002279E-09/,  00014390
     *WT(313)/-6.975706010727724E-09/,WT(314)/ 4.973118173767800E-09/,  00014400
     *WT(315)/-3.545433871815285E-09/,WT(316)/ 2.527609620402724E-09/,  00014410
     *WT(317)/-1.801982669579813E-09/,WT(318)/ 1.284668928008203E-09/,  00014420
     *WT(319)/-9.158657752100248E-10/,WT(320)/ 6.529387454724082E-10/,  00014430
     *WT(321)/-4.654928886728159E-10/,WT(322)/ 3.318590463615230E-10/,  00014440
     *WT(323)/-2.365888487920329E-10/,WT(324)/ 1.686688489780138E-10/,  00014450
     *WT(325)/-1.202473436969784E-10/,WT(326)/ 8.572669911362267E-11/,  00014460
     *WT(327)/-6.111625184367596E-11/,WT(328)/ 4.357097938028588E-11/,  00014470
     *WT(329)/-3.106260915694127E-11/,WT(330)/ 2.214514572223418E-11/,  00014480
     *WT(331)/-1.578771044574020E-11/,WT(332)/ 1.125536965278491E-11/,  00014490
     *WT(333)/-8.024174655104147E-12/,WT(334)/ 5.720592115753784E-12/,  00014500
     *WT(335)/-4.078322763576438E-12/,WT(336)/ 2.907516604461514E-12/,  00014510
     *WT(337)/-2.072825839268808E-12/,WT(338)/ 1.477758356855950E-12/,  00014520
     *WT(339)/-1.053523031162099E-12/,WT(340)/ 7.510773138515030E-13/,  00014530
     *WT(341)/-5.354578065181299E-13/,WT(342)/ 3.817384139735767E-13/   00014540
      DATA                                                              00014550
     *WT(343)/-2.721488321379581E-13/,WT(344)/ 1.940202613174297E-13/,  00014560
     *WT(345)/-1.383208647487461E-13/,WT(346)/ 9.861166815737169E-14/,  00014570
     *WT(347)/-7.030219999306174E-14/,WT(348)/ 5.011982269661039E-14/,  00014580
     *WT(349)/-3.573140851051000E-14/,WT(350)/ 2.547362471478360E-14/,  00014590
     *WT(351)/-1.816064866065287E-14/,WT(352)/ 1.294708403175416E-14/,  00014600
     *WT(353)/-9.230231147441696E-15/,WT(354)/ 6.580413537615677E-15/,  00014610
     *WT(355)/-4.691306385976846E-15/,WT(356)/ 3.344524699139429E-15/,  00014620
     *WT(357)/-2.384377515096891E-15/,WT(358)/ 1.699869681321967E-15/,  00014630
     *WT(359)/-1.211870567970942E-15/,WT(360)/ 8.639663908659455E-16/,  00014640
     *WT(361)/-6.159386524219969E-16/,WT(362)/ 4.391147937677918E-16/,  00014650
     *WT(363)/-3.130535830922384E-16/,WT(364)/ 2.231820637286411E-16/,  00014660
     *WT(365)/-1.591108879130287E-16/,WT(366)/ 1.134332850476810E-16/,  00014670
     *WT(367)/-8.086882252666564E-17/,WT(368)/ 5.765297596814659E-17/,  00014680
     *WT(369)/-4.110194181301590E-17/,WT(370)/ 2.930238365717713E-17/,  00014690
     *WT(371)/-2.089024630262032E-17/,WT(372)/ 1.489306793905684E-17/,  00014700
     *WT(373)/-1.061756139317321E-17/,WT(374)/ 7.569468587585003E-18/,  00014710
     *WT(375)/-5.396423206478743E-18/,WT(376)/ 3.847216364856985E-18/,  00014720
     *WT(377)/-2.742756302043894E-18/,WT(378)/ 1.955364975352951E-18/,  00014730
     *WT(379)/-1.394018194041768E-18/,WT(380)/ 9.938230201715542E-19/   00014740
      DATA                                                              00014750
     *WT(381)/-7.085160004787920E-19/;WT(382)/ 5.051150081588374E-19/,  00014760
     *WT(383)/-3.601064355624574E-19/,WT(384)/ 2.567269687909788E-19/,  00014770
     *WT(385)/-1.830257112802977E-19/,WT(386)/ 1.304826335452211E-19/,  00014780
     *WT(387)/-9.302363879800099E-20/,WT(388)/ 6.631838383372175E-20/,  00014790
     *WT(389)/-4.727968171282729E-20/,WT(390)/ 3.370661607632613E-20/,  00014800
     *WT(391)/-2.403011031359726E-20/,WT(392)/ 1.713153881218087E-20/,  00014810
     *WT(393)/-1.221341134602363E-20/,WT(394)/ 8.707181429542280E-21/,  00014820
     *WT(395)/-6.207521077792512E-21/,WT(396)/ 4.425463988842000E-21/,  00014830
     *WT(397)/-3.155000395835719E-21/,WT(398)/ 2.249261879409991E-21/,  00014840
     *WT(399)/-1.603543051538933E-21/,WT(400)/ 1.143197379660855E-21/,  00014850
     *WT(401)/-8.150078813546612E-22/,WT(402)/ 5.810351225960625E-22/,  00014860
     *WT(403)/-4.142313344616877E-22/,WT(404)/ 2.953136303480927E-22/,  00014870
     *WT(405)/-2.105348607743343E-22/,WT(406)/ 1.500944118463614E-22/,  00014880
     *WT(407)/-1.070052322309586E-22/,WT(408)/ 7.628611471159240E-23/,  00014890
     *WT(409)/-5.438585792837369E-23/,WT(410)/ 3.877274007553577E-23/,  00014900
     *WT(411)/-2.764184653478325E-23/,WT(412)/ 1.970641801894714E-23/,  00014910
     *WT(413)/-1.404909843147521E-23/,WT(414)/ 1.001588663330109E-23/,  00014920
     *WT(415)/-7.140532395899525E-24/,WT(416)/ 5.090636832712442E-24/,  00014930
     *WT(417)/-3.629226189021914E-24/,WT(418)/ 2.587357217905591E-24/   00014940
      DATA                                                              00014950
     *WT(419)/-1.844586958423583E-24/,WT(420)/ 1.315049505312082E-24/,  00014960
     *WT(421)/-9.375296774269097E-25/,WT(422)/ 6.683863590894837E-25/,  00014970
     *WT(423)/-4.765072157206098E-25/,WT(424)/ 3.397118112049352E-25/,  00014980
     *WT(425)/-2.421872841072187E-25/,WT(426)/ 1.726603263516931E-25/,  00014990
     *WT(427)/-1.230938988644950E-25/,WT(428)/ 8.775817915630044E-26/,  00015000
     *WT(429)/-6.256819583036690E-26/,WT(430)/ 4.461171435353558E-26/,  00015010
     *WT(431)/-3.181251616647866E-26/,WT(432)/ 2.269032642804270E-26/,  00015020
     *WT(433)/-1.618955905684386E-26/,WT(434)/ 1.155727651063629E-26/,  00015030
     *WT(435)/-8.256240401524400E-27/,WT(436)/ 5.903135682837950E-27/,  00015040
     *WT(437)/-4.224588415734394E-27/,WT(438)/ 3.025918616503642E-27/,  00015050
     *WT(439)/-2.168627753535370E-27/,WT(440)/ 1.554288235465150E-27/,  00015060
     *WT(441)/-4.937813102320317E-28/                                   00015070
C                                                                       00015080
C FOLLOWING CODE FOR STARTING WEIGHT=214 FROM TOTAL WTS=441.             00015090
C                                                                       00015100
```

```
      NONE=0
C-----INITIALIZE KERNEL ABSCISSA GENERATION FOR GIVEN B
      Y1=0.13142582398223379lD1/DBLE(B)
  100 SQJ1=0.0E0
      CMAX=0.0E0
      NF=0
      Y=Y1
C-----BEGIN RIGHT-SIDE CONVOLUTION AT WEIGHT 214
      ASSIGN 110 TO M
      I=214
      Y=Y*E
      GO TO 200
  110 TMAX=AMAX1(ABS(T),TMAX)
      I=I+1
      Y=Y*E
      IF(I.LE.250) GO TO 200
      IF(TMAX.EQ.0.0E0) NONE=1
C-----ESTABLISH TRUNCATION CRITERION (CMAX=TMAX)
      CMAX=TOL*CMAX
      ASSIGN 120 TO M
      GO TO 200
C-----CHECK FOR FILTER TRUNCATION AT RIGHT END
  120 IF(ABS(T).LE.TMAX) GO TO 130
      I=I+1
      Y=Y*E
      IF(I.LE.441) GO TO 200
  130 Y=Y1
C-----CONTINUE WITH LEFT-SIDE CONVOLUTION AT WEIGHT 213
      ASSIGN 140 TO M
      I=213
      GO TO 200
C-----CHECK FOR FILTER TRUNCATION AT LEFT END
  140 IF(ABS(T).LE.TMAX.AND.
     * NONE.EQ.0) GO TO 190
      I=I-1
      Y=Y*ER
      IF(I.GT.0) GO TO 200
C-----NORMALIZE BY B TO ACCOUNT FOR INTEGRATION RANGE CHANGE
  190 SQJ1=SQJ1/B
      RETURN
  200 C=FUN(SNGL(Y))*WT(I)
      NF=NF+1
      SQJ1=SQJ1+C
      GO TO M,(110,120,140)
      END
      SUBROUTINE WARN(MSG,ISKIP,IUNIT1,IUNIT2,*)
C
C  GENERAL WARNING MESSAGE OUTPUT AND RETURN 1 ON VAX-11/780
C
C  MSG*(*) = VARIABLE-LENGTH 'MESSAGE'
C  ISKIP = 0 FOR NO BLANK LINE BEFORE OUTPUT TO IUNIT1 & IUNIT2
C         > 0 FOR ONE BLANK LINE BEFORE.
C  IUNIT1 = 0 TO SUPPRESS OUTPUT ON IUNIT1 (>0 TO WRITE ON IUNIT1).
C  IUNIT2 = 0 TO SUPPRESS OUTPUT ON IUNIT2 (>0 TO WRITE ON IUNIT2).
C
C  MESSAGES ARE WRITTEN IN THE FORM:
C
```

```
C  {WARN}: _MSG_HERE_
C
      CHARACTER*(*) MSG
      I=LEN(MSG)
      DO 1 J=1,2
        IF(J.EQ.1) THEN
       JUNIT=IUNIT1
          ELSE
       JUNIT=IUNIT2
        ENDIF
        IF(JUNIT.GT.0) THEN
          IF(ISKIP.EQ.0) THEN
            WRITE(JUNIT,2) MSG
            ELSE
       WRITE(JUNIT,3) MSG
          ENDIF
      ENDIF
1     CONTINUE
      RETURN 1
2     FORMAT(1X,'{WARN}: ',A<I>)
3     FORMAT(/1X,'{WARN}: ',A<I>)
      END
        REAL FUNCTION RLAGFO(X,FUN,TOL,L,NEW)
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*COS(G*B)*DG' DEFINED AS THE
C  REAL FOURIER COSINE TRANSFORM WITH ARGUMENT X(=ALOG(B))
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....
C
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.
C
C--PARAMETERS:
C
C     * X     = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM
C               'RLAGFO' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,
C               SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT,
C               THEN SUBPROGRAM 'RFOURO' IS ADVISED FOR GENERAL USE.
C               (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN
C                 CALLING PROGRAM AND IN SUBPROGRAM FUN.
C                 THE REAL FUNCTION FUN SHOULD BE A MONOTONE
C                 DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...
C       TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,
C                 IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.
C                 THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,
C                 TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON
C                 THE FUNCTION FUN AND PARAMETER X...IN GENERAL,
C               A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'
C               BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY
C               RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN
C               APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,
C               ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...
C       L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE
C                 CONVOLUTION (L DEPENDS ON TOL AND FUN).
C               MIN.L=24 AND MAX.L=281--WHICH COULD
C                 OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING
```

```
C                 VERY FAST...
C     * NEW=    1 IS NECESSARY 1ST TIME OR BRAND NEW X.
C               O FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20
C                 IS ASSUMED INTERNALLY BY THIS ROUTINE.
C                 NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL
C                 STILL ASSUME X=(LAST X)-0.20 ANYWAY...
C                 IT IS THE USERS RESPONSIBILITY TO NORMALIZE
C                 BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).
C               THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT
C                 TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A
C                 SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING
C                 ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...
C                 THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED
C                 KERNELS WILL BE USED IN THE LAGGED CONVOLUTION
C                 WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS
C                 WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)
C
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGFO; THE FOURIER
C  TRANSFORM IS THEN RLAGFO/B WHICH IS TO BE COMPUTED AFTER EXIT FROM
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...
C
C--USAGE-- 'RLAGFO' IS CALLED AS FOLLOWS:
C     ...
C     EXTERNAL RF
C     ...
C     R=RLAGFO(ALOG(B),RF,TOL,L,NEW)/B
C     ...
C     END
C     REAL FUNCTION RF(G)
C     ...USER SUPPLIED CODE...
C     END
C
C--NOTES:
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:
C               ...
C               NB=AINT(5.*ALOG(BMAX/BMIN))+1
C               NB1=NB+1
C               XO=ALOG(BMAX)+.2
C               NEW=1
C               DO 1 J=1,NB
C               I=NB1-J
C               X=XO-.2*J
C               ARG(I)=EXP(X)
C               ANS(I)=RLAGFO(X,RF,TOL,L,NEW)/ARG(I)
C     1         NEW=0
C               ...
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,
C      TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)          00016820
C      SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.               00016830
C      (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY          00016840
C      ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW              00016850
C      BMAX,BMIN AND BY SETTING NEW=1....                                00016860
C      (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE00016870
C                                                                        00016880
C                                                                        00016890
        DIMENSION KEY(281),SAVE(281)                                     00016900
        DIMENSION YT(281),Y1(76),Y2(76),Y3(76),Y4(53)                    00016910
        EQUIVALENCE (YT(1),Y1(1)),(YT(77),Y2(1)),(YT(153),Y3(1)),        00016920
     1 (YT(229),Y4(1))                                                   00016930
C--COS-EXTENDED FILTER WEIGHT ARRAYS:                                    00016940
       DATA Y1/                                                          00016950
     1 5.1178101E-14, 2.9433849E-14, 2.5492522E-14, 1.9034819E-14,       00016960
     2 6.4179780E-14, 1.3085746E-15, 1.1989957E-13,-1.2216234E-14,       00016970
     3 1.7534103E-13, 7.9373498E-15, 2.1235658E-13, 7.9981520E-14,       00016980
     4 2.3815757E-13, 1.9714260E-13, 2.8920132E-13, 3.4161340E-13,       00016990
     5 4.0349917E-13, 5.2203885E-13, 5.9837223E-13, 7.8015306E-13,       00017000
     6 8.8911655E-13, 1.1709731E-12, 1.3165595E-12, 1.7578463E-12,       00017010
     7 1.9538564E-12, 2.6289768E-12, 2.9167697E-12, 3.9044344E-12,       00017020
     8 4.3927341E-12, 5.7526904E-12, 6.6569552E-12, 8.4555678E-12,       00017030
     9 1.0063229E-11, 1.2487964E-11, 1.5134682E-11, 1.8501488E-11,       00017040
     1 2.2720051E-11, 2.7452598E-11, 3.4025443E-11, 4.0875985E-11,       00017050
     2 5.0751668E-11, 6.1094382E-11, 7.5492982E-11, 9.1445759E-11,       00017060
     3 1.1227336E-10, 1.3676464E-10, 1.6720269E-10, 2.0423244E-10,       00017070
     4 2.4932743E-10, 3.0470661E-10, 3.7198526E-10, 4.5449934E-10,       00017080
     5 5.5502537E-10, 6.7793669E-10, 8.2810001E-10, 1.0112626E-09,       00017090
     6 1.2354800E-09, 1.5085255E-09, 1.8432253E-09, 2.2503397E-09,       00017100
     7 2.7499027E-09, 3.3569525E-09, 4.1025670E-09, 5.0077487E-09,       00017110
     8 6.1205950E-09, 7.4703399E-09, 9.1312760E-09, 1.1143911E-08,       00017120
     9 1.3622929E-08, 1.6623917E-08, 2.0324094E-08, 2.4798610E-08,       00017130
     1 3.0321709E-08, 3.6992986E-08, 4.5237482E-08, 5.5183434E-08/       00017140
       DATA Y2/                                                          00017150
     1 6.7491070E-08, 8.2317946E-08, 1.0069271E-07, 1.2279375E-07,       00017160
     2 1.5022907E-07, 1.8316969E-07, 2.2413747E-07, 2.7322865E-07,       00017170
     3 3.3441046E-07, 4.0756197E-07, 4.9894278E-07, 6.0793233E-07,       00017180
     4 7.4443665E-07, 9.0679753E-07, 1.1107379E-06, 1.3525651E-06,       00017190
     5 1.6573073E-06, 2.0174273E-06, 2.4728798E-06, 3.0090445E-06,       00017200
     6 3.6898816E-06, 4.4879625E-06, 5.5059521E-06, 6.6935820E-06,       00017210
     7 8.2160716E-06, 9.9828691E-06, 1.2260527E-05, 1.4888061E-05,       00017220
     8 1.8296530E-05, 2.2202672E-05, 2.7305154E-05, 3.3109672E-05,       00017230
     9 4.0751046E-05, 4.9372484E-05, 6.0820947E-05, 7.3619571E-05,       00017240
     1 9.0780005E-05, 1.0976837E-04, 1.3550409E-04, 1.6365676E-04,       00017250
     2 2.0227521E-04, 2.4398338E-04, 3.0197018E-04, 3.6370760E-04,       00017260
     3 4.5083748E-04, 5.4213338E-04, 6.7315347E-04, 8.0800951E-04,       00017270
     4 1.0051938E-03, 1.2041401E-03, 1.5011708E-03, 1.7942344E-03,       00017280
     5 2.2421056E-03, 2.6730676E-03, 3.3490681E-03, 3.9815050E-03,       00017290
     6 5.0028666E-03, 5.9285668E-03, 7.4730905E-03, 8.8233510E-03,       00017300
     7 1.1160132E-02, 1.3119627E-02, 1.6653199E-02, 1.9472767E-02,       00017310
     8 2.4800811E-02, 2.8793704E-02, 3.6762063E-02, 4.2228780E-02,       00017320
     9 5.3905163E-02, 6.0804660E-02, 7.7081738E-02, 8.3874501E-02,       00017330
     1 1.0377190E-01, 1.0377718E-01, 1.1892208E-01, 9.0437429E-02/       00017340
       DATA Y3/                                                          00017350
     1 7.1685138E-02,-3.9473064E-02,-1.5078720E-01,-4.0489859E-01,       00017360
     2-5.6018995E-01,-6.8050388E-01,-1.5094224E-01, 6.6304064E-01,       00017370
     3 1.3766748E+00,-8.0373222E-01,-1.0869629E+00, 1.2812892E+00,       00017380
     4-5.0341082E-01,-4.4274455E-02, 2.0913102E-01,-1.9999661E-01,      00017390
     5 1.5207664E-01,-1.0920260E-01, 7.8169956E-02,-5.6651561E-02,      00017400
     6 4.1611799E-02,-3.0880012E-02, 2.3072559E-02,-1.7311631E-02,      00017410
     7 1.3021442E-02,-9.8085025E-03, 7.3943529E-03,-5.5769518E-03,      00017420
     8 4.2073164E-03,-3.1745026E-03, 2.3954154E-03,-1.8076122E-03,      00017430
     9 1.3640816E-03,-1.0293934E-03, 7.7682952E-04,-5.8623518E-04,      00017440
     1 4.4240399E-04,-3.3386183E-04, 2.5195025E-04,-1.9013541E-04,      00017450
     2 1.4348659E-04,-1.0828284E-04, 8.1716174E-05,-6.1667509E-05,      00017460
     3 4.6537684E-05,-3.5119887E-05, 2.6503388E-05,-2.0000904E-05,      00017470
     4 1.5093768E-05,-1.1390572E-05, 8.5959318E-06,-6.4869407E-06,      00017480
     5 4.8953713E-06,-3.6942830E-06, 2.7878625E-06,-2.1038241E-06,      00017490
     6 1.5875917E-06,-1.1980090E-06, 9.0398030E-07,-6.8208296E-07,      00017500
     7 5.1458650E-07,-3.8817581E-07, 2.9272267E-07,-2.2067921E-07,      00017510
     8 1.6623514E-07,-1.2514102E-07, 9.4034535E-08,-7.0556837E-08,      00017520
     9 5.2741581E-08,-3.9298610E-08, 2.9107255E-08,-2.1413893E-08,      00017530
     1 1.5742032E-08,-1.1498608E-08, 8.7561571E-09,-7.2959446E-09/      00017540
      DATA Y4/                                                          00017550
     1 6.8816619E-09,-8.9679825E-09, 1.4258275E-08,-1.9564299E-08,      00017560
     2 2.0235313E-08,-1.4725545E-08, 5.4632820E-09, 3.5995580E-09,      00017570
     3-9.5287133E-09, 1.1460041E-08,-1.0250532E-08, 7.4641748E-09,      00017580
     4-4.4703465E-09, 2.0499053E-09,-4.4806353E-10,-4.0374336E-10,      00017590
     5 7.0321001E-10,-6.7067960E-10, 4.9130404E-10,-2.8840747E-10,      00017600
     6 1.2373144E-10,-1.5260443E-11,-4.2027559E-11, 6.1885474E-11,      00017610
     7-5.9273937E-11, 4.6588766E-11,-3.2054182E-11, 1.9831637E-11,      00017620
     8-1.1210098E-11, 5.9567021E-12,-3.2427812E-12, 2.1353868E-12,      00017630
     9-1.8476851E-12, 1.8438474E-12,-1.8362842E-12, 1.7241847E-12,      00017640
     1-1.5161479E-12, 1.2627657E-12,-1.0129176E-12, 7.9578625E-13,      00017650
     2-6.2131435E-13, 4.8745900E-13,-3.8703630E-13, 3.1172547E-13,      00017660
     3-2.5397802E-13, 2.0824130E-13,-1.7123163E-13, 1.4113344E-13,      00017670
     4-1.1687986E-13, 9.7664016E-14,-8.2977176E-14, 7.2515267E-14,      00017680
     5-5.6047478E-14/                                                   00017690
C--$$ENDATA                                                             00017700
      IF(NEW) 10,30,10                                                  00017710
10    LAG=-1                                                            00017720
      X0=-X-30.30251236                                                 00017730
      DO 20 IR=1,281                                                    00017740
20    KEY(IR)=0                                                         00017750
30    LAG=LAG+1                                                         00017760
      RLAGF0=0.0                                                        00017770
      CMAX=0.0                                                          00017780
      L=0                                                               00017790
      ASSIGN 110 TO M                                                   00017800
      I=149                                                             00017810
      GO TO 200                                                         00017820
110   CMAX=AMAX1(ABS(C),CMAX)                                           00017830
      I=I+1                                                             00017840
      IF(I.LE.170) GO TO 200                                            00017850
      IF(CMAX.EQ.0.0) GO TO 150                                         00017860
      CMAX=TOL*CMAX                                                     00017870
      ASSIGN 120 TO M                                                   00017880
      I=148                                                             00017890
      GO TO 200                                                         00017900
120   IF(ABS(C).LE.CMAX) GO TO 130                                      00017910
      I=I-1                                                             00017920
      IF(I.GT.0) GO TO 200                                              00017930
130   ASSIGN 140 TO M                                                   00017940
      I=171                                                             00017950
      GO TO 200
140   IF(ABS(C).LE.CMAX) GO TO 190
      I=I+1
      IF(I.LE.281) GO TO 200
      GO TO 190
150   ASSIGN 160 TO M
      I=1
      GO TO 200
160   IF(C.EQ.0.0) GO TO 170
      I=I+1
      IF(I.LE.148) GO TO 200
170   ASSIGN 180 TO M
      I=281
      GO TO 200
180   IF(C.EQ.0.0) GO TO 190
      I=I-1
      IF(I.GE.171) GO TO 200
190   RETURN
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)
200   LOOK=I+LAG
      IQ=LOOK/282
      IR=MOD(LOOK,282)
      IF(IR.EQ.0) IR=1
      IROLL=IQ*281
      IF(KEY(IR).LE.IROLL) GO TO 220
210   C=SAVE(IR)*YT(I)
      RLAGF0=RLAGF0+C
      L=L+1
      GO TO M,(110,120,140,160,180)
220   KEY(IR)=IROLL+IR
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))
      GO TO 210
      END
        REAL FUNCTION RLAGF1(X,FUN,TOL,L,NEW)
C--*** A SPECIAL LAGGED* CONVOLUTION METHOD TO COMPUTE THE
C  INTEGRAL FROM 0 TO INFINITY OF 'FUN(G)*SIN(G*B)*DG' DEFINED AS THE
C  REAL FOURIER SINE TRANSFORM WITH ARGUMENT X(=ALOG(B))
C  BY CONVOLUTION FILTERING WITH REAL FUNCTION 'FUN'--AND
C  USING A VARIABLE CUT-OFF METHOD WITH EXTENDED FILTER TAILS....
C
C--REF: ANDERSON, W.L., 1975, NTIS REPT. PB-242-800.
C
C--PARAMETERS:
C
C     * X     = REAL ARGUMENT(=ALOG(B) AT CALL) OF THE FOURIER TRANSFORM
C               'RLAGF1' IS USEFUL ONLY WHEN X=(LAST X)-.20 *** I.E.,
C               SPACED SAME AS FILTER USED--IF THIS IS NOT CONVENIENT,
C               THEN SUBPROGRAM 'RFOUR1' IS ADVISED FOR GENERAL USE.
C               (ALSO SEE PARM 'NEW' & NOTES (2)-(4) BELOW).
C     FUN(G)= EXTERNAL DECLARED REAL FUNCTION NAME (USER SUPPLIED).
C             NOTE: IF PARMS OTHER THAN G ARE REQUIRED, USE COMMON IN
C                 CALLING PROGRAM AND IN SUBPROGRAM FUN.
C                 THE REAL FUNCTION FUN SHOULD BE A MONOTONE
C                 DECREASING FUNCTION AS THE ARGUMENT G BECOMES LARGE...
C       TOL=      REAL TOLERANCE EXCEPTED AT CONVOLVED TAILS--I.E.,
C                 IF FILTER*FUN<TOL*MAX, THEN REST OF TAIL IS TRUNCATED.
C                 THIS IS DONE AT BOTH ENDS OF FILTER.  TYPICALLY,
C                 TOL <= .0001 IS USUALLY OK--BUT THIS DEPENDS ON
C                 THE FUNCTION FUN AND PARAMETER X...IN GENERAL,
C               A 'SMALLER TOL' WILL USUALLY RESULT IN 'MORE ACCURACY'
C               BUT WITH 'MORE WEIGHTS' BEING USED.  TOL IS NOT DIRECTLY
C               RELATED TO TRUNCATION ERROR, BUT GENERALLY SERVES AS AN
C               APPROXIMATION INDICATOR... FOR VERY LARGE OR SMALL B,
C               ONE SHOULD USE A SMALLER TOL THAN RECOMMENDED ABOVE...
C       L=        RESULTING NO. FILTER WTS. USED IN THE VARIABLE
C                 CONVOLUTION (L DEPENDS ON TOL AND FUN).
C               MIN.L=20 AND MAX.L=266--WHICH COULD
C                 OCCUR IF TOL IS VERY SMALL AND/OR FUN NOT DECREASING
C                 VERY FAST...
C     * NEW=    1 IS NECESSARY 1ST TIME OR BRAND NEW X.
C               0 FOR ALL SUBSEQUENT CALLS WHERE X=(LAST X)-0.20
C                 IS ASSUMED INTERNALLY BY THIS ROUTINE.
C                 NOTE:  IF THIS IS NOT TRUE, ROUTINE WILL
C                 STILL ASSUME X=(LAST X)-0.20 ANYWAY...
C                 IT IS THE USERS RESPONSIBILITY TO NORMALIZE
C                 BY CORRECT B=EXP(X) OUTSIDE OF CALL (SEE USAGE BELOW).
C               THE LAGGED CONVOLUTION METHOD PICKS UP SIGNIFICANT
C                 TIME IMPROVEMENTS WHEN THE KERNEL IS NOT A
C                 SIMPLE ELEMENTARY FUNCTION...DUE TO INTERNALLY SAVING
C                 ALL KERNEL FUNCTION EVALUATIONS WHEN NEW=1...
C                 THEN WHEN NEW=0, ALL PREVIOUSLY CALCULATED
C                 KERNELS WILL BE USED IN THE LAGGED CONVOLUTION
C                 WHERE POSSIBLE, ONLY ADDING NEW KERNEL EVALUATIONS
C                 WHEN NEEDED (DEPENDS ON PARMS TOL AND FUN)
C
C--THE RESULTING REAL CONVOLUTION SUM IS GIVEN IN RLAGF1; THE FOURIER
C  TRANSFORM IS THEN RLAGF1/B WHICH IS TO BE COMPUTED AFTER EXIT FROM
C  THIS ROUTINE.... WHERE B=EXP(X), X=ARGUMENT USED IN CALL...
C
C--USAGE-- 'RLAGF1' IS CALLED AS FOLLOWS:
C     ...
C     EXTERNAL RF
C     ...
C     R=RLAGF1(ALOG(B),RF,TOL,L,NEW)/B
C     ...
C     END
C     REAL FUNCTION RF(G)
C     ...USER SUPPLIED CODE...
C     END
C
C--NOTES:
C       (1).  EXP-UNDERFLOW'S MAY OCCUR IN EXECUTING THE SUBPROGRAM
C     BELOW; HOWEVER, THIS IS OK PROVIDED THE MACHINE SYSTEM SETS
C     ANY & ALL EXP-UNDERFLOW'S TO 0.0....
C     (2).  AS AN AID TO UNDERSTANDING & USING THE LAGGED CONVOLUTION
C     METHOD, LET BMAX>=BMIN>0 BE GIVEN. THEN IT CAN BE SHOWN
C     THAT THE ACTUAL NUMBER OF B'S IS NB=AINT(5.*ALOG(BMAX/BMIN))+1,
C     PROVIDED BMAX/BMIN>=1. THE USER MAY THEN ASSUME AN 'ADJUSTED'
C     BMINA=BMAX*EXP(-.2*(NB-1)).  THE METHOD GENERATES THE DECREASING
C     ARGUMENTS SPACED AS X=ALOG(BMAX),X-.2,X-.2*2,...,ALOG(BMINA).
C     FOR EXAMPLE, ONE MAY CONTROL THIS WITH THE CODE:
C                 ...
C               NB=AINT(5.*ALOG(BMAX/BMIN))+1
C               NB1=NB+1
```

```
C               XO=ALOG(BMAX)+.2
C               NEW=1
C               DO 1 J=1,NB
C               I=NB1-J
C               X=XO-.2*J
C               ARG(I)=EXP(X)
C               ANS(I)=RLAGF1(X,RF,TOL,L,NEW)/ARG(I)
C     1         NEW=0
C               ...
C     (3).  IF RESULTS ARE STORED IN ARRAYS ARG(I),ANS(I),I=1,NB FOR
C     ARG IN (BMINA,BMAX), THEN THESE ARRAYS MAY BE USED, FOR EXAMPLE,
C     TO SPLINE-INTERPOLATE AT A DIFFERENT (LARGER OR SMALLER)
C     SPACING THAN USED IN THE LAGGED CONVOLUTION METHOD.
C     (4).  IF A DIFFERENT RANGE OF B IS DESIRED, THEN ONE MAY
C     ALWAYS RESTART THE ABOVE PROCEDURE IN (2) WITH A NEW
C     BMAX,BMIN AND BY SETTING NEW=1....
C     (5). ABSCISSA CORRESPONDING TO WEIGHT IS GENERATED TO SAVE STORAGE
C
C
        DIMENSION KEY(266),SAVE(266)
        DIMENSION WT(266),W1(76),W2(76),W3(76),W4(38)
        EQUIVALENCE (WT(1),W1(1)),(WT(77),W2(1)),(WT(153),W3(1)),
     1 (WT(229),W4(1))
C--SIN-EXTENDED FILTER WEIGHT ARRAYS:
      DATA W1/
     1-1.1113940E-09,-1.3237246E-12, 1.5091739E-12,-1.6240954E-12,
     2 1.7236636E-12,-1.8227727E-12, 1.9255992E-12,-2.0335514E-12,
     3 2.1473541E-12,-2.2675549E-12, 2.3946842E-12,-2.5292661E-12,
     4 2.6718110E-12,-2.8227693E-12, 2.9825171E-12,-3.1514006E-12,
     5 3.3297565E-12,-3.5179095E-12, 3.7163306E-12,-3.9256378E-12,
     6 4.1464798E-12,-4.3794552E-12, 4.6252131E-12,-4.8845227E-12,
     7 5.1582809E-12,-5.4474462E-12, 5.7530277E-12,-6.0760464E-12,
     8 6.4175083E-12,-6.7783691E-12, 7.1595239E-12,-7.5618782E-12,
     9 7.9864477E-12,-8.4344110E-12, 8.9072422E-12,-9.4067705E-12,
     1 9.9349439E-12,-1.0493731E-11, 1.1084900E-11,-1.1709937E-11,
     2 1.2370354E-11,-1.3067414E-11, 1.3802200E-11,-1.4575980E-11,
     3 1.5390685E-11,-1.6249313E-11, 1.7155934E-11,-1.8115250E-11,
     4 1.9131898E-11,-2.0209795E-11, 2.1352159E-11,-2.2561735E-11,
     5 2.3840976E-11,-2.5192263E-11, 2.6618319E-11,-2.8122547E-11,
     6 2.9709129E-11,-3.1382870E-11, 3.3149030E-11,-3.5013168E-11,
     7 3.6981050E-11,-3.9058553E-11, 4.1251694E-11,-4.3566777E-11,
     8 4.6010537E-11,-4.8590396E-11, 5.1314761E-11,-5.4193353E-11,
     9 5.7236720E-11,-6.0455911E-11, 6.3861222E-11,-6.7461492E-11,
     1 7.1265224E-11,-7.5279775E-11, 7.9512249E-11,-8.3971327E-11/
      DATA W2/
     1 8.8668961E-11,-9.3621900E-11, 9.8851764E-11,-1.0438319E-10,
     2 1.1024087E-10,-1.1644680E-10, 1.2301979E-10,-1.2997646E-10,
     3 1.3733244E-10,-1.4510363E-10, 1.5330772E-10,-1.6196550E-10,
     4 1.7110130E-10,-1.8074257E-10, 1.9091922E-10,-2.0166306E-10,
     5 2.1300756E-10,-2.2498755E-10, 2.3763936E-10,-2.5100098E-10,
     6 2.6511250E-10,-2.8001616E-10, 2.9575691E-10,-3.1238237E-10,
     7 3.2994314E-10,-3.4849209E-10, 3.6808529E-10,-3.8878042E-10,
     8 4.1063982E-10,-4.3372666E-10, 4.5811059E-10,-4.8386049E-10,
     9 5.1105728E-10,-5.3977672E-10, 5.7011632E-10,-6.0215516E-10,
     1 6.3601273E-10,-6.7175964E-10, 7.0955028E-10,-7.4942601E-10,
     2 7.9161025E-10,-8.3606980E-10, 8.8317110E-10,-9.3270330E-10,
     3 9.8533749E-10,-1.0404508E-09, 1.0993731E-09,-1.1605442E-09,
     4 1.2267391E-09,-1.2942905E-09, 1.3691677E-09,-1.4429912E-09,      00019670
     5 1.5288164E-09,-1.6077524E-09, 1.7085998E-09,-1.7890471E-09,      00019680
     6 1.9129068E-09,-1.9857116E-09, 2.1491608E-09,-2.1926779E-09,      00019690
     7 2.4312660E-09,-2.3959044E-09, 2.7872500E-09,-2.5610596E-09,      00019700
     8 3.2762318E-09,-2.6082940E-09, 4.0261453E-09,-2.3560563E-09,      00019710
     9 5.3176554E-09,-1.3960161E-09, 7.7708747E-09, 1.1853546E-09,      00019720
     1 1.2760851E-08, 7.4264707E-09, 2.3342187E-08, 2.1869851E-08/      00019730
      DATA W3/                                                          00019740
     1 4.6306744E-08, 5.4631686E-08, 9.6763087E-08, 1.2823337E-07,      00019750
     2 2.0832812E-07, 2.9280540E-07, 4.5580888E-07, 6.5992437E-07,      00019760
     3 1.0056815E-06, 1.4779183E-06, 2.2284335E-06, 3.2994604E-06,      00019770
     4 4.9485823E-06, 7.3545473E-06, 1.1001083E-05, 1.6380539E-05,      00019780
     5 2.4469550E-05, 3.6469246E-05, 5.4441527E-05, 8.1176726E-05,      00019790
     6 1.2113828E-04, 1.8066494E-04, 2.6954609E-04, 4.0202288E-04,      00019800
     7 5.9969995E-04, 8.9437312E-04, 1.3338166E-03, 1.9886697E-03,      00019810
     8 2.9643943E-03, 4.4168923E-03, 6.5773518E-03, 9.7855105E-03,      00019820
     9 1.4539361E-02, 2.1558670E-02, 3.1871864E-02, 4.6903518E-02,      00019830
     1 6.8559512E-02, 9.9170152E-02, 1.4120770E-01, 1.9610835E-01,      00019840
     2 2.6192603E-01, 3.2743321E-01, 3.6407406E-01, 3.1257559E-01,      00019850
     3 9.0460168E-02,-3.6051039E-01,-8.6324760E-01,-8.1178720E-01,      00019860
     4 5.2205241E-01, 1.5449873E+00,-1.1817933E+00,-2.6759896E-01,      00019870
     5 8.0869203E-01,-6.2757149E-01, 3.4062630E-01,-1.5885304E-01,      00019880
     6 7.0472984E-02,-3.1624462E-02, 1.4894068E-02,-7.4821176E-03,      00019890
     7 4.0035936E-03,-2.2543784E-03, 1.3160358E-03,-7.8636604E-04,      00019900
     8 4.7658745E-04,-2.9125817E-04, 1.7885105E-04,-1.1012416E-04,      00019910
     9 6.7910334E-05,-4.1914054E-05, 2.5881544E-05,-1.5985851E-05,      00019920
     1 9.8751880E-06,-6.1008526E-06, 3.7692543E-06,-2.3287953E-06/      00019930
      DATA W4/                                                          00019940
     1 1.4388425E-06,-8.8899353E-07, 5.4926991E-07,-3.3937048E-07,      00019950
     2 2.0968284E-07,-1.2955437E-07, 8.0046336E-08,-4.9457371E-08,      00019960
     3 3.0557711E-08,-1.8880390E-08, 1.1665454E-08,-7.2076428E-09,      00019970
     4 4.4533423E-09,-2.7515696E-09, 1.7001092E-09,-1.0504494E-09,      00019980
     5 6.4904567E-10,-4.0102999E-10, 2.4778763E-10,-1.5310321E-10,      00019990
     6 9.4600354E-11,-5.8453314E-11, 3.6119400E-11,-2.2320056E-11,      00020000
     7 1.3793460E-11,-8.5242656E-12, 5.2675102E-12,-3.2543076E-12,      00020010
     8 2.0097689E-12,-1.2405412E-12, 7.6530538E-13,-4.7191929E-13,      00020020
     9 2.9084993E-13,-1.7923661E-13, 1.1018948E-13,-6.7885902E-14,      00020030
     1 4.2025050E-14,-2.1314731E-14/                                    00020040
C--$$ENDATA                                                             00020050
C                                                                       00020060
      IF(NEW) 10,30,10                                                  00020070
10    LAG=-1                                                            00020080
      X0=-X-38.30455704                                                 00020090
      DO 20 IR=1,266                                                    00020100
20    KEY(IR)=0                                                         00020110
30    LAG=LAG+1                                                         00020120
      RLAGF1=0.0                                                        00020130
      CMAX=0.0                                                          00020140
      L=0                                                               00020150
      ASSIGN 110 TO M                                                   00020160
      I=191                                                             00020170
      GO TO 200                                                         00020180
110   CMAX=AMAX1(ABS(C),CMAX)                                           00020190
      I=I+1                                                             00020200
      IF(I.LE.208) GO TO 200                                            00020210
      IF(CMAX.EQ.0.0) GO TO 150                                         00020220
      CMAX=TOL*CMAX                                                     00020230
      ASSIGN 120 TO M
      I=190
      GO TO 200
120   IF(ABS(C).LE.CMAX) GO TO 130
      I=I-1
      IF(I.GT.0) GO TO 200
130   ASSIGN 140 TO M
      I=209
      GO TO 200
140   IF(ABS(C).LE.CMAX) GO TO 190
      I=I+1
      IF(I.LE.266) GO TO 200
      GO TO 190
150   ASSIGN 160 TO M
      I=1
      GO TO 200
160   IF(C.EQ.0.0) GO TO 170
      I=I+1
      IF(I.LE.190) GO TO 200
170   ASSIGN 180 TO M
      I=266
      GO TO 200
180   IF(C.EQ.0.0) GO TO 190
      I=I-1
      IF(I.GE.209) GO TO 200
190   RETURN
C--STORE/RETRIEVE ROUTINE (DONE INTERNALLY TO SAVE CALL'S)
200   LOOK=I+LAG
      IQ=LOOK/267
      IR=MOD(LOOK,267)
      IF(IR.EQ.0) IR=1
      IROLL=IQ*266
      IF(KEY(IR).LE.IROLL) GO TO 220
210   C=SAVE(IR)*WT(I)
      RLAGF1=RLAGF1+C
      L=L+1
      GO TO M,(110,120,140,160,180)
220   KEY(IR)=IROLL+IR
      SAVE(IR)=FUN(EXP(X0+FLOAT(LOOK)*.20))
      GO TO 210
      END
$
```